The ANY-maze reference

Introduction

The ANY-maze reference contains detailed information about all parts of the system. This section isn't designed to be read like a book but rather is intended to be used as a reference when you're looking for the answer to a specific question. You will often find the easiest way to find something in the reference is to use the ANY-maze Help's Search facility.

Contents

The File page

Overview

As the name implies, the File page brings together all the functions relating to ANY-maze files. As you may be aware, ANY-maze stores each experiment you perform in a separate file, so the file page is the place to go to when you want to do things like create a new experiment, open an existing experiment, save an experiment, etc.

It's worth noting that in ANY-maze a single experiment file contains all the tests in the entire experiment. For example, in a water-maze experiment you might train 30 animals across six trials, so in total the experiment would contain 180 tests, and all of these will be stored in the same file. I mention this because it contrasts with some other video tracking programs which store each test in an individual file.

- The startup page
- Creating new experiments
- Opening existing experiments
- Opening example experiments
- Viewing experiment information
- Saving experiments
- Reverting to a saved experiment
- Exporting data from experiments
- Using 'Places' to quickly access your files

Startup page

Overview

The startup page is shown whenever you start ANY-maze and whenever you close an open experiment. This page allows you to either quickly load a recent experiment or create a new one.

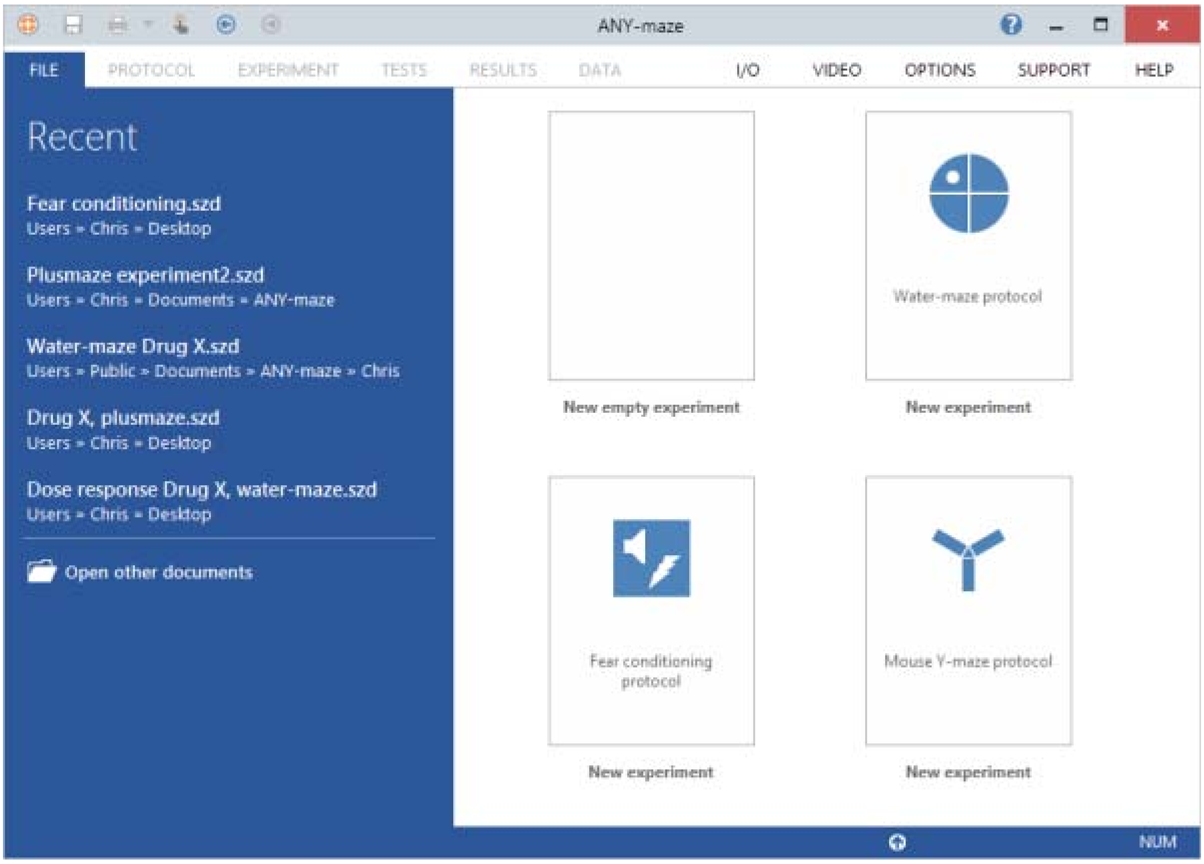


Figure 1. The left side of the startup page lists recently accessed experiments, while on the right are recently accessed protocols - selecting one will create a new experiment with the protocol loaded.

Opening recent experiments

The left side of the startup page lists experiments you have recently accessed. The number of experiments listed will be whatever value you have set in the Options > Filing page (the default is 20), although if this list is longer than will fit down the page, then the list will simply be truncated.

Of course, you might want to open an experiment which isn't listed - perhaps one you last worked on some time ago - and in this case you should select the last option in the list: *Open other documents*.

If you want to remove an experiment from the list, you simply need to right-click on the relevant item and select *Remove this experiment from the list* on the menu which appears.

Creating a new experiment

The right side of the page shows a list of protocol 'documents'. This always starts with a document titled *New empty experiment* and then includes one document for each recently used protocol. Clicking the *New empty experiment* document will, as you'd expect, create a new experiment with nothing in it; clicking any of the other documents will create a new experiment with the relevant protocol already loaded.

As you can see in figure 1, above, the protocol documents include a graphic which is intended to make it easier for you to find a specific protocol. The graphic shown is specified when you set up the protocol itself; details can be found here.

If you want to remove a document from the list (perhaps because you don't expect to use that protocol any more), then you should right-click on the document and select *Remove this protocol from the list* on the menu which appears.

See also:

- Creating new experiments
- Opening existing experiments
- Filing options

Creating new experiments

When you first start ANY-maze, or when you have just closed a file the startup page *is shown - details of how to create a new experiment from the startup page can be found* here*.*

Overview

Before you can do anything in ANY-maze you need to create an experiment file.

Creating an experiment file is extremely simple, just switch to the File page and select *New experiment* from the menu shown on the left. This will cause a list of 'documents' to be shown on the right side of the page, the first of which will be titled *New empty experiment;* selecting it will create a new experiment.

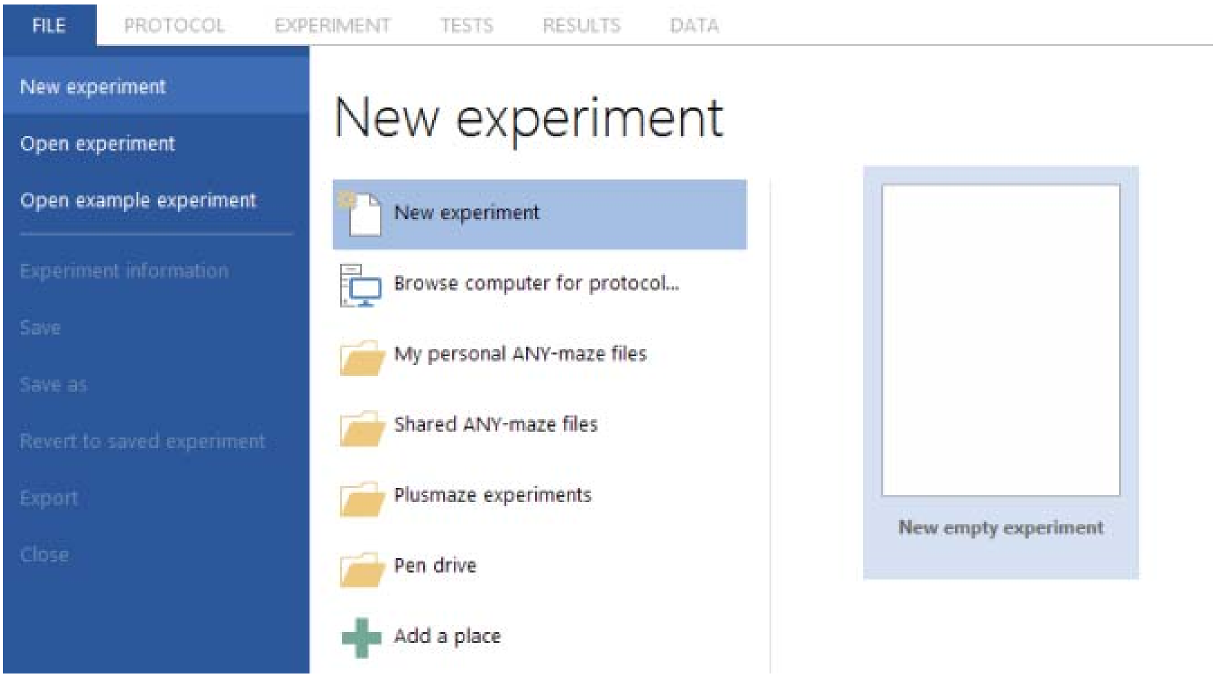


Figure 1. To create a new experiment, select New experiment *on the* File page *and then select the* 'New empty experiment' *document.*

Creating a new experiment based on an existing protocol

As you probably know, all experiments include a protocol which tells ANY-maze everything it needs to know about *how* the experiment is to be performed (if you're not already familiar with protocols, then I recommend you read the introductory tutorial on protocols).

Typically, you will create a number of different protocols for the different types of experiments you perform, and most often you will simply load one of these protocols into any new experiments that you create. For example, you might have a water-maze, a plusmaze and a Y-maze in your lab. You'd therefore have (at least) one protocol for each of them. So, when you want to perform a plusmaze experiment, you'd *first* create a new experiment file and *then* load the plusmaze protocol into it. But wouldn't it be easier if you could simply tell ANY-maze that you'd like to create a 'New plusmaze experiment' - it could then create the experiment file and load the protocol all in one step. Well you can; indeed there are a few ways to do this:

Creating a new experiment using the documents listed on the New Experiment page

The simplest method is to select one of the documents listed on the *New experiment* page. These documents represent the protocols you have used recently, and so it's likely that the protocol you want will be listed. When you select a document, ANY-maze will create a new experiment and load the protocol into it.

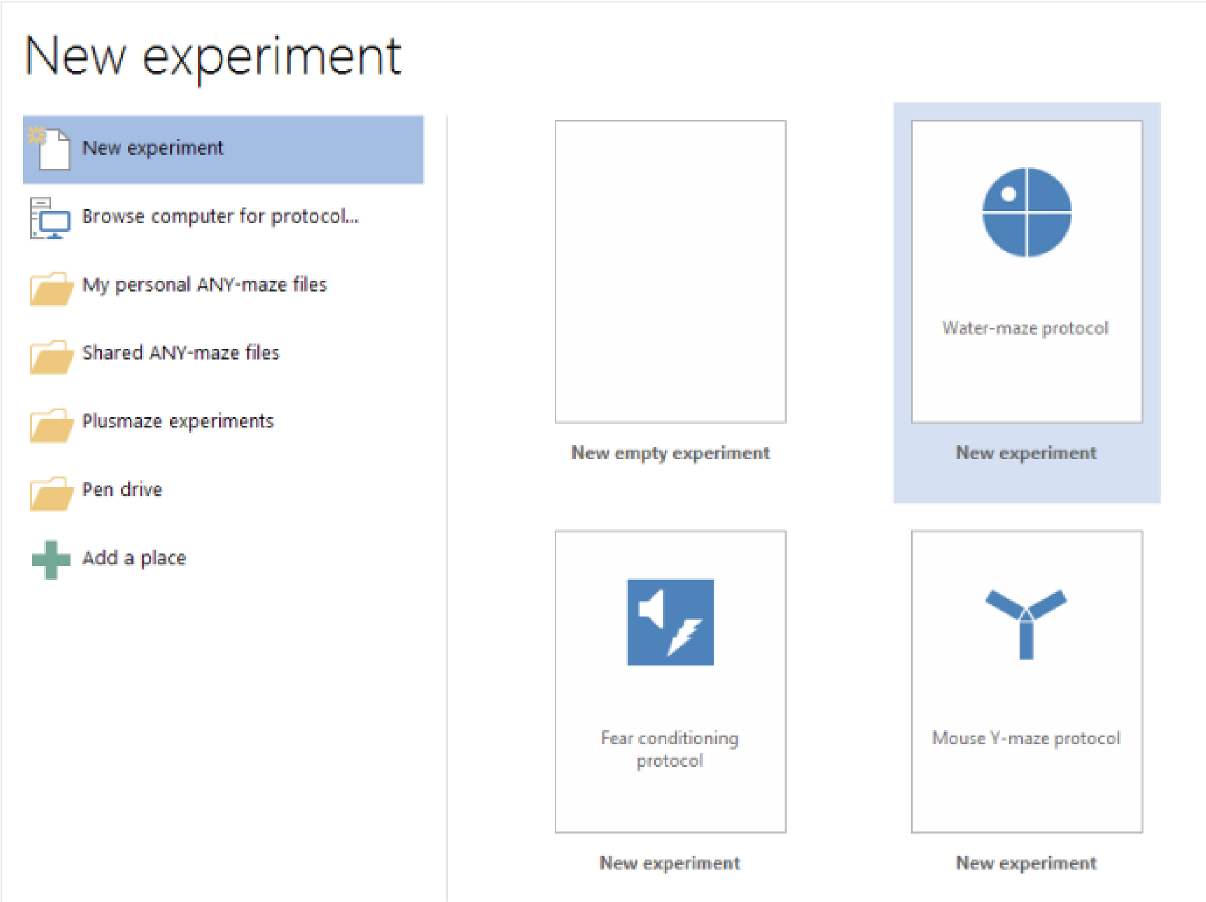


Figure 2. Selecting a protocol document will create a new experiment and load the protocol into it.

As you can see in figure 2, the protocol documents include a graphic which is intended to make it easier for you to find a specific protocol. The graphic shown is specified when you set up the protocol itself; details can be found here.

If you want to remove a 'document' from the list of protocols (perhaps because you don't expect to use that protocol any more), you should right-click on the document and select *Remove this protocol from the list* from the menu which appears.

Creating a new experiment using some other protocol or experiment file

In fact, you can create a new experiment based on any protocol (or experiment) file, not just those in the recent protocols document list. To do this you should select the *Browse computer for protocol...* option (see figure 2). This will cause the Open protocol window to appear, where you can choose a protocol file. When you select a file, ANY-maze will create a new experiment and load the chosen protocol into it.

In fact, you're not limited to loading a protocol from a protocol file - you can also load a protocol directly from another experiment file. In this case you just need to choose an *experiment,* rather than a protocol, file. This is described in more detail here.

An alternative to selecting *Browse computer for protocol...* is to select one of the 'Places' listed on the *New experiment* page, such as *Shared ANY-maze files.* Selecting a 'Place' will list all the protocol and experiment files in it - making it very quick to select a file. You can learn more about 'Places' here.

Creating a new experiment by double clicking a protocol file in Windows

The ultimate shortcut method for creating a new experiment with a pre-loaded protocol is to double click the relevant protocol file in Windows. For example, you might have a protocol file on your desktop - if you double click it then:

- If necessary, ANY-maze will start
- A new experiment will be created
- The protocol will be loaded into the experiment

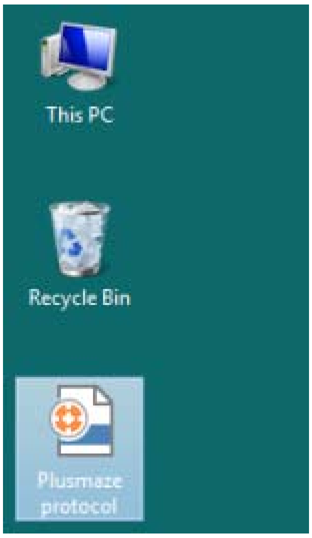


Figure 3. Double clicking a protocol file is a convenient way to open ANY-maze, create a new experiment and load a protocol - all with one action.

Creating a new experiment from an experiment design file

Opening an Experiment design file will create a new experiment, load a protocol into it, and add animals to to the experiment - i.e. everything required to set-up an experiment ready to be performed.

As the name suggests, the Experiment design file specifies which protocol to load and specifies the experiment's animals, it can also define various other options such as whether the protocol can be edited, whether tests can be 'undone' etc. Full details about Experiment design files can be found here.

See also:

- An introductory tutorial on protocols
- Saving and loading protocols
- The startup page
- Using 'Places' to quickly access your files

The Open protocol window

Overview

When you use the *Browse computer...* option to open a protocol on which to base a new experiment, the *Open protocol window* will be displayed. Here you can navigate around your computer or network to find the file you want to open.

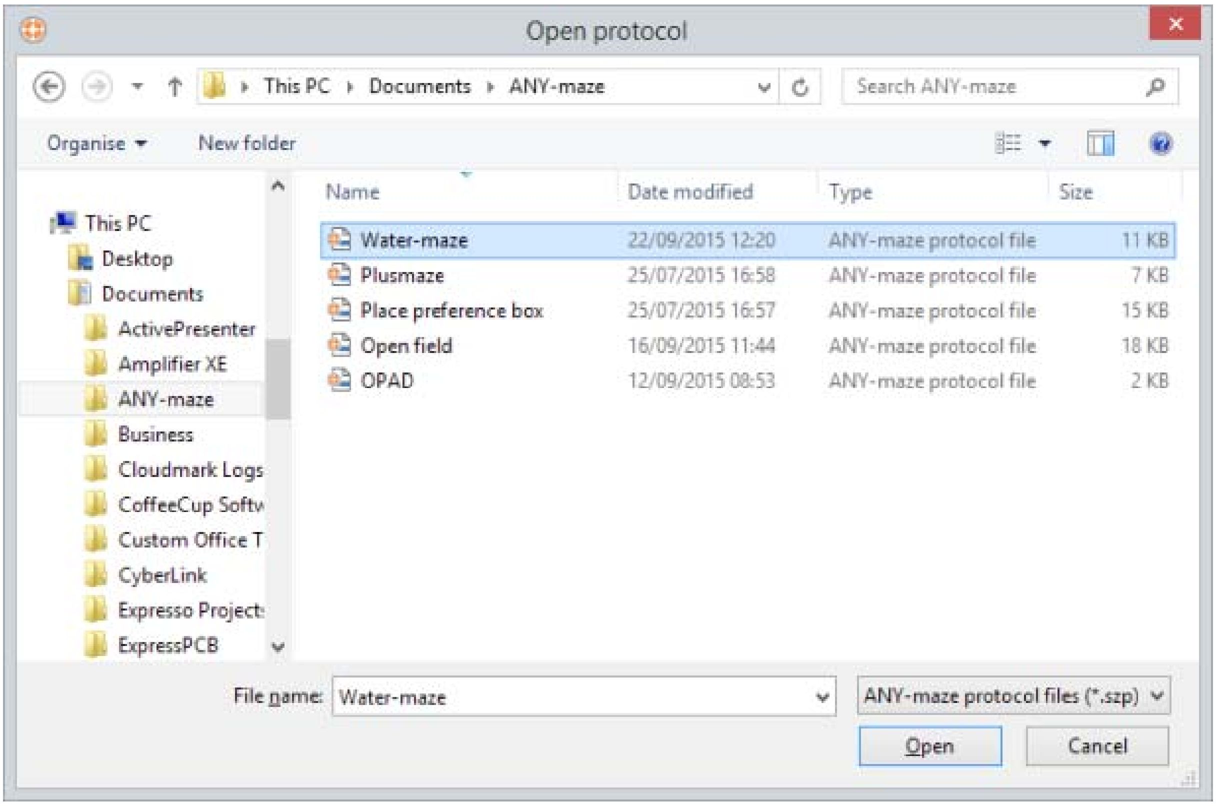


Figure 1. The Open protocol *window.*

Details

You will probably be familiar with this window already, as it's based on the standard 'Open file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list	The files list shows the folders and experiment files stored in the current location. You can double click a folder to open it so you can access the files inside. If you double click a protocol file, then ANY-maze will open it.
File name	You can use this field to type in the location and name of the file you want to open - but usually it's much easier to open a file by double clicking on it in the *Files list*.
Type of file	The window will always start with *ANY-maze protocol files* selected as the *Type of file,* but you can change this so that the *Files list* will show other files types instead. The options are:

ANY-maze protocol files	The default option. ANY-maze will create a new experiment and load the chosen protocol into it.
ANY-maze experiment files	If you choose an experiment file, ANY-maze will read the protocol directly from the experiment, then create a new experiment and write the protocol to it. This operation will not affect the chosen experiment file in any way.
All files	The *Files list* will show all the files in the selected location, but you should only open an ANY-maze protocol, experiment (or backup) file; opening any other type of file will fail.

See also:

- Creating new experiments
- Saving and loading protocols

ANY-maze help topic H0063

Experiment design files

Introduction

Experiment design files provide a mechanism by which you can predefine an experiment and then create it with a single action. This can be useful if the person who will create the experiment is not the same person who will perform it. The 'creator' can specify all the details of the experiment within the Experiment design file, and the 'performer' then just has to load the design file and run the experiment.

- Creating an Experiment design file
- Using an Experiment design file

Creating an Experiment design file

Experiment design files are just text files and can be created using any text editor, such as Notepad in Windows.

The file has a specific, although simple format: The first line must contain the text '[ANY-maze]' (without the quotation marks). Subsequent lines contain a series of settings, one per line, in any order. Each setting consists of a name followed by an equals sign, followed by a value. The names of the settings are not case sensitive, nor are the values. Some of the settings are required, but most are optional. For example:

```
 [ANY-maze]
 Protocol           = C:\Experiments\ANY-maze protocols\watermaze.szp
 ExperimentTitle    = Watermaze test of drug X at three concentrations
 ExperimentFile     = C:\Experiments\ANY-maze experiments\Test of Drug
X.szd
 AnimalList         = C:\Experiments\ANY-maze experiments\Test of Drug X
- animals.csv
 LockProtocolEdits = Y
```

The full list of Experiment design file settings is:

Protocol	(Required) The full path and name of the protocol file that will be used by the experiment. For example: 'C:\Experiments\ANY-maze protocols\watermaze.szp'
LockProtocolEdits	(Optional) Y or 1 to prevent the protocol from being edited by anyone other than an administrator. N or 0 to allow anyone to edit the protocol. If

	this setting is not included then anyone will be able to edit the protocol.
ExperimentTitle	(Optional) The title of the experiment. This will be written to the experiment title field on the Experiment page. If no title is specified then the experiment simply won't have a title.
ExperimentFile	(Optional) The full path and name of the experiment file to be created. For example: 'C:\Experiments\ANY-maze experiments\Test of Drug X.szd'. If an experiment file is specified then the experiment will be saved to this file immediately after it has been created. If no experiment file is specified then the user will need to specify the experiment file path and name when he first saves the file. In this case, if an 'ExperimentTitle' has been specified then it will be *suggested* as the file name.
AnimalList	(Optional) The full path and name of a file containing a list of the animals to be tested in the experiment. This file should be formatted in exactly the same way as an animal import file; full details of this format can be found here.
DisallowUndoLastTest	(Optional) Y or 1 to prevent the user from undoing tests, N or 0 to allow the user to undo tests. If this setting is not specified, the user will be able to undo tests.
AlwaysSaveTestResults	(Optional) Y or 1 to force test results to always be saved (if the user ends the test manually). N or 0 to allow the user to choose whether or not to save the test results. If this settings is not specified, the user will be able to choose whether or not to save the test results.
BarcodeVerification	(Optional) Y or 1 to force animal IDs to be verified by scanning the animal's barcode or chip before every test. N or 0 to not require animal IDs to be verified in this way. If this setting is not specified, the animal ID will not need to verified.
BarcodePrefix	(Optional) If 'BarcodeVerification' is being used, then this specifies the character code of the barcode prefix character used by the barcode or chip scanner. If this setting is not specified then a default barcode prefix of character code 3 (ASCII code ETX) will be used. Note that this is different to the default that is normally used by ANY-maze, which is 2 (ASCII code STX). If 'BarcodeVerification' is not being used then this setting is ignored.
BarcodeSuffix	The suffix character for barcodes If 'BarcodeVerification' is being used, then this specifies the character code of the barcode suffix character used by the barcode or chip scanner. If this setting is not specified then a default barcode suffix of character code 3 (ASCII code ETX) will be used. Note that this is different to the default that is normally used by ANY-maze, which is 4 (ASCII code EOT). If 'BarcodeVerification' is not being used then this setting is ignored.

Below is an example of an Experiment design file which uses every setting:

```
 [ANY-maze]
 Protocol              = C:\Experiments\ANY-maze protocols\watermaze.szp
 LockProtocolEdits     = Y
 ExperimentTitle       = Watermaze test of drug X at three
concentrations
 ExperimentFile        = C:\Experiments\ANY-maze experiments\Test of
Drug X.szd
 AnimalList            = C:\Experiments\ANY-maze experiments\Test of
Drug X - animals.csv
 DisallowUndoLastTest  = N
 AlwaysSaveTestResults = Y
 BarcodeVerification   = Y
 BarcodePrefix         = 2
 BarcodeSuffix         = 3
```

Using an Experiment design file

To use an experiment design file, you just need to open the file in the same way as you'd normally open an experiment file; for example, you could select *Open experiment* on the File page, change the type of file on the file open window to 'Experiment design file' and then select the file. Alternatively, you could drag an Experiment design file into ANY-maze, or you could simply double click on an Experiment design file from within Windows. In all cases the Experiment design file will be opened.

When you open a design file ANY-maze will create an experiment based on the design file settings. If during this process something goes wrong, for example, the protocol file specified doesn't exist, or the Animal list contains an invalid entry, then a message will be displayed explaining what the problem is and no experiment will be created.

After successfully creating an experiment from the design file, ANY-maze will note that the design file has been used and will then prevent it from being used again. In other words, you can only create an experiment from a design file once. If you try to use it again, a message will be displayed stating this and detailing who used the design file and when.

See also:

- Importing animals
- Undoing a performed test
- Choosing whether results will always be saved when a test is ended manually
- Choosing whether or not to verify animal IDs before each test
- Barcode readers or chip scanners that can be used to verify animal IDs

Opening existing experiments

When you first start ANY-maze, or when you have just closed a file, the startup page *is shown - details of how to open an experiment from the startup page can be found* here*.*

Overview

To open an experiment in ANY-maze, you should switch to the File page and there select *Open experiment* from the menu shown on the left. This will cause a list of *Recent experiments* to be shown on the right side of the page, and you'll often find that the experiment you want is listed there - just select it and it will open.

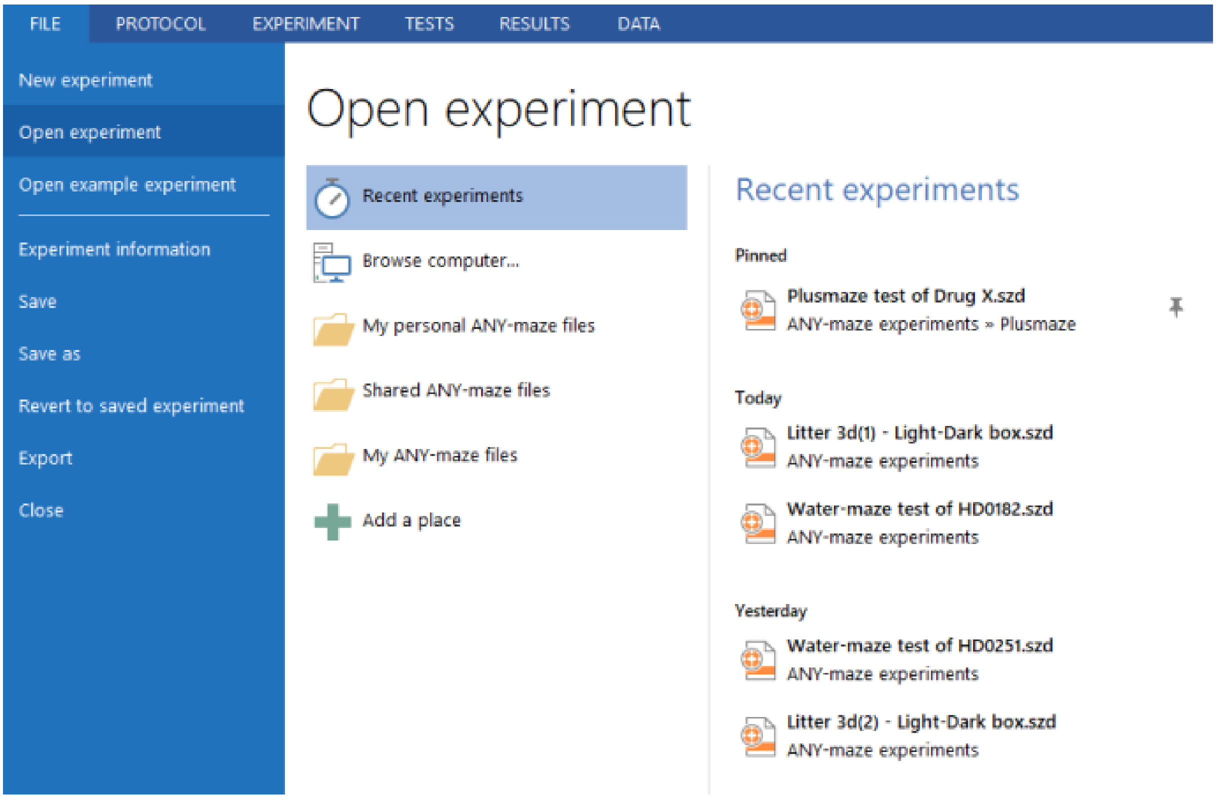


Figure 1. To open a new experiment, select Open experiment *on the* File page, *you'll often find the experiment you want in the* Recent experiments *list.*

Pinning experiments

You can pin experiments to the recently used experiments list so they are always shown, no matter how recently you used them. To pin an experiment, move the mouse over it, so it is highlighted, and then click the pin button shown to the far right of the experiment's name. Pinned experiments appear at the top of the Recent Experiments list.

Sorting the recently used experiments list

The recently used experiments list can be sorted either by date (the default option) or alphabetically by file name. To choose how the list should be sorted, simply right click over the list and selecting the appropriate sort option from the menu that appears.

Opening experiments not in the Recent experiments list

Although you'll often find that the experiment you want to open is in the *Recent experiments* list, there will obviously be occasions when it isn't. In these cases you should select the option to *Browse computer...;* this will cause the Open experiment window to open, where you can navigate to the file.

An alternative to using the *Browse computer...* option is to select one of the 'Places' listed on the *Open experiment* page, such as *Shared ANY-maze files* (see figure 1). Selecting a 'Place' will list all the experiment files in it - making it very quick to select a file. You can learn more about 'Places', including how to add more of them, here.

Opening password protected files

As you may know, when you save an experiment file, you can optionally protect it with a password. If you try to open a file which has been protected in this way, ANY-maze will display the File password window where you will have to enter the password before you can access the file.

In fact, if the password you used to protect the file is the same as your ANY-maze user password and you're logged on, ANY-maze won't require you to enter the password and the protected file will open immediately.

See also:

- Opening example experiments
- Using 'Places' to quickly access your files
- The File password window
- Saving experiments

The Open experiment window

Overview

When you use the *Browse computer...* option to open an experiment, the *Open experiment window* will be displayed. Here you can navigate around your computer or network to find the file you want to open.

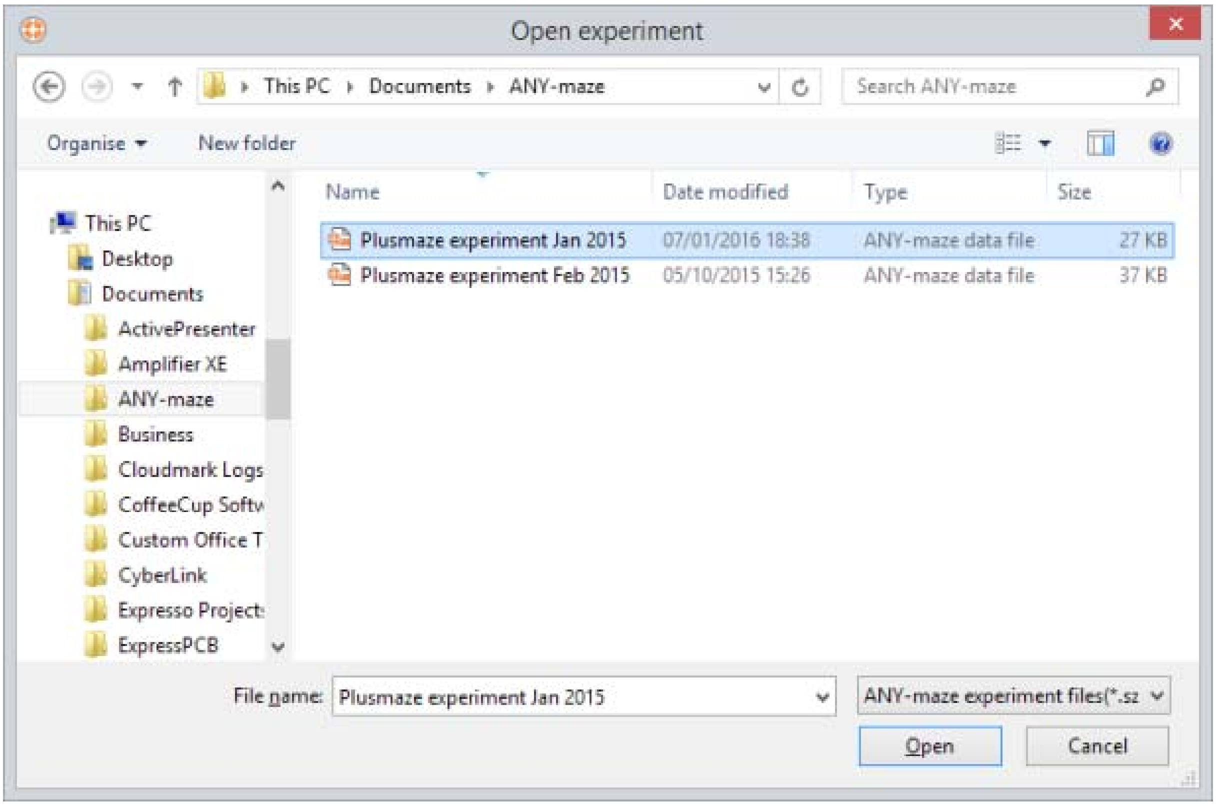


Figure 1. The Open experiment *window.*

Details

You will probably be familiar with this window already, as it's based on the standard 'Open file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list	The files list shows the folders and experiment files stored in the current location. You can double click a folder to open it so you can access the files inside. If you double click an experiment file, then ANY-maze will open it.
File name	You can use this field to type in the location and name of the file you want to open - but usually it's much easier to open a file by double clicking on it in the *Files list.*
Type of file	The window will always open with *ANY-maze experiment files* selected as the *Type of file,* but you can change this so that the *Files list* will show other files types instead. The options are:

ANY-maze experiment files	The default option.
ANY-maze backup files	You can open a backup file just like you can open experiment files.
ANY-maze protocol files	If you open a protocol file, ANY-maze will create a new experiment and load the protocol into it.
ANY-maze experiment design files	If you open a design file, ANY-maze will create an experiment, load a protocol (defined in the design file) into it, and then import animals and tests into the experiment (the files to import from are defined in the design file). Experiment design files are described fully here.
All files	The *Files list* will show all the files in the selected location, but you should only open an ANY-maze experiment, backup, protocol or design file - opening any other type of file will fail.

See also:

- Opening existing experiments

File password

Overview

When you save an ANY-maze experiment or protocol, you can optionally specify a password which will be used to protect the file. When you try to open a file which has been protected in this way, the *File password window* will open where you must enter the correct password before you can access the file.

In fact, if the password you used to protect the file is the same as your ANY-maze user password and you're logged on, then ANY-maze won't require you to enter the password and the protected file will open immediately.

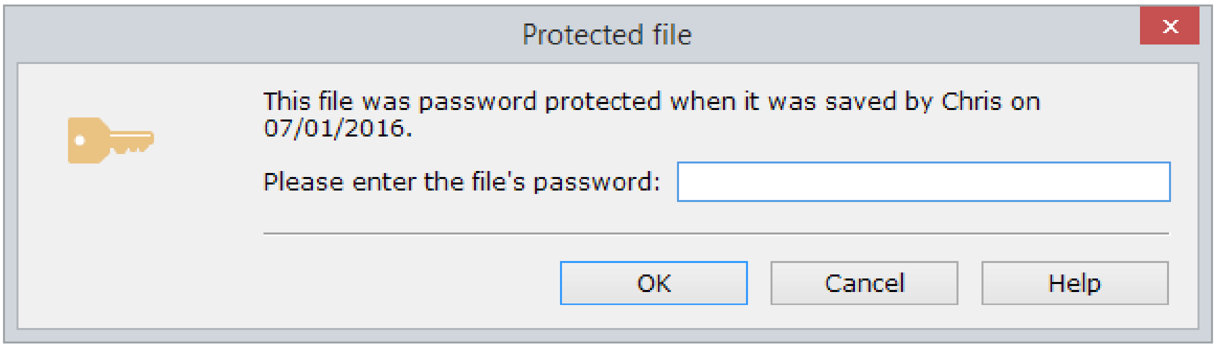


Figure 1. The File password *window where you must enter the correct password before you can access a protected file.*

Details

Simply enter the correct password into the *Password* field. Note that passwords are case sensitive.

If you enter the wrong password, there will be a short delay before you can try again - each time you enter the wrong password this delay will double. This is a security feature which makes it very hard for someone to try to guess your password.

Opening example experiments

Overview

ANY-maze is provided with a number of example experiments (exactly how many depends on the version you've installed) which are designed to help you quickly evaluate the system and learn how to use it.

Details

To open an example experiment, switch to the File page and select *Open example experiment* from the menu on the left. A list of example experiments will appear; just click on the one you want - see figure 1.

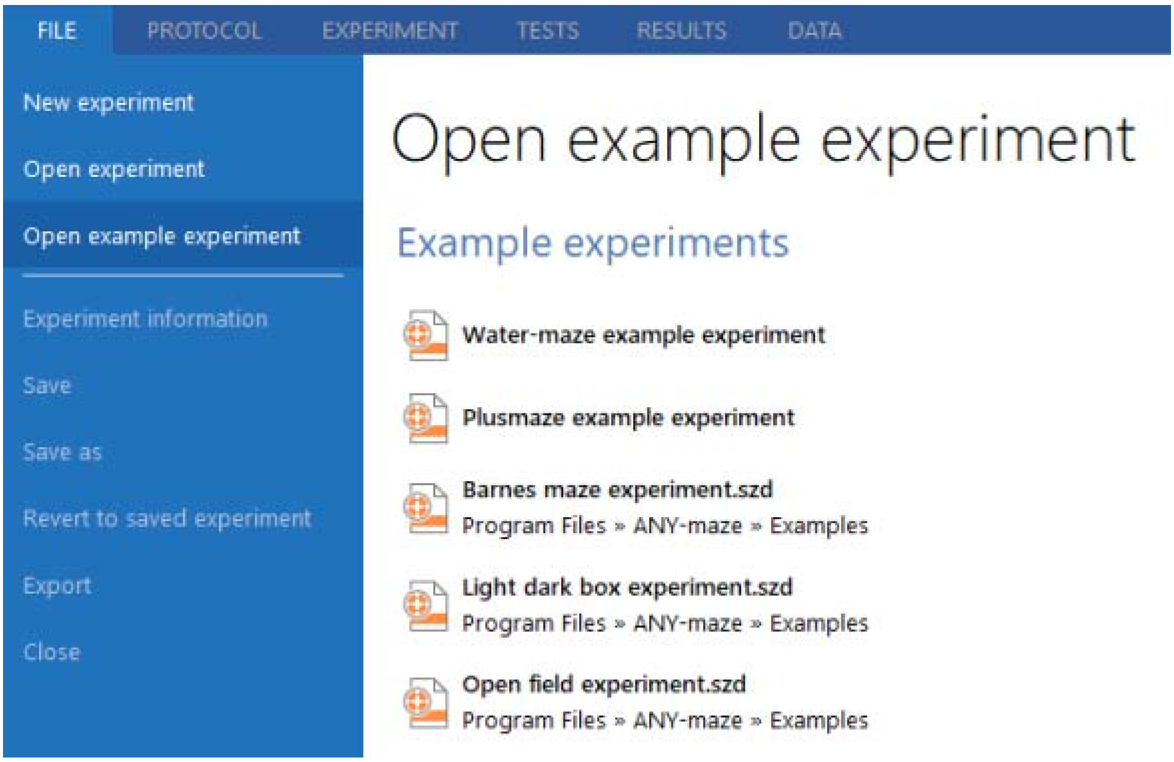


Figure 1. Choosing an example experiment to open.

The Open example experiment *option will only appear in the menu on the left of the* File page *if it is set to do so in the* Options > Features for new users *page.*

Viewing experiment information

Overview

Whenever you switch to the File page when an experiment is open, ANY-maze will display the *Experiment information*. This not only provides details about the experiment but also allows you to alter the file's protection (for example, to add a password), view details of related files, and review all backup copies of the file.

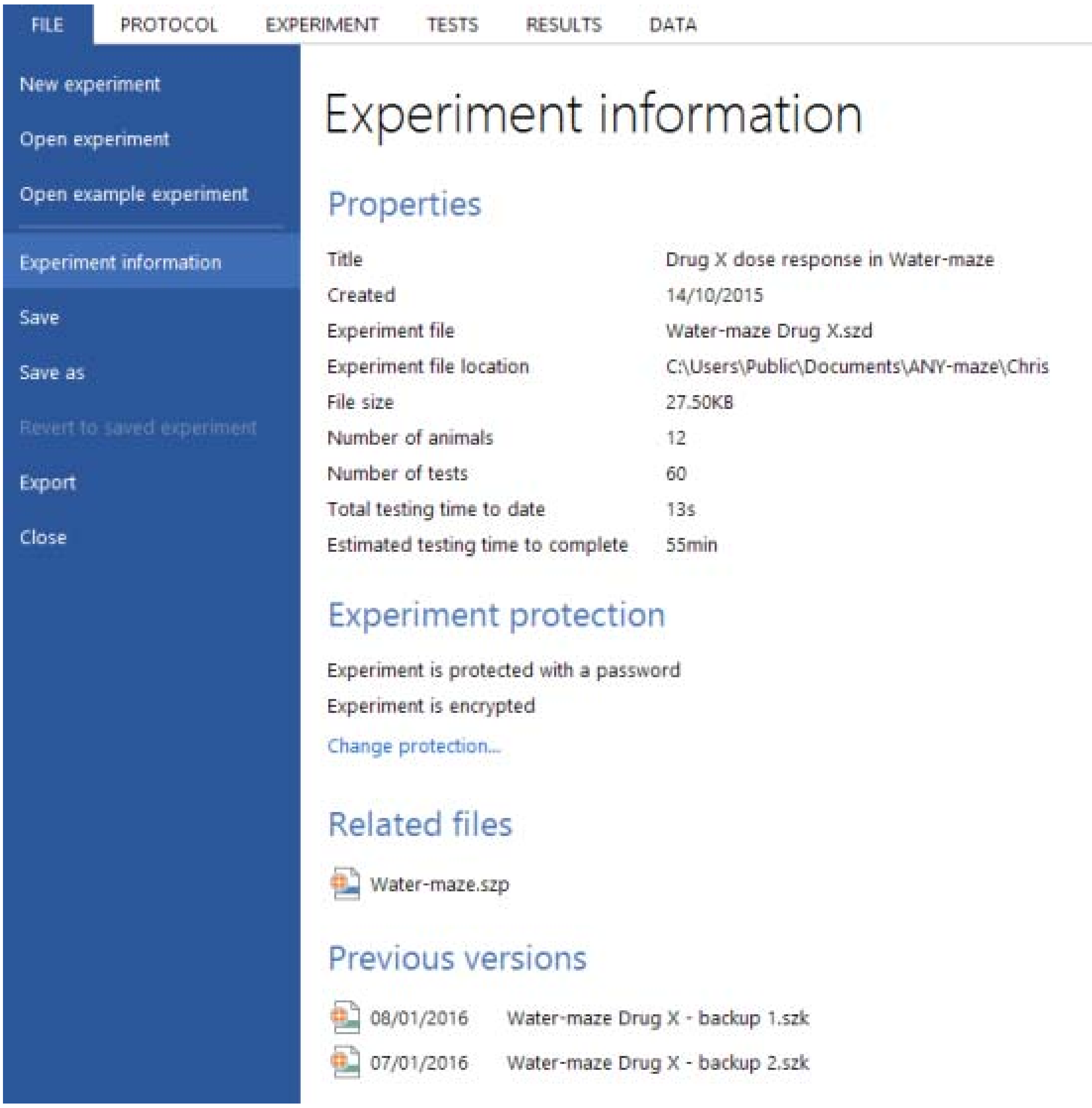


Figure 1. The experiment information shows a wealth of useful information about an experiment.

Details

Properties	The Properties section contains much useful information about the experiment, such as the location where the file is saved, the number of tests and animals in the experiment, the total amount of time spent testing so far and an estimate of how long the remaining tests will take to complete.
Experiment protection	Experiment files can be password protected and/or encrypted, and this section will show the file's current protection. Usually if you want to protect a file you will specify a password when you first save it, but you can use the *Change protection* option to access the File protection window where you can add, edit or remove the file's password and/or encrypt the file.
Related files	This section will list files related to this experiment. This includes any protocol file used to load a protocol into the experiment, any files from which animals or tests were imported, the *folder* to which test videos are being saved automatically and, if relevant, the experiment design file. Clicking on any of the files (or folders) shown here will locate the file in Windows Explorer (i.e. show you where on your computer the file is).
Previous versions	Whenever you save an experiment, ANY-maze will (usually) store the previous version of the file as a backup. If you save the file again, then you'll have two previous versions; save again and you'll have three - i.e. you'll have a history of all the previous versions of the file. You probably don't want this to go on forever, otherwise you could end up with a lot of previous versions taking up a lot of space on your disk drive, so you can tell ANY-maze how many previous versions it should retain. By default the value is 3, but you can change it in Options > Filing to any number up to 9 (you can also turn off this storing of previous versions altogether if you want). As you'll doubtless have realised, the previous versions section of the Experiment information lists all the previous file versions. Right-clicking on one of these files will give you two options: Restore the backup file (this will load the backup file in place of the current experiment), or locate the file in Windows Explorer (i.e. show you where on your computer the file is).

See also:

- Filing options

ANY-maze help topic H0069

File protection window

Although ANY-maze can provide a reasonable level of casual *security for your ANY-maze files, it is neither highly secure (an expert could probably 'crack' an ANY-maze file quite quickly) nor does it provide a complete solution. For example, while ANY-maze can protect your files so other users can't* open *them, it can't prevent someone else from moving, copying or deleting them - for that you will need to use Windows' native file security features.*

Overview

The *File protection window* is accessed from the Experiment information shown on the File page. Here you can add, edit or delete the password used to protect a file and optionally encrypt the file contents.

Figure 1. Using the File protection window, you can password protect and encrypt a file.

Password protecting a file

When you save an experiment in ANY-maze, you have the option to enter a password, but what if you forgot to enter a password at the time, or you want to change the password, or you want to remove the password you had set? Well, all these things can be done using the *File protection window*.

Here you can specify that you do (or don't) want the file to be protected, and you can enter the password you want to use. You can enter anything you like as the password, but it must be between 5 and 15 characters long.

By the way, if you expect you will usually want to secure your files and you are **using ANY-maze users** then you can set ANY-maze to *automatically* protect files using your ANY-maze user password. This has the added benefit that ANY-maze won't require you to enter your password each time you open a protected file, provided you're logged on under your own user account and you haven't changed your user password since you saved the file. For more details, refer to Editing your user account details.

Encrypting a file

Password protecting a file simply means that it can't be opened in ANY-maze without first entering the password. But that doesn't mean that someone couldn't open the file in another program (such as a binary file viewer) and look at your data that way. In fact, much of the data would be quite hard to interpret, but some things (such as notes) are stored as plain text and could be read very easily. To address this, ANY-maze can encrypt your files.

In fact, it will do this automatically if you save the file with a password, but you can use the options on the *File protection window* to change this; for example, you could save a file with a password but without encrypting it.

Saving experiments

Overview

Saving an experiment in ANY-maze is as simple as switching to the File page and selecting *Save experiment* or *Save experiment as* from the menu on the left. If the experiment is new, i.e. it's never been saved before, then the Save experiment window will open where you can specify a file name and choose the location to save it to.

If the file has been saved before, then selecting *Save experiment* will immediately save the experiment - this is usually so fast that it may not be too obvious that it has happened.

There's also a shortcut method for saving an experiment; you can simply click the button in the top left of the ANY-maze window title bar. You can do this at any time, even when a test is running.

ANY-maze includes other facilities related to saving experiments, and these are described in the sections below:

- The Save experiment window
- Saving experiments without ANY-maze asking for confirmation
- Backing up experiment files
- Protecting experiment files with a password
- Recovering files in the event of a "crash"

Saving experiments without ANY-maze asking for confirmation

Whenever you close an experiment, ANY-maze will check whether it contains any changes which haven't yet been saved and if it does, it'll ask you whether you'd like to save them - probably you'll almost always reply *Yes*. For this reason, you can use the Filing options to tell ANY-maze to always save changes without asking you - thus avoiding this question which can otherwise become a bit irritating.

Using this facility does mean that it's harder to cancel changes to a file if you don't want to save them. For example, imagine you opened an experiment and added 10 animals to it. You then noticed that you'd opened the wrong file. If you simply close the file, ANY-maze will save your changes automatically (here we're assuming you *are* using the option to save changes without asking for confirmation). To avoid this you should select the File page option to Revert to saved experiment - this will simply discard the changes you've made.

Backing up experiment files

We've all been told a thousand times how important it is that we keep backups of our files - but let's be honest, most of us don't, or if we do, we make backups less frequently than we should.

ANY-maze includes a simple option which, while by no means representing a full backup facility, does at least add a little extra protection. Essentially, the system can retain the previous version of an experiment file as a backup each time you save a new version.

If you want to use this facility, then you should switch it on in the Filing options - it's entirely transparent but, of course, you'll find you use more disk space because you'll be keeping multiple copies of each experiment file. You can see a list of an experiment's backup files by viewing the Experiment information.

By the way, backup files are given the extension **.szk** and you can open them by using *File page > Open experiment > Browse computer* and then changing the *Type of file* in the 'Open file' window to *ANY-maze backup files*.

Protecting experiment files with a password

If you work in a lab where lots of people have access to the ANY-maze computer, you may wish to protect your experiment files using a password.

To do this is very easy - at the bottom of the Save experiment window is a field titled *Password* where you simply need to make an entry in order to protect the file. When you, or someone else, tries to open the file, the File password window will open and they'll need to enter the password in order to access the file.

In fact, if you expect you'll usually use this facility, then you can tell ANY-maze to automatically protect all your files using your ANY-maze password*. This has two benefits - you won't need to enter a password each time you save a file and, provided you're logged on, ANY-maze will open your files without asking you for the password as you had to provide it to log-on in the first place.

If you want to use this facility, you should check the box titled *Protect my files by default...* which is shown in Options > My account.

It's important to understand that protecting a file in this way will prevent other people from opening it, but it won't prevent them from using Windows to move it, copy it, delete it, etc. - to prevent that, you should protect the file using Windows' native security options.

**This facility is only available if you're using ANY-maze to manage its own users - see An introduction to managing users for more information.*

Recovering files in the event of a crash

Unlike most computer programs which only save your work when you explicitly select a 'Save' option, ANY-maze saves everything you do as you do it. This means that your work is always secure - even if there's a power-cut.

This isn't like the 'auto-save' option in Microsoft Word, which saves your work at set time intervals; in ANY-maze, everything is saved as you do it. There's just one exception to this rule - test results are only saved at the end of a test.

Of course you might wonder why, if ANY-maze saves everything all the time, it still has a 'Save' option at all. The answer is that ANY-maze saves your work to a temporary file, and it only makes this temporary file permanent when you 'save' it. This means that you can still abandon changes by simply not 'saving' an experiment.

So what happens if disaster strikes? Well, if your computer goes wrong (a crash) or there's a power-cut, then ANY-maze will exit without removing the temporary file in which it saves your actions. As a result, next time you start ANY-maze it will find the temporary file and ask you if you'd like to recover the data it contains - see figure 2. In general, you should answer **YES** - ANY-maze will then recover all the data and you can continue working as if the crash, power-cut or whatever had never occurred. By the way, if you answer NO, ANY-maze will delete the temporary file and your experiment file will be left in the same state as it was on the last occasion you explicitly saved it.

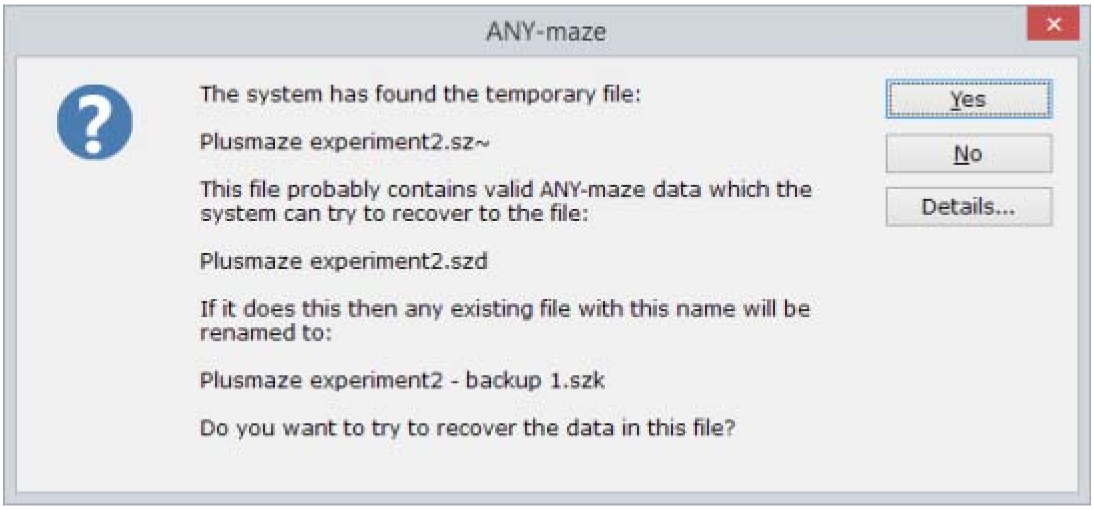


Figure 2. If your computer crashes, ANY-maze will detect a temporary experiment file when it restarts. Recovering the data in the file will recover everything you did between the last time you saved and the time the computer crashed.

Temporary files are user-specific. This means that to recover a file, you must log back on using the *same user account as you were using when the computer crashed.*

See also:

- Filing options
- My account option

The Save experiment window

Overview

When you save an experiment for the first time, or you use *Save as...*, the *Save experiment* window will be displayed. Here you can provide a name for the experiment, and select where on your computer or network you'd like it to be saved.

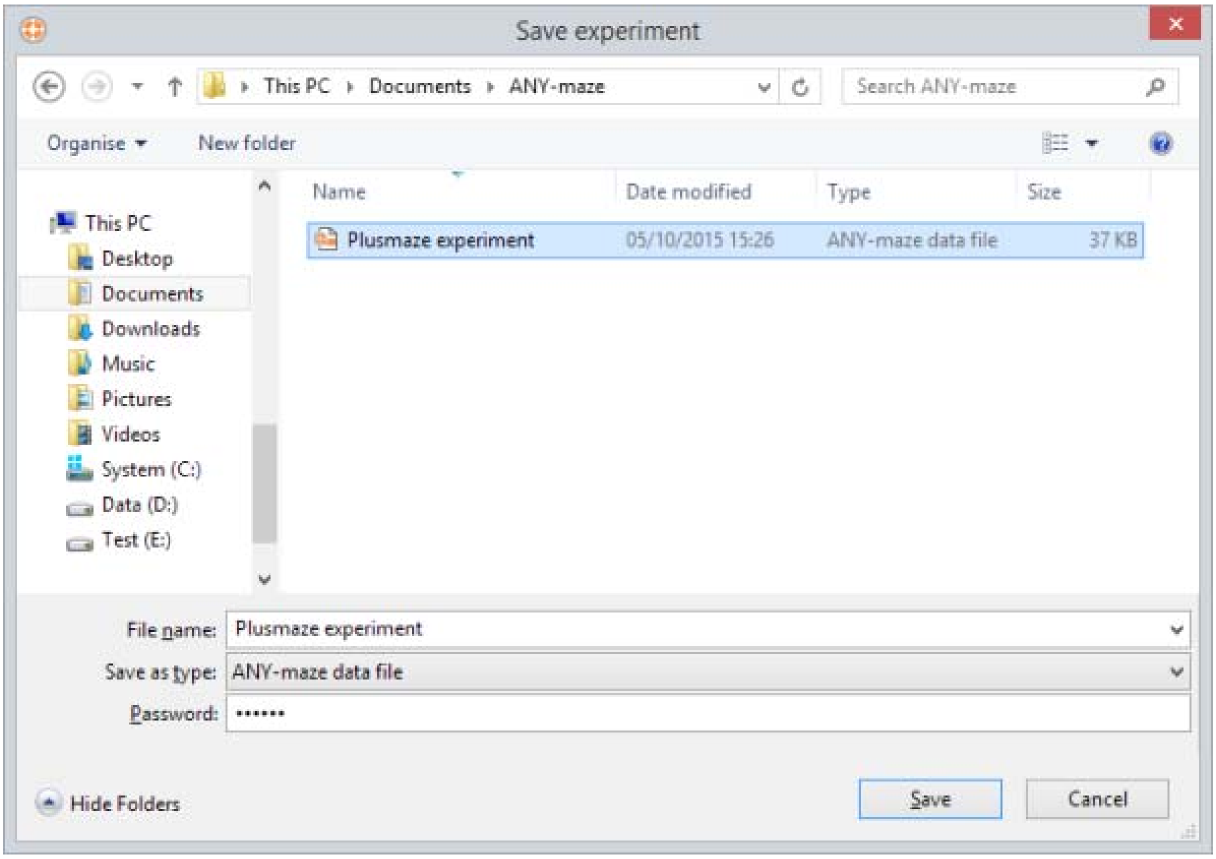


Figure 1. The Save experiment *window.*

Details

You will probably be familiar with this window already, as it's based on the standard 'Save file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list — The files list shows the folders and experiments files stored in the current location. You can double click a folder to open it and you can click a file to transfer its name to the File name field.

File name — Use this field to enter a name for the experiment file. ANY-maze will automatically fill in this field with the *Experiment's title,* assuming you've given it one, but you can delete this and type in something else if you wish to.

File names can be up 255 characters long, but cannot contain any of the following characters: \ / : * ? " < > |. If the experiment's title includes any of these characters, they'll be replaced by a space when the title is copied to this field.

Type of file — Experiments are always saved as *ANY-maze experiment (.szd)* files, so you can ignore the file types listed.

Password — You can optionally enter a password to protect the file. If you do so, then you will need to re-enter the password when you try to open the file in the future.

You can enter anything you like as the password, but it must be between 5 and 15 characters long. If you don't enter anything at all, the file will not be protected. You can change a file's protection at any time using the File protection window.

In fact, if you expect you'll usually password protect your files, then you can tell ANY-maze to automatically protect all your files using your ANY-maze password. This has two benefits - you won't need to enter a password each time you save a file and, provided you're logged on, ANY-maze will open your files without asking you for the password as you had to provide it to log-on in the first place. If you want to use this facility, you should check the box titled *Protect my files by default...* which is shown in Options > My account.

It's important to understand that protecting a file in this way will prevent other people from opening it, but it won't prevent them from using Windows to move it, copy it, delete it etc. - to prevent that, you should protect the file using Windows' native security options.

See also:

- Creating new experiments
- Saving and loading protocols

Reverting to a saved experiment

Overview

Sometimes you might decide you want to throw away all the changes you made since you opened a file. For example, you might be editing the wrong file, or perhaps you added some more animals to an experiment and now you've changed your mind. Whatever the reason, it's easy to revert to the saved copy of a file; you just need to switch to the File page and select *Revert to saved experiment* from the menu on the left.

Details

When you *revert* an experiment, the changes you made since the file was last saved are irretrievably lost - no backup is made.

If you are using the option to save experiments without ANY-maze asking for confirmation (described in detail here), then the ability to *revert* is very useful, because if you simply close a file, ANY-maze will automatically save it - so if you *don't* want to save the file, you should *first* revert the file and *then* close it.

Exporting experiment data

Overview

ANY-maze data files use a proprietary format which means they can't be read by other programs. This can be a problem if you want to analyse an experiment's data using a program such as Matlab.

To address this, ANY-maze includes an export facility. To use it, switch to the File page and select *Export* from the menu on the left. The right-hand side of the page will change to show the following options:

- Export experiment as XML
- Export test data

Exporting an experiment in XML format

The ability to export an experiment in XML format is a work in progress. At present, it is targeted at saving the tracking data for performed tests; however we will gladly include more data, so if there's something you would like included, just contact us at ANY-maze technical support.

Introduction

ANY-maze experiments are saved in a proprietary binary format. This format is optimized for speed of access and file size (some of the data is compressed) which is ideal for ANY-maze itself, but means that it is virtually impossible for any other programs to read the experiment data.

To address this, you can export an experiment in XML format. XML is a standard, text-based format which can be used to record data about virtually anything. Many programs (for example, Excel and Matlab) can read XML files.

- How to export an experiment as an XML file
- Specifying the XML file name
- Selecting the data to export
- Exporting coordinates relative to the centre of the apparatus
- Choosing the format for animal 'images'
- Excluding tests performed on retired animals
- Exporting an XSD (schema) file
- The XML file contents

How to export an experiment as an XML file

The option to export an experiment as an XML file is included on the File page. To use it, switch to the file page, select *Export > Export experiment as XML*. When you do this, the right side of the page will show a field for you to enter the name of the XML file, a list for you to select the data to be included in the XML file, and various options - see figure 1.

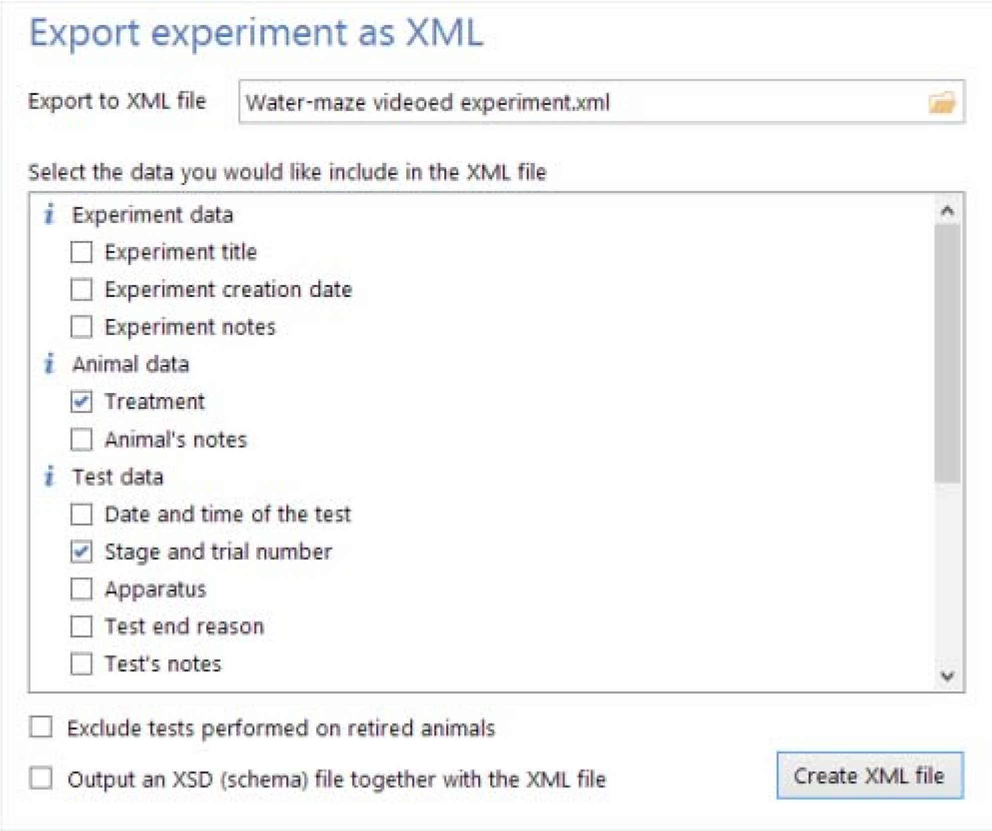


Figure 1. The options available when exporting a file as XML.

Specifying the XML file name

By default, ANY-maze will save the XML file in the same location as the experiment (.szd) file and with the same name - just the file extension will change to .xml.

However, you can alter the folder and/or XML file name simply by clicking the button (on the right of the name field), which will open the standard 'Save file' window.

Selecting the data to export

The export function will always export an experiment and the animals and tests within it, but you can choose what details of each item you would like included. This is quite self-explanatory - just check the boxes next to the data you would like included. Note that the animal number and the test number are always included in the XML file, so they're not listed as options.

Exporting coordinates relative to the centre of the apparatus

By default exported coordinates are relative to the top/left of the video image in which the test was performed, with positive x moving to the right and positive y moving down.

In some situations this coordinate system can be a bit unsatisfactory and you may find that exporting coordinates relative to the centre of the apparatus is a better option. In this case the origin of the coordinate system is the centre of the apparatus map, with positive x moving to the right and positive y moving up (in other words this coordinates system is the same as the Cartesian coordinate system usually used in graphs).

Choosing the format for animal 'images'

In tests where ANY-maze tracks an animal, it will usually save the 'image' of the animal together with every position (this behaviour can be switched off using the option *Don't record animal images* in the What to record while testing element of the protocol). These 'images' represent the area that ANY-maze considers to be the animal - this is the area that it can shade in blue during a test.

These animal images can be saved as part of the XML file, but doing so will create a dramatically larger file - so you should carefully consider whether you want to include them or not. There are three options:

- Don't save animal images. This is the default option. The animals' position data will be saved in the XML file, but the images will not.
- Output animal images as Base64 encoded bitmaps. This option encodes the images to help reduce the XML file size, but the resulting data is less easy to process than the 'pixel' option (see below). The exact format of the data is described in the XML file contents section of this document.
- Output animal images as unencoded pixels. This option creates the largest files, but the image data is most easily processed. Be warned, using this option can create very large XML files.

Excluding tests performed on retired animals

ANY-maze will normally export all the animals and all the tests in the experiment except for animals which are marked as deleted and tests which are marked as cancelled. If you also wish to exclude animals which are marked as retired then you should choose this option.

Exporting an XSD (schema) file

If you need an XSD (schema) file for the exported XML file, select the option to *Output an XSD file*. The XSD file will be saved to the same location as the XML file and will have the same name, just the extension will be different - it will be .xsd rather than .xml.

The XML file contents

We will gladly extend the data included in an exported XML file - just contact ANY-maze technical *support.*

The XML file that is exported by this function does NOT include ALL the data that is held in the ANY-maze experiment file; rather it targets the tracking data of performed tests.

The XML file consists of a hierarchy of elements as follows:

Experiment
Animal
Test
Result
Animal image data

Experiment

The experiment element is the root. It contains the experiment title, creation date, notes, and all the animals.

Animal

There is one animal element for each animal which that has any performed tests (note that animals which have not yet had an tests performed will, of course, be in your experiment file, but they will NOT be exported).

The element contains the number, ID (if you have specified an animal ID field), treatment, notes (if any), and all of the animal's performed tests.

Tests

There is one test element for each test which has been performed on the animal (tests which have not yet been performed are NOT included in the XML file). The test element contains the test number, the date and time the test was performed, the test's stage and trial, the name of the apparatus the test was performed on, the reason the test ended, test notes (if any), the position of all zones, scaling (i.e. the number of pixels per metre) and all of the test's results.

The information about the position of zones deserves a little explanation. First, the position of the zones is defined for each test so as to cope with any *movable* zones - the position provided will be the position the zone had in the specific test. Second, the position is provided as two values; the coordinates of the centre of the zone and the zone's *bounding box*. The centre is the zone's *centre of mass* and is the mean x and mean y coordinate of every pixel that is in the zone. It is important to understand that the centre may not actually be in the zone; for example, a ring-shaped zone would have its centre of mass in the centre of the hole in the middle of the ring. The zone's bounding box is a box which exactly encloses the zone and is specified as the top-left coordinate of the box and its width and height.

Results

Only tracking results are included in the XML file (so, for example, results from I/O devices are not exported). There is one result for each position of the animal. Each result contains the time of the position (in seconds relative the start of the test) and the position of the animal's centre (relative to the top left of the video image). The results may also contain the position of the animal's head and tail (if the tracking system was able to determine them) and details of any zones that the position implies the animal has entered or exited.

While most of the tags used in the XML file are self-explanatory, the result tags are abbreviated to keep file sizes to a minimum. The abbreviated tags are:

- r = result
- tm = time
- c = position of the animal's centre
- h = position of the animal's head
- t = position of the animal's tail
- np = no position

The 'no position' tag is output when the animal was not tracked in a frame and the option to Record frames in which the animal was not tracked is turned on.

If you choose to output the animal images (see above), then each result will also include animal image data.

Animal image

When selected, each result will include animal image data - this being the area of the video image which ANY-maze considered to be the animal. The image data consists of two parts: The first part defines a rectangular area by providing values for x, y, w and h; where x, y are the location of the top left of the rectangle in the video image (relative to the top left corner) and w and h are the rectangle's width and height respectively. (Note that the area will always have a width that is a multiple of 8).

The second part defines the contents of the rectangular area as a series of pixels - a pixel that is on is deemed to be part of the animal, a pixel that is off is not part of the animal. There are two ways in which the pixels can be defined:

The first method defines the rows of the rectangular area with each row consisting of a text string. A '0' in the string represents an off pixel and a '1' represents an on pixel; there are as many 0s and 1s as the width of the area.

For example:

```
<image>
  <x>107</x>
  <y>336</y>
  <w>16</w>
  <h>6</h>
  <row>0000000111100000</row>
  <row>0000011111111000</row>
  <row>0000001111110000</row>
  <row>0000000111100000</row>
```

```
  <row>0000000011000000</row>
  <row>0000000011000000</row>
</image>
```

Note how the area is defined as having a width of 16, so each row has 16 0s or 1s in it, and a height of 6, so there are 6 rows.

The second method encodes the image data in Base64. In this case all the row data is viewed as continuous (the w value can be used to determine the row length) and the 0 and 1 values are encoded as bits in a series of bytes. The resulting series of bytes is then encoded in Base64.

Using the data from the above example, we would therefore get something like this:

```
The first row is 0000000111100000 which can be viewed as two bytes:
  00000001 = 0x01,  11100000 = 0xE0

Processing the other rows in the same way gives :
  00000111 = 0x07,  11111000 = 0xF8
  00000011 = 0x03,  11110000 = 0xF0
  00000001 = 0x01,  11100000 = 0xE0
  00000000 = 0x00,  11000000 = 0xC0
  00000000 = 0x00,  11000000 = 0xC0

So the complete bytes series is:
  0x01,0xE0,0x07,0xF8,0x03,0xF0,0x01,0xE0,0x00,0xC0,0x00,0xC0

Which encoded in Base64 is:
  AeAH+APwAeAAwADA
```

So in this case the resulting XML would be:

```
<image>
  <x>107</x>
  <y>336</y>
  <w>16</w>
  <h>6</h>
  <bitmap>AeAH+APwAeAAwADA</bitmap>
</image>
```

As can be seen, this is much more compact than the pixel format (it's about 20% of the size), but the data will need to be decoded before it can be processed.

Exporting test data

Data can also be exported from ANY-maze in XML format.

Introduction

Most of the results that ANY-maze provides for a test are generated by processing the raw tracking data recorded while the test is running. For example, the distance the animal travels during a test is (more or less) the sum of the distances between each of its positions throughout the test. Clearly, these are the sorts of results you'll generally want, but there may be occasions when you'd like to access the raw data directly.

As you may know, this can easily be done using the Test data report, which presents the raw data in a spreadsheet format. You can easily save this report in a range of different formats, for example CSV, which can be read by programs such as Excel and Matlab. However, while this mechanism is great for saving the raw data for an individual test, it's not very satisfactory if you want to save the raw data for an entire experiment, which might include hundreds of tests. This is where the ability to Export test data comes in.

Details

Exporting the Test data will automatically generate one file for each test in the experiment, with the file contents matching that of the Test data report spreadsheet.

To export the data in this way is very simple:

- Switch to the File page
- Select the Export option
- Select the *Export test data* option
- The right-hand side of the File page will show a field where you can specify the *Base file name* for the export (described below), set it.
- Click the *Save data* button

Setting the base file name

As described above, exporting the test data will create one file for every test in the experiment. So, clearly, this will generate multiple files. These files will all be saved to the same place and will be called 'XXX - test 1', 'XXX - test 2', etc., where 'XXX' is whatever you specify as the *Base file name*.

By default, ANY-maze uses the experiment file name as the base file name (without the .szd extension). So if your experiment is called 'Wonder drug Plusmaze' then the export will generate files called 'Wonder drug Plusmaze - test 1', etc. However, you can change the base file name to anything you like.

Choosing where the files will be saved

You can also use the button on the right of the *Base file name* field to open a standard 'Save file' window which will allow you to select where the files should be saved. By default, they'll be saved to the same folder as the experiment file.

Choosing the file format to save in

The 'Save file' window, accessed from the button on the right of the *Base file name* field, also allows you to specify the file format to save the spreadsheet in. The default is CSV, but options include SYLK, DBASE III and tab-separated values.

Using 'Places' to quickly access your files

Introduction

Most of us tend to organise our computer files into some sort of logical structure within folders. For example, you might have a folder somewhere on your computer called 'ANY-maze experiments', and within that you might have separate folders organised by project, or perhaps by apparatus type.

Whenever you want to access a file in ANY-maze, you'll obviously need to navigate through this structure to the relevant folder (unless the file happens to be in a 'recently used' list). *Places* are designed to make this navigation quicker. Essentially, a place is just a shortcut to a specific folder, and selecting one will display the files the folder contains, directly in ANY-maze.

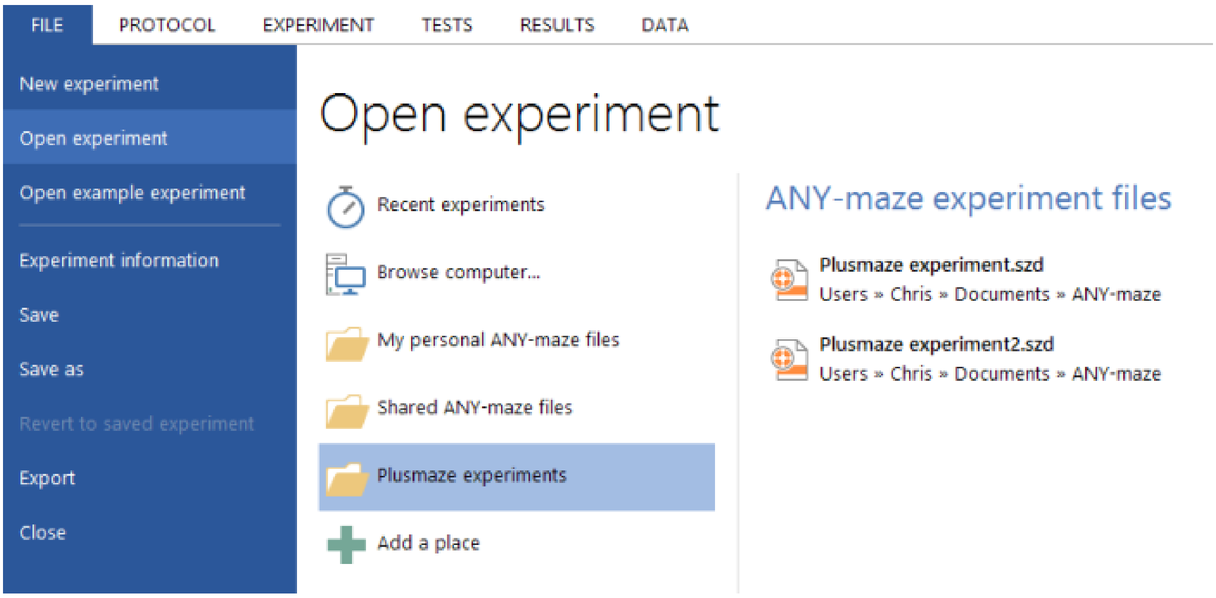


Figure 1. An example of a place; the list on the right is showing all of the experiment files in the 'Plusmaze experiments' place.

Adding a place

You can add a place simply by selecting the *Add a place* option shown when either *New experiment* or

Open experiment is selected on the File page. Doing this will cause the Add a place window to open - see figure 2.

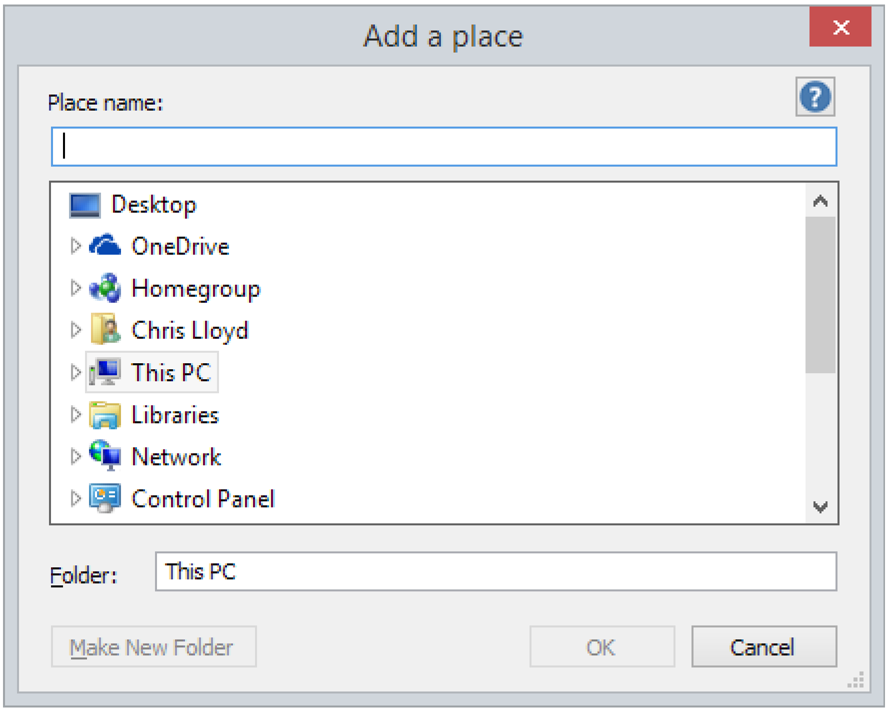


Figure 2. The Add a place window. *Here you simply need to enter a name for the place and select the folder it will refer to.*

Here you simply need to choose the folder you want the place to refer to and give the place a name, which can be any text you want.

After adding a place it will appear in the list of places shown when you select either *New experiment* or *Open experiment* on the File page (so it doesn't matter where you added the place; it will appear in both lists).

Using a place

To use a place, simply select it in the list. The right-hand side of the File page will then show a list of all the appropriate ANY-maze files in that place - see figure 1, above, for an example.

Sorting the files in a place

The list of files in a place can be sorted either by date (the default option) or alphabetically by file name. To choose how the list should be sorted, simply right click over the list and selecting the appropriate sort option from the menu that appears.

Editing or removing a place

To edit a place, right click it in the list and select *Properties* from the menu which appears. This will cause the Add a place window to open, where you can edit both the place name and the folder the place refers to.

To remove a place, right click it and select *Remove this place* from the menu which appears. This will remove the place from the list, but won't affect the folder the place refers to.

Automatically included places

ANY-maze includes two places automatically; these are *My personal ANY-maze files* and *Shared ANY-maze files.* The folders these places refer to are set up as part of the Filing options. These places can't be removed.

See also:

- Add/edit a place window
- Filing options

Add/edit a place window

Overview

The *Add a place window* is used to add a new place to ANY-maze. For an introduction to places, see Using 'Places' to quickly access your files.

Details

To add a place, you need to specify two things: the place name and the folder it refers to.

The name can be any text you want - there's no length limit.

You can choose any folder for a place, except for Windows 'virtual folders' such as the 'Libraries' folder. Until you select a valid folder, the OK button on the window will be disabled.

The Protocol page

Introduction

The *Protocol* is the part of an ANY-maze experiment which defines *how* the experiment will be performed. For example, it specifies where the video pictures used for tracking will come from, what the experiment's apparatus looks like, how long tests should last etc.

As you can imagine, the protocol is completely fundamental to an experiment and, in fact, you can't do anything in ANY-maze without first setting one up.

Protocols are divided into *elements*, each of which defines a specific feature; for example, there's an element for Apparatus, another one for Stages of the experiment, and another for Analysis. In fact, there are more than 40 elements altogether but you are very unlikely to need to define all of them - indeed, in the simplest protocol, you just need to define five.

For more information about protocols, refer to the topics listed below.

- An introductory tutorial on protocols
- The elements of a protocol
- Adding elements to a protocol
- Editing the elements of a protocol
- Deleting elements from a protocol
- Saving and loading protocols
- Viewing the protocol report

An introductory tutorial on protocols

Introduction

Protocols are perhaps the most important part of ANY-maze, because it's through them that the system gains its power and flexibility. Clearly then, a good understanding of protocols is important if you're going to get the most out of the system.

In this tutorial, I'll explain how a protocol is structured and I'll introduce some of their more fundamental parts. We'll then go on to create a simple protocol for a water-maze which will contain everything necessary to track an animal.

It's a good idea to print this tutorial as you'll find it easier to actually perform the steps described if you're working from paper. Just click the Print button above to print it now.

- The structure of a protocol
- The Protocol element itself
- Adding a video source
- Adding apparatus
- Setting the tracking options
- Setting the test duration
- Finished - a simple but complete protocol

The structure of a protocol

Before we do anything else, we need to create a new experiment file. To do this:

- Select the File page
- Select *New experiment* from the list on the left.
- The right side of the page will then show a list of protocol 'documents', the first of which is *New empty experiment* - click it.

If you follow the above instructions, ANY-maze will create a new empty experiment and automatically switch to the Protocol page. On the left of this page is what we call the *Protocol list* which contains all of the parts of the protocol - see figure 1.

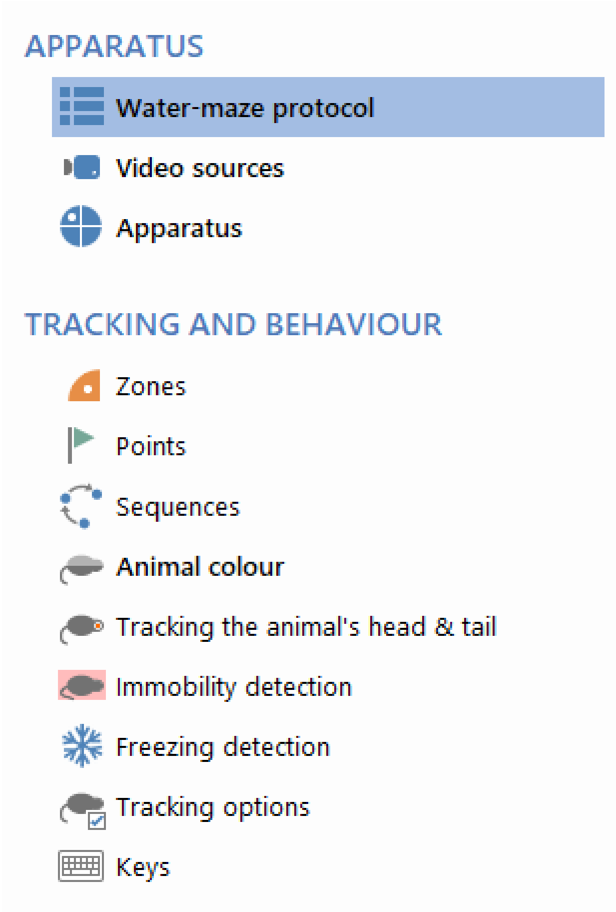


Figure 1. Part of the 'Protocol list'.

As you can see, the list is broken into different sections, each with a number of items inside it - for example, the *Tracking* section contains *Zones, Points, Sequences,* etc. These are what we call the *elements* of a protocol, and there are more than 40 in total. Fortunately, a protocol will almost never use all of them; rather they're like files in a filing cabinet where you *can* record information if you need to. For example, the element called *Sequences* can be used to record information about sequences of movements that you want ANY-maze to measure, but if your experiment doesn't involve any sequences which interest you, you simply won't include any.

In fact, there are a few elements which must be present in a protocol, otherwise ANY-maze won't have any idea what it's supposed to be doing (they're shown in bold in the protocol list), and it's these elements which we're going to look at here; the first of them is the *Protocol element* itself.

The Protocol element

The first element in all protocols is the Protocol element itself, and this is the element which will currently be selected in the *Protocol list.* Just to right of the *Protocol list* is a pane which we call the *Settings pane,* as it shows the settings of whatever element is currently selected in the *Protocol list.* So, the *Settings pane* will currently be showing the settings of the protocol itself - see figure 2

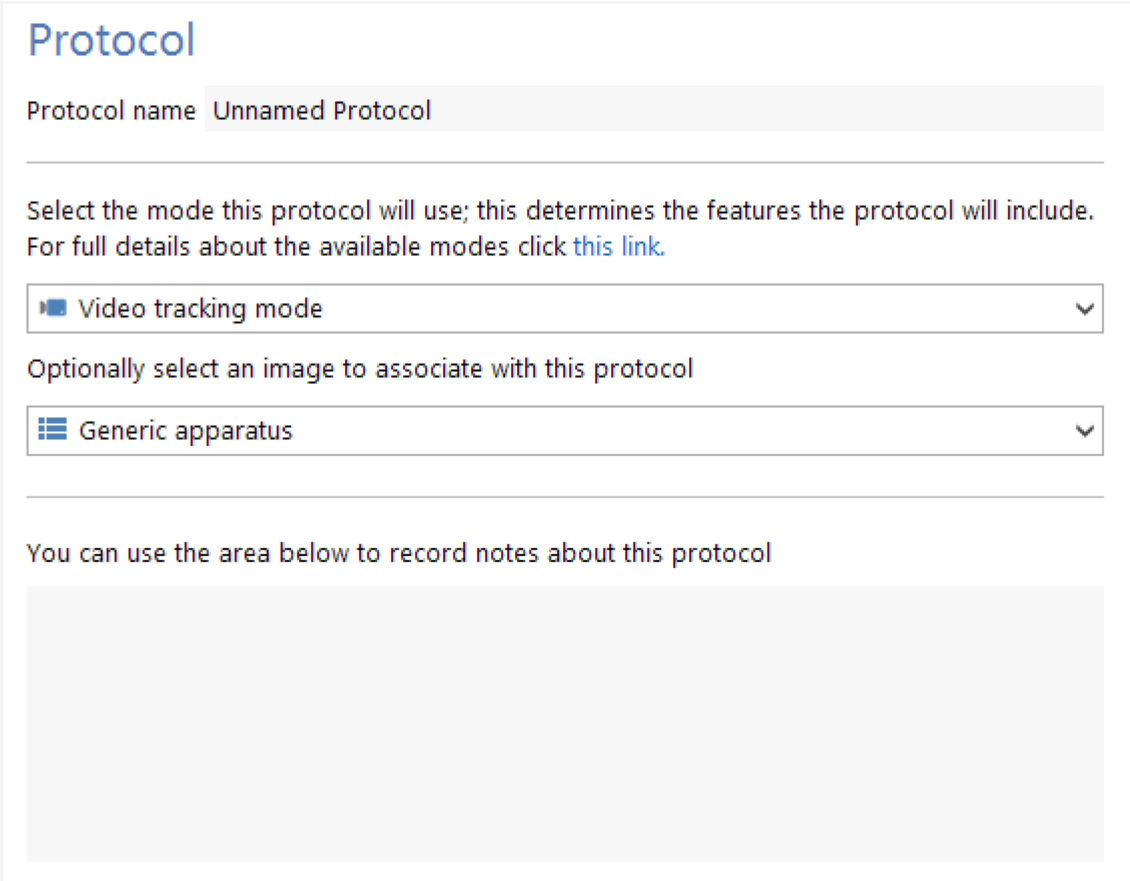


Figure 2. The 'Settings pane' shows details of the item that's currently selected in the Protocol list; in this case the Protocol element itself.

As you can see, the first field is the Protocol name, which for all new protocols defaults to 'Unnamed protocol'. You don't have to change this (although you do have to enter something), but it's usually a good idea to give the protocol a descriptive name - let's change it to 'Water-maze protocol'.

Just below the protocol name is a drop-down list which you can use to set the *Protocol mode;* opening the list will show you the available modes - see figure 3.

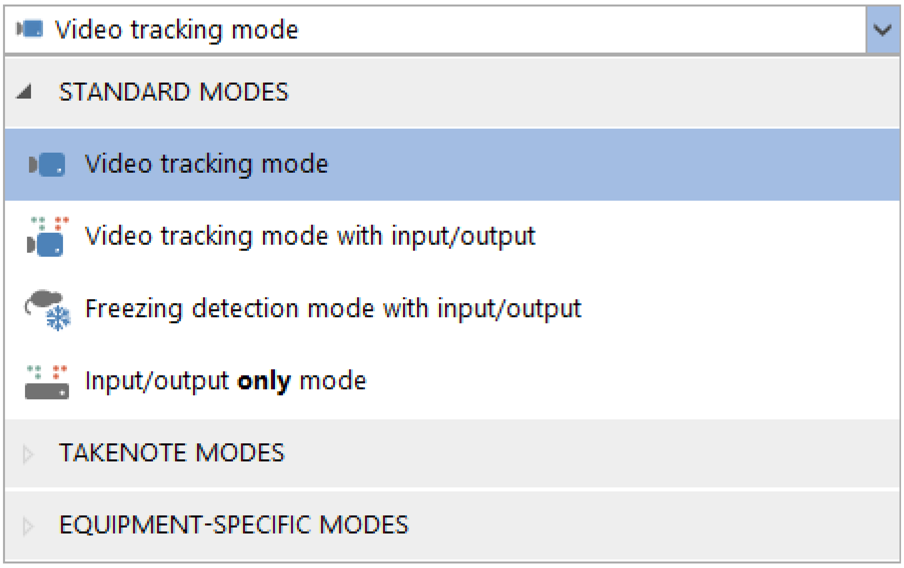


Figure 3. The Protocol mode *drop-down list. The modes that are available depend on your ANY-maze licence.*

As can be seen in the above figure, there are three groups of modes - *Standard modes, TakeNote modes* and *Equipment-specific modes.* For now, we will just consider the standard modes which essentially control what elements are included in the protocol - for example, changing to *Input/Output only mode* will remove the *Video sources* and *Sequences* elements (amongst others), as these are specific to video tracking.

The rest of this introduction assumes that the mode is *Video tracking mode*, which is usually the mode that is selected by default.

That's all we're going to do with the protocol element. Now we're ready to tell ANY-maze about our actual water-maze, and we'll start by adding a *Video source*.

Adding a video source

To add a video source, simply click the ✚ *Add item* button shown in the ribbon bar and select *New video source* from the menu which appears. In fact, you use this same technique to add any element to a protocol, so it's worth remembering how to do this - click the plus-sign button and then select the new item from the menu which appears.

Our new Video source element will now appear in the Protocol list and the Settings pane will show information about it - a video picture might also appear in the *Image* pane on the right-hand side of the screen.

OK, but what's a video source actually for? Well, as the name implies, a video source is essentially the source of a video picture but the really important thing is that it's the source of a video picture which **shows a single piece of apparatus**. This might not sound too important, but consider a situation in which you have two activity boxes placed close together - a single camera could probably see into them both. However, if we just used the video picture direct from the camera, ANY-maze would see two boxes, and presumably two animals, and this would make it rather complicated to perform tests in the two boxes independently. Video sources help resolve this and other issues related to the video pictures our camera (or cameras) can actually see. Using video sources:

- You can choose the camera, digitiser or video file from which images will be captured.
- You can crop the image, so just your apparatus is visible.
- If an image shows two pieces of apparatus, you can crop the image so just one of them is visible. You can then create a second video source, specify that it captures images from *the same camera,* and crop its image so that just the other piece of apparatus is visible. In fact, you can do this with any number of apparatus, not just two.
- You can also use video sources to glue-together images from two cameras - this is called a *montage* and is very useful if a single camera can't see all of your apparatus.

The important thing is that each of the video sources we create **shows just a single piece of apparatus** (this is important, so it's worth taking a moment to remember it) - see figure 4.

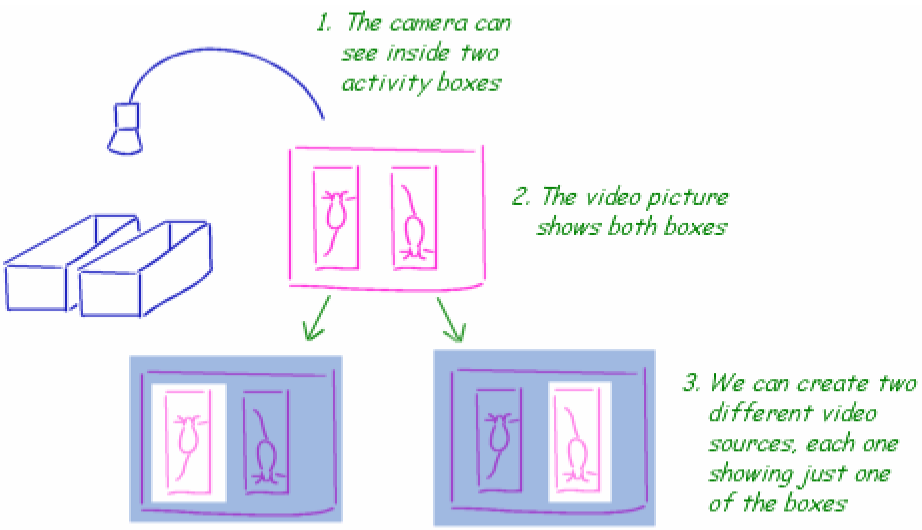


Figure 4. Using a video source, you can select just the part of a video image which shows a single piece of apparatus (the unshaded areas in the figure) - this is then the 'video source' for that apparatus.

Of course, a video source will often be extremely simple - you have a single camera, and it can see all of your apparatus and nothing else. In this case, the video source will just use the entire video picture from the camera and that's it. In fact, this is what we're going to do here.

1. In the Settings pane, change the name shown for the video source from 'New video source' to 'Water-maze'.
2. In the drop-down list titled *Source of video images,* choose the entry called

'Water-maze example image'. A picture of a water-maze will appear in the *Image pane* on the right. (If this entry doesn't appear in the list, it's because example images have been switched off in the Options > Features for new users page. To remedy this, switch the option on again.)

This example image is included within ANY-maze to help you learn how to use the system but, as it's just a static image and not a video, we won't be able to use it for tracking. In fact, you might see an entry in the list for 'Water-maze example video', if you do then you can select that instead - you'll now see a video of a water-maze playing on the right-hand side of the screen.

Ordinarily, of course, you wouldn't use a static image or even a digital video; instead, you'd use the *Source of video images* list to choose the camera or digitiser which shows the apparatus you want to use.

I'm not going to go into more details about video sources now, but if you want to learn more then you'll find full details in the Video sources topic.

So now we have a video source in our protocol which shows just our apparatus. But all the video source does is supply a video picture; it doesn't tell ANY-maze anything about what the picture actually shows - for that we need to add an *Apparatus* element to the protocol.

Adding apparatus

As I mentioned earlier, you always use the same technique to add things to a Protocol; click the *Add item* button in the ribbon bar and select the element you want to add from the menu which appears. So now; add a new *Apparatus* element.

As was the case when we added the *Video source* element to the Protocol, the new Apparatus element will appear in the Protocol list and the Settings pane will change to show details of it.

Before we do anything else, change the name of this new element from 'New apparatus' to 'Water-maze' - I imagine you're getting the idea about naming things by now!

Now take a closer look at the Settings pane, and you'll see a list titled *Video source*. If you had more than one video source (which you don't in this example), you could use this to choose the video source which shows **just** this piece of apparatus - see figure 4 above.

In our case, as we only have a single Video source in the protocol, ANY-maze will already have selected it for this apparatus and will already be showing the image from it in the Image pane on the right side of the page.

Now let's turn our attention to the *Image pane* on the right side of the page, because it's here that most of the work needs to be done. As you'll see, in the ribbon bar there's a section called *Apparatus map* which contains a number of drawing tools, and we're going to use these to draw a *map* of our apparatus on top of the video picture. If you've ever used any drawing software before then you shouldn't find this too hard to do, but I'll take you through it a step at a time anyhow:

1. Click the *Ellipse tool* button.
2. Move the mouse pointer over the top left quadrant of the video image, press the left

button down and (holding it down), drag the mouse down and to the right - an orange ellipse will appear. Try to make it roughly the same size as the water-maze, but don't worry too much because we'll move and resize it next. Release the left button.

3. Move the mouse pointer so it's pointing at any part of the ellipse you just drew (the pointer should change shape into a four-headed arrow) and click the left-hand button. Eight little boxes should appear, one in each *corner* of the ellipse and one on each pole point (if you're wondering how an ellipse can have a corner, just try it and you'll see what I mean).
4. Point the mouse at one of these boxes and click and hold down the left button. Now as you move the mouse, you'll be able to resize the ellipse.
5. If you want to move the ellipse, point at a part which *isn't* a little box and click and hold down the left button - you'll be able to move the entire ellipse.
6. Use these techniques to resize and move the ellipse until it nicely outlines the water-maze in the video picture, as in figure 5.

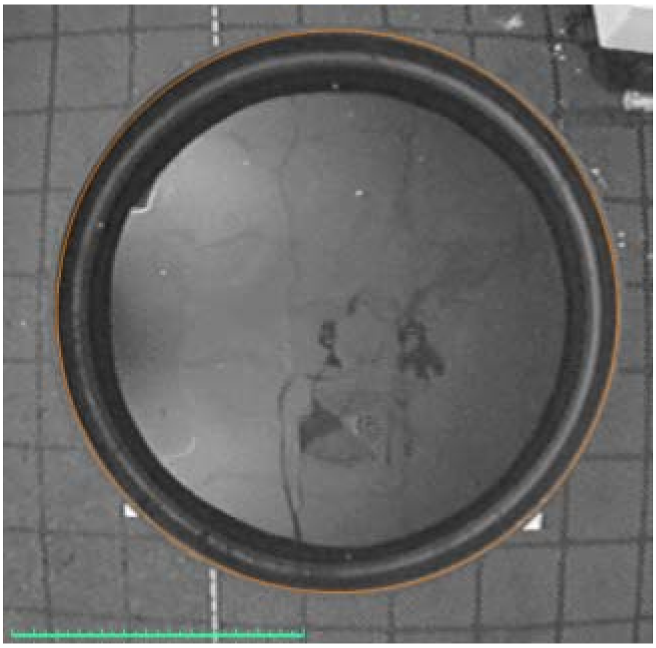


Figure 5. The apparatus map after we've drawn the outline of the water-maze.

As you might expect, the other *drawing tools* work in much the same way. Using them you can draw boxes, lines, and multi-lines. The latter will draw one line after another - I won't go into the details here, but you'll find full information in the Drawing the apparatus map topic.

One other point; if you want to remove something you've drawn, click it with the mouse (so those little boxes appear) and then press the Delete key on the keyboard.

OK, returning to our water-maze, you should have a nice orange circle drawn around the outside of the maze. This is great, but the apparatus map isn't only intended to define the edges of the apparatus; it also defines areas within the apparatus which might be of interest to us.

For example, let's imagine that our water-maze includes an island - we'd want to draw this on our map too. No problem, click the ellipse button again and draw another small circle somewhere in the maze where you think your island might be, say in the top-left quadrant. But that's not all; let's say we're also interested to know how much time the animal spends in each of the quadrants of the maze - therefore we'll need to draw these quadrants on the map so ANY-maze can analyse this too. You'd probably think that an easy way to do this would be to simply draw two lines, one horizontally and one vertically, that divide the apparatus into four - and you'd be right - but in fact there's an even easier way; you can use a grid.

Click the  *Grid settings* button and a window will appear in which you can select three types of grid. I won't go into the details; they can be found in Apparatus map grids. Just click the option to *Include a radial grid* and change the spacing from 30 to 90 degrees, then click OK - you should now have an apparatus map that looks like the one in figure 6.

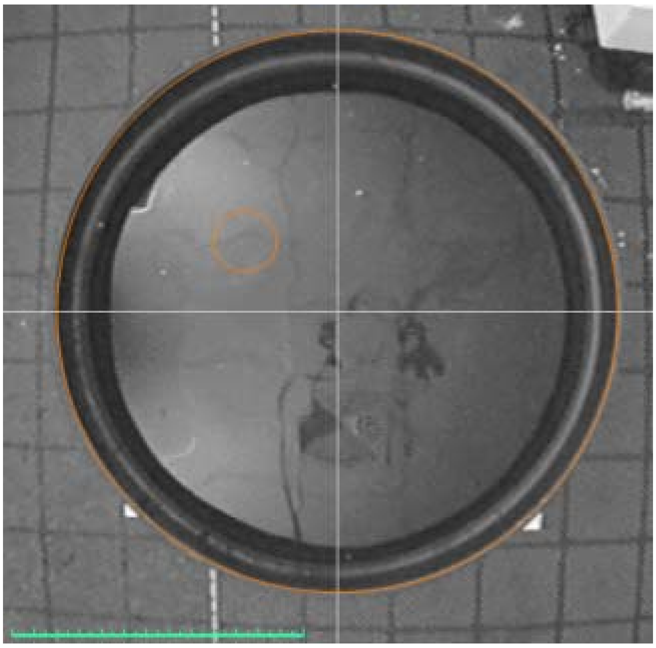


Figure 6. The final apparatus map with an island and a 90-degree radial grid.

And that's our apparatus map. Of course, you can create much more sophisticated maps than this. Indeed, once you learn more about ANY-maze you'll see that many of the system's abilities stem from the apparatus map - it's an absolutely critical part of the protocol.

Now is probably a good time to mention that almost everything in a Protocol, including the apparatus map, is editable. Not just before you run an experiment, but during and afterwards too. I won't explain all the repercussions of this, but it's good to know that what you're doing isn't cast in stone and you can come back and make adjustments if you want to.

You may have noticed a green line at the bottom of the video picture which looks like a ruler or a scale of some sort, and wondered what it's for. Well, it is a scale - in ANY-maze it's called the *drawing ruler* - and you use it to tell ANY-maze how big your apparatus is.

If you point at it with the mouse, the pointer will change to a four-headed arrow and if you click and hold down the left mouse button, you can drag the ruler around the screen. Drag it up the screen and place its left-hand end over the 'west pole' point of the ellipse you drew around the water-maze. Now release the mouse button and move the pointer to the right-hand end of the ruler - the pointer will change to something like a cross. Click and hold down the left mouse button and you'll be able to move the right-hand end of the ruler while the left-hand end stays anchored where it is - effectively, you can stretch the ruler. Stretch it until the right-hand end is over the 'east pole' point, and release the mouse button - your map should now look like figure 7.

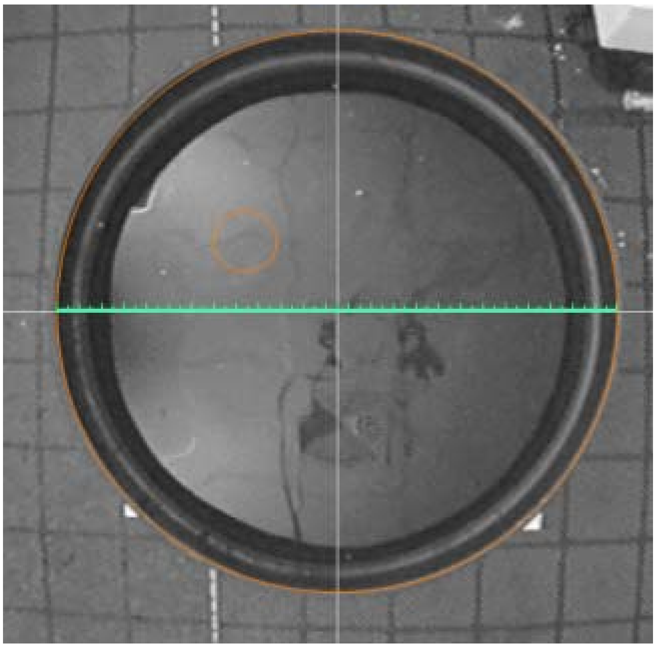


Figure 7. The apparatus map showing the drawing ruler *positioned across the diameter of the maze.*

Fine, but what does this *drawing ruler* actually do? Well, what you've done is to place the ruler across the diameter of the water-maze, which is a distance you'll probably know. Now in the Settings pane you need to enter this distance, in millimetres, in the field at the bottom of the page titled *The length of the 'ruler' line is.* Our water-maze is 1.2m across, so enter '1200'. Now that ANY-maze knows that the distance represented by the drawing ruler is 1200mm, it can calculate any other distance it needs. Thus all distances and speeds in ANY-maze will be reported in metres and metres per second, respectively.

In summary, you should position the drawing ruler along a known distance in the apparatus map, and then enter that distance in the Settings pane. One word of warning, however - make sure the ruler is measuring something that's in the same horizontal plane as the animals will be. For example, in an elevated plusmaze, measure the length of an open arm rather than measuring something on the floor of the room (which will be lower than the maze), because as you get further from the camera so the distances will appear smaller.

Enough about the apparatus! There are other things you can set, but we won't worry about them for now. In fact we're now close to having the essentials of a working Protocol; we just need to do a couple more things - the first is to specify the animal colour.

Specifying the animal colour

ANY-maze tracking is essentially automatic - you don't need to train it how to track animals, nor do you need to set any thresholds or other complex values. Nevertheless, there is one thing the system needs to know - the colour of the animals you intend to track.

The animal colour is specified using the protocol's Animal colour element, which is the fifth element in the *Tracking* group - select it. As usual, the Settings Pane will change, in this case to show the different options available for the animal's colour.

In fact, ANY-maze isn't exactly interested in the *colour* of the animal, what it really needs to know is whether the animal is lighter or darker than the background of the apparatus. In our water-maze, the animals are lighter so you should select this option in the Settings pane (you'll find more details about the other *animal colour* options in the Animal colour topic).

And that's the only setting that you have to make in order for ANY-maze to track your animals.

Setting the test duration

The final step in setting up our simple protocol is to specify the duration of the tests. This is done using the Stages element of the protocol, which is the first item in the *Testing* group.

This element is so named because you can use it to set up different *stages* in a single experiment; however, in our example we're going to create just a single stage.

In fact, as you'll find if you click the *Stages* element in the protocol list, ANY-maze automatically includes one stage in all protocols when it creates them. This *default stage* is called 'First stage', although you can of course edit this to something more appropriate if you wish.

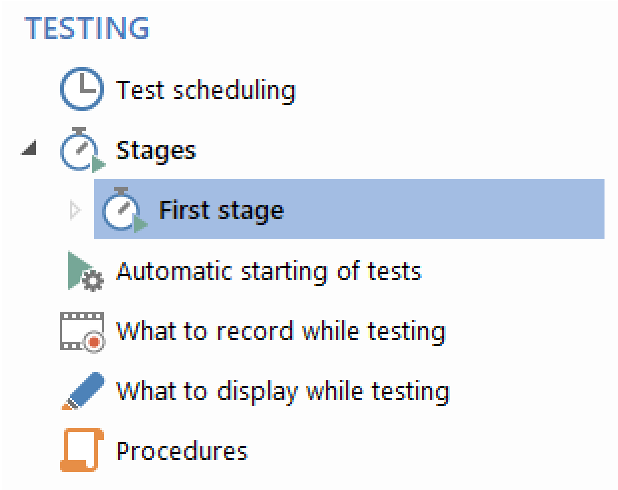


Figure 8. All protocols are automatically created with a default stage called 'First stage'.

Click this element in the protocol, and the stage's details will be displayed in the Settings pane.

As you'll see, apart from the stage's name, there are essentially two things to set - the test duration and the number (and order) of trials in the stage. For our simple protocol, we're going to have just a single trial (i.e. each animal will only ever be tested once), so the default value is fine, but we do need to set the *Test duration*. Our protocol is going to use 5 minute tests, so you can simply enter '5 min'.

Throughout ANY-maze, times can be specified using units of 'ms' for milliseconds, 's' for seconds, 'min' or 'm' for minutes, 'h' for hours and 'd' for days. You can mix these units if you wish, for example '3h 30min', and you can use decimal points, for example '3.5h'.

Finally, we're going to move on to the last compulsory element, *Treatment groups*.

Treatment groups

Despite its name, the Treatment groups element of the protocol is *not* where you assign treatments to your animals. Rather, it is where you can choose whether or not you want your animals to be put into treatment groups and, if you do, how the animals will be *assigned* to the groups. This is all explained in detail here, but for now you can just choose the last option, *Randomly assign the animals amongst the groups* (which is quite likely to be what you'll want to do anyway).

Finished - a simple but complete protocol

We're done! You've now created a simple protocol which includes everything ANY-maze needs to know to actually run an experiment in our water-maze.

To recap:

- In the *Protocol* element, you specify the *protocol mode* you want to use.

- In the *Video source* element, you tell ANY-maze where the video images are coming from and, when necessary, you select just the part of the image which shows a single piece of apparatus.
- In the *Apparatus* element, you select the *Video source* which shows your apparatus and then draw an apparatus map over the top of it. You also use the *drawing ruler* to measure a known distance in the apparatus, so ANY-maze can work in real-world units such as metres and metres/second.
- In the *Animal colour* element, you need to specify the *colour* of the animals you intend to track.
- In the *Stages* element, you need to at least set the test duration of the default stage to the correct duration for your tests.

Of course, as you can see in the Protocol list, there are many other Protocol element,s each of which adds certain abilities to the system (for example, in our water-maze we'd probably want to add a Zone to define the island) but they all follow the same structure and methodology as the elements we've looked at here.

See also:

- The elements of a protocol

The elements of a protocol

Introduction

Protocols are divided into a number of different elements, each of which defines a specific feature. There are 44 elements in total, but in the simplest protocol you only *have to* define six of them. Specifically, the elements are:

Apparatus

The elements in this group define the apparatus in which ANY-maze will be tracking animals.

Protocol settings	Defines overall features of the protocol such as its name, and most importantly, the protocol mode.
Video source	You can add as many individual video source elements to the protocol as you like. Each one defines the video picture of a single piece of apparatus. Generally, you must have at least one Video Source element in a protocol otherwise you won't have any video picture to track in!
I/O device	You can add any number of I/O device elements to a protocol; each one defines a specific I/O device you intend to use. This will usually be something like an ANY-maze interface, but it may be the actual apparatus the animal will be tested in, such as an OPAD cage. Anyhow, the important thing is that if you want to read inputs or control outputs, you must include an I/O device element for the device the inputs and outputs are part of.
Apparatus	You can include up to 40 individual apparatus elements in a protocol. Each one defines a single piece of apparatus in which you will track animals during your experiment. For example, you could define four activity boxes and then track animals simultaneously in all of them. All protocols must include at least one apparatus element.

Tracking and behaviour

The elements in this group allow you to set up a range of tracking-related options as well as defining keys which can be used to score behaviours that ANY-maze can't detect automatically. Note that the group's name is usually *Tracking and behaviour,* but changes to just *Behaviour* when a non-tracking protocol mode is being used.

Zones	You can include as many individual zone elements in a protocol as you like. Each one defines a part of the apparatus which is of interest to you. For example, in a plusmaze the open arms would be one zone, and the closed arms another. You can also define zones which move about between tests and/or between animals, such as an island in a Water-maze or a reward arm in a T-maze. You don't have to include any zones in a protocol if you don't want to.
Points	You can include as many points in a protocol as you want. Each one defines a specific position within the apparatus that is of interest to you - for example, the position of a novel object.
Sequences	You can include as many sequence elements in a protocol as you like. Each one defines a sequence of movements around the apparatus which an animal can perform. For example, in a water-maze you could define a sequence to count rotations around the maze and another to count centre crossings. You don't have to include any sequences in a protocol if you don't want to.
Animal colour	ANY-maze tracking is essentially automatic - you don't need to set any thresholds or other complex values. Nevertheless, there is one thing the system needs to know - the colour of the animals you intend to track - and you set this here.
Tracking the animal's head...	ANY-maze will always track the animal's centre point; you can use the options here to also track the head and tail.
Immobility detection	In some experiments you'll want to determine when the animal is

mobile (i.e. moving from place to place) vs. when it is immobile. This element provides options to do this.

Freezing detection

ANY-maze can detect when an animal freezes (no movement except respiration) which, amongst other things, is very useful in fear conditioning experiments. This option allows you to switch freezing detection on and to fine-tune it.

Tracking options

This element contains various advanced options related to tracking; these won't be required for most protocols.

Keys

You can include up to 46 key elements in a protocol. Each one defines a key on the computer's keyboard which you can press to register behaviours that ANY-maze can't score automatically. For example, you could define one key to record Grooming and another to record Rearing. You don't have to include any keys in a protocol if you don't want to.

Inputs and outputs

The elements in this group take ANY-maze beyond just *tracking* animals:

Electrophysiology synchro...

Used to output TTL signals to synchronise tests in ANY-maze with an electrophysiology recording system.

On/off Inputs

On/off input elements can be used to detect things like lever presses or photobeam breaks.

Rotary encoders

As the name implies, rotary encoders detect rotations - this might be rotations of a running wheel, or could even be rotations of a tethered animal.

Signals

Signals allow ANY-maze to capture analogue data from devices such as ECGs, gas analysers, etc.

Sensors

ANY-maze supports various sensors, such as temperature sensors, humidity sensors and weight sensors (i.e. scales).

Movement detectors	This element is only included in protocol modes which do not include tracking. A movement detector is able to determine when an animal is moving vs. when it is stationary, without using a video camera. (Note that movement detectors do not track the animal).
Output switches	Output switches allow ANY-maze to *control* devices; for example, an output switch could be used to control a pellet dispenser or a cue light.
Speakers	The speaker element provides a way for ANY-maze to play sounds, including fixed frequency tones, white noise, or any sound file.
Shockers	Through the shockers element, ANY-maze can control the onset, duration and also the *intensity* of shocks delivered during a test.
Pellet dispensers	Pellet dispensers element allow you to control pellet dispensers during tests. For example, dispensing a pellet as a reward in an operant conditioning experiment.
Laser controllers	Laser controllers can be used for optogenetics to control lasers.
Syringe pumps	The syringe pumps element gives ANY-maze the ability to control the direction and flow rate of a wide range of syringe pumps from manufacturers such as KDS, Harvard and New Era.
Analogue outputs	Analogue outputs can be used to control external equipment based on a voltage which ANY-maze generates. The voltage can be directly related to the animal's position, or can be any value based on what's happening in a test.
Temperature controllers	As the name implies, temperature controls give ANY-maze the

ability to control the temperature of something, for example, a thermal stimulus or the temperature of the apparatus floor.

Lighting controllers

Lighting controllers allow ANY-maze to control the intensity of lights - they're like a dimmer switch that ANY-maze can set.

Testing

The testing group elements are used to manage the actual process of performing tests:

Test scheduling

By default, ANY-maze will use the information you specify in the protocol to automatically build a Test schedule for you. Sometimes this may not be what you want, in which case you can use the option here to manually schedule your tests instead.

Stages

All protocols always include at least one stage but you can include up to 45 if you like. Each stage defines a series of trials in an experiment. For example, in a plusmaze experiment, you will probably have a single stage consisting of a single trial, whereas in a water-maze experiment you might have two stages - 'Training', consisting of perhaps 4 trials, and 'Probe', consisting of a single trial.

Within stages, you can also define stage-end rules. These are rules which can cause a stage to end automatically for an animal (i.e. the animal will have no more trials in the stage). They're particularly useful in 'training' stages; for example, 'choosing the correct arm of a T-maze in 3 consecutive training trials' might be a stage end rule.

Automatic starting of tests

When tracking, ANY-maze can automatically start tests when it sees you, the experimenter, leave the camera's view. This element allows you to set up this option.

What to record while testing

Obviously, ANY-maze will record a test's results (which include such things as the animal's position), but this element allows you to both fine-tune how often results are recorded and also to optionally specify that tests should be video-recorded too.

What to display while testing

While a test is actually running, ANY-maze can show you a wealth of information such as the location of the animal's centre, head, and tail; what zones the animal is in; the state of I/O devices; etc. This element allows you to choose what information you want displayed.

Procedures

Procedures are used to detect specific situations during a test and to take some action when they occur. For example, you could use a procedure to end a test when the animal enters the Island zone in a water-maze. Procedures are especially useful with I/O devices, for example, a procedure could determine when an animal presses a lever and decide (based, if necessary, on complex criteria) whether to reward the animal with a food pellet or generate an aversive shock.

Events and actions

Events and actions have been replaced by Procedures. They are documented here as they can still be used in legacy experiments.

Like procedures, events and actions are used to detect certain situations during a test and to take some action when they occur. Specifically, an event determines when something occurs and an action does something because of it.

Additional information

The additional information group provides options to record and utilise additional information about your animals and tests.

Treatment groups

Allows you to choose whether or not you will organise your animals into Treatment groups and, if you do, to specify how ANY-maze should assign animals to the groups.

Animal ID

ANY-maze gives all the animals in your experiment unique identifiers (IDs) called *animal numbers*. However, your animals probably already have IDs that you use in your lab - the options here allow you to tell ANY-maze that you want to use your own IDs rather than its animal numbers.

Animal weights

Allows you to choose whether to record the weight of your animals and whether you'll do this manually or using the ANY-maze Animal scales.

Treatment doses

If you choose to record the weights of your animals in ANY-maze (see previous item), then ANY-maze can automatically calculate the treatment doses for the animals.

Fields

You can include up to 24 different field elements in a protocol. Each one can define a new *entry field* in which you can record information about individual animals or tests. For example, you could include fields to record the weight or sex of the animals, or to record the light level or temperature in each test. ANY-maze can analyse field entries and can also use them as independent variables in statistical tests. You don't have to include any fields in a protocol if you don't want to.

Analysis and results

The last group of protocol elements provide powerful options for analysing and reporting on the results gathered during an experiment.

Virtual switches

Virtual switches can be used to determine on/off type events based on other information ANY-maze already knows. For example, a virtual switch could be used to determine when the animal is moving faster than a certain speed - when it is, the 'switch' will be on, when it isn't, the 'switch' will be off. Virtual switches can also be used to determine when two or more on/off type events occur - for example, when two specific behaviours detected are *both* occurring.

Calculations

You can include as many calculation elements in a protocol as you like. Each one defines a calculation which derives new a measure from those already defined by ANY-maze. For example, in a plusmaze you would probably define an 'Open arms' zone, and this would automatically cause ANY-maze to report a *Time in open arms* measure. You could then use a calculation to derive a 'Percentage time in open arms' measure from this.

Calculations can also be used to derive results from multiple tests, for example an average result across a number of trials, or to derive results for segments of a test; for example, distance travelled in the first 2 minutes of a 5 minute test. You don't have to include any calculations in a protocol if you don't want to.

Time segments

Time segments allow you to divide up a test so you can perform analysis across time. Tests can either be divided into equal duration segments, for example, segments of 1 minute each, or they can be divided into unequal segments which can even be based on the occurrence of something in the test, for example, a segment might start 10 seconds before the animal enters a certain zone and end 20 seconds afterwards.

By default, all tests will be divided into equal duration segments of 30 seconds and there's no need to do anything if you either don't need time segments or the default setting suits you.

Analysis

Each protocol always includes a single analysis element, which defines various options related to how ANY-maze analyses test results.

Charts

Charts are used to show a plot of some value across time. A wide range of values can be plotted, including such things as the animal's speed, its distance from a zone, the value being read from some I/O device, etc. Charts can be viewed while a test is running (providing live feedback of the value) and can be viewed, copied, saved or printed post-test.

Results, reports and data

A protocol always includes a number of standard report elements which define the information shown in the experiment management reports. However, you can also add your own elements to define new results reports - for example, in a plusmaze protocol, you could define a report which would use an ANOVA to analyse the 'Percentage of time in the open arms' between the different treatment groups.

See also:

- Adding elements to a protocol
- Editing the elements of a protocol

- Deleting elements from a protocol
- Saving and loading protocols
- Viewing the protocol report

ANY-maze help topic H0081

Protocol settings

Introduction

For a general introduction to protocols, see An introduction to Protocols.

You can use the Protocol settings element to specify the name of your protocol and, most importantly, to set the Protocol mode. You can also choose an image to represent the protocol and record some notes.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the protocol's name
2. Select the protocol's mode
3. Optionally select an image to represent the protocol
4. If necessary, record some notes about the protocol

What next?

After completing these steps, you should consider whether you want to include any Video sources in your protocol.

See also:

- Editing elements of a Protocol

Protocol name

In brief

Enter a name for the protocol in the *Protocol name* field in the protocol's settings pane. You must make an entry, but it can be anything you like.

Details

It's a good idea to enter a concise and meaningful name such as '4 trial water-maze' or 'Rat plusmaze' - something which both describes the protocol and differentiates it from other similar protocols.

If you choose to save the protocol, ANY-maze will use the name you enter here as the *default* file name for the protocol file - of course, as it's only a default, you'll be able to change it.

Limits

You can enter anything you like up to a maximum of 32 characters.

The protocol mode

Introduction

The protocol mode alters how ANY-maze operates. It does this in three ways:

- It limits the protocol elements included in the protocol list to those which are relevant to the mode in use. For example, when *Input/Output only mode* is selected, all the protocol elements used for video tracking are removed.
- It adjusts the features available within ANY-maze depending on the type of licence you own.
- It provides short-cuts to building protocols for ANY-maze specific equipment

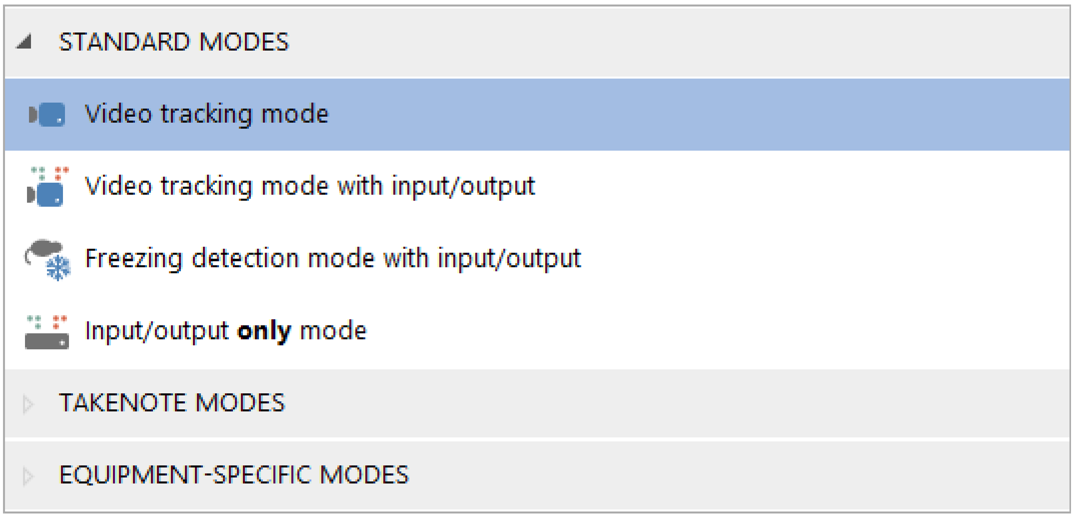


Figure 1. The protocol mode list. Modes are grouped into three categories, clicking a category title opens it to show its modes.

Full details about the protocol mode are given in the sections below:

- The different protocol modes
- Setting the protocol mode
- How should I choose which protocol mode to use?
- How licences affect the available modes

The different protocol modes

There are four *Standard* protocol modes, three *TakeNote* modes and some *Equipment-specific* modes.

Standard modes

Video tracking mode	This is the default mode, and provides all the features needed to video-track animals within your apparatus.
Video tracking mode with input/output	This is the most complete mode and enables all the features of ANY-maze, including those used to interface to input/output devices, such as photobeams, shockers, etc. This mode is appropriate when you not only want to track the animal within your apparatus, but also want to record the input from some type of I/O device, such as a micro-switch or photobeam, and/or control some sort of output such as a speaker, shocker, lamp etc.
Freezing detection with input/output	This mode is much like the *Video tracking with input/output* mode except that it doesn't actually *track* the animals, rather it only detects when they are freezing. This mode is appropriate for tests such as fear conditioning, where you are interested in detecting freezing but you don't need to track the animal's position. Note that in *Freezing detection mode* you will still be using video, and so you will still need to set up Video sources and you will still need to include Apparatus items in your protocol. You will also need to draw the *outline* of the apparatus map, so ANY-maze knows where in the video image it should be detecting freezing. When you perform tests in this mode, ANY-maze will show the video image of your apparatus (although it won't track the animal).
Input/output only mode	In this mode, ANY-maze will not use a video image of your apparatus at all. This is appropriate when you simply don't need to see an image of the animal, but instead you want the system to record and/or respond to inputs and control external apparatus via outputs. This mode is appropriate in situations such as the automation of an operant chamber -

where you could use ANY-maze to detect lever presses or nose pokes, and control the chamber's loudspeaker, shocker, food hopper, light, etc.

When a protocol is using this mode, all options related to video and video tracking will disappear from the protocol list - this includes Video sources, Zones, Points and Sequences.

Note that in this mode you will still need to create Apparatus items in your protocol, although you won't have to select a video source for them. During testing, ANY-maze won't show a video picture of your apparatus; in its place it will show the state of the various inputs and outputs included in your protocol.

TakeNote modes

In the TakeNote modes, ANY-maze does not track the animal; rather *you* will observe the animal and score its behaviour using keys.

TakeNote video observation mode

In this mode, ANY-maze will not track the animals in the video picture of the apparatus; rather, you will use keys to score the animal's behaviour manually. This mode is appropriate when you either don't care where the animal is within the apparatus (because you just want to score what it is doing), or your apparatus is such that the video tracking won't work.

When a protocol is using this mode, options that require video tracking will disappear from the protocol list - this includes Zones, Points and Sequences as well as the options to detect immobility and freezing, and those to track the animal's head and to automatically start tests. Furthermore, you won't need to draw any apparatus maps, but all other features of ANY-maze will still be available.

Note that in *TakeNote video observation mode* you will still be using video, and so you will still need to set up video sources and you will still need to include Apparatus items in your protocol. When you perform tests in this mode, ANY-maze will show the video image of your apparatus (although it won't track the animal) and you will be able to press keys to score behaviours manually.

TakeNote video observation mode with input/output

	This is similar to the previous mode, except that features used to control input/output devices, such as photobeams, shockers, etc. will be available. This mode is appropriate when you want to both observe the animals and record data from inputs and/or control outputs.
TakeNote direct observer mode	This is similar to the *TakeNote video observation mode* except that you won't be observing the animal in a *video picture,* but rather observing it *directly* (i.e. you'll be looking directly at the animal). In this mode, all elements relating to video and to tracking will disappear from the protocol list - this includes Video sources, Zones, Points, etc., but you will still need to include Apparatus items in your protocol (the apparatus being where the animal that you're observing is). This mode is appropriate when you want to observe animals without using video - for example, you might just be sitting in a lab watching the animals in a cage, or you could be observing them in the field.

Equipment-specific modes

As the name implies, the equipment-specific modes are used with specific pieces of equipment, such as the OPAD cage.

ANY-maze includes various *Equipment-specific modes* such as OPAD cage mode and RAPC mode. These modes are usually selected automatically based on your ANY-maze licence.

As well as limiting the elements included in the protocol, these modes also make it easy to build your protocol because the system knows the features that the equipment provides. For example, an OPAD cage includes (amongst other things) two touch switches, one of which is activated when the animal makes contact with the 'thermal elements' and the other when it licks the spout of the water bottle. Because ANY-maze knows this, it means that when you add an OPAD cage into your protocol, the system can automatically include appropriate on/off inputs for these items.

In fact, the equipment-specific modes not only automatically build the protocol, but they can also cause the protocol to include some elements which are only relevant to the equipment whose mode is chosen. For example, when an OPAD cage mode is selected, the *Analysis* element of the protocol includes a sub-element for *Analysis of OPAD results.*

As mentioned above, the equipment-specific modes limit the elements included in the protocol to those that are relevant to the specific equipment. However, you may encounter situations in which you want to use other features of ANY-maze in an equipment-specific mode - for example, you might want to track the animal within the OPAD cage, but this won't be possible because all the protocol elements related to tracking will have been removed from the protocol. There is, however, a solution to this: if you own a *full* or *I/O only* ANY-maze licence, then when you select an equipment-specific mode a second list will displayed where you can select a *sub mode.* The sub modes allow you to switch on additional features of ANY-maze whilst remaining within the equipment-specific mode - for example, you can remain in OPAD mode, but switch on the features for video tracking. Further details about sub modes are given in the next section.

Setting the protocol mode

To set the protocol mode, you should first select the Protocol element in the protocol list (it's the first item) and then select the mode you want from the drop-down list shown in the *Settings pane*.

The default mode, which ANY-maze will select automatically when you create a new protocol, will depend on your ANY-maze licence. For example, if you own a licence for the OPAD cage, then the default mode will be *OPAD cage mode*. On the other hand, if you own a TakeNote licence, then the default mode will be *TakeNote video observation mode*. But it's most likely that you own a Full licence (or no licence, because you are trialling ANY-maze) in which case the default mode will be *Video tracking mode*.

To change the mode, simply open the list and select a new mode from those shown. You may find that some modes are disabled; this means that your licence doesn't allow you to use those modes. For example, if you own a TakeNote licence, then the *Video tracking* modes will be disabled.

While you are designing a protocol, you can change the mode at any time. So you might start off intending to use *Input/output only mode* but then decide that you would like to track the animals too, and so change to *Video tracking mode with input/output*. However, after you start the experiment - that is, after you have performed at least one test - you can no longer alter the mode. This is a hard rule and there is no way to work round it.

If you own a *full* or *I/O only* ANY-maze licence, then when you select an equipment-specific mode, a second list will displayed where you can select a *sub mode*. The options shown will be:

- Standard sub mode
- I/O sub mode
- Full sub mode

⚠ *WARNING: Changing the sub mode causes the protocol to be cleared - that's to say, all the entries in the protocol are deleted - so it's a very good idea to select the correct sub mode before you start setting up the rest of the protocol.*

Standard sub mode

This is the default sub mode, and will cause the protocol to only include the elements that are required to use the relevant equipment. For example, if you are using OPAD thermal mode, then the standard sub mode will include protocol elements for *On/off inputs, Temperature controllers,* etc. as these are inputs and outputs that OPAD includes; but it won't include such things as *Switch outputs, Zones* or *Points,* as these aren't required to run OPAD tests.

This sub mode is the best one to use if your experiment doesn't actually require any of the extra features that the other sub modes provide, as it keeps the protocol simpler.

I/O sub mode

In this sub mode, *all* the input/output elements are included in the protocol. For example, normally when using an OPAD mode, the protocol will include some input/output elements (being those that are required for OPAD) but it won't include such things as *switch outputs, signal inputs, syringe pumps* etc. Selecting the *I/O sub mode* changes this, so that all the input/output elements are included, allowing you, for example, to use a syringe pump with OPAD.

This sub mode will only be available if you have a *full* or *I/O only* ANY-maze licence.

Full sub mode

In this sub mode, *all* the protocol elements are available. This means that you can use all the features of ANY-maze. For example, if you are using OPAD mode, then selecting this sub mode would allow you to use any input or output, to track the animal, to detect freezing, etc.

This sub mode will only be available if you have a *full* ANY-maze licence.

How should I choose which protocol mode to use?

If you own a full licence for ANY-maze, then all the protocol modes will be available to you, so how should you choose the one to use? The answer depends on what you're trying to do, but there are a couple of simple rules you can apply:

- Choose a mode which includes all the features you need. This sounds rather obvious, but clearly, if you want to track your animals then you must use one of the *Video tracking* modes.
- Choose a mode which *only* includes features you need. This is a little less obvious, but consider this scenario: You want to score how often an animal in a skinner box presses a lever; which mode should you use? Well, you need an input so applying the first rule, you will need a mode that includes Input/Output. But there are four of those, so which of the four should you use? Applying this rule you would choose the *Input/output only mode*, because you don't need to track an animal or detect freezing (features the other modes provide). It's useful to understand why this is important: consider what would happen if you chose the *Video tracking mode with input/output* - ANY-maze would expect you to track the animal, so it would expect you to set up a video source, draw an apparatus map, etc., but as your apparatus doesn't include a camera, you wouldn't be able to do this.

Of course there's one other rule - if you're using a piece of equipment which has its own mode, such as an OPAD cage, then you should use the equipment's specific mode.

As mentioned above, if you own a *full* or *I/O only* ANY-maze licence, then when you select an equipment-specific mode, you will have the option to choose a sub mode too. You should choose the simplest sub mode which provides the features you need for your experiment, as this will make the protocol simpler. For example, if you don't need any additional input/output (over and above that which the equipment itself provides) then you should choose the *Standard sub mode*. However, if you need some additional inputs or outputs then you should choose the *I/O sub mode,* and only if you need to track the animals should you choose the *Full sub mode*.

How licences affect the available modes

As you probably know, to actually record the results of tests performed in ANY-maze, you need an ANY-maze licence. The most common licence is a 'Full licence' which causes all the protocol modes to be available. But there are also (cheaper) licences available such as the 'TakeNote' licence, which makes the TakeNote modes available but which won't allow you to select any of the *Video tracking* modes.

Some ANY-maze licences are apparatus-specific; for example, the OPAD cage is provided with a copy of ANY-maze, but the licence is specific to the OPAD cage. Similarly, the RAPC apparatus is provided with a RAPC-specific licence. These licences only enable their respective *Equipment-specific* modes.

Selecting an image to represent the protocol

In brief

The Startup page and the New experiment page both show lists of recently used protocol 'documents' on their right. Selecting one of the documents in these lists creates a new experiment with the specified protocol already loaded into it.

As can be seen in figure 1 (below), these documents include an image which makes it easier to identify a protocol quickly. The image that's shown is set as part of the protocol element's settings.

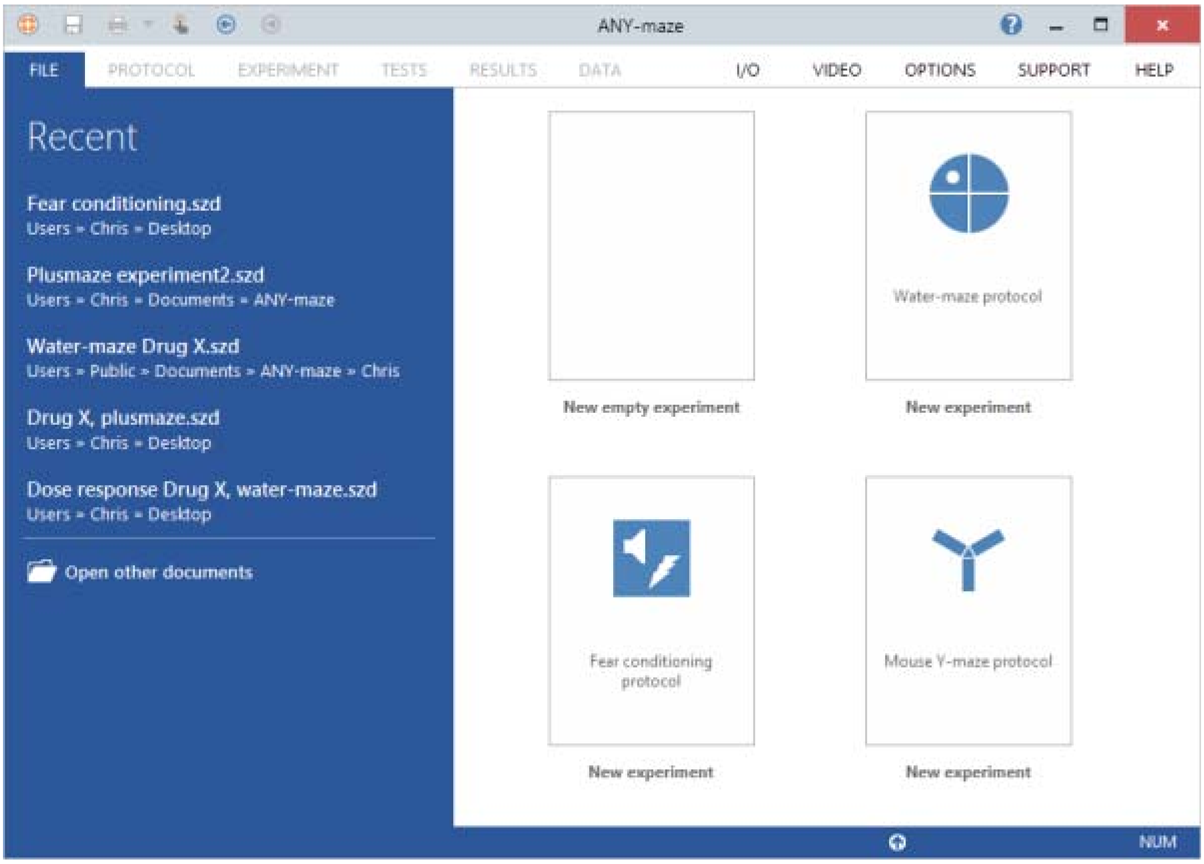


Figure 1. The Startup page lists recently accessed protocols on the right, and these protocol 'documents' are each shown with an image in them. These images make it easier to quickly identify the protocols.

Details

To set the protocol image, you should first select the Protocol element in the protocol list (it's the first item) and then select the image you want from the drop-down list shown in the *Settings pane*.

Figure 2. Some of the protocol images available.

There's quite an extensive range of images available (see figure 2) and you can choose any one you want. Although the images have 'names', for example *Elevated plusmaze,* only the image is actually used, so you should choose an image that best represents your apparatus, without worrying about what it's called. For example, if you have a piece of apparatus that you've created which is in the form of a cross, then you might choose to use the *Elevated plusmaze* image, even though the apparatus is not elevated and is not being used as a traditional plusmaze.

A protocol does need to have an image, but if you don't want to select one, you can just leave it set to the default of *Generic apparatus.*

Recording protocol notes

In brief

If you want to, you can record notes about a protocol in the notes field on the Protocol's settings page - there are no limits to what you can enter.

Details

You may wish to use the protocol notes to record a brief synopsis of the protocol, together with a list of any changes you make. This can be particularly useful if the protocol is to be shared between different users.

Formatting

Since the protocol notes field is designed to help out other users of ANY-maze, you might want to format the text in such a way as to make it easier to read. For example, you might want to underline some titles, or make sections of the text bold, or a different colour (or both!).

When the protocol notes field is currently active (i.e. the cursor is in that field), then the *Protocol notes format* section of the ribbon bar will be enabled. This contains a number of options which apply to any text that is currently selected in the notes field. (If no text is currently selected, then the formatting you select will apply to new text when you start typing).

Reset formatting	Resets the formatting of the selected text to the default formatting (i.e. removes any colour, bold, italic, etc.)
Text colour	Opens a dialog box allowing you to select a colour for the selected text.
Bold	Makes the selected text **bold.**
Italic	Makes the selected text *italic.*
Underline	Makes the selected text <u>underlined.</u>
Increase text size	Increases the size of the selected text by one point size.
Decrease text size	Decreases the size of the selected text by one point size.

Limits

There are no limits to what you can enter.

Video sources

Video sources are a fundamental protocol element, and any protocol which uses a video-based protocol mode must include at least one. A video source's job is to provide a video image of your apparatus.

For more information about video sources, refer to the topics below.

- An introduction to video sources
- Setting up a video source
- Setting up a montage video source
- Editing a video source
- Deleting a video source

An introduction to video sources

Introduction

A video source is something which, as the name implies, acts as the source of video images. The important thing about a video source is that it shows an image of *a single piece of apparatus*.

- Why video sources are necessary
- Every apparatus needs a video source
- Working with apparatus which can't be viewed with a single camera
- How montage video sources work

Why video sources are necessary

The easiest way to understand why video sources are necessary is to consider a simple example. Imagine you have two activity boxes and you'd like to track in them both at the same time. Of course, you could do this using two cameras, one over each box, but you might find that if you place the two boxes next to each other just one camera could see into them both - see figure 1.

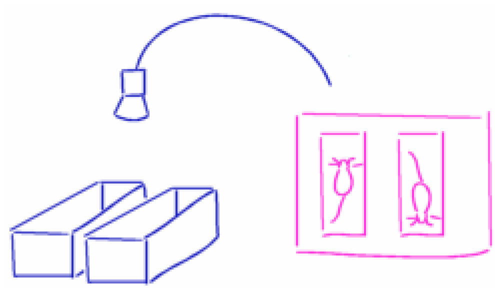


Figure 1. One camera can see into both activity boxes creating a cheaper solution than using one camera for each box.

Obviously, using one camera would be a cheaper solution than using two - however, there'd be a problem. The picture from the single camera would show two pieces of apparatus, but for ANY-maze to track animals in two boxes, it needs two distinct images (one of each box) - this is where video sources help out. In this example, you would create two video sources; they'd both use the picture from the same camera but each one would use a different part of the picture as the 'image' of its apparatus - see figure 2.

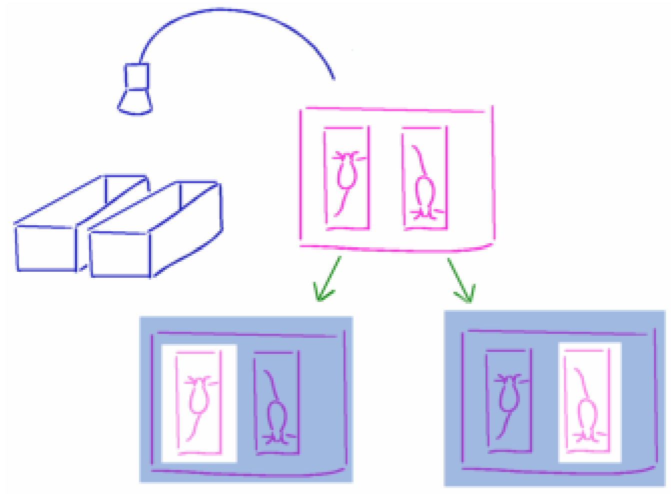


Figure 2. The two video sources share the picture from the camera, but each one only uses the part of the picture which shows its apparatus (shown unshaded in the above figure).

In fact, you're not limited to two pieces of apparatus; you can use the same technique to divide up images showing up to 40 pieces of apparatus if you want to (although that would be rather extreme).

Every apparatus needs a video source

You might think that if your camera shows just a single piece of apparatus, then you won't need a video source at all - however, this isn't the case. ANY-maze always expects the image of your apparatus to come from a video source, so you'll still need to create a video source for your apparatus even if it's the only thing in the picture. In fact, this is quite useful because it means you can crop any part of the image which you don't really need - see figure 3.

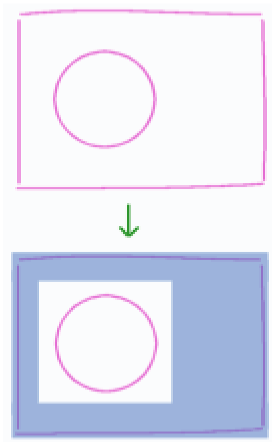


Figure 3. The water-maze doesn't fill the video picture, so the video source uses

just a part of the picture, reducing the amount of information ANY-maze has to process.

Removing excess parts of an image is a good idea, as it reduces the amount of information that ANY-maze has to process making it run faster.

Working with apparatus which can't be viewed with a single camera

In some circumstances, you might find that your apparatus can't be fully viewed with a single camera, this occurs most often in apparatus which is in some type of enclosure and/or includes some type of dividing wall, see figure 4 - it might also occur if your apparatus is simply very large.

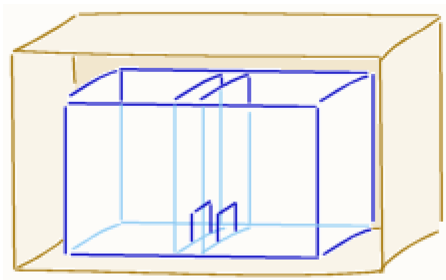


Figure 4. Using one camera, it will be impossible to see fully into both sides of the place preference box because the surrounding sound-proof enclosure means that the camera can't be mounted far enough away. However, mounting one camera on either side would solve the problem.

In these situations you can use two cameras to view the apparatus and then use a special *montage video source* to join the images from the two cameras together.

How montage video sources work

A montage video source joins together the images supplied by *two normal video sources,* thus creating an image which shows an entire piece of apparatus - see figure 5.

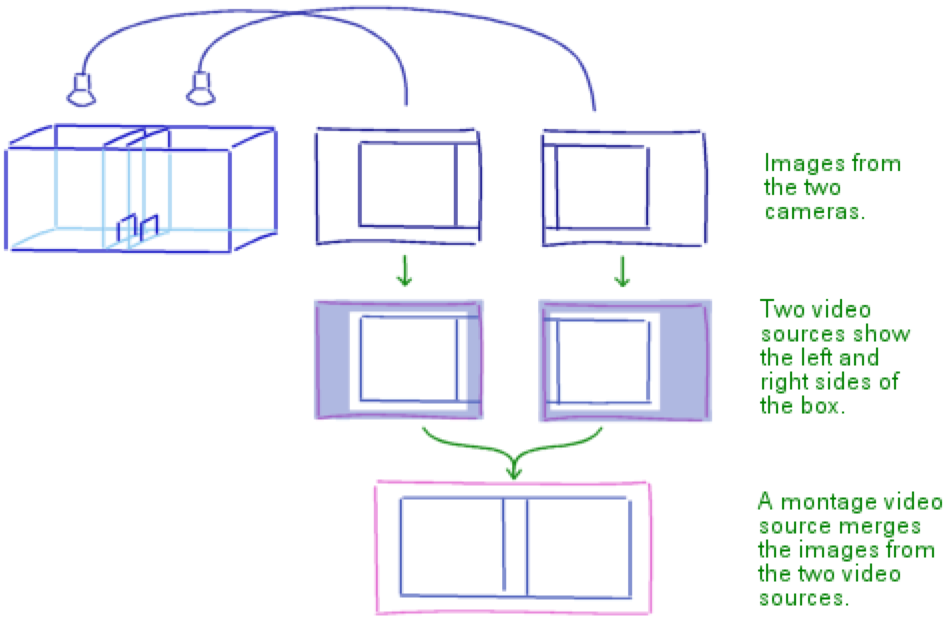


Figure 5. The pictures from the two cameras are first used to create two separate video sources. These are then merged using a montage video source, to create a final image of the entire box.

At present, it's not possible to join two *montage sources* together, so if your apparatus requires three or more cameras to be fully visible then you won't be able to use it in ANY-maze - however, we're planning to include this feature in the future.

See also:

- Setting up a video source
- Setting up a montage video source

Setting up a Video Source

Introduction

For a general introduction to video sources, see An introduction to Video sources.

To add a video source to a protocol, click the *Add item* button shown in the ribbon bar and select *New Video Source* from the menu which appears.

The video source's settings page is used to set up a video source for a piece of apparatus. To do this, you should first select the camera, digitiser or video file which is the source of the raw video pictures and then define the part of the image which shows the apparatus - of course, this might just be the entire image. You may also need to alter the video image's orientation, size, brightness, contrast etc.

If you are working with a piece of apparatus which requires two cameras in order to be viewed entirely, then you will need to create two video sources (each showing half of the apparatus), and one montage video source which will join the two video sources together to create a final image of the entire apparatus.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the video source name
2. Select the source of video images
3. Specify whether the apparatus is being viewed from above or from the side
4. If necessary, flip the image horizontally or vertically
5. If necessary, adjust the video image size and/or frame rate
6. If necessary, zoom in to enlarge the apparatus in the image
7. If necessary, select the area of the video picture which shows the apparatus
8. If necessary, adjust the video image brightness, contrast, etc.
9. If necessary, erase any cage bars, grid lines or cables in the image.

What next?

After completing these steps, you should consider whether you want to include any I/O devices in your protocol.

See also:

- Problems with cameras
- Problems with video pictures
- Adding elements to a protocol
- The Camera/Digitiser properties window
- Setting up a montage video source
- Editing a video source
- Deleting a video source

Video source name

In brief

Enter a name for the video source in the *Video source name* field on the Video source's settings page. You must make an entry, but it can be anything you like.

Details

- As a video source will show a single piece of apparatus, the name of the apparatus is usually a good choice - perhaps *Plusmaze*.
- However, if you will be using more than one piece of apparatus in this protocol then you may wish to call the video sources something like '*Plusmaze Room 1*', '*Plusmaze Room 2*' etc.
- On the other hand, if the video source shows just part of a piece of apparatus, then it's a good idea to call it something like '*Place preference box - left side*'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Selecting the source of video images

In brief

Use the *Source of video images* list on the Video source's settings page to select the camera or digitiser which shows the apparatus. As you make a selection in the list, the image from that camera or digitiser will appear in the *Image pane* on the right-hand side of the screen.

Details

This list shows, in order :

- Any digital cameras connected to your computer.
- Any digitisers connected to your computer.
- Any shared devices on other computers connected to your network.
- Any DVD drives on your computer, provided you've enabled them in the Set up video devices window of the video page.
- A single entry which allows you to choose an ANY-maze video file or any standard format video file (such as an AVI or MPEG file) as the video source.
- Any example ANY-maze video files installed on your computer*.

  **This item will only be shown if the* Include example experiments... *option is selected under Features for new users on the Options page.*

If the picture from the camera or digitiser shows more than one piece of apparatus, don't worry - you'll sort that out in a moment by selecting just the relevant part of the image.

If the picture shows just part of the apparatus, then you'll also need to create another video source which shows the rest of the apparatus, and a montage video source which joins the two video sources together.

Problems

If you're not sure which item to select, just work through the list selecting one at a time until the correct image appears on the right-hand side of the page.

If your camera or digitiser isn't listed and it's a USB or FireWire device, then check it's plugged in. If it isn't, then just plug it in now and it will appear in the list within a few seconds.

If you've already selected a video file as the source, and now you'd like to change to use a different file, just reselect *ANY-maze video files* in the list.

For other devices and/or other problems, refer to:

- Problems with cameras
- Problems with video pictures

Specifying whether the apparatus is being viewed from above or from the side

In brief

Use the two buttons on the Video source's settings page to specify whether the video source shows the apparatus from the side or from above. Note that by default, ANY-maze assumes that an apparatus is being viewed from above, as this is the most common scenario.

Details

When a piece of apparatus is viewed from the side, ANY-maze makes two changes to how it works:

- The system takes into account the fact that the target animal will appear bigger and smaller as it moves towards and away from the camera. This is important, because normally (i.e. when viewed from above) the system assumes that the animal will remain generally the same size and it uses this assumption to help avoid erroneously detecting artefacts (such as reflections in a water-maze) as the animal.
- ANY-maze will automatically detect rearing by the target animal. In fact, this only works well if there is good contrast between the animal and the background of the apparatus - this is because the system analyses the animal's shape, and this is only well defined if there's good contrast.

Note that although rearing is automatically detected when apparatus is viewed from the side, you obviously don't have to actually use the rearing results. So if rearing is irrelevant to you (for example, you're tracking a fish or the need for good contrast is not met) then you can simply ignore this analysis.

If rearing is of interest, you may wish to have the system indicate when it considers the animal to be rearing - this is one of the options that can be specified in the What to display while testing protocol element.

Flipping an image horizontally or vertically

In brief

By clicking the *Flip horizontal* or *Flip vertical* buttons (which are shown in the ribbon bar when a video source is selected in the protocol), you can flip the video image horizontally and/or vertically.

Details

Changing the image orientation (i.e. flipping the picture vertically or mirroring it horizontally) doesn't actually affect ANY-maze tracking in any way, but it might help you as incorrect orientation can be very confusing.

Adjusting a video picture's size and/or frame rate

Changing image size and/or frame rate actually alters the images being provided by the relevant camera / digitiser. This means that any other video sources which also use the same camera / digitiser will be affected too.

In brief

By clicking the *Properties* button (which is shown in the ribbon bar when a video source is selected in the protocol), you can open the properties window for the video source's camera or digitiser. This window includes a page called Image format, where you can alter the size and/or frame rate of the video images the device provides.

Details

Altering the image size is particularly useful when you're planning to use the video source as part of a montage video source. In this case, joining two full-sized images would create an extremely large image - reducing the two images to half size and then joining them would be a better option.

It's also useful to reduce the image size and/or frame rate when:

1. You're using a USB device and USB bandwidth is an issue.
2. You're working with multiple apparatus simultaneously.

In all these cases, a reduction in image size or frame rate will help because it reduces the amount of information that is transmitted to and processed by the computer.

See also:

- Camera / Digitiser Properties window : Image format page

Zooming in to enlarge the apparatus in an image

In brief

By clicking the *Zoom* button (which is shown in the ribbon bar when a video source is selected in the protocol), you can zoom-in in the video picture.

Details

When you click the *Zoom* button, a slider opens that allows you to alter the level of *zoom* from x1 up to x3. Of course, an effect of zooming is that you will no longer be able to see all of the picture from the camera and this might mean you want to pan the image so the apparatus is centred - you can do this by clicking the *Pan* button and then dragging the image with the mouse - try it and you'll soon get the idea.

Having zoomed in, you can quickly cancel the zoom simply by clicking the *Cancel zoom* button.

Note that this zoom function uses what's known as a *digital zoom* - that's to say, ANY-maze enlarges the captured picture digitally. The quality of a digital zoom is never as good as an optical zoom (in which a lens is used to zoom in) and therefore if your camera has a zoom or vari-focal lens, you should use it in preference to this function.

Selecting the area of a video picture which shows a piece of apparatus

In brief

You should use the mouse to select the part of the video picture which shows **just** this video source's apparatus. To do this, simply left click in the video picture and drag the mouse. The area you define is known as the video source's *region of interest.*

Note: To select the area, you should first ensure that the appropriate video source is selected in the protocol list.

Details

You can edit the region of interest after you've drawn it by dragging its edges or by picking it up and sliding it around over the top of the video picture.

To remove the region of interest, either click the *Remove region of interest* button, or just double click the left mouse button on the video picture.

If the video picture only shows a single piece of apparatus, then you don't *have* to define any region of interest - you could just use the whole picture. Nevertheless, if the apparatus doesn't fill the picture then it's still a good idea to define a region of interest, as you can use it to remove the surplus parts of the image - thus reducing the amount of information ANY-maze has to process. This can be very useful if you are tracking in more than one piece of apparatus at the same time.

However, take care not to crop the image to the very edge of the apparatus as this may prevent the auto-start feature from working - see figure 1.

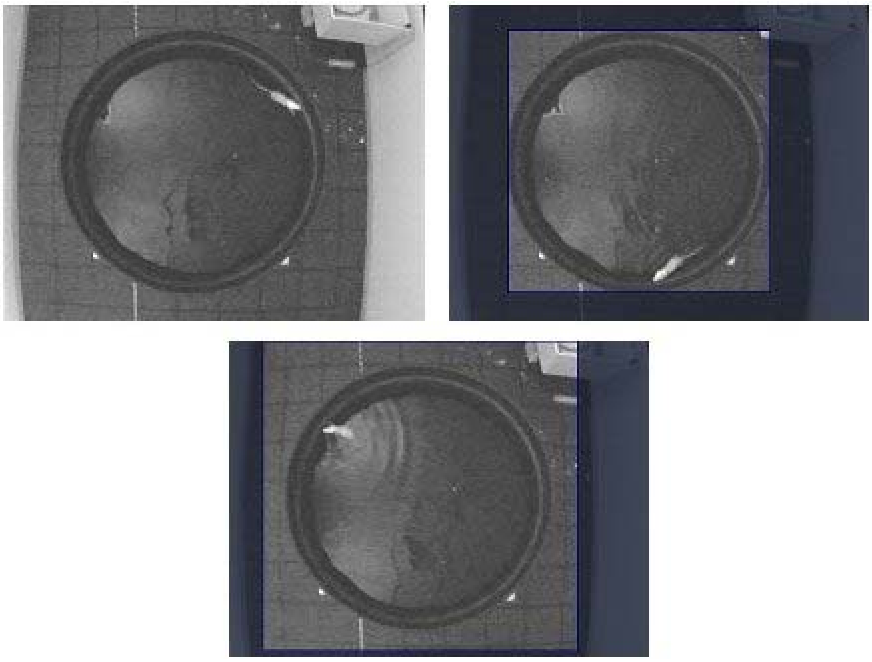


Figure 1. The first image shows the full picture from the camera. As there's a lot of the image which isn't part of the water-maze, it would be a good idea to crop the excess. In the second image, too much of the excess has been cropped, so auto-start probably won't work. In the final image, the selected area is just right.

Adjusting a video picture's brightness, contrast etc.

When you change the brightness, contrast etc. of a picture, you actually change the image supplied by the camera / digitiser. This means that any other video sources which also use the same camera / digitiser will be affected too.

In brief

By clicking the *Properties* button (which is shown in the ribbon bar when a video source is selected in the protocol), you can open the properties window for the video source's camera or digitiser. This window includes a page called Image control where you can alter the brightness, contrast etc. of the picture - the exact options available will depend on the capabilities of your camera or digitiser. You should use the controls to produce a clear image with a good contrast between the animal and the apparatus background - see figure 1.

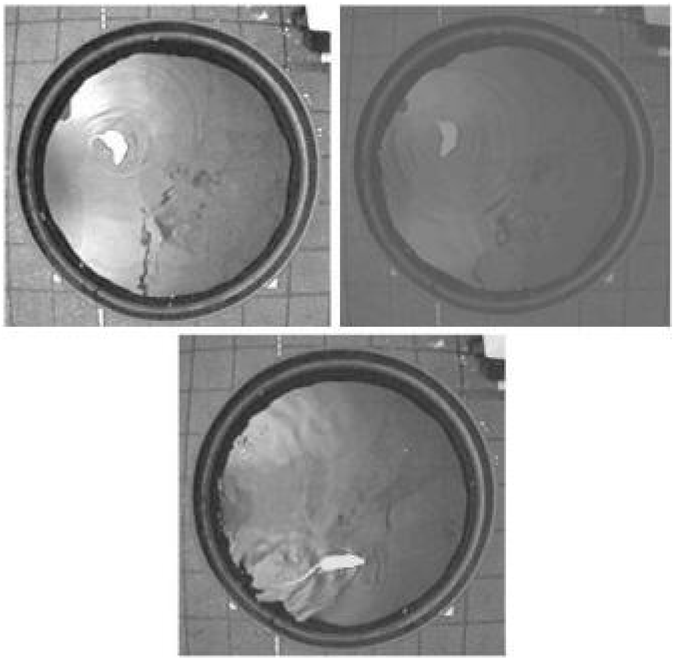


Figure 1. In the first image, there is too much contrast; in the second, not enough. In the third, the contrast is just right. In fact, ANY-maze would track without errors in all three images, but would identify the animal in more frames in the third image than in the other two.

Details

When adjusting the image, you should aim to create a *good* level of contrast between the animal and the background of the apparatus. This doesn't mean you should necessarily try to create the *maximum* contrast, as this can actually result in worse tracking if there are reflections in the image. In the figure above, the first image has too much contrast, whereas the second has too little - the last image is just right for tracking. You'll probably notice that it's also the image which is the most satisfactory to human eyes; in general, what looks good to you will look good to the software.

However, there are two situations when you will need to pay careful attention to these settings if tracking is going to be at its best. These are when the animal and apparatus background are very similar, for example, a white rat on a white background, and/or when you're tracking in low light conditions - see the following topics for details

- Tracking with low contrast between the animal and apparatus background
- Working in low light or darkness

See also:

- Camera / Digitiser Properties window : Image control page

Erasing cage bars, grid lines or cables from a video image

In brief

By clicking the *Erase lines* button (which is shown in the ribbon bar when a video source is selected in the protocol), you can open the Erase grid lines, cage bars or cables window. Using the controls on this window, you can electronically remove cage bars, grid lines or cables from the image of your apparatus; generally speaking, this will improve the tracking.

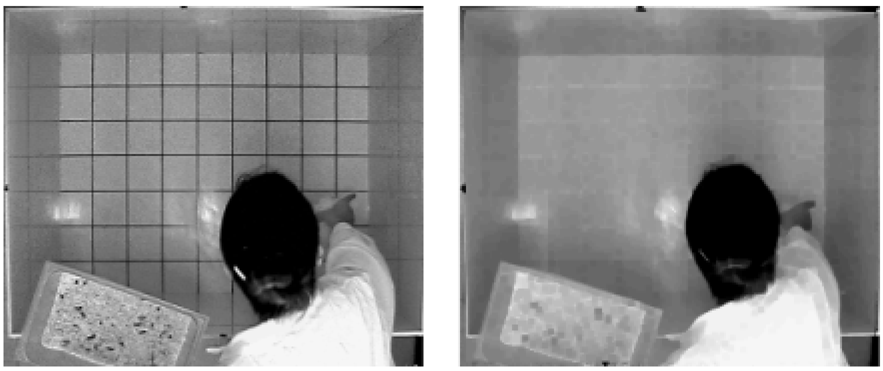


Figure 1. This open field has grid line painted in its base, and these will interfere with the tracking. In the image on the right, the grid lines have been electronically erased.

Details

If your apparatus is being viewed through cage bars, then as you'd expect, the animal will appear to be broken into pieces by the bars. Although this doesn't inconvenience a human observer, ANY-maze will often see all the pieces as separate animals and won't track (or if it does track, not very well). The solution is to electronically erase the bars so the animal is then viewed as a whole. This sounds like some kind of magic, but it's actually quite a mundane image processing technique that expands the parts of the image either side of the bars until they join up, effectively making the bars disappear.

Although it's quite easy to understand why you might want to erase cage bars, it's harder to appreciate why it is necessary to delete grid lines painted on the apparatus - after all, the animal is standing *on top* of the lines so the camera can see all of it!

This is of course true, but in this case the problem relates to how ANY-maze detects where the animal is. To do this, it compares an image of the apparatus *without* the animal in it to the one that *does* have the animal in. So, taking the image above as an example, it will be looking for areas of the image that are darker than the background (the animal being a touch darker) and it will consider that these are the animal. However, where the animal is standing over a grid line, it will be *lighter* than the background and the effect of this will be that ANY-maze won't consider the animal to be in those places, so the grid lines will seem to *break the animal into pieces.* A simple solution to this problem is to tell ANY-maze that the background is non-uniform - then it looks for places where the animal is either lighter or darker than the background (i.e. just different). However, in the above example this wouldn't work very well because the animal is so similar to the background colour. This would mean that the system would also detect the shadow cast by the animal and the tracking would therefore be quite poor. So in this case, erasing the grid lines and then tracking the animal as a target that's just *darker than the background* would be the best solution (and indeed in the above example it works very well).

Another common scenario is where there's a cable or tube in the picture - for example, a cable coming from an electrophysiology head-stage. In this case, the cable will be very like cage bars and will potentially break the animal into two parts - erasing it will overcome this problem.

Actually erasing the lines is done by clicking the *Erase lines* button in the ribbon bar and using the options on the Erase grid lines, cage bars or cables window that appears.

ANY-maze will erase lines in the images it uses internally to track the animal; however, the image you see on the screen will continue to be the unadjusted picture, and so the lines will still be shown.

See also:

- The Erase grid lines, cage bars or cables window

The Erase grid lines, cage bars or cables window

In brief

You can use the options in this window to erase grid lines, cage bars, cables, wires, tubes, etc. from the video image.

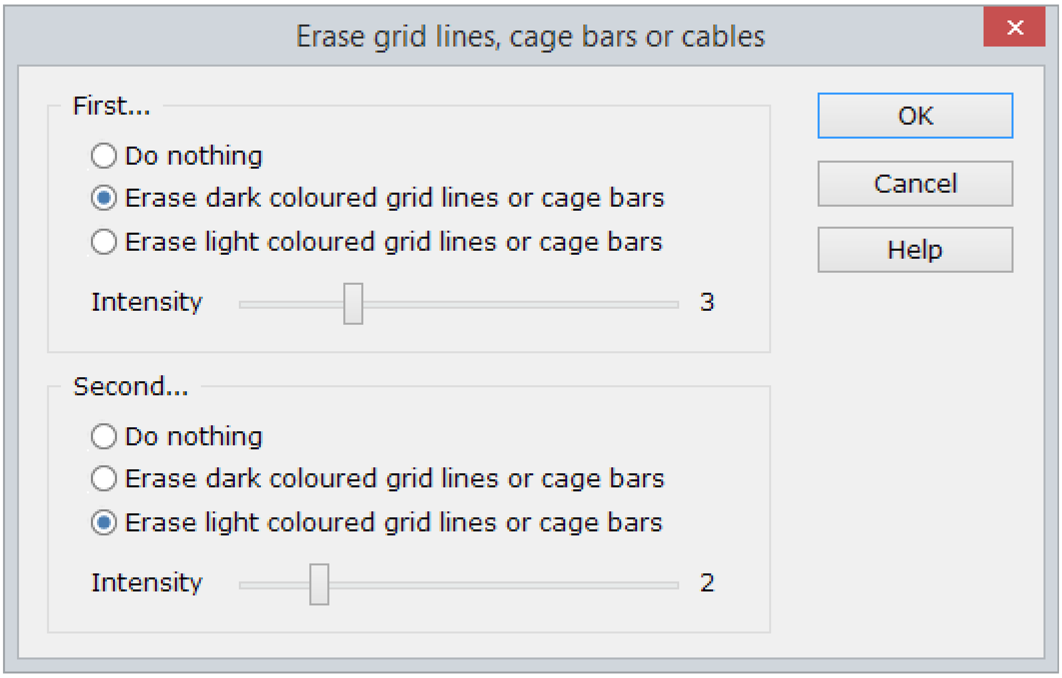


Figure 1. This Erase grid lines, cage bars or cables window

Details

To erase lines that are darker than the background of the image, you should first select the *First...Erase dark coloured grid lines or cage bars* button and then adjust the *Intensity* slider until the lines just disappear. Surprisingly, this may cause light coloured lines to appear in their place; if this occurs, then select the *Second...Erase light coloured grid lines or cage bars* button and then adjust the *Intensity* slider until the pale lines just disappear too.

If the lines are lighter than the background of the image, you should simply reverse the above process; i.e. first erase light coloured lines and then, if necessary, erase any dark lines that appear.

Take care not to use an unnecessarily high intensity; this will simply reduce the image quality.

⚠ *Although ANY-maze will erase lines in the images it uses internally to track the animal, the image you see on the screen will be the unadjusted image, and will continue to have the lines shown. The only exception is when you're using this window to actually set the erase parameters.*

Special considerations when using montage video sources

As you may know a montage video source shows images from two other video sources joined together. Usually, any adjustments you make to the constituent sources will appear in the montage source, but erasing lines is an exception.

You *can* use this function in any *constituent* video source, but the image used by the montage source *won't* have the lines erased. To erase them in a montage source, you should select the montage source and then erase them there - see figure 1.

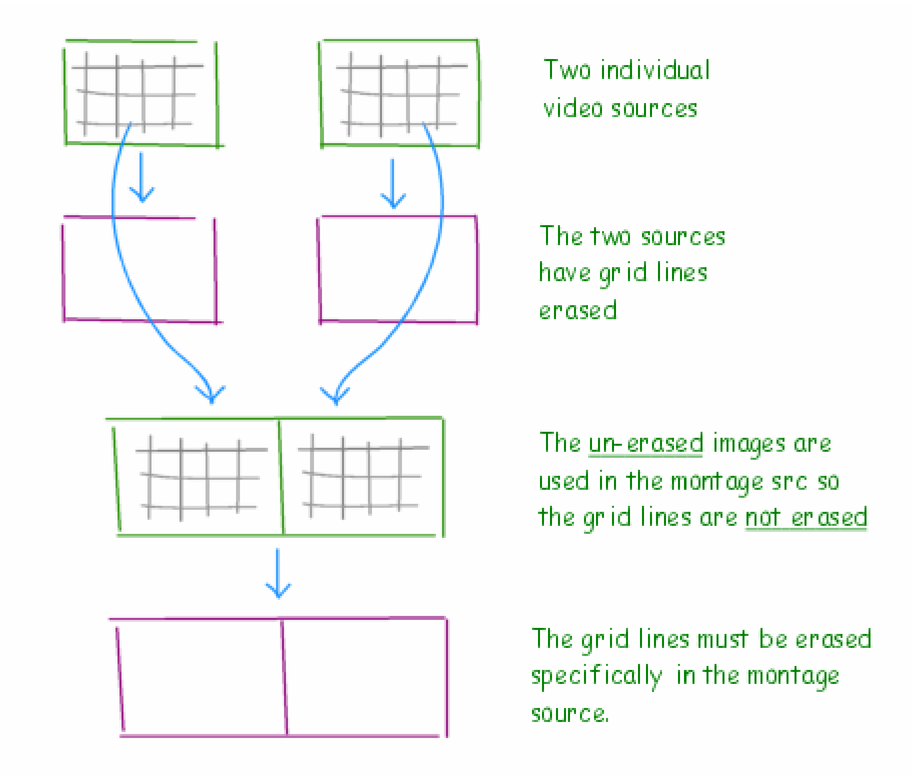


Figure 1. A montage source uses the images from the constituent sources before any lines are erased in them. Therefore to erase lines in a montage source, you must apply the erasing to the montage source itself.

See also:

- Erasing cage bars, grid lines or cables from a video image

The Camera/Digitiser properties window

Introduction

The camera/digitiser properties window allows you to alter the low-level settings of a camera or digitiser. Specifically you can alter such things as contrast and brightness, image size, frame rate, etc.

These settings actually directly control the physical camera or digitiser and this has two repercussions:

1. The settings available will depend on the camera or digitiser. For example, some cameras include an ability to zoom while others don't - obviously you'll only be able to use this setting if your camera actually has this ability.
2. Changes you make will affect *all* video sources which capture images from the camera or digitiser. This is actually fairly obvious when you think about it - for example, if you increase a camera's brightness, then anything which asks the camera for a picture is going to receive a brighter image.

How to open this window

To open the camera / digitiser properties window, you simply need to right click over a video image and select *Properties* from the menu which appears.

If the properties option is disabled, it means that you can't edit the properties because a test is actually running.

You can also open this window when you're editing a video source by clicking the *Properties* button in the ribbon bar.

Details

The camera/digitiser properties window can include six different pages, although exactly which ones are available will depend on the camera/digitiser's abilities. The pages are:

- The General information page
- The Camera control page
- The Video signal page
- The Image format page
- The Image control page
- The Lighting level page

General information page

In brief

This page of the Camera/Digitiser properties window shows general information about your camera or digitiser, such as the type of device and the version number of its driver software. In fact this page is also available for ANY-maze video files and _amexampleimages_, where it shows information such as video file size and compression ratio.

Details

The information on the general page is self-explanatory except for one thing - for some devices, the page will include an *Advanced settings* button.

The Advanced settings

Devices which include an advanced settings button support more options than those which are available through the standard properties window - for example, some cameras include the ability to alter their shutter speed.

By clicking the Advanced Settings button, you will open the properties window provided by the camera or digitiser's manufacturer - obviously, the options available will be specific to the device and therefore I can't include details about them. Nevertheless, you might find options such as:

Auto mode

An auto mode is sometimes available on digital cameras and causes the camera to automatically adjust its shutter speed, gain, etc. depending on the lighting conditions. Although auto modes can work well in ANY-maze, they sometimes causes problems because the overall brightness of the picture can change as the picture contents alters. For example, if you enter a picture wearing a white lab coat, the entire picture may automatically be made darker to compensate for the increased white area (your lab coat) now in the image - and as you leave again, the reverse can occur. Such adjustments probably won't affect the tracking, but they can cause ANY-maze to take longer to identify the animal at the start of a test.

Other situations in which auto modes can be unsatisfactory are low light and/or low contrast tracking. In these cases you may find that manual adjustments are better able to maximize the contrast between the animal and apparatus background.

Backlight compensation

If your device supports backlight compensation, then it's best to switch it off in ANY-maze as your animals are unlikely to be backlit. That said, if you

	have a bright background and a darker animal, then it's sometimes worth experimenting with this setting as it can help to create a more natural image.
Colour controls	As ANY-maze uses black and white images, you can safely ignore any colour controls your device provides. In fact, if it includes an option to provide just black and white images then it's a good idea to switch it on.
Shutter speed / gain	You may find it useful to use controls such as *shutter speed, gain* or *exposure* to improve images in low light conditions.
Other options	Feel free to experiment with any other options your device provides.

Camera control page

In brief

This page of the Camera/Digitiser properties window contains controls to alter camera settings such as focus and iris.

Details

Most cameras don't support all the possible options, and therefore it's normal for some of them to be disabled. In fact, for many cameras this page simply won't be included at all because they don't support any of the settings.

We've noticed that some devices report that they support certain options when in fact they don't. In this case, the setting will be enabled but using it won't have any effect on the picture.

Video signal page

In brief

This page of the Camera/Digitiser properties window only applies to digitisers, and allows you to specify the type of video signal that your digitiser will receive from your camera.

Details

This page allows you to choose between the four most common analogue video signal types. If you're not sure which one to choose, try each one in turn until a good image appears.

Depending on your digitiser, selecting the wrong signal type may simply cause the image to show the text *The camera or VCR is either disconnected or switched off*, or it might cause the image to be unstable causing it to 'roll' down or across the screen. Selecting the correct signal type will resolve this.

Some digitisers automatically detect the signal type they're receiving, and therefore they won't include this page in their properties window.

Image format page

In brief

This page of the Camera/Digitiser properties window allows you to change the size and frame rate of the images provided by a camera or digitiser.

The image format can only be altered from the Video sources element of the protocol. If you access this page from anywhere else, it will show the correct information, but you won't be able to change any of the settings. For more information see the restrictions *section below.*

Details

The main benefit of altering the Image size, and indeed of altering the Image frame rate, is that you can control how much information is provided by the camera or digitiser to the ANY-maze software.

Altering the image size

By default ANY-maze will ask a camera or digitiser to provide 30 images per second at a size of 640 x 480 pixels (or as close to it as possible). For most apparatus this is a good size to use: the image is large enough for you to clearly see how the animal is behaving without it being so large as to require an excessive amount of processing. However, there may be situations when you want to alter the image size:

- If the animal is relatively small compared to the apparatus, then using a larger image size will provide more detail.
- If the camera shows multiple apparatus (for example, four open fields) then using a larger image size will provide more detail for each of the individual apparatus.
- If you are working with multiple cameras then using a smaller image can be a good idea as it reduces the amount of information sent from the camera to the computer (which can help to solve USB bandwidth issues, as described here) and it reduces the amount of data ANY-maze needs to process.

Only image sizes that your camera or digitizer supports are listed. For most USB cameras this is usually quite an extensive list, with sizes often varying from as much as 1920 x 1080 to as small as 160 x 120, however, for some cameras, and most digitizers, there may only be one size shown, usually 640 x 480.

For cameras which support high-definition images, it is often tempting to choose the largest size listed, but this generally isn't a good idea, as it simply means ANY-maze will have much more data to process without (usually) providing any real benefit. Also you may find that at large size, the camera's frame rate is reduced - for example, a camera which can send 30 images of 640 x 480 per second, may only be able to send 7 images of 1920 x 1080 per second.

Altering the image scaling

In the previous section we saw how you can alter the image size provided by your camera or digitizer, but that the list of available sizes are limited to those that the camera or digitizer supports. So what should you do if, for example, you have a digitizer which only provides HD images of 1920 x 1080 and you want to use a smaller size? The answer is to use the *image scaling* to reduce the image to a more manageable size - perhaps by scaling it to 50%, so the final image is 960 x 540.

It's important to understand that scaling is performed on the image received from the camera and not by the camera itself. This is relevant because it means that (in our earlier example) the camera would still send an image of 1920 x 1080, but that ANY-maze would then scale the image down by 50%. And this is relevant because it means that ANY-maze has to do a lot more processing than it would have to do if the image was actually sent as 960 x 580. For this reason it's best to choose an image size which is as close to the size you want as possible, and then use image scaling to reduce it further if necessary.

Increasing the frame rate

In some situations you may want to increase the number of frames sent from the camera, which you can do by simply setting the *Frame rate* to *Maximum*. If your camera can supply 90 frames per second, for example, then this is what it will then start to do.

Although it might seem appealing to always set the frame rate to maximum, you should really only use this option if there's some specific reason why you need it. Using it unnecessarily just means that ANY-maze has to analyse many more images per second, without providing much (if any) benefit, as the animal will usually have moved very little from one frame to the next.

That said, if you do increase the frame rate, then you may want to also adjust:

- The frequency with which ANY-maze records the animal's position. By default, this is a maximum of 10 positions per second. This can be changed using the options in the What to record while testing element of the protocol.
- The number of frames recorded per second in ANY-maze format videos (assuming you want to record a video). The default 'Medium quality' video recorder settings will record 15 frames per second, even if the camera is providing many more. You can change the video recorder settings in the What to record while testing element of the protocol. If you want to record at a high frame rate, then you should edit one of the standard video recorder settings schemes so the frame rate is blank - these means the recorder should record every frame it receives from the camera.

Reducing the frame rate

In general ANY-maze will ask a camera or digitiser to provide 30 images per second at a size of 640 x 480 pixels (or as close to it as possible). With a little maths, you'll find that this is about 10 million pixels per second which need to be transferred to the computer and processed. Fortunately, for modern computers this isn't much of a problem, but there are circumstances when this amount of information may be too much:

1. *If you're using multiple USB devices*

   If your device connects to your computer using USB and you have four or more devices connected to the same USB host controller then you may have problems with bandwidth (the cameras will be trying to send more data than the USB cable can transmit). By reducing the frame size to say 320 x 240 pixels and dropping the frame rate to 15 frames per second you'll reduce the amount of information by a factor of 8 which can help resolve USB bandwidth issues, as described here.

   This technique doesn't always work, because some cameras will continue to transmit the same amount of data and only adjust it once it arrives at the PC.

3. *If you're tracking in multiple apparatus simultaneously*

   It's self-evident that when tracking in multiple apparatus, the processor has more work to do. Normally, this isn't a problem as modern processors can easily cope with tracking in up to 4 or even 8 apparatus, but as you get to higher numbers of apparatus, things can slow down. Reducing the image size can help to resolve these sorts of performance problems - for example, tracking sixteen images of 320x240 is not too different to tracking four of 640x480.

Restrictions

The options on this page will only be enabled if you access the page when you are setting up or editing a video source. In other words, if you want to alter these settings then you should go to the protocol, select the relevant video source and then edit them there.

Image control page

In brief

This page of the Camera/Digitiser properties window allows you to change the appearance of the video image provided by your camera or digitiser.

Details

Changing the appearance of a video image can be critical in ANY-maze, as it will be hard for the system to track in a very dark image or one with very low contrast. When using the controls on this page, you should try to create an image which is pleasing to *you* - in other words, not too bright, not too dark, nice realistic contrast, etc. In general, an image which looks good to you will also look good to the software.

However, there are two situations when you may need to use these settings to create a less natural image: when there's low contrast between the animal and the apparatus background, and/or when you're tracking in low light. For full details on the considerations in these situations, refer to: Tracking with low contrast between the animal and apparatus background and Working in low light or darkness.

The actual settings available are:

Brightness This is self-explanatory.

Contrast This is self-explanatory, but be aware that some devices can report that they support changes to contrast but then don't actually alter it.

White level Setting the white level is supported by some digitisers and can be **very useful in low light**. Essentially, the white level is the level of the video signal above which the digitiser considers everything to be fully white. If an image has low light then it will, naturally, be rather dark and won't have any area which is 'fully white'. By reducing the white level, you can alter this so that whatever the brightest thing in the image is (even if it's actually quite dark) is considered to be fully white and everything darker is considered to be progressively darker shades of grey.

Black level Setting the black level, which is supported by some digitisers, is much like setting the white level only this time you're setting the level of the video signal below which the digitiser will consider everything to be fully black. This is less useful than setting the white level, but can still be beneficial in low light as by increasing the black level you can concentrate the digitiser's resolution on a narrower band of the video signal, thus improving the image.

Gain Altering the image gain effectively amplifies the video signal. Again, in low light this can be useful, although as it also amplifies noise, the image will become noisier - fortunately, ANY-maze is fairly immune to image noise so this shouldn't be a problem.

Lighting level page

In brief

This page of the Camera/Digitiser properties window only applies to some cameras. It allows you to alter the actual lighting of the scene being viewed by the camera.

Details

Some cameras have built-in lights which illuminate the scene they are viewing, and this page allows you to alter the intensity of these lights. The page is only displayed if the camera supports this functionality.

Infrared lighting

In some cameras, the lights used to illuminate the scene are infrared. In this case, the camera usually doesn't detect the visible light because it is fitted with a visible-block/infrared-pass filter - this has the advantage that the visible light levels can change without affecting the camera's view of the scene.

In this case, the adjustments you make using this page will alter the intensity of the infrared lights which *will* affect the camera's view of the scene. Therefore you will see the video image become brighter or darker, while the illumination under the camera will appear to be unaffected.

Setting up a montage video source

Introduction

For a general introduction to video sources, see An introduction to Video sources.

To add a montage video source to a protocol, click the *Add item* button shown in the ribbon bar and select *New Montage Video Source* from the menu which appears.

The New Montage Video Source *option will be disabled until the protocol includes at least **two** normal video sources.*

Details

A montage video source is used when a single piece of apparatus requires two cameras to be fully viewed. The montage source joins together the images from two 'normal' video sources each of which is showing an image of just part of the apparatus. Therefore, before you create a montage source, you will need to have already created the two constituent image sources.

Use this page to set up the montage video source by simply selecting the two constituent video sources and then specifying how to join them together.

As you would expect, when you join two images you create a new image which is the sum of their sizes - however, this may be a very large image. If this is a problem in your apparatus, then you may want to reduce the size of the images in the two video sources that you're joining - see Adjusting a video picture's size and/or frame rate for more details.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Select the two video sources you want to join
2. Choose how to join them

What next?

After completing these steps, you should consider whether you want to include any I/O devices in your protocol.

See also:

- Adding elements to a protocol
- Setting up a video source
- Editing a video source
- Deleting a video source

Selecting the constituent sources of a montage video source

In brief

You should use the two lists shown on the Montage video source's settings page titled *Source 1* and *Source 2* to choose the video sources which show the two halves of your apparatus.

Details

If the two constituent sources are different sizes:

- If they're shown side-by-side, the height of the larger one will be reduced to the height of the smaller.
- If they're shown one on top of the other, the width of the larger one will be reduced to the width of the smaller.

You can't join two *montage* video sources together, nor can you join a normal video source to a montage source - however, we're planning to add this functionality in the future - please let us know if you'd find it useful.

Problems

The images don't align correctly.

You can fix this by altering the image area of one, or both, of the constituent video sources - see Selecting the area of a video picture which shows a piece of apparatus for details.

The image is very large

If the picture you create is very large, you should alter the sizes of the images of the constituent video sources - see Adjusting a video picture's size and/or frame rate for details.

Choosing how to join the constituent sources of a montage video source

In brief

You can choose to join the two images which make up a montage video source so they are shown side-by-side or one on top of the other. Simply select the appropriate options on the Montage video source's settings page.

Problems

The orientation of the two images is such that they don't join correctly.

You may be able to fix this by flipping one, or both, of the constituent video sources. However, this may not work if the image needs to be *rotated* - in this case, you'll have to physically rotate the camera.

The images don't align correctly.

You can fix this by altering the image area of one, or both, of the constituent video sources - see Selecting the area of a video picture which shows a piece of apparatus for details.

Editing a video source

Introduction

Editing a video source is just like setting it up - for example, you can simply type in a different name or alter the area of the image being used by the source - there are no restrictions.

See also:

- Editing a montage video source
- Setting up a video source
- Setting up a montage video source
- Deleting a video source
- Editing protocol elements

Editing a montage video source

Introduction

Editing a montage video source is exactly like setting it up - for example, you can simply type in a different name, or alter the selected constituent sources - there are no restrictions.

See also:

- Editing a video source
- Setting up a montage video source
- Deleting a video source
- Editing protocol elements

Deleting a video source

In brief

To delete either a *normal* or a *montage* video source, you simply need to select it in the protocol list and then click the *Remove item* button shown in the ribbon bar.

Restrictions

You can't delete a video source which is used by an apparatus in which at least one test has been performed - in fact, you can't delete the apparatus either. However, this restriction is entirely benign as you can simply ignore the video source - there's no situation in which you would absolutely have to delete it.

See also:

- Deleting protocol elements
- Editing a video source

I/O Devices

The *I/O device* elements in a protocol represent the physical input/output devices that you wish to use in your experiment.

For more information about I/O devices, refer to the topics below.

- An introduction to I/O devices
- Setting up an I/O device
- Editing an I/O device
- Deleting an I/O device

An introduction to I/O devices

Introduction

In a protocol, an I/O device represents the physical device which connects inputs and outputs to your computer.

For example, imagine you want to connect a temperature sensor to your computer - how would you do that? Obviously you'd need the sensor itself, but you'd also need some type of interface device which would connect the sensor to your PC - this is what ANY-maze calls an *I/O device*.

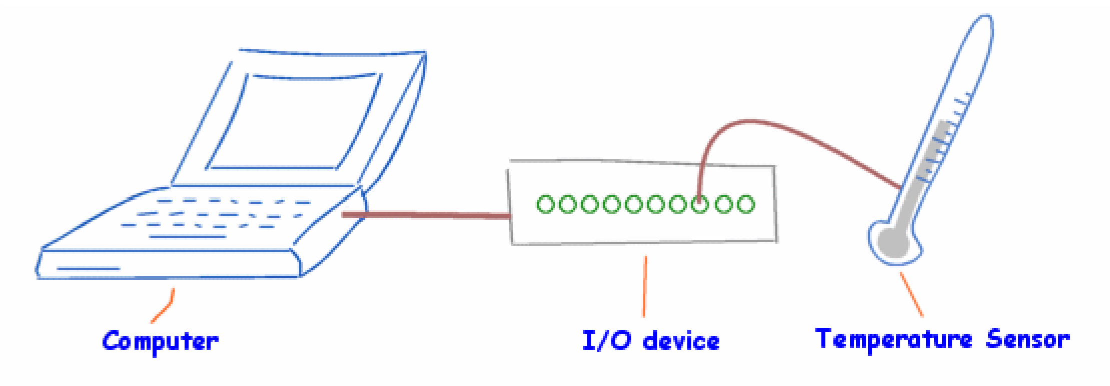


Figure 1. An I/O device being used to connect a temperature sensor to your PC.

A good example of this type of *interfacing I/O device* are the AMi-2 family of devices which can interface all sorts of inputs and outputs to your machine.

In fact, in some cases the *I/O device* and the actual *inputs* or *outputs* are the same physical thing. For example, the OPAD cage is an I/O device that has two touch-switch inputs, a weight sensor, a temperature sensor and a temperature controller built-in. ANY-maze views this single *physical* device as consisting of both an *I/O device* and various individual *inputs and outputs*.

The important thing to understand about I/O devices is that in order to include individual inputs (such as photobeams or sensors) or outputs (such as switches or speakers) in a protocol, you *must* first have included the device they connect to (or are part of). For example, in order to include the temperature sensor shown in figure 1 in your protocol, you would first have to add the I/O device and then add the temperature sensor itself.

Why I/O devices are necessary

There are two reasons why I/O devices are part of the protocol. First, including a device in the protocol ensures that you can then include the physical inputs or outputs connected to it (or that are part of it). So in the example in figure 1, if you didn't include the interface in the protocol, ANY-maze would not know about the temperature sensor (connected to it) and so you wouldn't be able to sense any temperatures in your experiment.

The second reason is that the *I/O device* element in the protocol allows you to configure the device. For example, the AMi-2 Digital interface includes 6 *General Purpose I/O ports* (known as GPIO ports). Each GPIO port can be configured to work in a number of different ways; for example, a port can be configured (amongst other things) as a *TTL output* or as a *photobeam input*. Now imagine you want to connect 6 photobeams to ANY-maze - you would first need to include the AMi-2 Digital interface in your protocol, then configure it so that the 6 GPIO ports were all set to be *photobeam inputs*, and only then would you be able to add the individual photobeams to the protocol.

Using multiple I/O devices

You can include any number of I/O devices in a protocol. They don't all need to be the same type - for example, you could include two Switch and Sense 8/8s and an AMi-2 Digital interface. This would mean that all the inputs and outputs on all these devices would be available for use in your experiment.

See also:

- Setting up an I/O device
- Editing an I/O device
- Deleting an I/O device

Setting up an I/O device

Introduction

For a general introduction to I/O devices, see An introduction to I/O devices.

To add an I/O device to a protocol, click the *Add item* button shown in the ribbon bar and select *New I/O device* from the menu which appears.

The New I/O device menu option is only included if the protocol mode *includes input/output.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the I/O device's name.
2. Choose the physical I/O device.
3. If necessary, configure the I/O device.

What next?

After completing these steps, you will need to add an apparatus element to the protocol.

See also:

- Adding elements to a protocol
- Editing an I/O device
- Deleting an I/O device

I/O Device name

In brief

Enter a name for the I/O device in the *I/O device name* field shown on the I/O device's settings page. You must make an entry, but it can be anything you like.

Details

For interface-type I/O devices, such as the ANY-maze interface, it's best to use a name which reflects what the interface is doing - for example, 'Photobeam interface' would be a good name for a device which is interfacing a group of photobeams to your PC.

For I/O devices which include built-in inputs and outputs, such as the OPAD cage, then you should use a name which identifies the cage, such as 'Cage 1'. This can be the same name that you give the Apparatus element for the cage - this doesn't cause a problem, indeed it's usually a good idea as it makes it clear which I/O device and apparatus go together.

Limits

You can enter anything you like up to a maximum of 32 characters.

Choosing the physical I/O device.

In brief

An *I/O device* protocol element represents some physical device; perhaps an ANY-maze interface or an OPAD cage. You should select the physical device from the drop-down list of devices shown on the I/O device's settings page.

Details

The drop-down list of devices shows all the devices physically *connected* to your computer that ANY-maze supports. If a device is not listed, then either:

- It is not connected
- It is connected, but it has been set to - *Prevent this device from being used in ANY-maze* - in the I/O ports and devices set up window on the I/O page
- The connection to it is faulty (or perhaps the device is not switched on)
- The device's driver is not installed
- The device is not supported by ANY-maze - see I/O devices supported by ANY-maze for a complete list
- It has already been selected in a different I/O device element in the protocol

As the final point above implies, you can only select a physical I/O device into one *I/O device* protocol element - indeed, after you have included a physical device in one element, it will not even be listed in the physical devices list of any other elements.

Note that the device must be connected to your PC when you *add* it to the protocol, but thereafter you can edit the protocol without the device needing to be present (although you won't be able to configure it).

Configuring an I/O device.

In brief

To configure an I/O device, you should click the *Configure* button shown on the I/O device's settings page. Note that not all I/O devices are configurable; if a device cannot be configured, then the button is shown disabled.

Details

The actual process of configuring an I/O device depends on the device itself. Usually, a window will open with various configuration options - this window will include a help button, which you can use to get help specific to the device you are configuring.

Editing an I/O device

Introduction

You can edit all aspects of an I/O device at any time, whether before, during or after an experiment is performed. There are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up an I/O device
- Deleting an I/O device

Deleting an I/O device

Introduction

To delete an I/O device, you simply need to select it in the protocol list and then click the ✕ *Remove item* button in the ribbon bar.

Restrictions

If you delete an I/O device which includes a port that is used by an Input or Output in the protocol, then ANY-maze will display a warning message, alerting you to the fact that deleting the I/O device will cause the relevant Input or Output to stop working. This isn't exactly a restriction, as you can ignore the warning and delete the I/O device if you wish.

See also:

- Deleting protocol elements
- Setting up an I/O device
- Editing an I/O device

Apparatus

A protocol's *Apparatus* elements define the physical apparatus in which testing will be performed.

When using a video tracking protocol mode, the most important aspect of the apparatus is the *apparatus map*, which defines the part of a video source's image that corresponds to the apparatus and divides the apparatus into discrete areas.

Apparatus elements are fundamental to a protocol, and any protocol must include at least one.

For more information about apparatus, refer to the topics below.

- An introduction to apparatus
- Setting up a piece of apparatus
- Editing apparatus
- Deleting apparatus
- Apparatus measures

An introduction to apparatus

Introduction

As you would expect, all protocols in ANY-maze must include a definition of at least one piece of apparatus, as this is where testing is going to take place.

When video tracking, the most essential aspect of this definition is the apparatus map, which specifies both the part of the video image which corresponds to the apparatus and divides the apparatus into discrete areas.

- Apparatus maps
- Working with real-world distances
- Using multiple apparatus

Apparatus maps

As you probably know already, a video source provides images of a single piece of apparatus. This is fine, but how can ANY-maze know which part of the image is the apparatus, and within the apparatus, what parts are of particular interest to you?

Well, that's what the protocol's *Apparatus* element is for. Essentially, to set up a piece of apparatus, you select the video source which shows an image of the apparatus and then you draw an apparatus map over the top of the video picture - see figure 1.

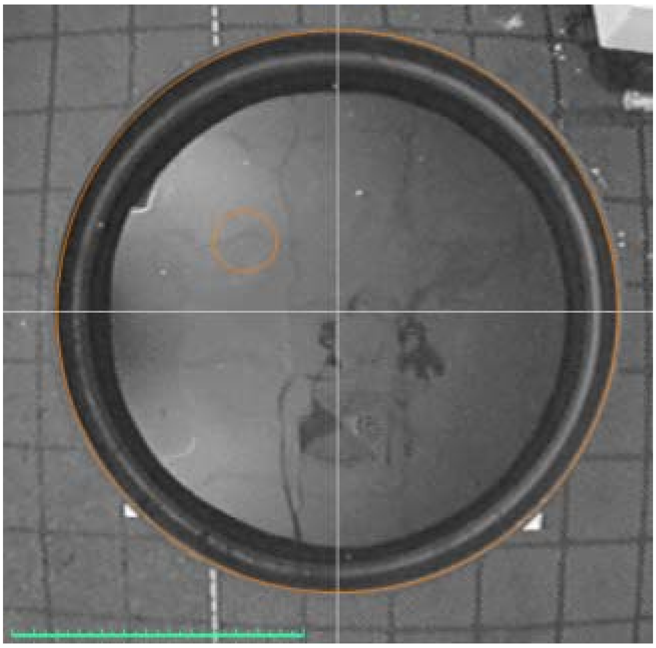


Figure 1. A simple apparatus map for a water-maze.

The apparatus map serves two purposes:

1. It defines the borders of the apparatus. This is important because while tracking, ANY-maze will ignore almost* everything that goes on outside these borders. For example, if you walk around in the video picture during a test, but you stay outside the apparatus's borders, then ANY-maze will simply ignore you.

   ** ANY-maze's auto-start feature actually tracks you (the experimenter) in the area* outside *the apparatus borders - when you leave the camera's view, ANY-maze automatically starts the test. However, after the test has started you can reappear in this area without causing any effect.*

2. It defines distinct areas *within* the apparatus which might be of interest to you. For example, in figure 2 the first map just defines the borders of the plusmaze, so how can ANY-maze know which are open and which are closed arms - or indeed which parts are arms at all? This is solved in the second map, where the apparatus is divided into different areas.

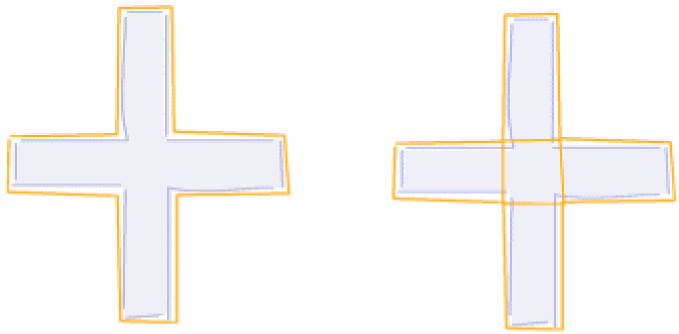


Figure 2. The apparatus map on the left only defines the borders of the plusmaze, whereas the one on the right divides the plusmaze into discrete areas.

Working with real-world distances

As well as defining the apparatus map, the apparatus element is also used to define the apparatus's real-world size. For example, imagine a round open field arena - in the video picture it would appear as something like a large bucket, but how big is it? Of course there's no way to know - a 1m diameter open field would look just like a 50cm diameter open field, particularly if viewed through a vari-focal (zoom) lens.

In fact, the actual size of the apparatus doesn't matter to ANY-maze, but it will to you - for example, you probably won't want measures such as *Distance travelled* reported in pixels!

To solve this, you need to set a scale for the video image which you can do using the *ruler line*. Essentially, you just position the ruler line along a known distance in the image, perhaps across the diameter of an open field, and then tell ANY-maze how long that distance actually is in millimetres. Armed with this information, the system can translate all distances from pixels to metres and thus report measures in units such as metres and metres/second.

Using multiple apparatus

As you're probably aware, ANY-maze can track simultaneously in up to 40 pieces of apparatus in a single experiment. To achieve this, you first need to create one video source for each piece apparatus and then add one apparatus element for each one. This means you'll need to draw one apparatus map for each one, and set the scale for each one too. This is necessary because each apparatus element corresponds to a physically different piece of apparatus possibly being viewed by a different camera and lens, with the result that the exact form of the apparatus and their exact scales will probably all differ to some extent. This might all sound like quite a lot of work, but remember, you can save protocols and then use them again and again, so you'd typically only have to draw all the apparatus maps once.

See also:

- Setting up a piece of apparatus
- Adding elements to a protocol
- An introduction to apparatus
- Saving and loading protocols

Setting up a piece of apparatus

Introduction

For a general introduction to apparatus, see An introduction to apparatus.

To add an apparatus to a protocol, click the *Add item* button in the ribbon bar and select *New Apparatus* from the menu which appears.

In protocol modes which use video, the New Apparatus menu option will be disabled until the protocol includes at least one video source.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the apparatus name.
2. Select the video source which shows an image of the apparatus.
3. Optionally select an 'F' key and/or a switch input to control tests in the apparatus.
4. Choose whether results will always be saved when a test is ended manually.
5. Draw the apparatus map.
6. Use the ruler line to define the scale of the apparatus map.

What next?

After completing these steps, you should consider whether you want to include any zones in this protocol.

See also:

- Adding elements to a protocol
- Editing apparatus
- Deleting apparatus

Apparatus name

In brief

Enter a name for the apparatus in the *Apparatus name* field shown on the apparatus's settings page. You must make an entry, but it can be anything you like.

Details

You can give the apparatus element the same name as a video source - this doesn't cause a problem; indeed, it's usually a good idea as it makes it clear which video source and apparatus go together.

If you will be using more than one piece of apparatus in this protocol, then you may wish to call them something like '*Plusmaze Room 1*', '*Plusmaze Room 2*' etc.

Limits

You can enter anything you like up to a maximum of 32 characters.

Selecting an apparatus's video source

In brief

Use the *Video source* list shown on the Apparatus settings page to select the video source which shows this apparatus. The apparatus image will then appear in the Image pane on the right-hand side of the screen.

Details

All apparatus should have a corresponding video source which shows a nicely cropped image of the entire apparatus with **no other apparatus visible** - see Setting up a video source for details about creating a video source.

Here, you simply select the appropriate video source from those listed. Note that you won't be able to alter an apparatus's video source after at least one test has been performed in the apparatus - for more details about this restriction, and how to overcome it, refer to Editing apparatus.

You can make adjustments to the apparatus's image by right clicking on the video picture and selecting *Properties* from the menu which appears.

See also:

- An introduction to video sources
- The Camera/Digitiser properties window

Choosing a test control key or switch

In brief

You can control tests in a piece of apparatus using one of the 'F' keys located at the top of your keyboard, or using an input switch or remote control of an I/O interface. You don't *have* to select a key or switch, as there are other ways to control tests too.

If you want to use a key or switch, and I recommend you do, then use the *Test control key* and *Test control switch* lists shown on the apparatus's settings page to choose the ones to use.

Another way to control tests is via the Griffin AirClick remote control. This device emulates the keys F1 to F5, and so to use it you simply need to select one of these 'F' keys as the test control key. For more details about the AirClick, refer to this topic. *[Note that the AirClick has been discontinued and is therefore classified in ANY-maze as a legacy device].*

Details

Setting a test control key

To specify a test control key, you simply need to select it in the list. If the key you select has already been assigned to another piece of apparatus, then a message will be displayed asking you whether you want to control multiple apparatus with the same key; if you do this, then selecting one key will start the tests in all the apparatus that use it - this can be useful if you want to synchronise the start of tests in multiple apparatus.

If you have multiple apparatus, and you don't want to control them all with the same key, then it's a good idea to try to choose keys which are not physically adjacent on the keyboard. This just helps to avoid the possibility of pressing the wrong key - good choices are F4, F5, F8 and F9 (there are gaps on most keyboards between F4 and F5 and F8 and F9).

As well as the 'F' keys F1 to F12, the list of keys includes an entry for *Media play/pause*. This is a special key which can be found on some 'multi-media' keyboards, but more importantly, it can also be emulated by some remote control devices. Thus, if you have such a remote control you can use it to control tests in ANY-maze by pressing its *Play/Pause* button. Exactly how this works will depend on the remote control you are using, but a good tip is to set the device to use its 'Windows Media Player' settings (assuming it has these settings).

As well as starting a test, the test control key can also be used to end a test (see Controlling testing for more details). Generally, this is a useful feature but it does have one disadvantage: in some circumstances, if you accidentally press the key twice then you will both start and then end the test. To prevent this from happening, you can deselect the option to *Allow the test control key to end tests* -

then the key will only ever start tests; it will never end them.

Setting a test control switch

A test control switch is either a button on a remote control or a physical switch (usually on a long cable) connected to some sort of I/O device.

Using a button on a remote control, such as the AMi-2 Remote control, is a great way to control tests at a distance from your computer, especially as these remote controls work through walls. An alternative is to use a switch on a long cable and mount it close to (or on) your apparatus.

To use a button on a remote control, you just have to select it in this list. To use a switch, you must be using a Protocol mode which supports inputs and you must have included in the protocol at least one I/O device that has an input switch, such as an AMi-2 Digital interface.

Switches that are already assigned to other apparatus or that are being used by an on/off input item in your protocol won't be listed.

If a switch or remote control button is shown with an X drawn over its icon, it means that the switch has been defined as the test control switch for the apparatus, but it's not currently available - for example, it might be a switch on a USB interface device and the device might have been unplugged.

If you would like to use a switch, but you're unsure how to connect it to your computer then contact ANY-maze technical support who will be happy to advise you.

See also:

- Controlling testing
- Choosing whether results will always be saved when a test is ended manually

Choosing whether results will always be saved when a test is ended manually

In brief

When a test is ended manually, ANY-maze will normally display a message box asking whether you want to save the results of the test or not. Selecting this option will suppress the message - the results will *always* be saved.

Details

Normally when you end a test manually, either by clicking the button or by pressing the Test control key or switch, ANY-maze will display a message asking whether you want to save the test results or not. This is useful if you are ending a test that should never have been started (for example, you accidentally clicked the start test button), since in this case you wouldn't want to save the results as they're meaningless.

Nevertheless, there may be circumstances when you want to be able to end tests without having to respond to this message, and you can do this by selecting this option.

Using this option means that test results will *always* be saved, which may be a problem if you really do accidentally start a test. To address this, you can simply 'undo' the test by clicking the button.

See also:

- Choosing a test control key or switch
- Controlling testing

Drawing the apparatus map

In brief

In order for ANY-maze to understand which parts of a video image constitute your apparatus, you must draw an apparatus map in the Image pane. This map should at least define the border of your apparatus, although you will probably want to divide the apparatus into different areas as well.

To draw the map, you should use the various drawing tools (line, circle, rectangle, etc.) which are shown at the top of the Image pane. These work much like drawing tools in other software you might have used.

- The structure of an apparatus map
- The drawing tools
- Drawing operations
- Drawing the apparatus border
- Dividing up the apparatus

The structure of an apparatus map

The apparatus map always defines the *borders* of your apparatus, and will usually divide the apparatus into different parts as well. The borders are very important, as ANY-maze only looks for an animal inside the borders that you draw, while dividing up the apparatus is optional.

An apparatus map in ANY-maze is made up of various *drawing objects,* such as lines, circles and rectangles. For example, the map in figure 1 consists of three objects; a circle and two lines. These objects are drawn using the *drawing tools*.

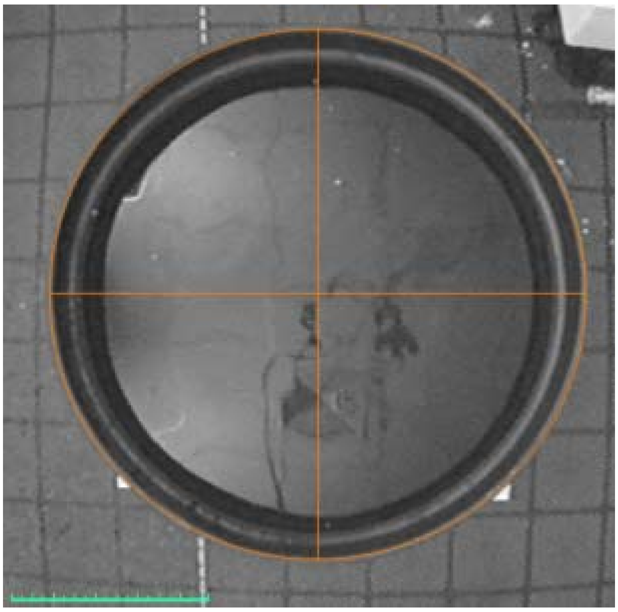


Figure 1. A simple apparatus map for a water-maze, consisting of a circle and two lines.

The drawing tools

The various drawing tools are shown in the ribbon bar when an apparatus element is selected - you select a tool by clicking it.

Some of the vertices of drawing objects are slightly magnetic, that's to say that they attract one another - this makes it very easy to neatly join objects together.

Select objects Use this tool to select objects in a drawing. To use it, simply point at a line of the object you want to select (the mouse pointer will change to a four-headed arrow) and click the left mouse button. Small boxes will be drawn on the object's vertices - as well as allowing you to resize the object, these also indicate that the object is selected.

Multiline tool Draws a multi-line. A multi-line is a series of connected lines. The first line starts where you click the left-hand mouse button, and ends where you release it. The next line starts where the first line ended, and ends where you click next - using this technique you can build up a series of any number of connected lines. To end a multi-line, double click the mouse. Multi-lines are very useful for drawing around apparatus such as a plusmaze. The *magnetic* vertices of a multi-line are the end points of each line. A multi-line's vertices will also attract each other, making it easy to *close* a multi-line object - see figure 2.

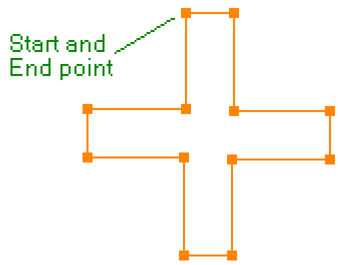


Figure 2. The borders of a plusmaze, drawn with a multi-line. The start and end vertices align perfectly, because they attract each other.

Rectangle tool — Draws a rectangle. The top left corner of the rectangle will be where you click the left-hand mouse button, and the bottom left corner will be where you release it. The *magnetic* vertices of a rectangle are its corners and the centre of each side.

Ellipse tool — Draws an ellipse. The top left 'corner' of the ellipse will be where you click the left-hand mouse button, and the bottom left 'corner' will be where you release it. Obviously an ellipse doesn't actually have any corners, but if you try drawing one you'll see what I mean. The *magnetic* vertices of an ellipse are its pole points.

Line tool — Draws a single line. The line will start where you click the left-hand mouse button, and will end where you release it. The *magnetic* vertices of a line are its end points.

Grid settings — The grid tool is used to add a regularly spaced grid to a map. For full details about grids, see the Apparatus map grids topic.

Select all — Causes *all* the drawing objects to be selected, allowing you to move, copy or delete them all simultaneously.

Points attract — Toggles the *magnetic* attraction of vertices on and off. Switching this off can be helpful when you want two vertices to be close together but NOT touching.

Drawing operations

Adding objects

To add an object to an apparatus map, you should first select an appropriate drawing tool - see above. Next, click and hold down the left mouse button somewhere in the video picture. Now with the button still held down, move the mouse - the object will appear and will get bigger and smaller as you move the mouse around. When you're happy with the object's size, release the mouse button.

Resizing and moving objects

Once an object has been included in the apparatus map, it's very easy to move and/or resize it. To do this, first check that the *Select objects* button is selected and then move the mouse over one of the object's lines - the mouse pointer will change to a four-headed arrow. If you now click the left-hand button, the object will become *selected* - some little boxes will appear around it. You can then drag the boxes to resize the object, or drag any of its lines to move it.

Deleting objects

Deleting an object is very simple. First select it (i.e. click on one of its lines) and then press the 'Delete' key on the keyboard.

Joining objects

The vertices of ANY-maze drawing objects are slightly *magnetic*, in other words they attract each other. If you move the vertices of two objects close together they'll stick to each other just like moving two magnets close together. This is very useful as it makes it easy to create neat drawings in which objects join up nicely; and this isn't just an aesthetic issue - gaps in drawings can mean that areas you intended to be separate are actually treated as one.

Copying and pasting objects

To copy an object, just select it (i.e. click on one of its lines) and then click the *Copy* button. (Alternatively, you can use Ctrl+C on the keyboard or you can right click and select *Copy* from the menu which appears.)

Having copied a drawing object you can paste it into the same apparatus map, or into a different one, simply by clicking the *Paste* button. (Alternatively you can use Ctrl+V or you can select *Paste* from the right click menu.)

It can be very useful to copy all the objects in a drawing, as you can then paste the complete drawing into another piece of apparatus. (Remember, you can select all the objects by clicking the *Select all* button).

Another way to copy an object is to click and drag while holding down the Ctrl key - you'll end up dragging a duplicate of the object you clicked on.

Drawing the border

You must draw a border around your apparatus in order for ANY-maze to work. The border doesn't actually *have* to go exactly round the apparatus; nevertheless you shouldn't just draw a big box with the apparatus somewhere inside. Ideally you should draw a border that neatly encloses the apparatus - see figure 1.

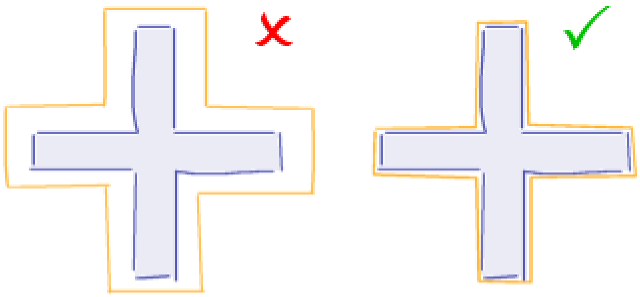


Figure 1. In the first picture the border is too loose, whereas in the second it's just right.

For a technical reason, it's better to draw the apparatus border using either a rectangle, an ellipse or a multi-line; in other words, avoid using lots of individual lines. That said, if you do use lots of lines ANY-maze will still work correctly.

Dividing up the apparatus

After you've drawn the border of the apparatus, you will probably want to divide it into different areas. For example, in a plusmaze you'll probably want to divide off the arms and possibly the ends of the arms too. Without this type of division, ANY-maze won't be able to report on activity in different parts of the apparatus; it will just treat the whole apparatus as a single area - see figure 2.

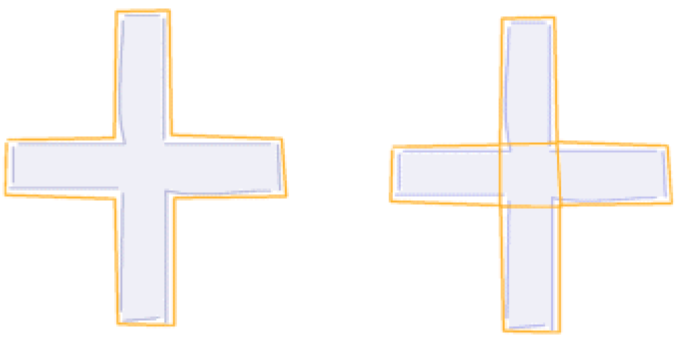


Figure 2. The apparatus map on the left doesn't include any internal divisions. ANY-maze will only be able to report on activity in the apparatus as a whole. The map on the right however, has the arms divided off. Now ANY-maze will be able to report on activity in the open and closed arms, as well as in the apparatus as a whole.

It's important to understand two things. Firstly, that the areas you create will be the *lowest* level at which ANY-maze will be able to report information, and secondly that it's very easy to group areas together. For example, if you *don't* divide off the ends of the arms in a plusmaze, then ANY-maze *won't* be able to report on activity for the ends of the arms as distinct to the rest of the arm. However, if you *do* divide off the ends of the arms, then you can easily group together the two parts of the arm to get information for the arm as a whole - see figure 3.

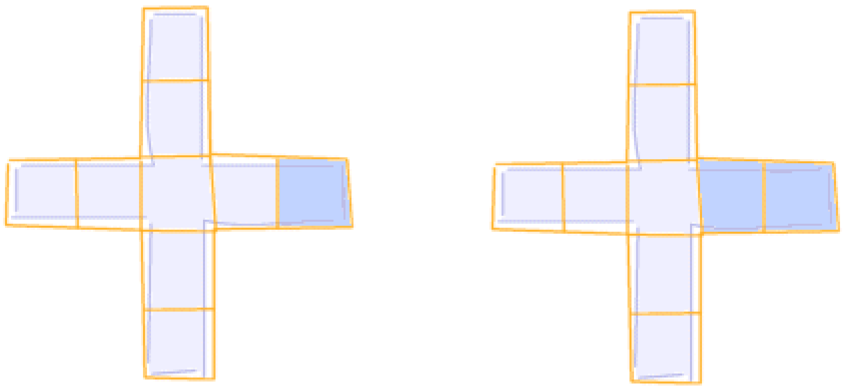


Figure 3. With this apparatus map, ANY-maze can report on activity in any individual area, such as the end of an arm (left), or any combination of areas, such as an entire arm (right).

Of course, you might be tempted to go crazy and divide your apparatus into tiny areas and then group them together so as to guard against the possibility that at some point you might want to know something about an area that isn't defined in your map. Don't worry! You don't need to do this, because you can **always** edit the apparatus map, before, during or even after an experiment - so if you realise you want information about an area which isn't divided off in your map, you can simply edit the map so it is.

In some cases, you may want to divide your apparatus into equal-sized areas, for example into 10cm squares. Of course, you could do this using lots of lines, but making the squares exactly 10cm might be challenging and drawing all the lines would be a little tedious! Fortunately, ANY-maze includes an ability to draw *grids* which make this type of operation very simple - see the Grid Settings topic for details.

There are a few other things to note when dividing up your apparatus.

- If a dividing line doesn't quite close an area, then ANY-maze won't consider the area as being a separate part of the apparatus - see figure 4. Fortunately, the 'magnetic attraction' between vertices often helps to eliminate this problem, as do grids - which will always fully divide areas. However, if neither of these help in your apparatus, then you can simply draw lines which are a little too long - as is the case in figure 4.

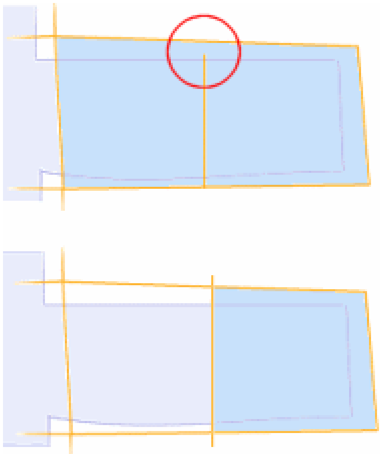


Figure 4. In the first map, the dividing line doesn't quite close off the end of the arm so ANY-maze considers all of the area in blue to be a single area. In the second map, the line fully closes off the end of the arm so it's now considered to be a separate area.

- There's a maximum limit of 250 different areas in a single apparatus. This sounds like a lot, but when using grids it's not too difficult to reach it - for example, a 1.5m x 1.5m open field with a 10cm grid would contain 225 areas. Nevertheless, it's usually easy to reduce the number of areas in these circumstances.
- You may be tempted to use a grid in your apparatus because you used one when you were scoring behaviours manually. However, you should carefully consider whether the grid is actually required. For example, you might have painted a grid onto an open field so you could use 'line crossings' as a way to measure locomotor activity. As ANY-maze can report locomotor activity by measuring the distance an animal travelled during a test, then the grid (at least for this purpose) is redundant. Of course, for the sake of consistency you may want to continue using grid line crossings to measure locomotor activity, in which case you can simply include the grid and ANY-maze will measure the crossings for you.

When using ANY-maze in Freezing detection only *mode, you only actually need to draw the border of the apparatus. However, you can still draw divisions within your apparatus map if you want to; they will just be ignored by the system.*

See also:

- Grid Settings

Apparatus map grids

You can't use a grid to define the borders of your apparatus - you must use one or more drawing objects to do this; see below for more details.

Introduction

It's quite common to want to divide apparatus up in a regular way, perhaps into a number of equal sized squares or into a series of concentric circles. Although this *can* be done using the normal apparatus map drawing objects like lines and rectangles, ANY-maze includes a simpler method - grids.

Apart from simplifying the process of drawing an apparatus map, grids have another advantage - they're defined in centimetres. Thus, if you want to divide your apparatus into say, 10cm squares, you simply have to specify a 10cm grid.

The fact that you specify a grid in centimetres means that ANY-maze needs to know the scale of your apparatus before it can correctly display a grid. Therefore, you should first set the scale - see Using the ruler line to define the scale of the apparatus map - and then add a grid. If you add the grid first, it's not a problem - it'll just be resized when you set the scale.

Adding a grid

To add a grid to an apparatus map, click the *Grid settings* button which is shown in the ribbon bar when an apparatus element is selected. This will open the *Grid Settings Window,* where you can specify the grid, or grids you want.

There are three types of grids available, and you can use them in any combination.

Square grid	As you'd expect a square grid divides a map into equal sized squares. In common with all grids, a square grid will by default be centred within the borders of your apparatus map. However, you can control exactly what 'centred' means. There are two choices; the grid can either have a square in the centre of the apparatus map or a grid crossing - see figure 1.

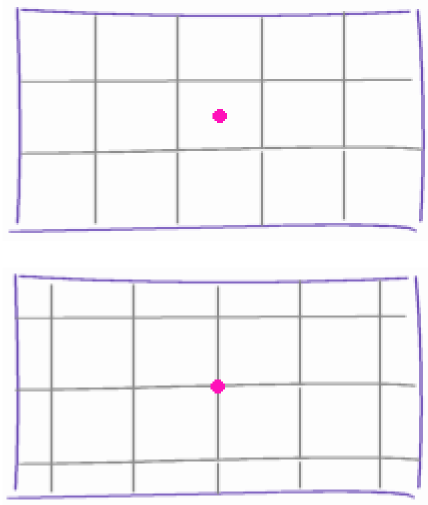


Figure 1. The upper box is divided up using a square grid with a grid square at the centre, while the lower one has a grid crossing at the centre.

Concentric grid	A concentric grid is a series of concentric circles. The grid 'size' is the amount by which the *radius* of the circles increases. For example, a 20cm concentric grid will have an inner circle with a radius of 20cm, the next circle will have a radius of 40cm, then 60cm, and so on. Concentric grids can be very useful in water-mazes. By default, the grid will be centred on the centre of your apparatus map.
Radial grid	Radial grids are usually used in conjunction with concentric grids, but they don't have to be. This type of grid draws 'spokes' outwards from a centre point. The grid 'size' is the angle between one spoke and the next. Of course, you will probably want to use an exact divisor of 360° such as 90°, 45° or 30° so that the grid is a regular size. A 90° radial grid offers an easy way to divide a water-maze, for example, into four quadrants.

As mentioned above, you can use more than one type of grid at the same time. If you do this, then all the grids will have the same origin. Therefore, using a concentric grid with a 90° radial grid will cause the radial grid to exactly divide the concentric grid's circles.

You might be tempted to use a grid to define your apparatus in its entirety without using any drawing objects; however, this won't work. For example, if you have a water-maze with a 2m diameter and you define a 1m concentric grid, the first circle will exactly fit around the water-maze and so you might think that this circle could be used to define the apparatus borders. In fact, ANY-maze requires that you always use drawing objects to define the *borders* of your apparatus, whereas you can only use grids to divide up the inside of the apparatus map.

You will probably notice that grids extend *outside* the borders of your apparatus and cover the entire video image - this is normal, but only occurs while you're drawing the apparatus map. Thereafter, ANY-maze will restrict the grid to the inside of the apparatus.

Editing a grid

You can return to the *Grid Settings* Window at any time to alter any aspect of your grid(s). You can also use the mouse to physically reposition a grid within your apparatus map by simply dragging it around. If you do this and then want the grid to go back to its default position of being centred within the apparatus map, simply double click one of the grid lines.

Removing a grid

To remove a grid, just switch it off in the Grid Settings Window.

Using the ruler line to define the scale of the apparatus map

In brief

Use the ruler line (see figure 1) to define the scale of your video image. To do this, simply position the line along a known distance in the video image and then type that distance into the field titled *The length of the ruler line is* which is shown on the apparatus's settings page.

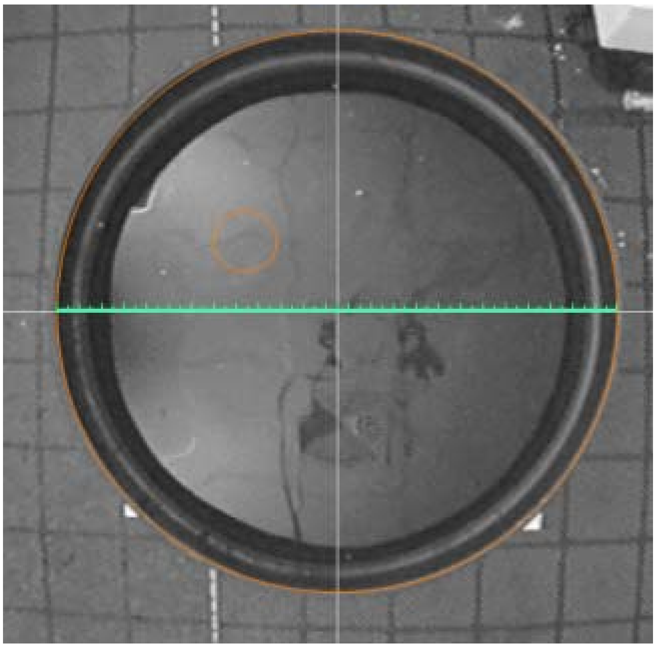


Figure 1. The ruler line positioned across the diameter of a water-maze. Telling ANY-maze the actual diameter, 1200mm in this case, allows it to report all distances in metres.

Details

When ANY-maze *looks* at your apparatus, what it actually sees is lots of dots, which together make up the video picture. This means that the smallest thing it can see is a dot - in other words the *resolution* of the system is the size of a dot. It makes sense, therefore, for the system to use dots as its unit of measure. Thus, as far as ANY-maze is concerned, your animals travel a certain number of dots during a test.

While dots - which are really called pixels - make a nice unit for the software, they're not much use to you - what you would probably like is for distances to be reported in real-world units such as metres.

Obviously, to achieve this ANY-maze needs to know how big a dot actually is - but this is not something it can automatically calculate, as it depends on many factors such as how close the camera is to your apparatus.

To resolve this problem, ANY-maze apparatus maps include something called the *ruler line*. Essentially, the ruler line shows ANY-maze the scale of the image it's looking at, just like the scale shown in microscope photographs do.

To use the ruler line, you have to position it within the apparatus map along a distance you know. For example, you might position it across the diameter of a water-maze (as in figure 1) or along an arm of a plusmaze. When you do this, you should take care to position the ruler along some feature in the video image which is *at the same distance from the camera as the animal will be*. For example, in an elevated plusmaze, position it along an arm - not along something on the floor.

By the way, to position the ruler line. you should drag its end points with the mouse. These, like the vertices of drawing objects, have a slight *magnetic attraction* which makes them stick to other vertices. This usually makes it very easy to position the ruler along a feature in your apparatus map.

Having positioned the ruler along some feature. you then simply need to enter the length of the feature into the field titled *The length of the ruler line is* which is shown in the settings pane. ANY-maze will then calculate how many dots there are along the ruler line in the video image, and use this together with the distance you entered to calculate the size of each dot. Thereafter, all distances will be translated from dots to metres.

Editing apparatus

Introduction

You can edit all aspects of an apparatus element, including the apparatus map, at any time - even after an experiment is complete. There are no restrictions.

This is very useful, as it means that you can, for example, alter the apparatus map to define areas for a new zone or sequence after you finish testing.

See also:

- Editing the elements of a protocol
- Setting up a piece of apparatus
- Deleting apparatus

Deleting apparatus

Introduction

To delete an apparatus element, you simply need to select it in the protocol list and then click the ✕ *Remove item* button in the ribbon bar.

Restrictions

You can't delete an apparatus element once it has been used in a test. Instead, you should disable it by checking the *Disabled* box shown in the settings pane. This will prevent the apparatus from being used in any future tests, but will still allow the existing tests to refer to it.

See also:

- Deleting protocol elements
- Editing apparatus

Apparatus measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for the apparatus as a whole:

- Test duration
- Total distance travelled
- Total distance travelled by the animal's head
- First zone entered
- Visited zone list

Speed measures

- Overall average speed
- Maximum speed

Freezing measures

- Total freezing episodes
- Total time freezing
- Latency to start of first freezing episode
- Average freezing score

Mobility measures

- Total time mobile
- Total time immobile
- Total mobile episodes
- Total immobile episodes
- Latency to start of first mobile episode
- Latency to start of first immobility episode

Activity measures

- Total time active
- Total time inactive

- Total active episodes
- Total inactive episodes
- Longest active episode
- Shortest active episode
- Longest inactive episode
- Shortest inactive episode

Turning and rotation measures

- Rotations of the animal's body
- Clockwise rotations of the animal's body
- Anti-clockwise rotations of the animal's body
- Absolute turn angle
- Absolute head turn angle

Rearing measures

- Number of rears
- Total time rearing
- Latency to first rear
- Average duration of a rear
- Maximum duration of a rear
- Minimum duration of a rear

Miscellaneous measures

- Path efficiency
- Number of line crossings
- On/off inputs positive reversal
- On/off inputs negative reversal
- Tracking quality
- Number of centre positions tracked
- Number of head positions tracked

RAPC measures

- RAPC - Type 1 errors
- RAPC - Type 2 errors

- RAPC - Door sequence

Test duration

Description	Reports the duration of a test.
Calculation method	The value of the test clock when the test ended.
Analysis across time	This measure can be analysed across time. The result for a period is the amount of the test duration which fell inside the period. This will be the full duration of the period for all periods except the last one in the test. This result is most useful in calculations.
Units	Seconds
Notes	None

Total distance travelled

Description	Reports the total distance that the animal travelled during the test.
Calculation method	Sum of the distance between each point in the track - see note below.
Analysis across time	This measure can be analysed across time. The result for a period is the distance that the animal travelled during the period.
Units	Metres
Notes	In some situations, tracks can have small oscillations in them which tend to generate unrepresentatively large values for distance travelled. This occurs most often when an animal travels slowly while moving its body a lot - for example, while exploring an open field. To overcome this, ANY-maze uses an adaptive smoothing algorithm to attenuate these oscillations when calculating distance travelled - see figure 1. Note: The definition of what's a *small* oscillation is based on the animal's size.

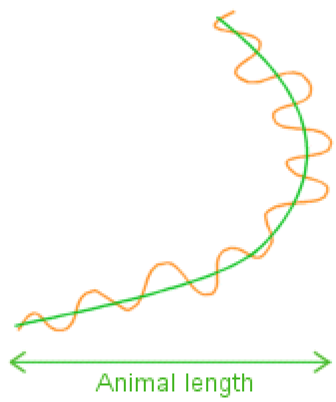


Figure 1. Measuring the length of the actual track (shown in orange) would yield an unrepresentatively large value for distance travelled. ANY-maze uses a 'smoothed' track (shown in green) to better estimate the true distance travelled. [Note: The oscillations in this track have been exaggerated to aid explanation.]

Total distance travelled by the animal's head

Description	Reports the total distance that the animal's head travelled during the test.
Calculation method	Sum of the distance between each point in the head track.
Analysis across time	This measure can be analysed across time.
Units	Metres
Notes	As for the Total distance travelled, ANY-maze will smooth the animal's head track to remove small oscillations which would otherwise distort the result of this measure.

First zone entered

Description	Reports the name of the first zone that the animal entered during the test.
Calculation method	Simply reflects the first zone entry. This is affected by the *Don't score any results in this zone until the first 'true' entry* option on the *Zone entry settings* page. See Choosing how ANY-maze should detect entries into a zone for more details.
Analysis across time	This measure cannot be analysed across time.
Units	None
Notes	When analysed, this value will be treated as a nominal value - see statistical tests included in ANY-maze. In some circumstances, it's possible that two or more zones could be the 'first zone entered'. For example, if you create a protocol in which a single area of the apparatus is included in two zones then, if the animal enters this area first, it will have entered both zones at the same time - meaning there are two 'first zones entered'. In this situation, ANY-maze will report the 'first zone entered' as being the first one of the zones in the zone list shown in the protocol.

Visited zone list

Description	Reports a comma-separated list of the names of the zones the animal visited, in the order in which they were visited.
Calculation method	Each time the animal enters a zone, the zone's name is added to the list.
Analysis across time	This measure cannot be analysed across time.
Units	None
Notes	It is possible to define an area of the apparatus as being part of two (or more) zones. If the animal enters such an area, then it is necessarily entering all the zones simultaneously and the list will therefore include them all. In this case, the zones are added to the list in the order in which they appear in the protocol.

Overall average speed

Description	Reports the average speed of the animal during a test.
Calculation method	Calculated by dividing the Total distance travelled by the Test duration.
Analysis across time	This measure can be analysed across time. The result for a period is the Total distance travelled during the period divided by the Test duration for the period. Note: The 'Test duration' for a period is the amount of the test duration which fell in the period - this is the period's duration for all periods except the last one in the test.
Units	Metres per second
Notes	If you want to know average speed while mobile (i.e. ignoring periods when the animal was stationary), then use a calculation of *Total distance travelled / Total time mobile*.

Maximum speed

Description	Reports the maximum speed of the animal.
Calculation method	The speed of the animal between positions is calculated and the maximum speed is found.
Analysis across time	This measure can be analysed across time.
Units	Metres per second
Notes	The calculation of maximum speed does not use *successive* positions but instead requires that the animal move at least a minimum distance (which is based on the animal's size) and the speed to cover *this* distance is calculated. This method of calculation is used to avoid reporting the speed of movements

that don't constitute locomotion of the animal. For example, if an animal scratches, its centre point may oscillate rapidly but this will not be reported as the animal's maximum speed.

Total freezing episodes

Description	Reports the number of times the animal froze during the test.
Calculation method	Each time the animal begins to freeze, a counter is incremented. The result is the counter's value at the end of the test.
Analysis across time	This measure can be analysed across time. The result for a time period is the number of freezing episodes that *started* during the period.
Units	None
Notes	None

Total time freezing

Description	Reports the total amount of time during the test that the animal was freezing.
Calculation method	The duration of each freezing episode is calculated and these values are summed.
Analysis across time	This measure can be analysed across time. The result for a time period is the amount of time the animal was freezing during the period. If the animal is freezing at the start of the period, then the result includes the time until the animal stops freezing (or the period ends). This means that the total freezing episodes during a time period can be zero when the time freezing during the period is non-zero.
Units	Seconds
Notes	None

Latency to start of first freezing episode

Description	Reports the latency to the start of the first moment in the test when the animal freezes. If the animal is freezing at the start of the test, this value will be zero.
Calculation method	The test time when the first freezing episode occurs.
Analysis across time	This measure can be analysed across time. The result for a time period is the time when the animal *first* froze during the time period. This means that if the animal is freezing at the start of the time period, then the latency is not

	reported as zero.
Units	Seconds
Notes	None

Average freezing score

Description	Reports the average of the animal's freezing score.
Calculation method	The freezing scores are summed and divided by their count.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	This measure is most useful when applied to time segments, as comparison of the score between segments gives an indication of the changes in the animal's activity across time.

Total time mobile

Description	Reports the amount of time the animal was mobile during the test.
Calculation method	Calculated by subtracting the Total time immobile from the Test duration.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Test duration of the period minus Total time immobile during the period. Note: The 'Test duration' for a period is the amount of the test duration which fell in the period - this is the period's duration for all periods except the last one in the test.
Units	Seconds
Notes	None

Total time immobile

Description	Reports the amount of time the animal was immobile during the test.
Calculation method	Sums the duration of each immobile episode in the test. The definition of immobility depends on the current tracking options - see Detecting immobility.
Analysis across time	This measure can be analysed across time. For any time period, the result is the sum of the duration of each immobile episode in the period. Episodes of immobility which fall partly in a period but which start or end outside it are calculated as if they started or ended at the start or end of the period, respectively. This means that it is possible to have a result for Total

	immobile episodes in the period which is zero and a result for *Total time immobile* in the period which is not zero. For example, if the animal is immobile at the start of a period and remains immobile throughout the period then the *Total time immobile* in the period will be the period's duration but the Total immobile episodes in the period will be zero because no transition from a mobile state to an immobile state occurred during the period.
Units	Seconds
Notes	None

Total mobile episodes

Description	Reports the number of times the animal was mobile during the test.
Calculation method	Counts the number of transitions from an immobile state to a mobile state during the test. For the purposes of this calculation, the animal is assumed to be immobile at the start of the test. The definition of immobility depends on the current tracking options - see Detecting immobility.
Analysis across time	This measure can be analysed across time. For any time period, the result is the count of the number of transitions from an immobile state to a mobile state during the period.
Units	Seconds
Notes	None

Total immobile episodes

Description	Reports the number of times the animal was immobile during the test.
Calculation method	Counts the number of transitions from a mobile state to an immobile state during the test. For the purposes of this calculation, the animal is assumed to be mobile at the start of the test. The definition of immobility depends on the current tracking options - see Detecting immobility.
Analysis across time	This measure can be analysed across time. For any time period, the result is the count of the number of transitions from a mobile state to an immobile state during the period.
Units	Seconds
Notes	None

Latency to start of first mobile episode

Description	Reports the latency to the start of the first moment in the test when the animal is considered to be mobile. If the animal is mobile at the start of the test, this measure's value will be zero.
Calculation method	The test time when the animal first becomes mobile.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Latency to start of first immobility episode

Description	Reports the latency to the start of the first moment in the test when the animal is considered to be immobile. If the animal is immobile at the start of the test, this measure's value will be zero.
Calculation method	The test time when the animal first becomes immobile.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Total time active

Description	Reports the amount of time the animal was active during the test.
Calculation method	Calculated by subtracting the Total time inactive from the Test duration.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Test duration of the period minus Total time inactive during the period. Note: The 'Test duration' for a period is the amount of the test duration which fell in the period - this is the period's duration for all periods except the last one in the test.
Units	Seconds
Notes	None

Total time inactive

Description	Reports the amount of time the animal was inactive during the test.
Calculation method	Sums the duration of each inactive episode in the test.

Analysis across time This measure can be analysed across time. For any time period, the result is the sum of the duration of the inactive episodes in the period.

Episodes of inactivity which fall partly in a period, but which start or end outside it, are calculated as if they started or ended at the start or end of the period, respectively. This means that it is possible to have a result for Total inactive episodes in the period which is zero and a result for *Total time inactive* in the period which is not zero. For example, if the animal is inactive at the start of a period and remains inactive throughout the period, then the *Total time inactive* in the period will be the period's duration, but the Total inactive episodes in the period will be zero (because no transition from an active state to an inactive state occurred during the period).

Units Seconds

Notes Inactivity is defined as NOT activity. An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Total active episodes

Description Reports the number of times the animal was active during the test.

Calculation method Counts the number of transitions from an inactive to an active state. For the purposes of this calculation, the animal is assumed to be inactive at the start of the test.

Analysis across time This measure can be analysed across time. For any time period, the result is the number of transitions from an inactive to an active state that occurred during the period.

Units None

Notes An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Total inactive episodes

Description Reports the number of times the animal was inactive during the test.

Calculation method Counts the number of transitions from an active to an inactive state. For the

	purposes of this calculation, the animal is assumed to be active at the start of the test.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of transitions from an active to an inactive state that occurred during the period.
Units	None
Notes	Inactivity is defined as NOT activity. An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Longest active episode

Description	Reports the duration of the longest continuous period of activity during the test.
Calculation method	The duration of each episode of activity is calculated when the episode ends. The longest one is found.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest episode of activity that occurred during the period.
Units	Seconds
Notes	An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Shortest active episode

Description	Reports the duration of the shortest continuous period of activity during the test.
Calculation method	The duration of each episode of activity is calculated when the episode ends. The shortest one is found.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest episode of activity that occurred during the period.
Units	Seconds
Notes	An animal is defined to be active if it is either mobile OR it's performing some

other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Longest inactive episode

Description	Reports the duration of the longest continuous period of inactivity during the test.
Calculation method	The duration of each episode of inactivity is calculated when the episode ends. The longest one is found.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest episode of inactivity that occurred during the period.
Units	Seconds
Notes	Inactivity is defined as NOT activity. An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Shortest inactive episode

Description	Reports the duration of the shortest continuous period of inactivity during the test.
Calculation method	The duration of each episode of inactivity is calculated when the episode ends. The shortest one is found.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest episode of inactivity that occurred during the period.
Units	Seconds
Notes	Inactivity is defined as NOT activity. An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the immobility detection element specifies that mobility should NOT be detected, then activity analysis will be based purely on the performance of other behaviours.

Rotations of the animal's body

Description	Reports the number of times the animal's body completed an entire rotation of 360°.
Calculation method	The animal's centre point is taken as a virtual origin; i.e. this origin is adjusted to be in the same place in each frame. A line is then taken from the animal's centre point to its head creating a vector. The angle between successive vectors is calculated and while the angle continues to have the same sign, the angles are accumulated - when the accumulated angle reaches 360°, the animal has completed a rotation. In fact this is a simplification, as the exact method used takes partial reversals of direction into account - see the figure below.

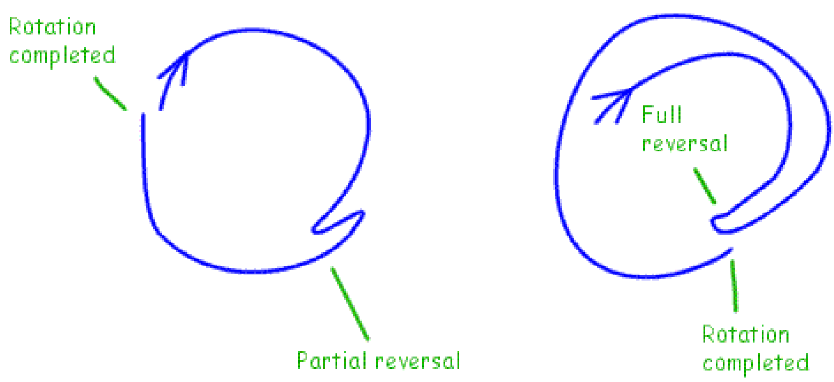


Figure 2. A partial reversal in direction doesn't alter the end of the rotation. A complete reversal however, means the animal has to rotate back to the reversal point to complete the rotation.

Analysis across time	This measure can be analysed across time. A rotation is deemed to occur at the time it is completed.
Units	None
Notes	None

Clockwise rotations of the animal's body

Description	Reports the number of times the animal's body completed an entire rotation of 360° in a clockwise direction.
Calculation method	This measure is calculated in the same way as rotations of the animal's body

	only it reports just the clockwise rotations.
Analysis across time	This measure can be analysed across time. A rotation is deemed to occur at the time it is completed.
Units	None
Notes	None

Anti-clockwise rotations of the animal's body

Description	Reports the number of times the animal's body completed an entire rotation of 360° in an anti-clockwise direction.
Calculation method	This measure is calculated in the same way as rotations of the animal's body, except that it reports only the anti-clockwise rotations.
Analysis across time	This measure can be analysed across time. A rotation is deemed to occur at the time it is completed.
Units	None
Notes	None

Absolute turn angle

Description	Reports the sum of the absolute angle between each movement vector of the animal.
Calculation method	A vector of movement from one position of the animal's centre point to the next is created. For each vector, the angle between it and the previous vector is calculated with anti-clockwise movement being negative and clockwise movement being positive (i.e. the angle is from -180° to 180°). The absolute value of this angle is summed for all the positions of the animal throughout the test or time period.
Analysis across time	This measure can be analysed across time. The result is the based on just those positions within the specific time period.
Units	Degrees
Notes	From this measure, it is easy to use calculations to derive measures such as *Meander* and *Angular velocity*. The former is the *Absolute turn angle* divided by the *Total distance travelled* and the latter is the *Absolute turn angle* divided by the *Test duration*.

Absolute head turn angle

Description	Reports the cumulative absolute angle through which the animal's head moved. For example, if the animal moved its head 30° to the left and then moved its head 45° to the right, the absolute head turn angle would be 75°.
Calculation method	For each position of the animal's head a vector is created from the animal's centre point to the head. The angle between this vector and the same vector for the previous position of the animal's head is calculate and the absolute value of this angle is summed throughout the test.
Analysis across time	This measure can be analysed across time.
Units	Degrees
Notes	None

Number of rears

Description	Reports the number of times the animal reared.
Calculation method	Counts the number of times the animal started to rear.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Total time rearing

Description	Reports the total amount of time for which the animal was rearing.
Calculation method	Sums the duration of each bout of rearing that occurred during the test. If the animal was rearing at the end of the test, then the last bout of rearing ends with the test end.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Latency to first rear

Description	Reports the latency to the first time that the animal reared.
Calculation method	The time when the first bout of rearing started.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Average duration of a rear

Description	Reports the average duration of the rearing bouts.
Calculation method	The result of Total time rearing divided by Number of rears.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Maximum duration of a rear

Description	Reports the duration of the longest bout of rearing.
Calculation method	The duration of each bout of rearing is calculated and the longest bout is found.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Minimum duration of a rear

Description	Reports the duration of the shortest bout of rearing.
Calculation method	The duration of each bout of rearing is calculated and the shortest bout is found.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Path efficiency

Description	This measure represents an index of the efficiency of the path taken by the animal to get from the first position in the test to the last position. A value of 1 indicates perfect efficiency - the animal moved in a straight line - values less than 1 indicate decreasing efficiency.
Calculation method	The straight line distance between the first position in the test and the last position is divided by the total distance travelled by the animal during the test.
Analysis across time	This measure cannot be analysed across time.
Units	None
Notes	This measure is intended for use in water-maze experiments, but is available in all tests.

Number of line crossings

Description	Reports the number of times the animal's centre point moved from one area of the apparatus map to another - i.e. crossed the lines which constitute the map.
Calculation method	The apparatus is divided into unique areas by the apparatus map. For each animal position recorded in the experiment, the area which contains the animal's centre point is found. Each time this changes the measure's value if increased by 1.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	This measure DOES NOT count transitions between ZONES. It counts transitions between areas of the apparatus map, irrespective of whether the areas are part of a zone. This measure is primarily intended to provide an easy way to measure 'grid line crossings' in a similar way to that commonly used

manually - viz: A regular grid is drawn on the apparatus and the experimenter counts the number of times the animal moves from one grid square to another. It's important to understand that ANY-maze uses the animal's centre point when calculating the measure and therefore it can be prone to 'spurious entries' if an animal straddles a line between two areas (i.e. by moving a very small amount, the animal can apparently cross a line many times). This problem can be overcome by setting one zone for each area, using the percentage of the animal that's in the zone to score zone entries, and then using a calculation to sum all the entries into these zones.

On/off inputs positive reversal

Description	The number of times the sequence in which on/off inputs were being activated changed from a decreasing sequence to an increasing one.
Calculation method	This measure is only applied to on/off inputs which have index values defined. As an input is activated, the system checks to see whether the index value of the newly activated input is greater or less than the index value of the previously active input. If the newly activated input has a lower index value, then the inputs are being activated in a decreasing sequence; whereas if it is higher then they are being activated in an increasing sequence. Once the 'direction' of the sequence has been determined, then any change in direction is deemed to be a *reversal*. This measure counts the number of times the sequence changes from decreasing to increasing.
Analysis across time	This measure can be analysed across time. The result for a time period is the number of positive reversals which occurred during the time period.
Units	None
Notes	This measure will only be available if the protocol includes two or more On/off inputs which have index values defined.

On/off inputs negative reversal

Description	The number of times the sequence in which on/off inputs were being activated changed from an increasing sequence to a decreasing one.
Calculation method	This measure is only applied to on/off inputs which have index values defined. As an input is activated, the system checks to see whether the index value of the newly activated input is greater or less than the index value of the previously active input. If the newly activated input has a lower index value, then the inputs are being activated in a decreasing sequence; whereas if it is higher then they are being activated in an increasing sequence. Once the

'direction' of the sequence has been determined, then any change in direction is deemed to be a *reversal*. This measure counts the number of times the sequence changes from increasing to decreasing.

Analysis across time This measure can be analysed across time. The result for a time period is the number of negative reversals which occurred during the time period.

Units None

Notes This measure will only be available if the protocol includes two or more On/off inputs which have index values defined.

Tracking quality

Description Classifies the tracking in a test as either *Good* or *Poor*.

Calculation method The determination of tracking quality is based on two things: how many frames captured by the camera were successfully tracked, and the total amount of time that the animal was not tracked for. The actual calculation is complex as it takes into consideration the duration of each non-tracked period, with long periods being seen as much worse than short periods. For example, in a 2-minute test the animal might not be tracked for half a second twenty times during the test - however, ANY-maze would never have 'lost' the animal for more than half a second so the tracking errors this would cause would be minor. On the other hand if in a 2-minute test the animal was not tracked for a single period of 10 seconds then the *total* time the animal was 'lost' would be the same as in the first example, but in this case the tracking error could be quite significant (the animal could have moved quite a lot in 10 seconds). The calculation also takes into consideration the duration of the test. For example. if an animal is 'lost' for 30 seconds in a 2-minute test then that is quite significant, but if it was lost for 30 seconds in a 24 hour test then probably this would make no real difference to the test results.

Units None

Notes ANY-maze's assessment of the tracking as *Good* or *Poor* is intended as a guide to the experimenter. You may, for example, wish to review tests ANY-maze classifies as having poor tracking.

This measure can be useful when the option to remove jumps is switched on, as the removal of jumps can leave a test with long untracked periods, which will cause the test's tracking to be classified as *Poor*.

Number of centre positions tracked

Description Reports the number of positions of the centre of the animal that were tracked and recorded in the results. Note that ANY-maze may have *tracked* the centre of the animal in more positions than are reported here, because not all position are necessarily stored in the test's results - this depends on the setting for the maximum number of positions to record each second.

Calculation method Counts the number of centre positions in the results.

Analysis across time This measure can be analysed across time. The result is the number of positions of the centre of the animal that fall within the time period.

Units None

Notes None

Number of head positions tracked

Description Reports the number of positions of the head of the animal that were tracked and recorded in the results. Note that ANY-maze may have *tracked* the head of the animal in more positions than are reported here, because not all position are necessarily stored in the test's results - this depends on the setting for the maximum number of positions to record each second.

Calculation method Counts the number of head positions in the results.

Analysis across time This measure can be analysed across time. The result is the number of positions of the head of the animal that fall within the time period.

Units None

Notes The number of tail positions tracked is identical to the number of head positions tracked.

RAPC - Type 1 errors

Description Reports the total number of *Type 1 errors* in the RAPC apparatus. A *Type 1 error* occurs when the animal tries to open a door that is latched shut.

Calculation method The number of door 'openings' for all the doors in the RAPC apparatus is analysed (note that doors which are latched shut will still be registered as 'opening' when the animal pushes against the door, because the door will move a few millimetres). The last door opened in each chamber is, necessarily, the non-latched door; therefore the other doors must be latched. The number of openings of the latched doors is summed and this is the total number of *Type 1 errors*.

Units None

Notes This measure is only available if the protocol includes 12 'switch inputs' with 'index' values of 1 - 12.

RAPC - Type 2 errors

Description	Reports the total number of *Type 2 errors* in the RAPC apparatus. A *Type 2 error* occurs when the animal opens a non-latched door but does not go though it into the next chamber.
Calculation method	The number of door 'openings' for all the doors in the RAPC apparatus is analysed (note that doors which are latched shut will still be registered as 'opening' when the animal pushes against the door, because the door will move a few millimetres). The last door opened in each chamber is, necessarily, the non-latched door; therefore the other doors must be latched. The number of openings of the non-latched door for each chamber less 1 is the number of *Type 2 errors* for that chamber. The sum for all the chambers is the total number of *Type 2 errors*.
Units	None
Notes	This measure is only available if the protocol includes 12 'switch inputs' with 'index' values of 1 - 12.

RAPC - Door sequence

Description	Reports the sequence of non-latched doors in the RAPC apparatus, where the doors in each chamber are numbered 1 through 3. Thus a value of 1321 would mean that door 1 between the first and second chamber was not latched, door 3 between the second and third chamber was not latched, and so on.
Calculation method	The number of door 'openings' for all the doors in the RAPC apparatus is analysed (note that doors which are latched shut will still be registered as 'opening' when the animal pushes against the door because the door will move a few millimetres). The last door opened in each chamber is, necessarily, the non-latched door.
Units	None
Notes	This measure is only available if the protocol includes 12 'switch inputs' with 'index' values of 1 - 12.

See also:

- Information measures
- Zone measures

- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

OPAD measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for OPAD:

- Temperature when contact broken
- Number of non-lick contacts
- Temperature of interest: time in contact
- Temperature of interest: number of times contact broken
- Temperature of interest: number of times contact made
- Temperature of interest: number of licks

Temperature when contact broken

Description	The average temperature at which the animal broke contact with the thermal elements.
Calculation method	Each time the animal breaks contact, the temperature of the thermal elements is noted. The result is the average of all the noted values.
Analysis across time	This measure can be analysed across time. The result for a time period is the average temperature at which the animal broke contact. for all the breaks which occurred during the time period.
Units	Seconds
Notes	If OPAD is being used with independent control of the left and right thermal elements, then this measure will be reported independently for the left and right sides.

Number of non-lick contacts

Description	Reports the number of licks that occurred when the animal was not making contact with the thermal elements.
Calculation method	Each time the animal licks, if it is not in contact with the thermal elements, a counter is increased.

Analysis across time	This measure can be analysed across time.
Units	None
Notes	None

Temperature of interest: time in contact

Description	Reports the amount of time the animal was in contact with the thermal elements when the elements' temperature was within the bounds of the specific temperature of interest.
Calculation method	Each time the animal makes contact and the thermal elements are within the bounds of the temperature of interest (or when the animal is already in contact and the temperature changes to be within the bounds of the temperature of interest), a timer is started. Whenever the animal breaks contact (or the animal is in contact and the temperature changes to be outside the bounds of the temperature of interest), then the timer is stopped. The duration of each timed bout is summed to give the total time in contact.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	If OPAD is being used with independent control of the left and right thermal elements, then this measure will be reported independently for the left and right sides.

Temperature of interest: number of times contact broken

Description	Reports the number of times that the animal broke contact with the thermal elements when the elements' temperature was within the bounds of the specific temperature of interest.
Calculation method	Each time the animal breaks contact with the thermal elements and the elements are within the bounds of the temperature of interest, a counter is increased.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	If OPAD is being used with independent control of the left and right thermal elements, then this measure will be reported independently for the left and right sides.

Temperature of interest: number of times contact made

Description	Reports the number of times that the animal made contact with the thermal elements when the elements' temperature was within the bounds of the specific temperature of interest.
Calculation method	Each time the animal makes contact with the thermal elements and the elements are within the bounds of the temperature of interest, a counter is increased.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	If OPAD is being used with independent control of the left and right thermal elements, then this measure will be reported independently for the left and right sides.

Temperature of interest: number of licks

Description	Reports the number of licks that occurred while the thermal elements' temperature was within the bounds of the specific temperature of interest.
Calculation method	Each time the animal licks and the thermal elements are within the bounds of the temperature of interest, a counter is increased.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	If OPAD is being used with independent control of the left and right thermal elements, then this measure will be reported independently for the left and right sides.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures

- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Pellet dispenser measures
- Syringe pump measures
- Laser controller measures
- Procedure measures
- Event measures
- Virtual switch measures

Zones

Zones define specific parts of your apparatus for which you want individual results. For example, in a plusmaze you will probably define at least two zones, one for the open arms and one for the closed arms. ANY-maze will then provide results such as *Time in the open arms*, *Number of entries into the open arms* etc.

For more information about zones, refer to the topics below.

- An introduction to zones
- Setting up a zone
- Editing a zone
- Deleting a zone
- Zone measures

An introduction to zones

Introduction

ANY-maze will always calculate results for a range of measures for a test, for example the total distance the animal travelled and the test duration. These measures, however, always apply to the apparatus as a whole, so how do you get results for just a certain part of the apparatus - for example, how long the animal spent in the open arms of a plusmaze?

The answer is that you use Zones. A zone defines a certain part of your apparatus for which you would like individual results. You can define any number of different zones, and ANY-maze will calculate the results separately for each one.

- Defining a zone
- Hidden zones
- Zones which change location
- Defining zone entries

Defining a zone

As you will probably recall, when you draw an apparatus map, you don't just define the border of the apparatus; you also divide it into discrete areas, and these areas will now be used to define your zones. Essentially, all you have to do to define a zone is select the area or areas of the apparatus map which will constitute the zone - see figure 1.

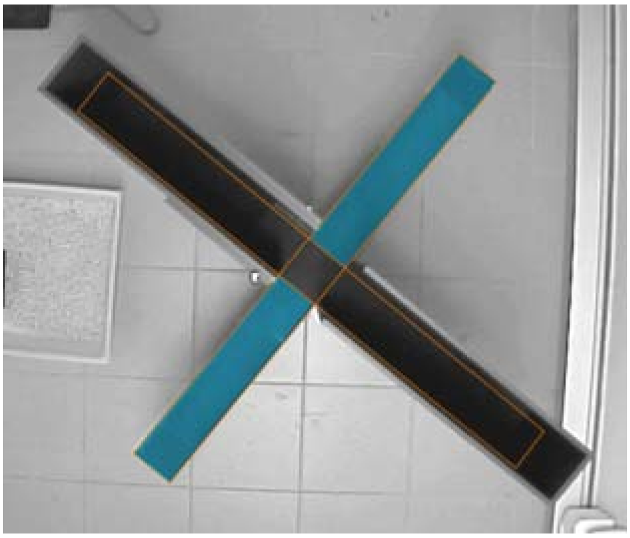


Figure 1. Defining the areas which constitute the 'Open arms' zone of a plusmaze. Note that the zone can consist of more than one area, and that the areas don't have to be contiguous.

There's no limit to how many different areas you can select to create a single zone, nor do the areas have to be contiguous. Furthermore, one area can appear in any number of different zones.

Hidden zones

In some apparatus, you may have areas in which the animal can disappear from the camera's view - a tunnel, for example, or a nest box. In such apparatus you can define a single 'hidden zone', and ANY-maze will assume that if it can't find the animal anywhere in the apparatus, then it must be in the hidden zone. Thus you will be able to score measures such as *Time in the tunnel* and *Number of entries into the tunnel*.

Zones which change location

It's not uncommon to have zones in your apparatus which are sometimes physically located in one part of the apparatus and sometimes in another. For example, a 'Known arm' in a Y-maze may sometimes be on the left and sometimes on the right; or an island in a water-maze might sometimes be in the NW quadrant, sometimes in the SW and sometimes in the NE. Clearly then, there needs to be a way to tell ANY-maze that a zone might move about.

Defining a zone which can change its position is actually very simple - instead of defining a single location for the zone, as described above, you define a number of different locations which the zone can adopt - see figure 2.

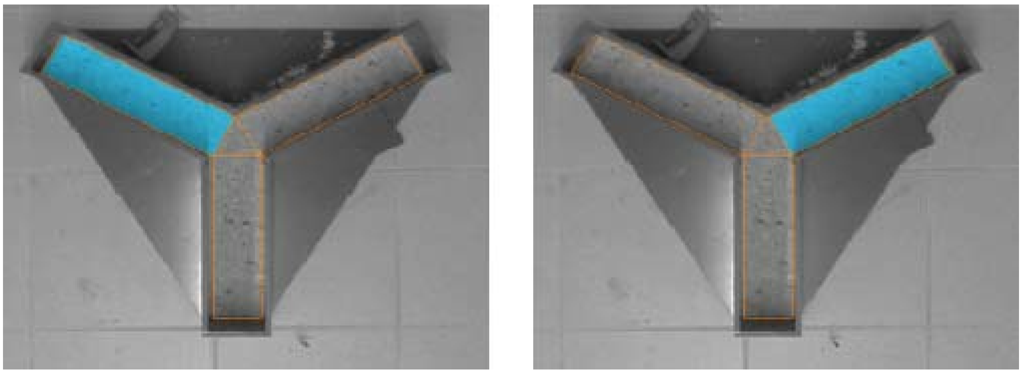


Figure 2. The two possible positions of the 'Known arm' zone in a Y-maze. The zone will always adopt one of these two positions in a test.

Clearly, if a zone can adopt different positions, then ANY-maze will need to know which position the zone is in when it runs a test. For example, is the known arm on the left or the right side of the Y-maze? For experiments in some apparatus, the position of a zone may just depend on the animal which is being tested, whereas in others it might depend on the animal and/or the trial. For example, in a water-maze you might locate the island in the NW for animal 1 and in the SW for animal 2, and throughout the animals' trials these positions won't change - for animal 1 the island will *always* be in the NW, and for animal 2 it will *always* be in the SW. On the other hand, in a T-maze you might alter the reward arm in a certain sequence so that, for example, in an animal's first trial it's on the left and in the second trial on the right.

The importance of this difference in how a zone alters position is related to how you tell ANY-maze where the zone will be:

- In the case where the zone can alter position between animals but not within an animal's trials, as in the water-maze described above, then you can simply tell ANY-maze what the zone's position is for each animal. It will then know when it's testing animal 2, for example, that the island is in the SW.
- In the case where a zone alters position in a fixed sequence, as in the T-maze described above, then you can tell ANY-maze what that sequence is - it will then know, for example, that in an animal's 2nd trial the reward arm is on the right.
- There's also a third possibility, which mixes the two above - in other words, the zone alters position both between the animals and within an individual animal's trials. In this case, ANY-maze will ask you to identify the location of the zone before you perform each test.

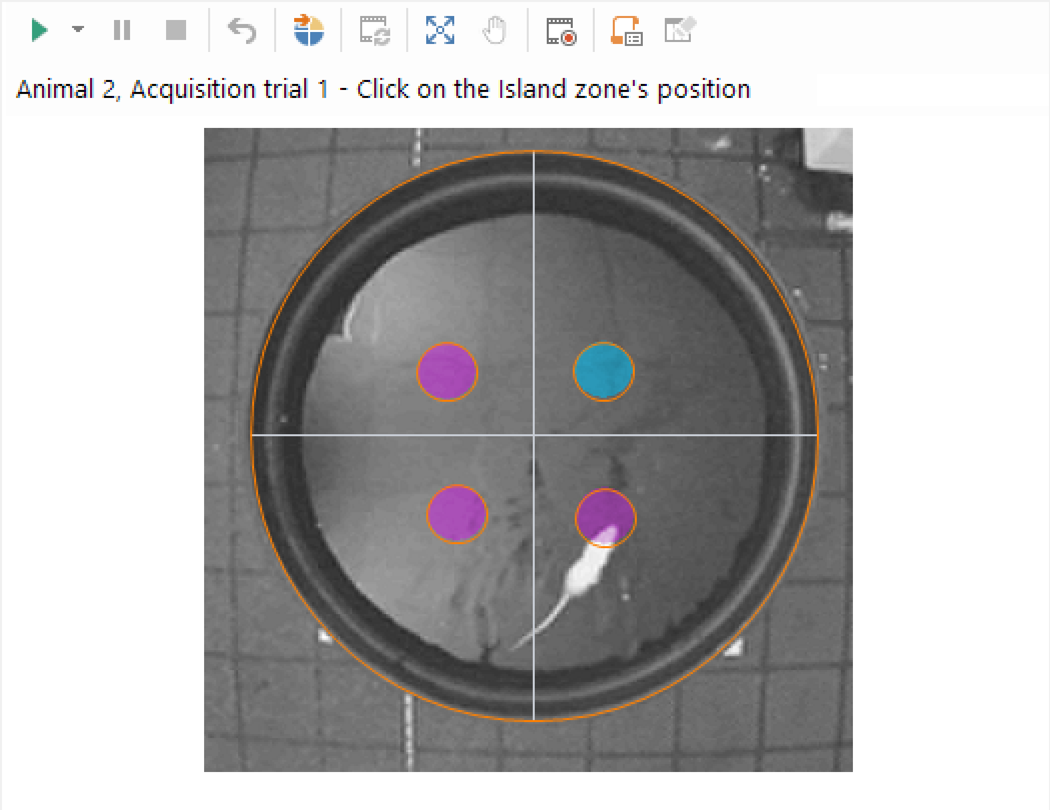


Figure 3. In this water-maze experiment, the position of the 'Island' zone differs both between and within the animals, so ANY-maze asks you to specify the position before each test. The purple areas indicate the possible positions of the island; the blue area indicates the position the user has selected.

ANY-maze also allows you to define relationships between the positions of movable zones. For example, in a Y-maze, you might test an animal twice, once with one of the arms closed off, and a second time with both the arms accessible. Let's assume you call the arm which is always accessible the 'Known arm', and the arm which is only accessible in the second trial the 'New arm'. You then randomly choose which side of the Y-maze will be the 'Known arm' for each animal, and you tell ANY-maze what they are: Animal 1, Known arm is on the left; Animal 2, Known arm is on the right; etc. But this implies that for animal 1, the New arm must be on the right and that for Animal 2, on the left. So rather than also having to tell ANY-maze the position of the New arm for every animal, you can instead just tell it that when the Known arm is on the right, the New arm is on the left and vice versa.

This method of defining the position of one zone based on the position of another is a great time saver, as it reduces the amount of information you have to enter each time you run an experiment - remember, this information is part of the protocol, so you just define it once and reuse it in all your experiments.

Defining zone entries

One of the problems with video tracking systems is that they tend to define an animal's position based on a single point, usually somewhere in the centre of the animal. This is usually fine in apparatus such as a water-maze, where you're interested in such things as the distance the animal travelled, but can create problems in apparatus such as a plusmaze.

Specifically, two problems often occur:

- First, in manual plusmaze tests, experimenters usually define an entry into the arm of a plusmaze as 'all four paws in the arm' - clearly this is not the same as specifying that a loosely-defined centre point should be in the arm.
- Secondly, if the animal stands across the entrance to an arm, the centre point can easily shift from one side of the dividing line (between the arm and the zone) to the other and back again - this type of 'rattle' can create large numbers of spurious zone entries. (Similar problems occur in other apparatus such a T-, Y- and radial-mazes)

ANY-maze addresses both these issues by including the ability to define a zone entry using the entire area of the animal, rather than just its centre point.

Specifically, you can define how much of the animal must enter the zone before an entry occurs - perhaps 85% - and how much of the animal must remain in the zone to *prevent* a zone exit from occurring - perhaps 75%. In this example, then, the value of 85% will equate nicely to the plusmaze '4 paws in the arm' rule - i.e. by the time 85% of the animal is in the arm, its 4 paws probably will be too - see figure 4. And the spurious entries problem will also be fixed, because we're saying that the animal has to get 85% of its body into the zone to count as an entry but then, provided 75% of it stays in the zone it won't exit. Effectively we have a 10% buffer between an entry and an exit, which means that the animal can shift about across the zone division without causing multiple entries.

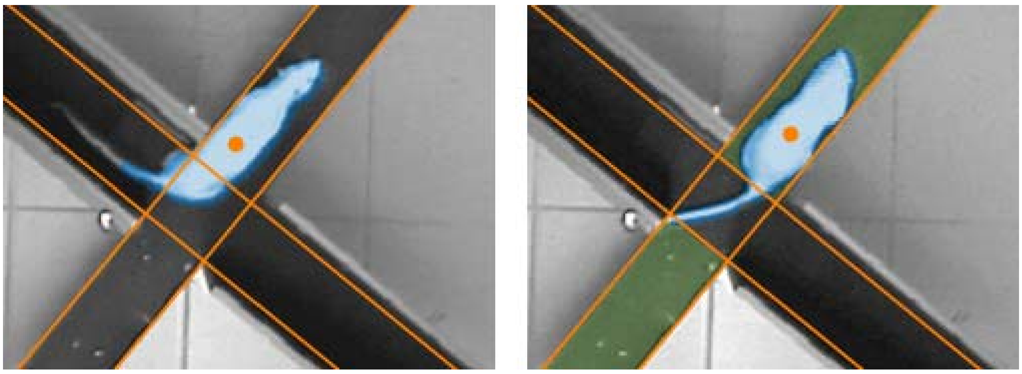


Figure 4. In the first image, the centre of the animal (indicated by the orange dot) has entered the open arm, but ANY-maze doesn't count this as an entry because less than the required amount of the animal - 85% in this case - is in the zone. The blue shading indicates the area that the system considers to be the animal. In the second image, ANY-maze does consider the animal to be in the zone,

which it indicates by shading the zone in green.

ANY-maze also allows you to define zone entries based on the position of the animal's head. This is very useful in things like Novel Object Recognition tests, where you're interested in how much time the animal spends exploring an object.

See also:

- Zone measures

Setting up a zone

Introduction

For a general introduction to zones, see An introduction to zones.

To add a zone to a protocol, click the *Add item* button in the ribbon bar and select *New Zone* from the menu which appears.

The New Zone menu option will be disabled until the protocol includes at least one apparatus item.

Choosing zones

When choosing what zones to include in your protocol, you should consider the *logical* division of your apparatus into zones rather than the *physical* division.

For example, in a T-maze you may want a 'Reward arm' zone which would sometimes be on the left and sometimes on the right side of the T. You *shouldn't* create a 'left side' zone and a 'right side' zone; instead you should create a single 'Reward arm' zone because this is the *logical* division you want - you'll sort out the physical location of the zone when you define it. An easy way to think about this is to consider the results you want - for example, you'd want to know 'Latency to first reward arm entry' and not 'Latency to first left side entry'.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the zone name.
2. Specify whether the zone is a hidden zone.
3. Specify whether the zone is a movable zone.
4. Optionally specify the width of the Whishaw's Corridor for this zone.
5. Choose whether to highlight the zone when the animal is in it.
6. For a zone which is not hidden and not movable, select the areas of the apparatus map which constitute the zone.
7. For a zone which is movable, set up the zone's positions.
8. For zone which is not hidden, choose how ANY-maze should detect entries into the zone.
9. For a hidden zone, choose how ANY-maze should detect exits from a hidden zone.

What next?

After completing these steps, you should consider whether you want to include any points in this protocol.

See also:

- Adding elements to a protocol
- Editing a zone
- Deleting a zone

Zone name

You don't need to include the word 'zone' at the end of the name you enter, because ANY-maze will include this automatically when required. If you do include it, the name may occasionally appear as something like 'North west zone zone'.

In brief

Enter a name for the zone in the *Zone name* field on the zone's settings page. You must make an entry, but it can be anything you like.

Details

Zone names should uniquely and clearly identify a zone, because they'll be used to label results. For example, ANY-maze will report 'Time in the [ZoneName] zone', so if you called the zone 'Zone 1' then the result would be 'Time in the Zone 1 zone'. This would be a rather ambiguous result; something like 'Time in the North West zone' would be much clearer.

To avoid having very long result names, you should try to keep zone names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Setting up hidden zones

In brief

If your apparatus includes areas where the camera can't see the animal - a tunnel or nest box, for example - then you can include *hidden zones* in your protocol. When it can't see the animal, the system will assume it's in one of these zones.

In order to use hidden zones, the tracking must be quite good (i.e. ANY-maze must be able to find the animal in most of the frames) because the software assumes, when it can't find the animal, that it's hidden. This usually won't be an issue, unless you're using apparatus with very low contrast and/or strong reflections.

Setting up a single hidden zone

In most cases where the animal can't be seen by the camera, there's only one place it could have gone - for example, into the dark side of a light/dark box. In these cases setting up a hidden zone is very simple: you should just select the option in the *Hidden zone drop list* (on the Zone settings page) for 'This is the only hidden zone'.

When you do this ANY-maze will automatically select the option *If the animal is hidden at the start of the test then assume it is in this zone,* which makes sense, because if the animal can't be seen at the start of the test then there's only one place it can be - in this zone.

Setting up multiple hidden zones

Imagine you have a light/dark box with a nest box in the light side - as in figure 1, below:

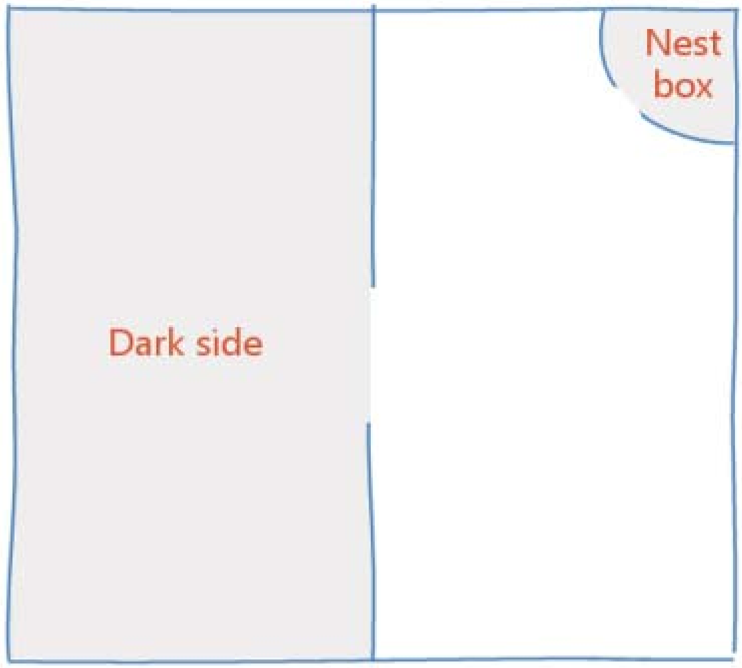


Figure 1. A light/dark box with a nest box in the light side. When the animal is in either the dark side or in the nest box it will be hidden from the camera's view.

Now we have two places where the camera won't be able to see the animal, the dark side and the next box, so (if we want to differentiate the time it spends in these two locations) we will need two hidden zones.

But how will ANY-maze know which hidden zone the animal is in when it can't be seen? The answer is that it considers which zone the animal was in immediately before it disappeared. So, if it was in the 'In front of nest box zone' (see figure 2, below) and then it disappears, it seems reasonable to assume it has gone into the nest box; but if it was in the 'In front of dark side entrance zone' then it seems reasonable to assume it has gone into the dark side.

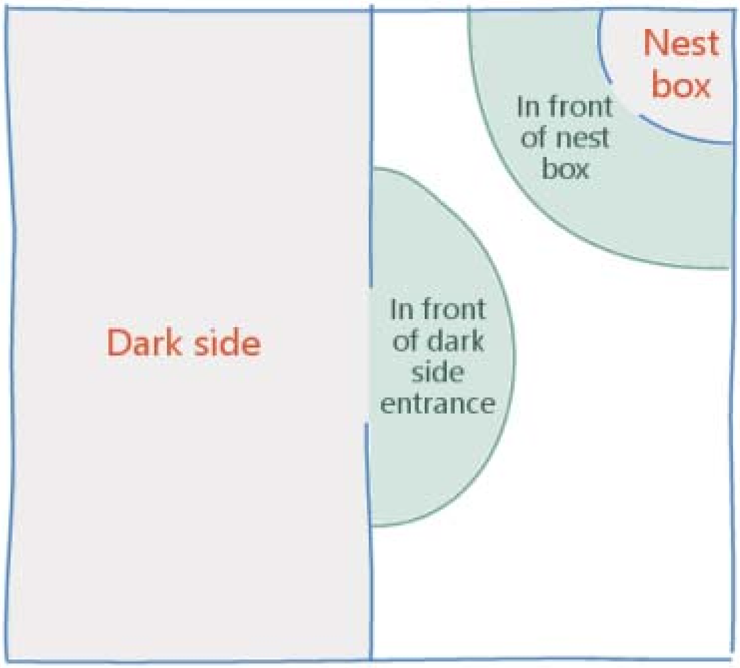


Figure 2. Zones placed in front of the entrances to the dark side and the nest box will allow ANY-maze to determine where the animal has gone when it disappears.

To actually set this up in the protocol you first need to create the zones that the animal will leave in order to enter the hidden zones - so in the above example, these would be a 'In front of nest box zone' and a 'In front of dark side entrance zone'. Then you can create a new zone, call it, for example, 'Nest box' and in the *Hidden zone drop down list* select the option *This is a hidden zone - the animal enters it from the In front of nest box zone.* Do the same thing again for a 'Dark side' hidden zone, and you'll have two hidden zones that ANY-maze can differentiate between.

There is one other thing to consider when setting up multiple hidden zones - what if the animal is hidden at the start of the test? Clearly ANY-maze won't be able to use the last zone the animal was in to determine which hidden zone it is in now, so instead it will simply not consider it to be in any zone at all... unless, you select the option *If the animal is hidden at the start of the test, assume it is in this zone* for one of the hidden zones you have created. (Note that you can only select this option for one zone).

There are a few other things to be aware of when setting up multiple hidden zones:

- You should avoid using very small zones as the zone the animal leaves in order to enter a hidden zone. The reason is that if the animal runs quickly through the zone and into the hidden zone, then ANY-maze may not see it as being in the zone - see figure 3.

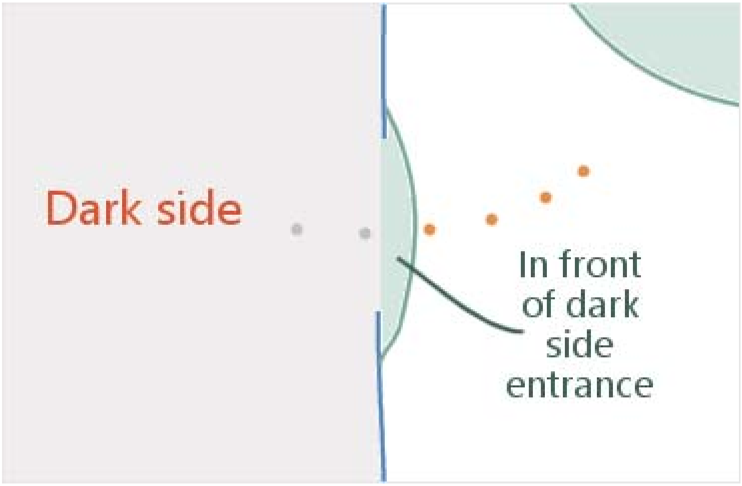


Figure 3. The dots represent the animal's track - i.e. the points at which ANY-maze determined its location. The zone in front of the dark side entrance is so small that, in this case, ANY-maze doesn't see the animal in this zone before it disappears into the dark side. (This effect has been exaggerated in this diagram to aid explanation).

- If, *immediately* before it disappears, the animal was not in any zone that is set to be a 'zone that the animal leaves in order to enter a hidden zone' then ANY-maze WON'T consider it to be in any of the hidden zones, although it WILL consider it to be hidden.
- In theory the animal could be in two hidden zones at the same time. This is possible because it can be in two normal (non-hidden) zones at the same time (as zones are not mutually exclusive). So if a particular area of the apparatus is in both 'Zone A' and in 'Zone B' and the animal can leave 'Zone A' to enter 'Zone Hidden A' and can leave 'Zone B' to enter 'Zone hidden B' then, if it is in this area of the apparatus and then disappears, ANY-maze will report it as being in 'Zone Hidden A' and in 'Zone Hidden B'.

Hidden zones and mobility

When an animal is in a hidden zone, ANY-maze will consider that it is mobile. This is an arbitrary decision, as we can't actually know whether the animal's mobile or not. An effect of this is that the result for the total time mobile will include the time the animal was in the hidden zones. If this doesn't suit your needs, then you can simply use a calculation to deduct the time in the hidden zones from the total time mobile.

Specifying a movable zone

In brief

In some apparatus, you may have a zone which is not always physically located in the same place. For example, the reward arm of a T-maze or the island in a water-maze. You can specify that a zone is one of these *movable zones* by selecting an appropriate option from the *Position...* list on the zone's settings page.

Details

The *Position...* list has a number of different entries which are explained below:

- *The position of the zone remains the same in all the tests*

  This is the option for a fixed (i.e. non-movable) zone. All zones are non-movable by default, and it's by selecting a different option to this one that you make a zone movable. If you want to change a zone back from being movable to being non-movable, simply revert to this option. Note: Changing a zone from movable to non-movable will automatically delete any positions you have defined for it.

- *The position varies between (but not within) the animals*

  Choosing this option indicates that the zone is movable and that it adopts different positions for different animals. However, if each animal has multiple trials, the zone will always be in the same position for any individual animal. For example, you might select this option for the Island in a water-maze. For animal 1, you might place the island in the North West, while for animal 2 it might be in the South West; however, these locations won't change between each animal's trials - i.e. however many trials animal 1 has, the island will always be in the North west for all of them.

- *The position varies within (and possibly between) the animals*

  Choosing this option indicates that the zone is movable and that it can adopt different positions in different trials for the same animal.

  In fact, there are two cases which are both included in this option. The first is where the zone adopts one position in the 1st trial, a different position in the 2nd trial, etc. but these positions are always the same irrespective of the animal being tested - e.g.: For trial 1 the zone is always in the NW, in trial 2 it's always in the SW and so on.

The second possibility is that the zone adopts different positions depending on both the trial AND the animal. For example, for trial 1 for animal 1 it's in the NW, for trial 2 for animal 1 it's in the SW; but for trial 1 for animal 2 it's in the SE and in trial 2 for animal 2 it's in the NW.

- *The position follows the positions of other movable zones*

  Note: This option only appears in the list when the protocol already includes at least one other movable zone.

  This option can be used to create 'virtual' movable zones. Effectively, such zones don't have any physical positions of their own; they just adopt the same position as another movable zone. The important point is that they can adopt the position of different zones in different trials. Here's an example of when this type of zone is useful:

  Imagine you have a place preference box in which you intend to condition animals so that one side of the box is paired with a drug. You decide to balance the drug-paired side between your groups, so for some animals the drug-paired side will be on the left while for others it will be on the right. This drug-paired zone will therefore be a 'classic' movable zone. Clearly, there will also be a non-drug-paired zone, and this would be defined as a movable zone whose position depends on the position of the drug-paired side (see next option below) - essentially, this would just mean that when the drug-paired side is on the left, the non-drug-paired side would be on the right and vice versa. Now, let's say you want to condition the animals over four days, giving them the drug and confining them to the drug-paired side on days 1 and 3, and giving them saline and confining them to the non-drug-paired side on days 2 and 4.

  So that's our procedure, but let's think about which side an animal will be placed in during each conditioning test. For animal 1, for example, you'll need to know which is his drug-paired side - let's say it's the left - and then you'll need to know which trial you're performing - let's say day 2. In this case, day 2 means a non-drug day so this animal will be placed in his non-drug-paired side, i.e. in the right. To avoid having to follow through this logic before each test, you could include in your protocol a 'Start side' zone whose *Position follows the positions of other movable zones*. You would then specify that for trial 1, the Start-side's position is the same as the drug-paired side; for trial 2, it's the same as the non-drug-paired side; etc. Now you could include this zone's position in your Test schedule report and ANY-maze would simply show you something like 'Animal 1, trial 2, Start side = Right'. This would simplify your testing and reduce the risk of mistakes. Also note that in this protocol the only thing you'd have to specify before running experiments would be which side is the drug-paired side for each animal. Remember, it's worth investing effort in your protocol, because you'll be able to reuse it in all your experiments.

- *The position depends on the position of the XXX zone*

  Note: This option only appears in the list when the protocol already includes at least one *other movable zone.*

The important point with this option is that it indicates a relationship between this zone's position and the position of another zone. This is best explained with an example: In a Y-maze, you might define two zones, one called the 'Known arm' and one called the 'New arm', and these two zones could be positioned in either of the two 'arms' of the 'Y'. This would mean that if the 'Known arm' was on the left then the 'New arm' would have to be on the right, and if the 'Known arm' was on the right then the 'New arm' would have to be on the left. In other words, the position of one arm would depend on the position of the other.

The advantage of using this option is that you will only need to tell ANY-maze about the position of one of the zones in order for it to deduce the position of the other one.

When you use this option for a zone, then you have to define the actual dependency when you define each position of the zone - see Setting up zone positions for more details.

Setting zone positions

After you specify that a zone is movable, you will need to add one zone position element to your protocol for each of the positions which the zone can adopt*. For example, for a water-maze island, you might need to add four zone positions - one for the North West position, one for the South West, one for the North East and one for the South East. See Setting up zone positions for more details.

** Except for zones whose 'Position follows the positions of other zones' - these are virtual zones which just borrow their positions from other zones.*

Specifying the width of a zone's Whishaw's Corridor

What is Whishaw's Corridor

Whishaw's Corridor is effectively a rectangular 'zone' of a specific width that runs from the animal's start position in a test to the centre of a zone - usually the escape platform in a water-maze (see figure 1, below). ANY-maze can report measures of the time and distance travelled in this 'corridor zone'.

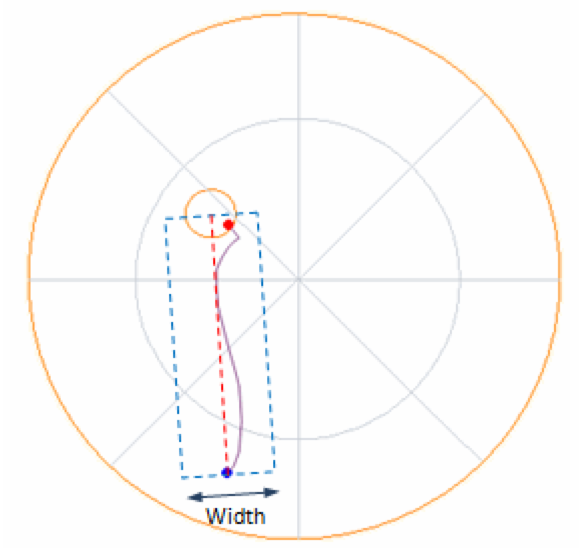


Figure 1. Example of Whishaw's Corridor in a water-maze. The corridor is centred on a line (shown in red) running from the animal's start position to the centre of the platform zone. The corridor itself (shown in blue) has a width specified in the protocol.

Creating a Whishaw's Corridor

In ANY-maze, any zone can have a Whishaw's Corridor associated with it simply by specifying the corridor width in the zone's settings. The width should be specified in millimetres. For example, specifying 200 will create a corridor that's 20cm wide (i.e. 10cm to one side of the line from the animal's start position to the centre of the zone and 10cm to the other side).

If you don't want to include a Whishaw's Corridor for a zone (and other than for water-maze platforms, you probably won't) then just leave the field blank.

See also:

- Zone measures

Specifying zone highlighting

In brief

ANY-maze can indicate when it considers an animal to be in a zone by shading the zone in green. This acts as a useful reassurance that the tracking is working correctly. To select zone highlighting, just check the *Highlight this zone during tracking when the animal is in it* option on the zone's settings page.

Details

Highlighting a zone causes it to be shaded when the system considers the animal to be in the zone. This shading *tints* the video image, without obscuring it. Nevertheless, you may find it's distracting if you're also trying to manually score some behaviours using keys. In this case you might want to switch this option off.

Because ANY-maze zones are not mutually exclusive, it's possible for an animal to be in two zones at the same time - for example, in a water-maze, the animal might be in the 'North West quadrant' and also in the 'Centre' of the maze. If both zones are set to be highlighted and this situation occurs, then they'll *both* be shaded, which can occasionally be a little confusing. In this case, you might want to consider only highlighting one of the zones.

While testing, you can choose to temporarily switch all zone highlighting off (by right-clicking on the video picture and selecting the appropriate option from the menu that appears). However, this will only affect the running test, i.e. the highlighting will be shown again in the following test. To permanently switch highlighting off, you will need to edit the zones' settings in the protocol.

See also:

- Specifying what's being displayed during a test
- Changing ANY-maze colours

Selecting the areas of the apparatus map which constitute a zone

If you have specified that a zone is movable, then you won't be able to select its areas. Instead you should add one Zone position element for each of the positions which the zone can adopt and then select the areas which constitute each position.

In brief

For zones which are not hidden and are not movable, you will need to select the area (or areas) of the apparatus map which constitute the zone. You do this by first ensuring that the relevant zone is selected in the protocol list, and then clicking in the relevant areas in the apparatus map(s) that are shown in the image pane on the right-hand side of the page. The areas that you click will change to be shaded in blue - see figure 1.

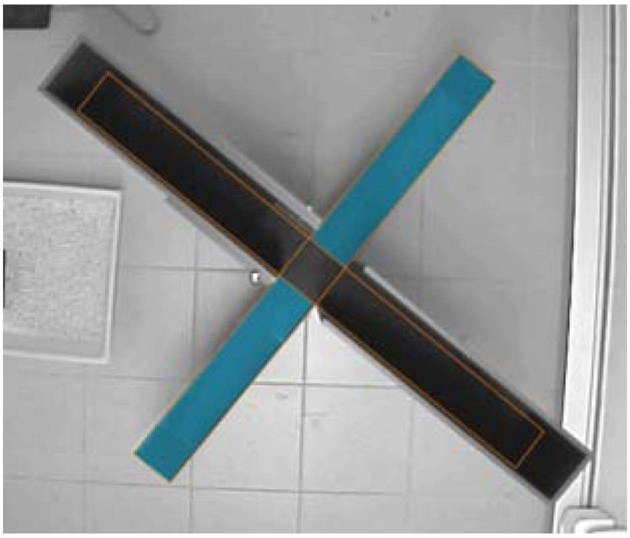


Figure 1. Selecting the areas which constitute the Open-arms zone in a Plusmaze.

Details

Selecting the areas which constitute a zone is very simple - you just click the relevant areas in the apparatus map. To deselect an area, just click it a second time. Of course, if you have more than one piece of apparatus in your protocol, then you will need to specify the zone's areas in each piece of apparatus separately - the Protocol page Image pane will be showing all the apparatus maps, so this is very simple to do.

There are no restrictions on selecting areas, thus:

- One area can appear in more than one zone
- The areas which constitute a zone don't have to be contiguous (i.e. they don't have to be touching each other).

Problems

An area I want to select for a zone is not divided off in the apparatus map

The solution is simply to go back to the apparatus element in the protocol and edit the apparatus map so the area is correctly divided off - see figure 2.

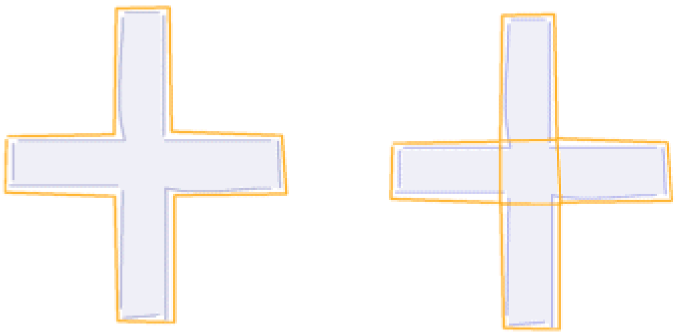


Figure 2. In the apparatus map on the left, the arms of the plusmaze haven't been divided off - so it's impossible to select the areas which, for example, constitute the Open arms. This is resolved in the second picture, where lines have been added to divide the map up.

When I click an area, it AND its neighbouring area are shaded in blue

This can occur when the dividing line between two areas doesn't quite fully separate them. In this case, you should go back to the apparatus element and edit the apparatus map so the areas are separated correctly - see figure 3.

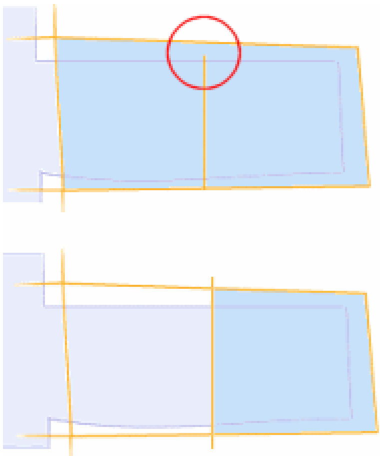


Figure 3. In the first map, the dividing line doesn't quite close off the end of the arm, so ANY-maze considers all of the arm to be a single area. In the second map, the line fully closes off the end of the arm, so it's now considered to be a separate area.

When I click the apparatus map nothing happens

This will occur if you've specified that the zone is a hidden or movable zone. In the former case, the zone simply doesn't have any areas; whereas in the latter, you need to add one Zone position element for each position that the zone can adopt and then select the areas which constitute each of the positions.

Setting up zone positions

Introduction

When you specify that a zone is *movable*, you need to add one *Zone position* element to the protocol for each of the positions that the zone can adopt. This is done by ensuring that the relevant zone is selected in the protocol list, then clicking the *Add item* button and selecting *New Zone Position* from the menu which appears.

The New Zone Position *menu option will be disabled if a zone is not selected in the protocol list, or the zone which is selected is not a movable zone.*

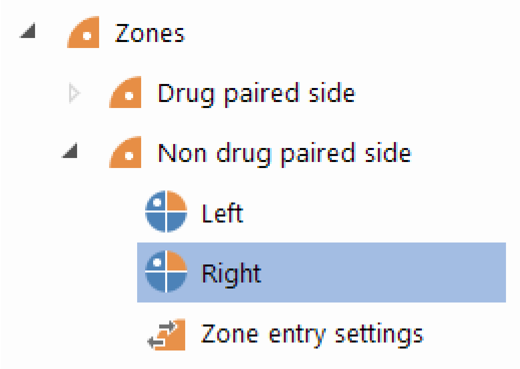


Figure 1. Part of the protocol list showing the positions of the 'Non-drug-paired side' zone in a Place Preference Box protocol.

Selecting the areas of the apparatus map which constitute a position

Selecting the areas which constitute a position is very simple - you just click the relevant areas in the apparatus map. To deselect an area, just click it a second time. Of course, if you have more than one piece of apparatus in your protocol then you will need to specify the position's areas in each piece of apparatus separately - the Protocol page's Image pane will be showing all the apparatus maps, so this is a straightforward process.

There are no restrictions on selecting areas, thus:

- One area can appear in more than one position either of the same zone, or of different zones.
- The areas which constitute a position don't have to be contiguous (i.e. they don't have

to be touching each other).

Specifying when the zone is in this position

In general, you will specify the positions of movable zones either as part of the protocol's Stage elements or when you set up or run an experiment. However, there is one type of movable zone whose positions you can specify when you create it, and this is described here.

One of the choices you have when you specify that a zone is movable is *Position depends on the position of the XXX zone* (where XXX is the name of some other movable zone). If you selected this option when setting up this zone, then you will need to specify, for each position of this zone, how it depends on the position of the XXX zone.

For example, if you are using a Place preference box then you might create two zones, one called 'Drug-paired side' and the other called 'Non-drug-paired side'. As your box only has two sides, if the 'Drug-paired side' is on the left then the 'Non-drug-paired side' must be on the right and vice versa.

So, in this example, you could specify that the position of the 'Non-drug-paired side' is dependent on the position of the 'Drug-paired side'. Therefore, when entering the two positions of the 'Non-drug-paired side' (left and right) you would have to specify when the zone is in each of these positions - for example, when entering the left position you would specify that 'This zone is in this position when the *Drug-paired side* zone is in the position *Right*'.

The advantage of doing this is that you will then only have to tell ANY-maze what the position of the 'Drug-paired side' zone is for each animal - it can then deduce the position of the 'Non-drug-paired side' for itself.

Choosing how ANY-maze should detect entries into a zone

Introduction

All zones include a sub-element in the protocol list called *Zone entry settings*. Selecting this will display options in the Settings pane which allow you to specify exactly how you want ANY-maze to detect entries into the zone.

There are various methods which ANY-maze can use to detect zone entries. The default method is to simply consider an animal to be in a zone when its centre point is in the zone. This, however, doesn't equate very accurately to zone entry rules used in some apparatus - for example, in the plusmaze, it's normal to count an arm entry when 'all four paws are in the arm'. Also, this method can easily create false entries if the animal's position oscillates slightly across the border of a zone.

Using the entire area of the animal to detect entries addresses these issues.

You specify how zone entries will be detected for each zone individually - so there's nothing to stop you using the animal's centre point for some zones, and its entire area for others.

Detecting zone entries using the centre of the animal

This is the default method for detecting zone entries. However, it can give rise to certain problems, as described in the Introduction above, so you should carefully consider whether these issues will affect your apparatus. For example, in an activity cage you might be interested in how much time the animal spends in the left and right sides of the cage, and how far it travels during the test. In this case, there would be no need to precisely control what constitutes an entry into one side or the other, and counting some spurious entries would make almost no difference to the *times* in the two sides. Therefore using the centre of the animal to detect entries would probably be fine.

The big advantage of using the centre of the animal to detect zone entries is that you don't need to record the entire area of the animal, just its centre point, and this will create dramatically smaller experiment files - usually about 95% smaller than recording the animal's area.

Detecting zone entries using the position of the animal's head

Selecting this method for zone entry detection will cause ANY-maze to consider that the animal is in the zone whenever its head is within the zone. Although this method of zone entry detection may seem very appealing, there are a couple of issues to consider before using it.

- As with using the animal's centre point to detect zone entries, this method can easily detect false entries if the animal's head position oscillates slightly across the border of a zone. In fact this is even more likely to occur when using the head position as this is generally less stable than the animal's centre point. Nevertheless, this may not be important as it will only affect the *count* of zone entries and not really alter measures such as the time in the zone, or the distance travelled in the zone.
- In order to track the animal's head, ANY-maze does require:
  - that there's a good level of contrast between the animal and the background of the apparatus,
  - that there's relatively little *noise* in the image (*noise* is that 'snow' type of effect you see in low light images),
  - and that the entire body of the animal is usually visible (which wouldn't be the case in a water-maze, for example).

You'll find that the protocol includes an element where you can choose whether or not ANY-maze should track the animal's head. It is **not** necessary to switch that option on in order to use the head position to detect zone entries, as the two functions are entirely independent.

Detecting zone entries using the entire area of the animal

This method of detecting zone entries requires you to specify two percentages: The first is the percentage of the animal which must *be in the zone for an entry to be counted.* For example, specifying 90% will require almost all of the animal to be in the zone before an entry is counted. The second value is the percentage of the animal which *must remain in the zone to prevent an exit from occurring*. In this case, it's normal to enter a value a little smaller than the *entry percentage* as this helps to prevent *spurious* entries.

There are a number of situations when using this method to detect zone entries is desirable, and some of them are described below:

- *Preventing spurious zone entries*

  A common problem with video tracking systems can occur when an animal straddles the border between two zones. If the system is using the centre point of the animal to count zone entries, and the animal moves even slightly, then the centre point can oscillate from one side of the zone border to the other - each time it does this, the system will count a zone entry. This is demonstrated in figure 1. Here, using the centre point to detect zone entries is approximated to using 50% of the animal to detect the entry. As 50% of the animal moves into the zone, an entry is counted, but if the amount of the animal in the zone drops even slightly then an exit is counted.

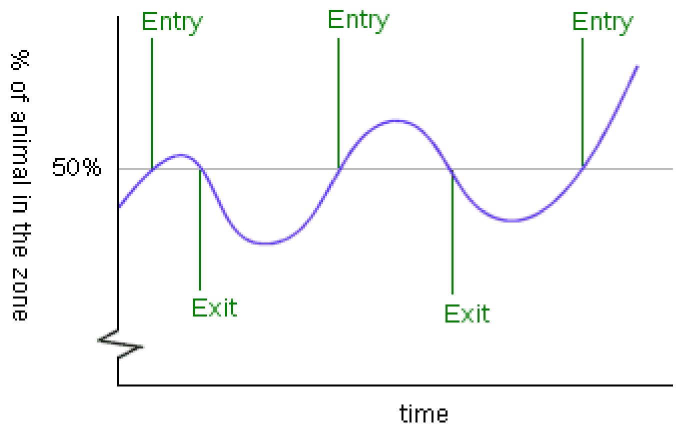


Figure 1. By using 50% of the animal's area to approximate tracking using the centre point, it can easily be seen how spurious zone entries can occur.

The problem of spurious entries can be resolved by using a zone entry *and* exit percentage. For example, setting the entry percentage to, say, 50% and the exit percentage to, say, 40% would resolve the problem shown in figure 1. Here, after the animal moves 50% of its body into the zone, it never moves more than 40% back outside it, so just a single entry is counted - see figure 2.

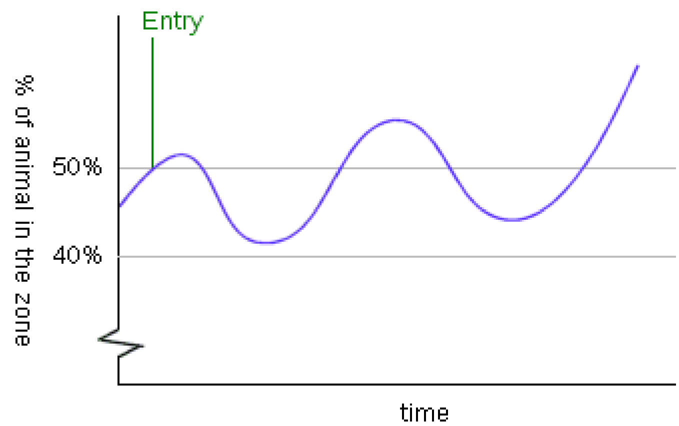


Figure 2. After 50% of the animal's area moves into the zone, an entry is counted. However, an exit will only occur if less than 40% of the animal is in the zone and

this doesn't occur - thus just a single entry is scored.

- *Controlling zone entries more precisely*

  In some apparatus, special rules are generally used to determine when an animal enters a zone. For example, in the plusmaze and Y-maze, an arm entry is only counted when the animal has *all four paws in the arm.* Using the animal's centre point to detect entries makes it difficult, or impossible, to approximate such rules in an automated system. However, using the entire area of the animal to detect entries can overcome this - in the case of the *fours paws* rule, an entry percentage of about 85% usually creates a good approximation - see figure 3.

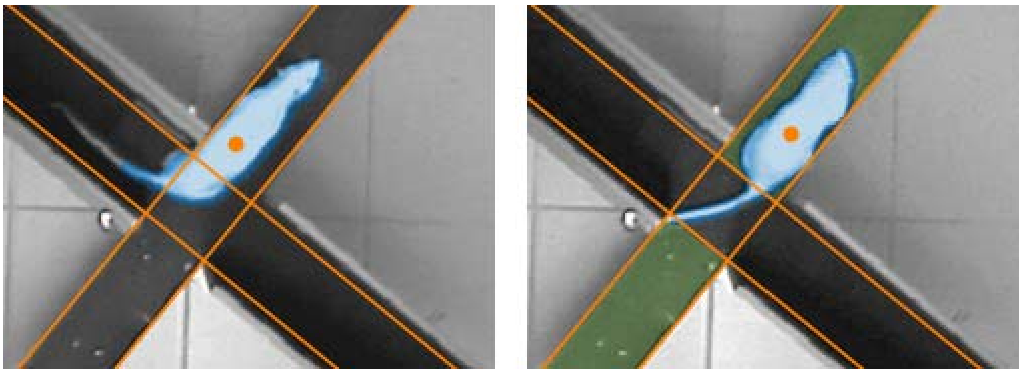


Figure 3. In the first image, the centre of the animal (indicated by the orange dot) has entered the open arm, but ANY-maze doesn't count this as an entry because less than the required amount of the animal - 85% in this case - is in the zone. The blue shading indicates the area that the system considers to be the animal. In the second image, ANY-maze does consider the animal to be in the zone, which it indicates by shading the zone in green. Here, the 85% entry value acts as a good approximation to the plusmaze 'four paws in the arm' rule.

  In this case, you will probably also want to use an exit percentage a little below the entry percentage to avoid spurious entries - 75% in the case of the plusmaze is usually about right.

- *Detecting zone 'touches'*

  Another problem with using the centre of the animal to detect zone 'entries' is that the animal has to be able to actually enter the zone. This might sound rather obvious, but consider the case of a novel object placed in an open field; perhaps a wooden block. In this case you might want to know when the animal touches, or perhaps gets very close to, the object. But unless the animal can climb onto the object, its centre point will never actually enter the zone. Of course, you could draw a wide zone around the object and count a 'touch' when the centre point enters this zone, but it's hard to determine exactly how wide this zone should be. A better solution is to use a very low zone entry percentage, perhaps just 1%, and a zone which tightly encloses the object - this way you can detect when the animal touches the object.

NOTE: ANY-maze electronically 'removes' animals' tails when tracking, so if a tail touches an object it won't be detected.

Why the animal can sometimes be 'nowhere'

A possible side effect of using percentages to detect zone entries is that you can define a protocol in which the animal can be nowhere, and this can sometimes lead to confusion.

Consider a simple open field which is divided into two zones, left and right. You would expect that if you added together all the time the animal spends in the left zone and all the time it spends in the right zone, the total would equal the test duration. And so it would, if you use the centre of the animal to detect the zone entries. But if you use entry/exit percentages, then you can create a situation when the animal will be in neither zone. For example, if the entry percentage for the two zones is 80% and the exit percentage is 70% then, if the animal has 50% of its body in the left zone and 50% in the right zone it won't be considered to be in either of them.

Defining one zone as being a 'not in any other' zone

In most cases, it's quite easy to choose the entry conditions for a zone, but it can become difficult or impossible to do this if you have a small zone surrounded by zones that use 'percentage' entries - the centre of a plusmaze, for example.

Consider the situation where the entry conditions for a plusmaze are set to 60% for an entry and 50% for an exit (see above). Now try to think what conditions you should use for an entry into the centre of the maze. An entry percentage of 50% and an exit of 40% might seem logical in this case, but the animal could quite easily put some of its body in an open arm and some in a closed arm, and have less than 40% in the centre - it would then be scored as not in the centre.

Of course, you could overcome this by simply not including a centre zone and calculating the results for the centre (for example time in centre = total time - time in the other zones). But this gets complicated for entries, and is impossible for latencies.

So the solution is to simply define the centre as 'not in any other zone'. As you'd expect, ANY-maze will then consider the animal to be in the zone when it isn't in any of the others.

You don't need to specify the areas of a 'not in any other' zone

By definition, ANY-maze knows where in your apparatus your zones are. Therefore, if you include a 'not in any other' zone, it will simply work out for itself which areas the zone includes.

One effect of this is that the 'not in any other' zone may move about, if your apparatus includes some movable zones. For example, if you have a target arm zone in a T-maze that might be on the left or might be on the right, then, if you define just this target zone and another 'not in any other' zone, then the 'not in any other' zone would sometimes consist of the stem of the T plus the left arm and sometimes the stem plus the right arm, depending on where the target was.

Using key presses to specify when the animal is in a zone

In some apparatus, it may simply be impossible to track the animal automatically, in which case you can use keys on your keyboard to indicate to ANY-maze when the animal has entered a zone.

You may think that you could achieve the same result using the Key element of the protocol, and you'd almost be right; however there are a number of benefits to specifically telling ANY-maze that you're using a key to score a zone entry:

- The system will automatically score an exit when the animal moves to any other zone, whether that's a zone you're scoring with keys or not.
- The system will be aware that the animal is in a ZONE, and this will be reflected in the results, where it will break key measures, such as grooming, into *Grooming in the X zone*, *Grooming in the Y zone* and so on.
- The system will report zone specific measures such as *Distance to the zone*.
- You can choose to highlight the zone when the animal is in it (i.e. when you press the key).

Setting up a key zone

To actually define a key zone, you simply need to check the appropriate box and then specify which key you want to use to signify an entry into the zone. If you have multiple apparatus, you'll need to specify one key for each of them. Of course, the keys you choose must be unique both within other key entry zones you've set up and also within any *Keys* in your protocol - don't worry too much though, because ANY-maze will tell you if you try to use the same key twice.

When using a key zone, the option to only score an exit from the zone when the animal enters another zone will be disabled. This is because key zones always score exits either when the animal enters another zone, or when you press the Esc key.

Although you'll be using a key to tell ANY-maze when the animal is in the zone, you should still indicate where the zone is within the apparatus by clicking on the relevant areas in the video picture. Note that ANY-maze *won't* use the area to determine entries or exits to/from the zone, but it will use it to report some measures, such as the distance of the animal from the zone, and it will highlight the area when the animal is in the zone (assuming the highlight option is selected). If you can't select the area of the zone because the zone isn't visible to the camera, then you should probably be using a hidden zone rather than a key zone.

Using a key zone

Actually using a key to score a zone entry is very simple - just press the key when the animal enters the zone; there's no need to hold it down. An exit will be scored when the animal moves to any other zone or, if you want to explicitly score an exit, when you press the Esc key on the keyboard.

Only scoring a zone exit when the animal enters another zone

Ordinarily, ANY-maze will score a zone exit based on the same method it uses to score a zone entry. For example, if the you use the centre point of the animal to score an entry into a zone, then, when the centre point exits the zone, an exit will be scored.

However, you might not want things to work like this. For example, in the plusmaze, you might prefer to ignore the exit from one arm until the animal enters another arm, thus avoiding the question of what to do with the time the animal spends in the centre of the maze.

If you want a zone to have its exits scored like this, you just need to check the box *Only score an exit from this zone when the animal enters another zone.* Thus ANY-maze will continue to consider that the animal is 'in' the zone (and, if set to do so, will continue to show the zone highlighted) until it satisfies the entry criteria for another zone. This can even mean that the animal is considered to be in the zone when no part of its body is physically in the zone's area.

By the way, if your protocol includes a zone with the entry criteria of *Not in any other zone* (let's call it zone A) and all the other zones have exit criteria of *Only score an exit when the animal enters another zone,* then ANY-maze will never consider that the animal has entered zone A, because it will only leave one zone when it enters another and thus will never be *Not in any other zone*.

Ignoring the apparent entry that occurs when an animal starts in a zone

When you start a test, the animal will (from ANY-maze's point of view) suddenly appear in the apparatus. Of course, it might *appear* in one of your zones, and ordinarily ANY-maze will count this as a zone entry.

However, you might wish to ignore this *apparent* entry. and only count an entry into the zone when the animal first enters it by actually crossing a border. If this is the case in your apparatus, then you should check the box titled *Don't score any results in this zone until the first 'true' entry*.

NOTE: Because ANY-maze won't count the entry into the zone, it won't consider the animal to be in the zone, therefore it won't update any of the zone's results - including such things as *Distance travelled in the zone, Time in the zone,* etc. However, it will only ignore these results in relation to the *zone*, not in relation to the apparatus as a whole, so measures such as *Test duration* and *Total distance travelled* **will** reflect the animal's activity in the zone.

Altering zone entry settings

You can change any aspect of the zone entry settings at any time. For example, after an experiment is complete, you might realise that your results include a number of spurious zone entries (as described above). To resolve this, you could simply alter the zone's entry settings to use an entry and exit percentage - all the zone results would then automatically update to reflect the change.

There is, however, one caveat to this ability to alter zone entry settings: Although you can change from using the centre of the animal to detecting entries using its entire area, the change won't affect any tests in which you didn't *record* the animal's 'images', (see What to record while testing). For these tests, ANY-maze will always use the centre point because it's the only information it has.

Choosing how ANY-maze should detect exits from a hidden zone

Introduction

All hidden zones include a *sub-element* in the protocol list called *Zone exit settings*. Selecting this will display options in the Settings pane that allow you to specify exactly how ANY-maze detects exits from the hidden zone.

- Specifying when ANY-maze should consider the animal to have exited a hidden zone
- Ignoring the apparent hidden zone entry that occurs when an animal starts a test hidden
- Altering hidden zone exit settings

Specifying when ANY-maze should consider the animal to have exited a hidden zone

When an animal exits a hidden zone it will, necessarily, become progressively more visible to the system. In some situations, you might want to consider the animal to have exited the hidden zone as soon as any part of it becomes visible, whereas in other cases you might want to ignore the exit until the entire animal has exited the hidden zone.

To control how ANY-maze detects hidden zone exits, you simply need to specify the amount of the animal that needs to be visible for the system to score an exit. Thus entering a value such as 15% will cause the system to consider that the animal has exited the hidden zone as soon as almost any of it becomes visible, whereas entering 100% will cause the system to wait until the entire animal has become visible before the exit is scored - this latter case means you would ignore partial exits.

As well as specifying how much of the animal needs to become visible for an exit to occur, you may also need to specify the approximate size of the animals you will be tracking. This is only actually necessary if the animals *might* start a test in the hidden zone. In this case, as ANY-maze will never have seen the animal, it won't know how large it is and therefore the concept of '15% of the animal's size' will be meaningless. To overcome this, you should specify the approximate size of the animal, excluding its tail - the program will then use this value to determine how large '15% of the animal' (or whatever value you specify) actually is.

As a rough guide, a value of 60mm for mice and 180mm for rats is usually about right, although of course these depend on the age, strain, etc. of the animals you are using.

If you know that the animal will never start a test in the hidden zone, then you can leave this value blank.

Ignoring the apparent hidden zone entry that occurs when an animal starts a test hidden

When you start a test, the animal may already be in the hidden zone. In this case, ANY-maze will immediately score an entry into the hidden zone.

However, you might wish to ignore this *apparent* entry, and only count an entry into the hidden zone when the animal first enters after actually becoming visible. If this is the case in your apparatus, then you should check the box titled *Don't score any results in this zone until the first 'true' entry*.

NOTE: Because ANY-maze won't count the entry into the zone, it won't consider the animal to be in the zone - therefore it won't update any of the zone's results, including such things as *Time in the zone* etc. However, it will only ignore these results in relation to the *zone,* not in relation to the apparatus as a whole, so measures such as *Test duration* **will** reflect the animal's time in the hidden zone.

Altering hidden zone exit settings

Unlike almost all other protocol settings, changes to the percentage of the animal which must be visible for an exit from the hidden zone to be scored **do not affect tests that have already been performed.** Therefore you should endeavour to set this value correctly before performing an experiment.

On the other hand, altering whether ANY-maze scores the apparent hidden zone entry that occurs when an animal starts a test hidden *will affect* both performed and unperformed tests.

ANY-maze help topic H0152

Editing a zone

Introduction

You can edit absolutely everything related to a zone at any time, whether before, during or after the experiment. However, although there are no restrictions on what you can edit, there are a few things to be aware of.

Points to note

1. If you edit a *movable* zone so it's no longer movable, then the zone's positions will automatically be deleted.
2. If you edit a *movable* zone so it becomes a *hidden* zone, then it will revert to being non-movable and, as a consequence, all of its positions will automatically be deleted.
3. Although you can change zone entry detection from using the centre of the animal to using the animal's entire area, this will only actually have an effect on existing tests if you recorded the animal's image while performing them.

See also:

- Editing the elements of a protocol
- Setting up a zone
- Deleting a zone

Deleting a zone

Introduction

You can delete a zone, or zone position, at any time whether before, during or after an experiment has been performed.

To delete a zone or position, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting zones.

See also:

- Deleting protocol elements
- Editing a zone

Zone measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a zone:

- Number of entries to the zone
- Number of exits from the zone
- Number of entries of the animal's head into the zone
- Time in the zone
- Time the animal's head was in the zone
- Was first zone entered

Distance measures

- Distance travelled in the zone
- Distance travelled until first entry into the zone
- Distance travelled by the animal's head in the zone

Latancy measures

- Latency to first entry to the zone
- Latency to first exit from the zone
- Latency to last entry to the zone
- Latency to the first entry of the animal's head into the zone
- Latency to the first exit of the animal's head from the zone

Speed measures

- Average speed in the zone
- Maximum speed in the zone

Zone visit measures

- Longest visit to the zone
- Shortest visit to the zone
- Average duration of visit to the zone

- List of the duration of each visit to the zone

Mobility measures

- Time mobile in the zone
- Time immobile in the zone
- Immobile episodes in the zone

Activity measures

- Time active in the zone
- Time inactive in the zone
- Inactive episodes in the zone

Distance to the zone (when outside the zone) measures

- Initial distance from the zone
- Average distance from the zone
- Maximum distance from the zone
- Minimum distance from the zone
- Cumulative distance from the zone
- Average distance of the animal's head from the zone
- Maximum distance of the animal's head from the zone
- Minimum distance of the animal's head from the zone

Distance to the zone border (when inside the zone) measures

- Average distance to the zone border
- Maximum distance to the zone border
- Minimum distance to the zone border
- Average distance from the animal's head to the zone border
- Maximum distance from the animal's head to the zone border
- Minimum distance from the animal's head to the zone border

Movement relative to zone measures

- Time getting closer to the zone
- Time getting further away from the zone

Heading/orientation measures

- Initial heading error to the zone
- Signed initial heading error to the zone
- Average absolute heading error to the zone
- Time moving towards the zone
- Time moving away from the zone
- Time oriented towards the centre of zone when inside zone

Turning measures

- Absolute turn angle while in the zone
- Absolute head turn angle while in the zone

Freezing measures

- Freezing bouts in the zone
- Time freezing in the zone

Rearing measures

- Number of rears in the zone
- Total time rearing in the zone
- Latency to first rear in the zone
- Average duration of a rear in the zone
- Maximum duration of a rear in the zone
- Minimum duration of a rear in the zone

Wishaw's corridor measures

- Time spent in Whishaw's Corridor
- Distance travelled in Whishaw's Corridor

Miscellaneous measures

- Time the animal's head was in the zone when its centre was outside the zone
- Path efficiency to first entry to the zone
- Corrected integrated path length
- Number of line crossings while in the zone

Number of entries to the zone

Description	Counts the number of times the animal entered the zone.
Calculation method	Depends on the method used to detect zone entries - see Choosing how ANY-maze should detect entries into a zone for more details.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	None

Number of exits from the zone

Description	Reports the number of times the animal exited from a zone.
Calculation method	Depends on the method used to detect zone entries - see Choosing how ANY-maze should detect entries into a zone for more details.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	None

Number of entries of the animal's head into the zone

Description	Counts the number of times the animal's head entered the zone.
Calculation method	Counts the number of times the animal's head position changed from being outside the zone to being inside it.
Analysis across time	This measure can be analysed across time. The result is based on just those head positions within the time period.
Units	None
Notes	This measure is only available if Head tracking is turned on.

Time in the zone

Description	Reports the total amount of time the animal spent in the zone.
Calculation method	Calculated by summing the duration of each visit to the zone where a visit starts at the time of a zone entry and ends at the time of a zone exit.

Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of the period that the animal spent in the zone. For example, if an animal entered a zone at time 45 seconds and exited it at time 80 seconds, then for the time period 30-60 seconds the result would be 15 seconds.
Units	Seconds
Notes	None

Time the animal's head was in the zone

Description	Reports the total amount of time that the animal's head was in the zone.
Calculation method	Calculated by summing the duration of each visit of the animal's head to the zone where a visit starts at the time the animal's head entered the zone and ends at the time the animal's head exited the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of the period that the animal's head spent in the zone. For example, if an animal's head entered a zone at time 45 seconds and exited it at time 80 seconds, then for the time period 30-60 seconds, the result would be 15 seconds.
Units	Seconds
Notes	This measure is only available if Head tracking is turned on.

Was first zone entered

Description	Reports whether the zone was the first zone the animal entered in the test.
Calculation method	Calculated by detecting the first zone entry in the test. This is affected by the *Don't score any results in this zone until the first 'true' entry* option on the *Zone entry settings* page. See Choosing how ANY-maze should detect entries into a zone for more details.
Analysis across time	This measure cannot be analysed across time.
Units	None
Notes	The result of this measure is either YES or NO, therefore when analysed it will be treated as a 2 level nominal value - see Statistical tests included in ANY-maze.

Distance travelled in the zone

Description	Reports the distance the animal travelled while in the zone
Calculation method	Calculated by summing the distance travelled during each visit to the zone. A visit starts when an animal enters the zone and ends when it exits the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the distance travelled within the zone during that time period.
Units	Metres
Notes	As the position of the animal prior to a zone entry must, by definition, be outside the zone and the position after it enters the zone must be inside, the distance between the two positions will be partly outside and partly inside the zone. ANY-maze adds all this distance to the distance travelled in the zone the animal's leaving. Although this can lead to inaccuracies, they are generally very small because: a) ANY-maze detects many positions per second so the distance between any two positions is usually very small; b) Any small distance 'lost' when the animal enters a zone entry is usually counterbalanced by a small distance which is 'gained' when it leaves the zone - see figure 1.

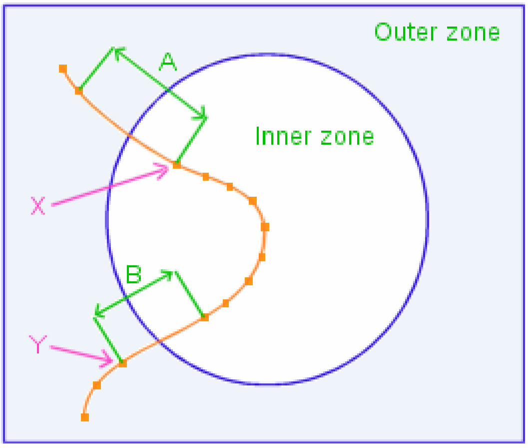


Figure 1. When the zone change at point 'X' is detected, ANY-maze adds all the distance 'A' to the distance travelled in the 'Outer zone'. When the second zone change at point 'Y' is detected, ANY-maze adds all the distance 'B' to the distance travelled in the 'Inner zone'. [Note: Distances A and B have been exaggerated in this diagram to aid explanation.]

In some situations, tracks can have small oscillations in them which tend to generate unrepresentatively large values for distance travelled. This occurs most often when an animal travels slowly while moving its body a lot - for example, while exploring an open field. To overcome this, ANY-maze uses an adaptive smoothing algorithm to attenuate these oscillations when calculating distance travelled - see figure 2. Note: The definition of what's a small oscillation is based on the animal's size.

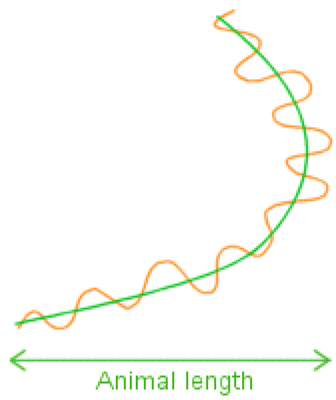


Figure 2. Measuring the length of the actual track (shown in orange) would yield an unrepresentatively large value for distance travelled. ANY-maze uses a 'smoothed' track (shown in green) to better estimate the true distance travelled. [Note: The oscillations in this track have been exaggerated to aid explanation.]

Distance travelled until first entry into the zone

Description	Reports the distance travelled by the animal up to its first entry into the specified zone.
Calculation method	The distance the animal travels is summed until it enters the zone. If the animal doesn't enter the zone during the test, then the result is undefined.
Analysis across time	This measure cannot be analysed across time.
Units	Metres
Notes	This measure is affected by the *Don't score any results in this zone until the first 'true' entry* option on the *Zone entry settings* page. See Choosing how ANY-maze should detect entries into a zone for more details.

Distance travelled by the animal's head in the zone

Description	Reports the distance the animal's head travelled while the head was in the zone.
Calculation method	Calculated by summing the distance travelled by the animal's head during each visit to the zone. A visit starts when the animal's head enters the zone and ends

	when it exits the zone.
Analysis across time	This measure can be analysed across time.
Units	Metres
Notes	As the position of the animal's head prior to a zone entry must, by definition, be outside the zone and the position after it enters the zone must be inside, the distance between the two positions will be partly outside and partly inside the zone. ANY-maze adds all this distance to the distance travelled by the animal's head in the zone that the animal is *leaving*. Although this can lead to inaccuracies, they are generally very small because: a) ANY-maze detects many positions per second so the distance between any two positions is usually very small; b) Any small distance 'lost' when the animal enters a zone entry is usually counterbalanced by a small distance which is 'gained' when it leaves the zone. This measure is only available if Head tracking is turned on.

Latency to first entry to the zone

Description	Reports the length of time which elapsed before the animal entered the zone for the first time.
Calculation method	This value is the time at which the first zone entry occurred. If the animal doesn't enter the zone during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is the time from the start of the period to the first entry. If the animal doesn't enter the zone during the time period, then the result is undefined.
Units	Seconds
Notes	This measure is affected by the *Don't score any results in this zone until the first 'true' entry* option on the *Zone entry settings* page. See Choosing how ANY-maze should detect entries into a zone for more details. This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis options element.

Latency to first exit from the zone

Description	Reports the length of time which elapsed before the animal exited the zone for the first time.
Calculation method	This value is the time at which the first zone exit occurred. If the animal doesn't exit the zone during the test then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is the

	time from the start of the period to the first exit. If the animal doesn't exit the zone during the time period then the result is undefined.
Units	Seconds
Notes	This measure is affected by the *Don't score any results in this zone until the first 'true' entry* option on the *Zone entry settings* page. See Choosing how ANY-maze should detect entries into a zone for more details. This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis options element.

Latency to last entry to the zone

Description	Reports the length of time that elapsed up to the moment when the animal made its last entry into the zone during the test.
Calculation method	This value is updated at the moment of each entry into the zone. The value at the end of the test is, necessarily, the latency to the last zone entry. If the animal doesn't enter the zone during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is the time from the start of the period to the last entry during that period. If the animal doesn't enter the zone during the time period, then the result is undefined.
Units	Seconds
	This measure is particularly useful in water-maze tests to report the time taken by the animal to find a platform zone. In such tests, it's common to only consider the animal as finding the platform if it remains on it for a certain period, for example 5 seconds. In this case, it's possible that the animal will enter the platform zone a number of times and therefore the time to 'find' the platform will be the latency to the *last* entry to the zone. This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis element.

Latency to the first entry of the animal's head into the zone

Description	Reports the length of time which elapsed before the animal's head entered the zone for the first time.
Calculation method	This value is the time at which the animal's head first entered the zone. If the animal's head doesn't enter the zone during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is the

	time from the start of the period to the first entry of the animal's head into the zone. If the animal's head doesn't enter the zone during the time period, then the result is undefined.
Units	Seconds
Notes	This measure is only available if Head tracking is turned on. This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis options element.

Latency to the first exit of the animal's head from the zone

Description	Reports the length of time which elapsed before the animal's head exited the zone for the first time.
Calculation method	This value is the time at which the animal's head first exited the zone. If the animal doesn't exit the zone during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is the time from the start of the period to the first exit of the animal's head from the zone. If the animal's head doesn't exit the zone during the time period, then the result is undefined.
Units	Seconds
Notes	This measure is only available if Head tracking is turned on. This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis options element.

Average speed in the zone

Description	Reports the average speed of the animal while it was in the zone.
Calculation method	Calculated by dividing the Distance travelled in the zone by the Time in the zone. If the animal was never in the zone during the test then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Distance travelled in the zone during the time period divided by the Time spent in the zone during the time period. If the animal was never in the zone during the time period then the result is undefined.
Units	Metres per second
Notes	If you want to know the average speed in the zone while mobile (i.e. ignoring periods when the animal was stationary), then use a calculation of *Distance travelled in the zone / Time mobile in the zone.*

This measure is affected by the option to *Use zero as the result for undefined averages* in the Analysis options element.

Maximum speed in the zone

Description	Reports the maximum speed of the animal while in the zone.
Calculation method	The speed of the animal between positions within the zone is calculated and the maximum speed is found. If the animal doesn't enter the zone during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. If the animal doesn't enter the zone during the time period, then the result is undefined.
Units	Metres per second
Notes	The calculation of maximum speed does not use *successive* positions but instead requires that the animal move at least a minimum distance (which is based on the animal's size) and the speed to cover *this* distance is calculated. This method of calculation is used to avoid reporting the speed of movements that don't constitute locomotion of the animal. For example, if an animal scratches, its centre point may oscillate rapidly but this will not be reported as the animal's maximum speed.

Longest visit to the zone

Description	Reports the duration of the longest single visit to the zone.
Calculation method	Each zone visit is delimited by a zone entry and a zone exit - the time between the two is the duration of the visit. The duration of each visit is calculated and the largest value is found. If the animal was never in the zone during the test then the result is zero.
Analysis across time	This measure can be analysed across time. For any time period, the result is the duration of the longest visit to the zone during the time period. If the animal spent the entire period in the zone, then the result will be the duration of the time period itself.
Units	Seconds
Notes	None

Shortest visit to the zone

Description	Reports the duration of the shortest single visit to the zone.

Calculation method	Each zone visit is delimited by a zone entry and a zone exit - the time between the two is the duration of the visit. The duration of each visit is calculated and the smallest value is found. If the animal was never in the zone during the test, then the result is zero.
Analysis across time	This measure can be analysed across time. For any time period, the result is the duration of the shortest visit to the zone during the time period.
Units	Seconds
Notes	None

Average duration of visit to the zone

Description	Reports the average duration of visits to the zone.
Calculation method	Calculated by dividing the Time spent in the zone by the Number of entries to the zone. If the animal was never in the zone during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. For any time period, the result is Time spent in the zone during the time period divided by the Number of entries to the zone in the time period. If the animal was never in the zone during the time period, then the result is undefined.
Units	Seconds
Notes	This measure is affected by the option to *Use zero as the result for undefined averages* in the Analysis options element.

List of the duration of each visit to the zone

Description	Reports a comma-separated list of the duration of each visit to the zone. For example, if the animal visited the zone three times during the test, for 1 second on the first occasion and for 20 seconds on the second and third occasions then the list would be '1.0, 20.0, 20.0'.
Calculation method	The time of the animal's entry to the zone is noted; then, when the animal exits the zone, the duration of the visit is calculated and added to the list. If the animal is in the zone at the end of the test, the time from the entry to the end of the test is used as the duration of the last visit.
Analysis across time	This measure cannot be analysed across time.
Units	The duration of all visits is reported in seconds.
Notes	When included on the Data page, this measure will show all the visits to the zone in a single cell. If the spreadsheet is saved in CSV format and then opened

in (for example) Microsoft Excel, then the visits will be listed in individual cells.

The length of the list is limited to 8192 characters, but usually at least 1,000 visits will be listed before the limit is reached.

Time mobile in the zone

Description	Reports the total time that the animal was mobile in the zone.
Calculation method	Calculated by subtracting the Time immobile in the zone from the Time spent in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Time spent in the zone during the period minus the Time immobile in the zone during the period.
Units	Seconds
Notes	None

Time immobile in the zone

Description	Reports the total time that the animal was immobile in the zone.
Calculation method	Sums the duration of each immobile episode in the zone - see notes for the definition of an immobile episode.
Analysis across time	This measure can be analysed across time. For any time period, the result is the sum of the duration of each immobile episode in the zone during the period.
Units	Seconds
Notes	An immobile episode in the zone starts when the animal becomes immobile after being mobile or when the animal is already immobile and enters the zone. Generally, it's unlikely that an immobile animal will enter a zone, because to enter the zone it will probably have to be mobile. Nevertheless, if the animal is immobile right on the border of a zone, it could enter the zone by moving very slightly but not by enough to end the immobile episode. An immobile episode in the zone ends when the animal becomes mobile or when it leaves the zone. The definition of immobility depends on the protocol - see Immobility detection.

Immobile episodes in the zone

Description	Reports the number of times the animal became immobile while in the zone.
Calculation method	Counts the number of times the animal changed from being mobile to being immobile while in the zone. If an *immobile* animal enters a zone (see note below), then the entry will be considered to start a new immobile episode in the zone, i.e. the count of immobile episodes in the zone will be incremented. This means that the sum of the immobile episodes in all the zones may be greater than the number of immobile episodes in the apparatus as a whole.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of immobile episodes that started in the zone during the time period. If an animal is already immobile in the zone at the start of the time period, then a new immobile episode is NOT counted for the period. This means that it's possible to have a period for which the result of this measure is zero but the result for the Time immobile in the zone is not zero.
Units	None
Notes	An immobile episode in the zone starts when the animal becomes immobile after being mobile or when the animal is already immobile and enters the zone. Generally, it's unlikely that an immobile animal will enter a zone, because to enter the zone it will probably have to be mobile. Nevertheless if the animal is immobile right on the border of a zone it could enter the zone by moving very slightly but not by enough to end the immobile episode. An immobile episode in the zone ends when the animal becomes mobile or when it leaves the zone. The definition of immobility depends on the protocol - see Immobility detection.

Time active in the zone

Description	Reports the total time that the animal was active in the zone.
Calculation method	Calculated by subtracting the Time inactive in the zone from the Time in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Time spent in the zone during the period minus the Time inactive in the zone during the period.
Units	Seconds
Notes	An animal is active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - grooming for example.

Time inactive in the zone

Description	Reports the total time that the animal was inactive in the zone.
Calculation method	Sums the duration of each inactive episode in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the sum of the duration of each inactive episode in the zone during the period.
Units	Seconds
Notes	An inactive episode in the zone starts when the animal becomes inactive after being active or when the animal enters the zone and is already inactive. An inactive episode in the zone ends when an animal becomes active or when it leaves the zone. Inactivity is defined as NOT activity. An animal is defined to be active if it is either mobile OR it's performing some other behaviour which has been specified as an activity - for example, grooming. If the protocol specifies that immobility should not be detected, then activity analysis will be based purely on the performance of other behaviours.

Inactive episodes in the zone

Description	Reports the total number of times the animal became inactive while in the zone.
Calculation method	Counts the number of times the animal changed from being active to being inactive while in the zone. If an *inactive* animal enters a zone, then the entry will be considered to start a new inactivity episode in the zone, i.e. the count of inactive episodes in the zone will be incremented. This means that the sum of the inactive episodes in all the zones may be greater than the number of inactive episodes in the apparatus as a whole.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of inactive episodes that started in the zone during the time period. If an animal is already inactive in the zone at the start of the time period, then a new inactive episode is NOT counted for the period. This means that it's possible to have a period for which the result of this measure is zero but the result for the Time inactive in the zone is not zero.
Units	None
Notes	An inactive episode in the zone starts when the animal becomes inactive after being active or when the animal enters the zone and is already inactive. An inactive episode in the zone ends when an animal becomes active or when it leaves the zone.

Initial distance from the zone

Description	Reports the distance from the animal's first position in the test to the zone
Calculation method	The straight line distance from the first position of the animal (see notes) to the nearest point of the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Metres
Notes	The position of the animal is either the animal's centre or the point of the animal that is closest to the zone. Which of these is used depends on whether or not the zone entry settings use the animal area to determine zone entries; when they do, the distance from the animal to the zone is based on the distance from the point of the animal that is closest to the zone; otherwise, the distance is based on the centre of the animal.

Average distance from the zone

Description	Reports the average distance from the animal to the zone when the animal is *outside* the zone.
Calculation method	ANY-maze calculates the distance from the animal to the zone for every position of the animal that is outside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand if zone entries are based on the centre of the animal, then the distance to the zone will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest part of the zone. Having determined the distance from the zone, ANY-maze maintains a cumulative sum of each distance multiplied by the time the animal remained at that distance. The final result for the average distance from the zone is this cumulative sum divided by the total duration of the test or period. The reason the system works this way is best explained using an example. Imagine the animal was 50cm from a zone and remained there for 55 seconds; it then moved to be 30cm from the zone and remained there for 5 seconds; the test then ended. Just taking the average of the two distances would imply that the average distance from the zone was 40cm, but this is very misleading as

the animal spent almost the entire test 50cm from the zone. Instead, ANY-maze would calculate the average distance as [(50 x 55) + (30 x 5)] / 60 = 48.33cm. Effectively, the system weights the distances depending on how long the animal remained there.

Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal that fall within the time period.
Units	Metres
Notes	If the animal spends the entire duration of the test (or of a time period) inside the zone, then the result will be zero (i.e. the animal was no distance from the zone).

Maximum distance from the zone

Description	Reports the maximum distance from the animal to the zone when the animal is *outside* the zone.
Calculation method	ANY-maze calculates the distance from the animal to the zone for every position of the animal that is outside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal, or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal then the distance to the zone will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest part of the zone. Having determined the distance from the animal to the zone, the system simply notes the maximum value during the test or time period.
Analysis across time	This measure can be analysed across time. The result is the maximum distance considering just those positions of the animal that fall within the time period.
Units	Metres
Notes	If the animal spends the entire duration of the test (or of a time period) inside the zone, then the result will be zero (i.e. the animal was no distance from the zone).

Minimum distance from the zone

Description	Reports the minimum distance from the animal to the zone when the animal is

outside the zone.

Calculation method	ANY-maze calculates the distance from the animal to the zone for every position of the animal that is outside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal, or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal then the distance to the zone will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest part of the zone. Having determined the distance from the animal to the zone, the system simply notes the minimum value during the test or time period.
Analysis across time	This measure can be analysed across time. The result is the minimum distance considering just those positions of the animal that fall within the time period.
Units	Metres
Notes	If the animal enters the zone, then this value is automatically set to zero.

Cumulative distance from the zone

Description	Reports the sum of the product of the distance from the zone and the time at that distance.
Calculation method	For every position of the animal, this calculates the distance from the zone multiplied by the time the animal stayed at that position. The final result is the sum of all these values.
Analysis across time	This measure can be analysed across time.
Units	Metres · seconds
Notes	This value represents the area under the curve of a graph of distance from zone vs. time.

Average distance of the animal's head from the zone

Description	Reports the average distance from the animal's head to the zone when the animal is *outside* the zone.
Calculation method	ANY-maze calculates the distance from the animal's head to the closest point on the zone border for every position of the animal's head that is outside the

zone. The system maintains a cumulative sum of each distance multiplied by the time the animal remained at that distance. The final result for the average distance of the animal's head from the zone is this cumulative sum divided by the total duration of the test or period.

The reason the system works this way is best explained using an example. Imagine the animal's head was 50cm from a zone and remained there for 55 seconds; it then moved to be 30cm from the zone and remained there for 5 seconds; the test then ended. Just taking the average of the two distances would imply that the average distance from the zone was 40cm, but this is very misleading as the animal spent almost the entire test 50cm from the zone. Instead, ANY-maze would calculate the average distance as [(50 x 55) + (30 x 5)] / 60 = 48.33cm. Effectively, the system weights the distances depending on how long the animal remained there.

Analysis across time — This measure can be analysed across time. The result is based on just those positions of the animal's head that fall within the time period.

Units — Metres

Notes — This measure is only available if Head tracking is turned on.

If the animal's head spends the entire duration of the test (or of a time period) inside the zone, then the result will be zero (i.e. the animal's head was no distance from the zone).

Maximum distance of the animal's head from the zone

Description — Reports the maximum distance from the animal's head to the zone when the animal is *outside* the zone.

Calculation method — For each position of the animal's head that is outside the zone, ANY-maze calculates the distance from the head to the closest point on the zone border. The maximum such distance is found.

Analysis across time — This measure can be analysed across time. The result is based on just those positions of the animal's head that fall within the time period.

Units — Metres

Notes — This measure is only available if Head tracking is turned on.

If the animal's head spends the entire duration of the test (or of a time period) inside the zone, then the result will be zero (i.e. the animal's head was no distance from the zone).

Minimum distance of the animal's head from the zone

Description	Reports the minimum distance from the animal's head to the zone when the animal is *outside* the zone.
Calculation method	For each position of the animal's head that is outside the zone, ANY-maze calculates the distance from the head to the closest point on the zone border. The minimum such distance is found.
Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal's head that fall within the time period.
Units	Metres
Notes	This measure is only available if Head tracking is turned on. If the animal's head enters the zone, then this value will be zero.

Average distance to the zone border

Description	Reports the average distance from the animal to the border of the zone when the animal is *inside* the zone.
Calculation method	ANY-maze calculates the distance from the animal to the zone border for every position of the animal that is inside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal then the calculation of the distance from the animal to the zone border will also be based on the entire area of the animal - specifically the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand if zone entries are based on the centre of the animal then the distance to the zone border will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest border of the zone. Having determined the distance to the zone border ANY-maze maintains a cumulative sum of each distance multiplied by the time the animal remained at that distance. The final result for the average distance to the zone border is this cumulative sum divided by the total duration of the test or period. The reason the system works this way is best explained using an example. Imagine the animal was 20cm from a zone border and remained there for 55 seconds; it then moved to be 10cm from the zone border and remained there for 5 seconds; the test then ended. Just taking the average of the two distances would imply that the average distance from the zone border was 15cm, but this is very misleading as the animal spent almost the entire test 20cm from the border. Instead, ANY-maze would calculate the average distance as [(20 x 55) +

(10 x 5)] / 60 = 19.16cm. Effectively, the system weights the distances depending on how long the animal remained there.

Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal that fall within the time period.
Units	Metres
Notes	If the animal never enters the zone, then the result is either undefined or zero depending on the setting *Use zero as the result for undefined averages* in the Analysis options element.

Maximum distance to the zone border

Description	Reports the maximum distance from the animal to the border of the zone when the animal is *inside* the zone.
Calculation method	ANY-maze calculates the distance from the animal to the zone border for every position of the animal that is inside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone border will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal then the distance to the zone border will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest border of the zone. Having determined the distance to the border, the system simply notes the maximum value during the test or time period.
Analysis across time	This measure can be analysed across time. The result is the maximum distance considering just those positions of the animal that fall within the time period.
Units	Metres
Notes	If the animal never enters the zone, then the result is undefined.

Minimum distance to the zone border

Description	Reports the minimum distance from the animal to the border of the zone when the animal is *inside* the zone.
Calculation method	ANY-maze calculates the distance from the animal to the zone border for every position of the animal that is inside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal or the

animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone border will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal then the distance to the zone border will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest border of the zone. Having determined the distance to the border, the system simply notes the minimum value during the test or time period.

Analysis across time	This measure can be analysed across time. The result is the minimum distance considering just those positions of the animal that fall within the time period.
Units	Metres
Notes	If the animal never enters the zone, then the result is undefined. If the animal exits the zone, this value is automatically set to zero.

Average distance from the animal's head to the zone border

Description	Reports the average distance from the animal's head to the border of the zone when the animal is *inside* the zone.
Calculation method	ANY-maze calculates the distance from the animal's head to the closest point on the zone border for every position of the animal's head that is inside the zone. The system maintains a cumulative sum of each distance multiplied by the time the animal remained at that distance. The final result for the average distance to the zone border is this cumulative sum divided by the total duration of the test or period. The reason the system works this way is best explained using an example. Imagine the animal's head was 20cm from a zone border and remained there for 55 seconds; it then moved to be 10cm from the zone border and remained there for 5 seconds; the test then ended. Just taking the average of the two distances would imply that the average distance from the animal's head to the zone border was 15cm, but this is very misleading as the animal's head spent almost the entire test 20cm from the border. Instead, ANY-maze would calculate the average distance as [(20 x 55) + (10 x 5)] / 60 = 19.16cm. Effectively, the system weights the distances depending on how long the animal remained there.
Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal's head that fall within the time period.
Units	Metres
Notes	If the animal's head never enters the zone, then the result is either undefined

	or zero depending on the setting *Use zero as the result for undefined averages* in the Analysis options element.
	This measure is only available if Head tracking is turned on.

Maximum distance from the animal's head to the zone border

Description	Reports the maximum distance from the animal's head to the border of the zone when the animal is *inside* the zone.
Calculation method	For each position of the animal's head that is inside the zone, ANY-maze calculates the distance from the head to the closest point on the zone border. The maximum such distance is found.
Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal's head that fall within the time period.
Units	Metres
Notes	If the animal's head never enters the zone, then the result is undefined.
	This measure is only available if Head tracking is turned on.

Minimum distance from the animal's head to the zone border

Description	Reports the minimum distance from the animal's head to the border of the zone when the animal is *inside* the zone.
Calculation method	For each position of the animal's head that is inside the zone, ANY-maze calculates the distance from the head to the closest point on the zone border. The minimum such distance is found.
Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal's head that fall within the time period.
Units	Metres
Notes	If the animal's head never enters the zone, then the result is undefined.
	This measure is only available if Head tracking is turned on.

Time getting closer to the zone

Description	Reports the total amount of time that the animal was outside the zone and was getting closer to it.
Calculation method	ANY-maze calculates the distance from the animal to the zone for every position of the animal that is outside the zone. Exactly how this is done

depends on whether zone entries *(sic)* are set to use the entire area of the animal or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal then the distance to the zone will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest part of the zone.

Having calculated the distance to the zone, ANY-maze compares it to the previous distance to the zone; if it is less, then the animal is getting closer to the zone and the time from the previous position to this one is added to the total time getting closer to the zone. Note that very small movements of the animal will be ignored by the system (the definition of very small being based on the animal's size).

Analysis across time	This measure can be analysed across time. The result is based on just those positions of the animal that fall within the time period.
Units	Seconds
Notes	This measure seems very similar to the Time moving towards the zone, but is calculated quite differently (see the definition of Time moving towards the zone for details on how it's calculated). The principal difference is that this measure relates to the animal's distance from the zone, whereas the *Time moving towards the zone* measure relates to the animal's heading. A good example of this difference is when a zone is in the form of a ring. In this case, if the animal is moving inside the ring it would always be 'moving towards' the zone as the zone surrounds it, but it could still either be getting closer or further from the zone. (In fact, it would always be moving both towards and away from the zone, so because of this ambiguity, ANY-maze simply wouldn't score either of these measures.)

Time getting further away from the zone

Description	Reports the total amount of time that the animal was outside the zone and was getting further away from it.
Calculation method	ANY-maze calculates the distance from the animal to the zone for every position of the animal that is outside the zone. Exactly how this is done depends on whether zone entries *(sic)* are set to use the entire area of the animal or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone will

also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal then the distance to the zone will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest part of the zone.

Having calculated the distance to the zone, ANY-maze compares it to previous distance to the zone; if it is greater, then the animal is getting further away from the zone and the time from the previous position to this one is added to the total time getting further away from the zone. Note that very small movements of the animal will be ignored by the system (the definition of very small being based on the animal's size).

Analysis across time This measure can be analysed across time. The result is based on just those positions of the animal that fall within the time period.

Units Seconds

Notes This measure seems very similar to the Time moving away from the zone, but is calculated quite differently (see the definition of Time moving away from the zone for details on how it's calculated). The principal difference is that this measure relates to the animal's distance from the zone, whereas the *Time moving away from the zone* measure relates to the animal's heading. A good example of this difference is when a zone is in the form of a ring. In this case, if the animal is moving inside the ring it would always be 'moving away from' the zone as the zone surrounds it, but it could still either be getting closer or further from the zone. (In fact, it would always be moving both towards and away from the zone, so because of this ambiguity, ANY-maze simply wouldn't score either of these measures.)

Initial heading error to the zone

Description Reports the angle between the animal's heading at the start of the test and a direct heading to the zone.

Calculation method Calculation of this measure depends on the settings in the Analysis options > Heading error sub-element of the protocol. Specifically, the options control both how the animal's heading at the start of the test is determined and what part of the zone is used to calculate the heading error.

There are two options for how the animal's heading at the start of the test is determined - one uses a specific time delay, the other a specific distance. In the first case, the animal's heading is taken to be the vector from the its first position in the test to the first position detected after the specified time interval has elapsed. In the second case, the heading is taken to be the vector from the animal's first position in the test to the first position that's more than the specified distance from it. In both cases, positions that are detected while the animal is considered to be immobile (if immobility detection is switched on) are ignored - thus in the first case, the animal must be *mobile* for the period that is specified.

Having determined the animal's initial heading, the system then calculates the heading from the first position in the test to the zone. To do this, ANY-maze can use one of two methods (again, these are specified using the Analysis options > Heading error sub-element in the protocol); it can either simply calculate the heading to the centre of the zone or it can calculate the heading to any part of the zone.

In the first case (see figure 3), the *centre* of the zone is taken to be the zone's 'centre of mass' (i.e. the mean x, y coordinate of all the points in the zone) and the heading error is defined as the angle between this heading and the animal's initial heading.

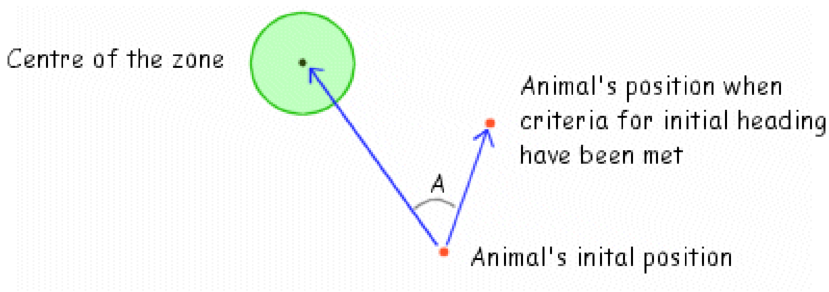


Figure 3. Calculation of the initial heading error using the centre of the zone: The animal's initial heading error is the angle 'A' between its initial heading and the direct heading to the centre of the zone.

It's important to understand that a zone's centre of mass may actually be outside the zone. For example, consider a ring-shaped zone; the centre of mass will be in the centre of the ring, but this point will not be within the zone.

The second method of calculating the heading to the zone is to consider the heading to every position on the zone's perimeter - in this case, the heading error is the smallest angle between the animal's heading and the heading to any perimeter point - see figure 4.

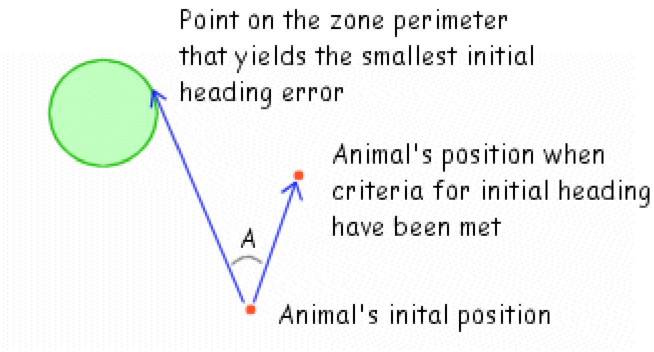


Figure 4. Calculation of the initial heading error using the entire area of the zone: The animal's initial heading error is the smallest angle 'A' between its initial heading and any point on the zone perimeter.

In the case of small zones the difference between the two calculation techniques is usually minimal, but for a large zone, it can make a substantial difference to the result.

Analysis across time This measure cannot be analysed across time.

Units Degrees

Notes None

Signed initial heading error to the zone

Description Reports the angle between the animal's heading at the start of the test and a direct heading to the zone. A positive initial heading error means the zone is to the animal's right, and a negative initial heading error signifies the zone is to the animal's left.

Calculation method Calculation of this measure depends on the settings in the Analysis options > Heading error sub-element of the protocol. Specifically, the options control both how the animal's heading at the start of the test is determined and what part of the zone is used to calculate the heading error.

There are two options for how the animal's heading at the start of the test is determined - one uses a specific time delay, the other a specific distance. In the first case, the animal's heading is taken to be the vector from its first position in the test to the first position detected after the specified time interval has elapsed. In the second case, the heading is taken to be the vector from the animal's first position in the test to the first position that's more than the specified distance from it. In both cases, positions that are detected while the animal is considered to be immobile (if immobility detection is switched on) are ignored - thus in the first case, the animal must be *mobile* for the period that is specified.

Having determined the animal's signed initial heading, the system then calculates the heading from the first position in the test to the zone. To do this, ANY-maze can use one of two methods (again, these are specified using the Analysis options > Heading error sub-element in the protocol); it can either simply calculate the heading to the centre of the zone, or it can calculate the heading to any part of the zone.

In the first case (see figure 9), the *centre* of the zone is taken to be the zone's 'centre of mass' (i.e. the mean x, y coordinate of all the points in the zone) and the heading error is defined as the angle between this heading and the animal's initial heading.

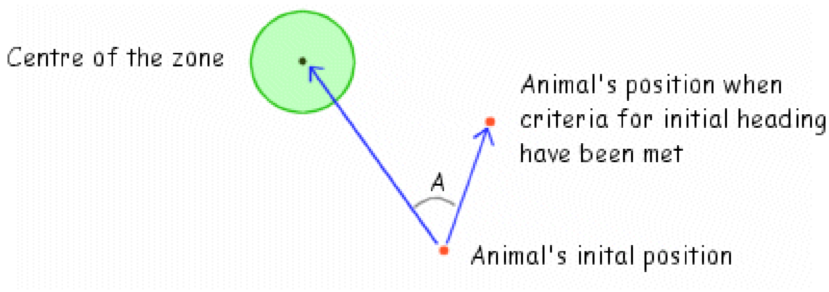


Figure 9. Calculation of the signed initial heading error using the centre of the zone: The animal's signed initial heading error is the angle 'A' between its initial heading and the direct heading to the centre of the zone. In this figure, the signed initial heading error will be a negative value, as the zone is to the animal's left.

It's important to understand that a zone's centre of mass may actually be outside the zone. For example, consider a ring-shaped zone; the centre of mass will be in the centre of the ring, but this point will not be within the zone.

The second method of calculating the heading to the zone is to consider the heading to every position on the zone's perimeter - in this case, the heading error is the smallest angle between the animal's heading and the heading to any perimeter point - see figure 10.

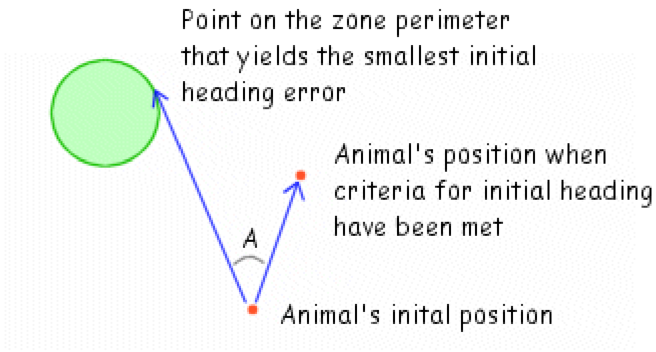


Figure 10. Calculation of the signed initial heading error using the entire area of the zone: The animal's initial heading error is the smallest angle 'A' between its initial heading and any point on the zone perimeter.

In the case of small zones, the difference between the two calculation techniques is usually minimal, but for a large zone it can make a substantial difference to the result.

Analysis across time This measure cannot be analysed across time.

Units Degrees

Notes None

Average absolute heading error to the zone

Description Reports the average absolute angle between the animal's heading and a direct heading to the zone.

Calculation method The method used to calculate this measure depends on the option specified in the Analysis options > Heading error sub-element in the protocol. Specifically, there are two ways to determine the heading to the zone - using the centre of the zone or using the entire zone area.

In the first case, the heading to the zone is taken to be the heading to the *centre* of the zone, where the centre is defined as the zone's 'centre of mass' (i.e. the mean x, y coordinate of all the points in the zone). It's important to understand that the centre of mass may actually be outside the zone. For example, consider a ring- shaped zone; the centre of mass will be in the centre of the ring, but this point will not be within the zone. With this definition of the heading to the zone, the heading error for a position is calculated as follows: The animal's heading is defined as the vector that joins the position with the next position in time. The heading to the zone is defined as the vector that joins the position to the centre of the zone and the heading error is defined as the angle between the two vectors - see figure 5.

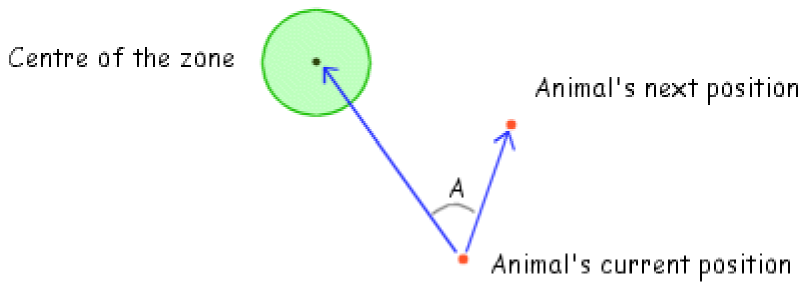


Figure 5. Calculation of the average heading error using the centre of the zone: The animal's heading error is the angle 'A' between its heading and the direct heading to the centre of the zone.

In the case where the heading to the zone is defined using the entire zone area, the calculation of the heading error is performed as follows: The animal's heading is defined as the vector that joins the position with the next position in time. The heading to the zone is then calculated for every point on the zone's perimeter, and the angle between this heading and the animal's heading is calculated. The smallest angle is the heading error - see figure 6.

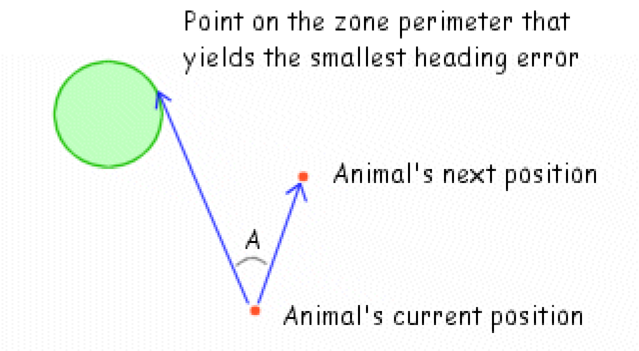


Figure 6. Calculation of the average heading error using the entire area of the zone: The animal's heading error is the smallest angle 'A' between its heading and the direct heading to any point on the zone perimeter.

Irrespective of which method is used to calculate the individual heading errors, the *average* absolute heading error is calculated in the same way: Each absolute heading error angle is multiplied by the time for which it persisted (i.e. the time from one position of the animal to the next). This product is then summed for the entire test (or time period). The final sum is then divided by the test duration (or the duration of the time period). This seemingly strange method of calculating the average is required because positions in ANY-maze are not necessarily recorded at a fixed frequency.

If immobility is being detected in a test, then all positions when the animal is deemed to be immobile are ignored in the calculation of the average heading error. If immobility is not being detected, then all positions are used with the caveat that a position must be at least a minimum distance from the previous position for it to be considered. The value used for this minimum distance is based on the size of the animal.

Analysis across time This measure can be analysed across time. The result is based on just those positions that fall within the time period.

Units Degrees.

Notes None

Time moving towards the zone

Description Reports the total amount of time for which the animal was moving towards the zone.

Calculation method The method used to calculate this measure depends on the option specified in the Analysis options > Movement towards and away from zones and points sub-element in the protocol. Specifically, the measure can either be based on the centre of the zone or on the zone's entire area.

In the first case, the measure is calculated as follows: For each position of the animal, a vector is created between the current position and the next position. A second vector is then created between the current position and the centre of the zone. The angle between these two vectors is calculated - see figure 7.

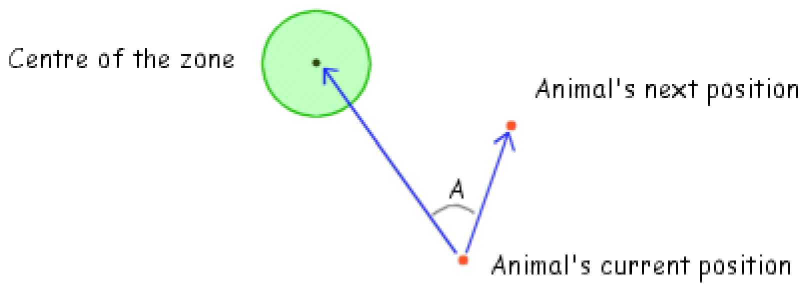


Figure 7. ANY-maze calculates the angle 'A' between the animal's heading and a direct heading to the zone; if this is less than a critical angle, the animal is deemed to be moving towards the zone.

Here, the *centre* of the zone is defined as the zone's 'centre of mass' (i.e. the mean x, y coordinate of all the points in the zone). It's important to understand that the centre of mass may actually be outside the zone. For example, consider a ring- shaped zone; the centre of mass will be in the centre of the ring, but this point will not be within the zone.

Having calculated the angle between the two vectors ('A'), it is compared to the critical angle for movement towards a zone (see notes). If the angle is less than this critical angle, then the animal is deemed to be moving towards the zone (unless it is *also* moving away from it - see notes) and the time it took to move from the current position to the next position is added to the total time moving towards the zone.

The second method used to calculate this measure uses the entire area of the zone. In this case, for each position of the animal, a vector is created between the current position and the next position. All possible vectors from the current position to the points on the zone's perimeter are then calculated, and the angle between each one and the animal's heading vector is calculated. The smallest of these angles is found. This angle is then compared to the critical angle for movement towards a zone in the same way as for the first calculation method (see above).

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Seconds

Notes The *critical angle* used to define whether the animal is moving towards the zone is also specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle described here, as this is more intuitive. The default critical angle is 90° (i.e. by default, the angle 'A' will be compared to 45°).

When calculating whether the animal is moving towards a zone, the system takes into consideration whether the animal is *also* moving away from the zone (see Time moving away from the zone). If it is, then the position is deemed to be ambiguous and the time from the position to the next position is NOT added to the overall result. A simple example of when this situation would arise is with a ring-shaped zone. If the animal is inside the ring, then no matter what direction it moves in, it will moving both towards the zone and away from it (as the zone surrounds the animal). The measures Time getting closer to the zone and Time getting further away from the zone offer alternatives that avoid this problem.

Time moving away from the zone

Description Reports the total amount of time for which the animal was moving away from the zone.

Calculation method The method used to calculate this measure depends on the option specified in the Analysis options > Movement towards and away from zones and points sub-element in the protocol. Specifically, the measure can either be based on the centre of the zone or on the zone's entire area.

In the first case, the measure is calculated as follows: For each position of the animal, a vector is created between the current position and the next position. A second vector is then created between the current position and the centre of the zone. The angle between these two vectors ('B') is calculated and deducted from 180°, yielding angle 'A' - see figure 8.

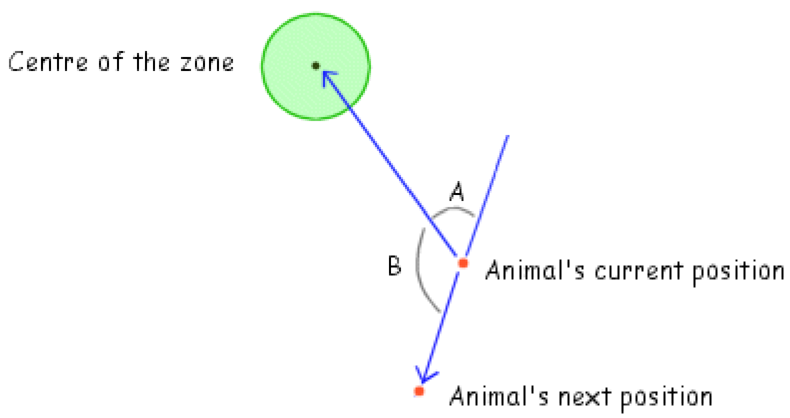


Figure 8. ANY-maze calculates the angle 'B' between the animal's heading

and a direct heading to the zone; this is deducted from 180°, yielding angle 'A'. If this angle is less than a critical angle, the animal is deemed to be moving away from the zone.

Here, the *centre* of the zone is defined as the zone's 'centre of mass' (i.e. the mean x, y coordinate of all the points in the zone). It's important to understand that the centre of mass may actually be outside the zone. For example, consider a ring- shaped zone; the centre of mass will be in the centre of the ring, but this point will not be within the zone.

Having calculated the angle 'A', it is compared to the critical angle for movement away from a zone (see notes). If the angle is less than this critical angle, then the animal is deemed to be moving away from the zone (unless it is *also* moving towards it - see notes) and the time it took to move from the current position to the next position is added to the total time moving away from the zone.

The second method used to calculate this measure uses the entire area of the zone. In this case, for each position of the animal, a vector is created between the current position and the next position. All possible vectors from the current position to the points on the zone's perimeter are then calculated, and the angle between each one and the animal's heading vector is calculated. The largest of these angles is found. This angle is deducted from 180° and the result is compared to the critical angle for movement away from a zone in the same way as for the first calculation method (see above).

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Seconds

Notes The *critical angle* used to define whether the animal is moving away from the zone is also specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle as described here, as this is more intuitive. The default critical angle is 90° (i.e. by default, the angle 'A' will be compared to 45°).

When calculating whether the animal is moving away from a zone, the system takes into consideration whether the animal is *also* moving towards the zone (see Time moving towards the zone). If it is, then the position is deemed to be ambiguous and the time from the position to the next position is NOT added to the overall result. A simple example of when this situation would arise is a ring-shaped zone. If the animal is inside the ring, then no matter what direction it moves it will be moving both towards the zone and away from it (as the zone surrounds the animal). The measures Time getting closer to the zone and Time getting further away from the zone offer alternatives that avoid this problem.

Time oriented towards the centre of the zone when inside zone

Description	Reports the amount of time the animal was oriented towards the centre of the zone while it was inside the zone.
Calculation method	The animal's orientation is a vector connecting its centre position to its head position. A second vector from the animal's head position to the centre of the zone is determined. and if the angle between the vectors is less than or equal to a *critical angle* (see notes). then the animal is deemed to be oriented towards the zone. The amount of time for which this is the case while the animal is inside the zone (as determined by the zone entry criteria) is summed.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	The *critical angle* used to define whether the animal is moving towards the zone is also specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact. the value entered in the analysis options is twice the critical angle as described here. as this is more intuitive. The default critical angle is 90° (i.e. by default. the angle between the vectors will be compared to 45°).

Absolute turn angle while in the zone

Description	Reports the sum of the absolute angle between each movement vector of the animal while it was inside the zone.
Calculation method	For each position of the animal that is inside the zone, a vector of movement from one position of the animal's centre point to the next is created. For each vector the angle between it and the previous vector is calculated with anti-clockwise movement being negative and clockwise movement being positive (i.e. the angle is from -180° to 180°). The absolute value of this angle is summed for all the positions of the animal within the zone throughout the test or time period.
Analysis across time	This measure can be analysed across time. The result is based on just those positions within the specific time period.
Units	Degrees
Notes	From this measure, it is easy to use calculations to derive measures such as *Meander in the zone* and *Angular velocity in the zone*. The former is the *Absolute turn angle while in the zone* divided by the *Distance travelled in the zone* and the latter is the *Absolute turn angle while in the zone* divided by the

Time in the zone.

Absolute head turn angle while in the zone

Description	Reports the cumulative absolute angle through which the animal's head moved while it was in the zone. For example, if while in the zone, the animal moved its head 30° to the left and then moved its head 45° to the right, the absolute head turn angle would be 75°.
Calculation method	For each position of the animal's head, a vector is created from the animal's centre point to the head. The angle between this vector and the same vector for the previous position of the animal's head is calculated, and the absolute value of this angle is summed whenever the animal is in the zone.
Analysis across time	This measure can be analysed across time.
Units	Degrees
Notes	As described above, calculation of this measure requires two vectors which are generated from two consecutive head positions. It is the second of these two positions which is used to determine whether the animal is in the zone. Thus, for example, the animal might be oriented North and standing outside the zone; if it then moves into the zone and is then oriented West, the 90° change in orientation will all be attributed to the zone that it has just entered. Whether the animal is in the zone is determined by the zone entry settings. Note that these may not require that either the head or the centre point are actually in the zone.

Freezing bouts in the zone

Description	Reports the number of times the animal *froze* while in the zone
Calculation method	Each time the animal starts to freeze, a check is made to determine whether it is in the zone. If it is, the count is incremented by one.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	None

Time freezing in the zone

Description	Reports the number of times the animal *froze* while in the zone.
Calculation method	The duration of each bout of freezing is calculated. If the animal was in the

	zone while frozen, then the duration is added to the result for the zone.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	None

Number of rears in the zone

Description	Reports the number of times the animal reared while in the zone.
Calculation method	Depends on the method used to detect zone entries (and thus by implication, zone exits too) - see Choosing how ANY-maze should detect entries into a zone for more details.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Total time rearing in the zone

Description	Reports the total amount of time for which the animal was rearing while it was in the zone.
Calculation method	Sums the duration of each bout of rearing that occurred while the animal was in the zone. If the animal enters the zone when it is already rearing, then the time will be counted from the time the animal entered the zone and not when the rearing bout started. If the animal exits the zone while rearing, then the time will stop at the time the animal exits the zone and not at the end of the rearing bout. For these reasons, it is possible for the result of this measure to be non-zero when the result for Number of rears in the zone is zero.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Latency to first rear in the zone

Description	Reports the latency to first time that the animal reared in the zone.
Calculation method	The time when the first bout of rearing started while the animal was in the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus. This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis options element.

Average duration of a rear in the zone

Description	Reports the average duration of the rearing bouts in the zone.
Calculation method	The result of Total time rearing in the zone divided by Number of rears in the zone.
Analysis across time	This measure can be analysed across time.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus. This measure is affected by the option to *Use zero as the result for undefined averages* in the Analysis options element.

Maximum duration of a rear in the zone

Description	Reports the duration of the longest bout of rearing in the zone.
Calculation method	The duration of each bout of rearing in the zone is calculated and the longest bout is found. Note that a bout of rearing in the zone starts when the animal is in the zone and begins to rear OR when the animal enters the zone when it is already rearing. Similarly, a bout ends when the animal is in the zone and stops rearing OR the animal exits the zone while rearing.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side.

ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Minimum duration of a rear in the zone

Description	Reports the duration of the shortest bout of rearing in the zone.
Calculation method	The duration of each bout of rearing in the zone is calculated and the shortest bout is found. Note that a bout of rearing in the zone starts when the animal is in the zone and begins to rear OR when the animal enters the zone when it is already rearing. Similarly, a bout ends when the animal is in the zone and stops rearing OR the animal exits the zone while rearing.
Units	Seconds
Notes	This measure is only available if the apparatus is being viewed from the side. ANY-maze actually detects rearing by analysing the shape of the animal, and therefore this measure will only work reliably if there is good contrast between the animal and the background of the apparatus.

Time spent in Whishaw's Corridor

Description	Reports the amount of time the animal spent in the Whishaw's Corridor for the zone.
Calculation method	The Whishaw's Corridor for the zone is determined based on the start position of the animal in the test, the centre point of the zone, and the width of the corridor specified in the protocol - see figure 11.

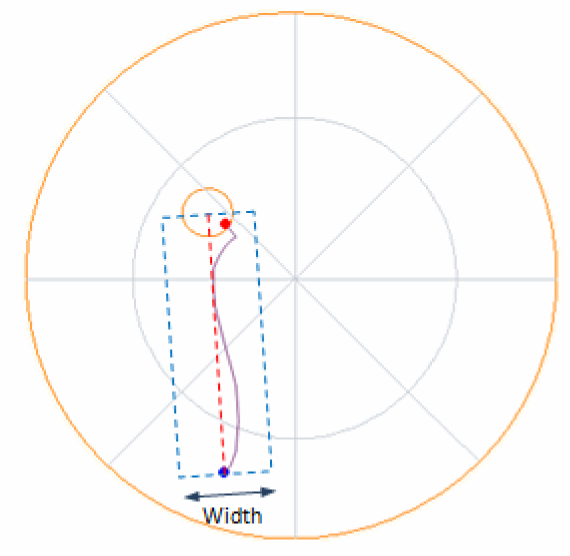


Figure 11. Example of the Whishaw's Corridor in a water-maze. The corridor is centred on a line (shown in red) running from the animal's start position to the centre of the platform zone. The corridor itself (shown in blue) has a width specified in the protocol.

The time spent in the corridor is then calculated by summing the duration of each visit to the corridor, where a visit starts at the time of entry and ends at the time of exit.

Analysis across time This measure can be analysed across time. For any time period, the result is the total amount of time the animal spent within the corridor during that time period.

Units Seconds

Notes This measure is only available if the Whishaw's Corridor width has been specified for the zone - see setting up a zone.

Distance travelled in Whishaw's Corridor

Description Reports the distance travelled by the animal in the Whishaw's Corridor for the zone.

Calculation method The Whishaw's Corridor for the zone is determined based on the start position of the animal in the test, the centre point of the zone, and the width of the corridor specified in the protocol - see figure 12.

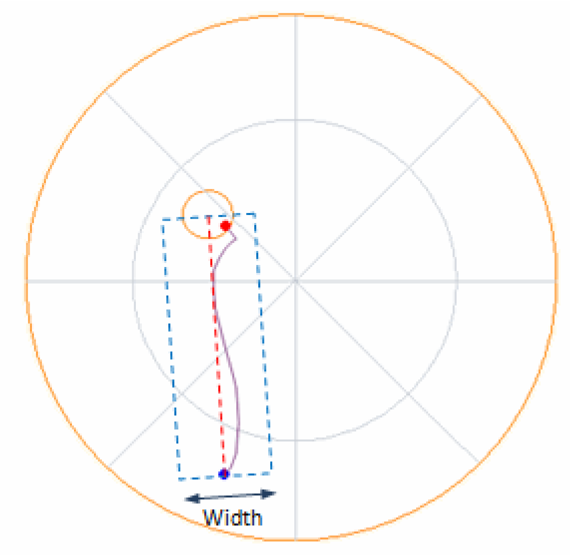


Figure 12. Example of Whishaw's Corridor in a water-maze. The corridor is centred on a line (shown in red) running from the animal's start position to the centre of the platform zone. The corridor itself (shown in blue) has a width specified in the protocol.

	The distance travelled in the corridor is then calculated by summing the distance travelled during each visit to the corridor, where a visit starts when an animal enters the zone and ends when it exits the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the distance travelled within the corridor during that time period.
Units	Metres
Notes	This measure is only available if the Whishaw's Corridor width has been specified for the zone - see Setting up a zone. In common with the way in which distance travelled in a zone is calculated, the distance travelled by the animal between a point outside the corridor and a point inside the corridor (i.e. when entering the corridor) is NOT included in the distance travelled in the corridor, whereas the distance travelled by the animal between a point inside the corridor and a point outside the corridor (i.e. when exiting the corridor) IS included in the distance travelled in the corridor.

Time the animal's head was in the zone when its centre was outside the zone

Description	Reports the amount of time for which the animal's head position was in the zone while its centre position was outside the zone
Calculation method	For every position of the animal's head. a check is made to determine whether

the head position is in the zone and the centre position is outside the zone (note that this determination does not use the zone entry criteria). The duration of each occurrence is calculated, and the total duration of all occurrences is summed.

Analysis across time This measure can be analysed across time

Units Seconds

Notes None

Path efficiency to first entry to the zone

Description This measure represents an index of the efficiency of the path taken by the animal to get from the first position in the test to the first position within the zone. A value of 1 indicates perfect efficiency - the animal moved in a straight line - values less than 1 indicate decreasing efficiency.

Calculation method The straight line distance between the first position in the test and the first position in the zone is divided by the total distance travelled by the animal until it first entered the zone

Analysis across time This measure cannot be analysed across time.

Units None

Notes This measure is intended for use in water-maze experiments but is available in all tests.

Corrected integrated path length

Description Reflects how efficiently the animal moved from the start position to the zone. A score of 0 implies perfect (straight line) efficiency.

Reference **Barnes CA, et al.** (1997) Multistability of cognitive maps in the hippocampus of old rats *Nature* **388**: 272-5

Calculation method The difference between the sum of the sampled distances from the target zone and the shortest possible sum, if the animal had swum directly to the platform at its mean speed.

Analysis across time This measure can be analysed across time.

Units Metres · seconds

Notes None

Number of line crossings while in the zone

Description	Reports the number of times the animal's centre point moved from one area of the apparatus map to another - i.e. crossed the lines which constitute the map, while the animal was in the specific zone.
Calculation method	The apparatus is divided into unique areas by the apparatus map. For each animal position recorded in the experiment, the area which contains the animal's centre point is found. Each time this changes, if the animal is in the zone, the measure's value if increased by 1.
Analysis across time	This measure can be analysed across time.
Units	None
Notes	When an animal enters a zone, it will usually cross a line; these line crossings are counted in the zone the animal has *entered.* This means that if the entire apparatus is divided into zones and the number of line crossings for all the zones is summed, it will equal the total number of line crossings for the apparatus as a whole. If you want to know the number of line crossings *within* the zone (i.e. excluding the line crossings that occur when the animal enters the zone), then use a calculation to subtract the number of zone entries from the number of line crossings in the zone. It's important to understand that ANY-maze uses the animal's centre point when calculating the measure and therefore it can be prone to 'spurious entries' if an animal straddles a line between two areas (i.e. by moving a very small amount, the animal can apparently cross a line many times). This problem can be overcome by setting one zone for each area, using the percentage of the animal that's in the zone to score zone entries, and then using a calculation to sum all the entries into these zones.

See also:

- Information measures
- Apparatus measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures

- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Points

Points define specific positions in your apparatus that are of interest to you. For each point that you define, ANY-maze will report such things as average distance from the point, time moving towards the point, time moving away from the point, etc.

For more information about points, refer to the topics below.

- An introduction to points
- Setting up a point
- Editing a point
- Deleting a point
- Point measures

An introduction to points

Introduction

As you may know, ANY-maze allows you to identify *areas* of your apparatus which are of interest by defining zones. But what if you're not really interested in an *area* of the apparatus, but rather in a specific position? Well, you could try to create a very small zone, but this would not be very satisfactory - instead, ANY-maze allows you to define a point.

- Defining the location of a point
- What ANY-maze reports for a point

Defining the location of a point

Defining a point could hardly be easier - you just click in the apparatus map at the point's location - see figure 1.

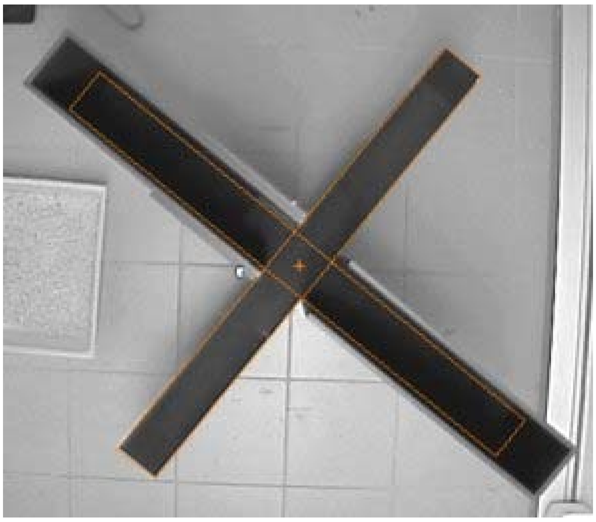


Figure 1. In this plusmaze a point has been defined in the centre of the apparatus.

You can define any number of points, and they can be located anywhere at all - even outside the apparatus map. This can be useful if, for example, you want to know how much time the animal spends oriented towards a cue outside the maze.

What ANY-maze reports for a point

If you defined a point in your apparatus, ANY-maze will report a range of measures related to it. These include:

- Average distance from the animal to the point
- The amount of time the animal was moving towards the point
- The amount of time the animal was moving away from the point

For a complete list, see Point measures.

See also:

- Setting up a point
- Point measures

Setting up a point

Introduction

For a general introduction to points, see An introduction to points.

To add a point to a protocol, click the *Add item* button in the ribbon bar and select *New Point* from the menu which appears.

The New Point menu option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the point name.
2. Specify whether to use the centre of the animal when calculating the distance to the point.
3. Define the location of the point in the apparatus.

What next?

After completing these steps, you should consider whether you want to include any sequences in this protocol.

See also:

- Adding elements to a protocol
- Editing a point
- Deleting a point

Point name

Don't use the word 'point' at the end of the name you enter, because ANY-maze includes this automatically when required - thus if you include it too, your name may occasionally appear as something like 'Novel object point point'.

In brief

Enter a name for the point in the *Point name* field on the point's settings page. You must make an entry, but it can be anything you like.

Details

Point names should uniquely and clearly identify a point, because they'll be used to label results. For example, ANY-maze will report *Average distance to the [PointName] point;* so if you called the point 'Point 1', the result would be *Average distance to the Point 1 point* - this would be a rather ambiguous result; something like *Average distance to the novel object point* would be much clearer.

To avoid having very long result names, you should try to keep point names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying whether to use the centre of the animal when calculating the distance to a point

In brief

Normally, when calculating the distance from the animal to a point, ANY-maze will use the distance from the part of the animal that's closest to the point (excluding the animal's tail). However, you may prefer to use the distance from the centre of the animal, in which case you should select this option.

Defining a point's position

In brief

Defining the position of a point within your apparatus could hardly be easier; just click the mouse at the relevant location.

Details

Having clicked the mouse, a *point marker* will be shown on the apparatus. If you want to change a point's location, you can either just click in a different place or you can drag the point marker.

Points don't necessarily have to be located inside the borders of the apparatus; they can be located anywhere in the video image. This can be useful if, for example, you want to know how much time the animal spends oriented towards a cue outside the maze.

Editing a point

Introduction

You can edit both the name and location of a point at any time, whether before, during or after the experiment - there are no restrictions and no unexpected repercussions.

See also:

- Editing the elements of a protocol
- Setting up a point
- Deleting a point

Deleting a point

Introduction

You can delete a point at any time, whether before, during or after an experiment has been performed.

To delete a point, simply select it in the protocol list and click the ❌ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting points.

See also:

- Deleting protocol elements
- Editing a point

Point measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a point:

- Average distance from the point
- Maximum distance from the point
- Minimum distance from the point
- Average distance of the animal's head from the point
- Maximum distance of the animal's head from the point
- Minimum distance of the animal's head from the point

Movement relative to point measures

- Time moving towards the point
- Time moving away from the point
- Time the animal's head was moving towards the point
- Time the animal's head was moving away from the point
- Average speed moving towards the point

Heading/orientation measures

- Initial heading error to the point
- Average absolute heading error to the point
- Time the animal's head was oriented towards the point
- Time the animal's head was oriented away from the point
- Number of times the animal's head was oriented towards the point

Average distance from the point

Description	Reports the average distance from the animal to the point.
Calculation method	For each position of the animal, the distance is calculated from the point to the animal; this distance is averaged throughout the entire test or time period. The method actually used to calculate this distance depends on the setting made

when setting up the point. Specifically, the distance can either be calculated based on the part of the animal which is closest to the point (excluding its tail), or based on the position of the centre of the animal.

Analysis across time	This measure can be analysed across time. The result is the average distance from the point to the animal during the time period.
Units	Metres
Notes	None

Maximum distance from the point

Description	Reports the maximum distance from the animal to the point.
Calculation method	For each position of the animal, the distance is calculated from the point to the part of the animal that's closest to the point (the animal's tail is excluded). The maximum distance during the entire test or time period is the result. The method actually used to calculate this distance depends on the setting made when setting up the point. Specifically, the distance can either be calculated based on the part of the animal which is closest to the point (excluding its tail), or based on the position of the centre of the animal.
Analysis across time	This measure can be analysed across time. The result is the maximum distance from the animal to the point during the time period.
Units	Metres
Notes	None

Minimum distance from the point

Description	Reports the minimum distance from the animal to the point.
Calculation method	For each position of the animal, the distance is calculated from the point to the part of the animal that's closest to the point (the animal's tail is excluded). The minimum distance during the entire test or time period is the result. The method actually used to calculate this distance depends on the setting made when setting up the point. Specifically, the distance can either be calculated based on the part of the animal which is closest to the point (excluding its tail), or based on the position of the centre of the animal.
Analysis across time	This measure can be analysed across time. The result is the minimum distance from the animal to the point during the time period.
Units	Metres
Notes	None

Average distance of the animal's head from the point

Description	Reports the average distance from the animal's head to the point.
Calculation method	Calculates the distance from the point to the position of the animal's head. Averages this distance through the entire test or time period.
Analysis across time	This measure can be analysed across time. The result is the average distance from the point to the animal's head during the time period.
Units	Metres
Notes	This measure is only available if Head tracking is turned on.

Maximum distance of the animal's head from the point

Description	Reports the maximum distance from the animal's head to the point.
Calculation method	Calculates the distance from the point to the position of the animal's head. The maximum distance during the entire test or time period is the result.
Analysis across time	This measure can be analysed across time. The result is the maximum distance from the animal's head to the point during the time period.
Units	Metres
Notes	This measure is only available if Head tracking is turned on.

Minimum distance of the animal's head from the point

Description	Reports the minimum distance from the animal's head to the point.
Calculation method	Calculates the distance from the point to the position of the animal's head. The minimum distance during the entire test or time period is the result.
Analysis across time	This measure can be analysed across time. The result is the minimum distance from the animal's head to the point during the time period.
Units	Metres
Notes	This measure is only available if Head tracking is turned on.

Time moving towards the point

Description	Reports the total amount of time for which the animal was moving towards the point.

Calculation method For each position of the animal, a vector is created between the current position and the next position. A second vector is then created between the current position and the point. The angle between these two vectors is calculated - see figure 1.

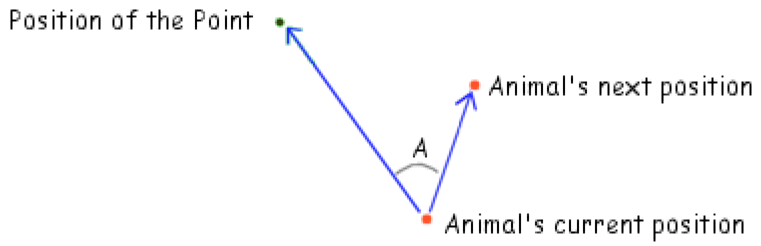


Figure 1. ANY-maze calculates the angle 'A' between the animal's heading and a direct heading to the point.

The absolute angle between the two vectors, 'A', is compared to the critical angle for movement towards a point (see notes). If the angle is less than this critical angle, then the animal is deemed to be moving towards the point. If the animal is moving towards the point, then the time taken to move from the current position to the next position is added to the total time moving towards the point.

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Seconds

Notes The *critical angle* used to define whether the animal is moving towards the point is specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle 'A' will be compared to 45°).

Time moving away from the point

Description Reports the total amount of time for which the animal was moving away from the point.

Calculation method For each position of the animal, a vector is created between the current position and the next position. A second vector is then created between the current position and the point. The angle between these two vectors is calculated and subtracted from 180° - see figure 2.

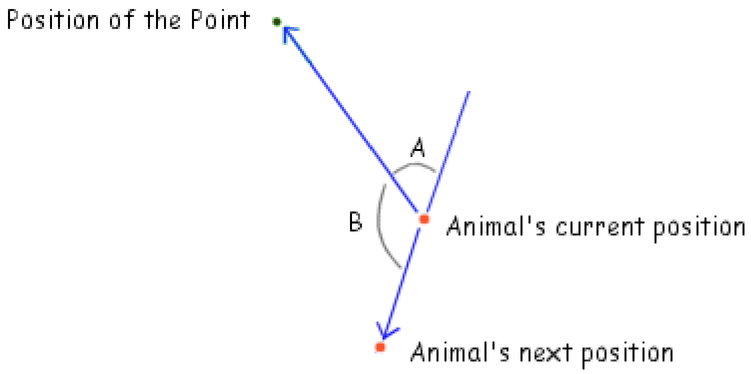


Figure 2. ANY-maze calculates the angle 'B' between the animal's heading and a direct heading to the point. This is then subtracted from 180°, yielding angle 'A'.

The angle 'A' is compared to the critical angle for movement away from a point (see notes). If the value is less than this critical angle, then the animal is deemed to be moving away from the point. If the animal is moving away from the point, then the time taken to move from the current position to the next position is added to the total time moving away from the point.

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Seconds

Notes The *critical angle* used to define whether the animal is moving away from the point is specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle 'A' will be compared to 45°).

Time the animal's head was moving towards the point

Description Reports the total amount of time for which the animal's head was moving towards the point.

Calculation method For each position of the animal's head, a vector is created between the current position and the next position. A second vector is then created between the current position and the point. The angle between these two vectors is calculated - see figure 3.

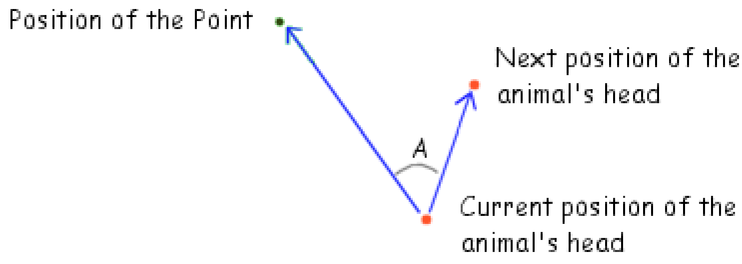


Figure 3. ANY-maze calculates the angle 'A' between the heading of the animal's head and the direct heading to the point.

The angle 'A' is compared to the critical angle for movement towards a point (see notes). If the angle is less than this critical angle, then the animal's head is deemed to be moving towards the point. If the animal's head is moving towards the point, then the time it took to move from the previous position to the current position is added to the total time moving towards the point.

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Seconds

Notes The *critical angle* used to define whether the animal is moving towards the point is specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle *beta* is compared to 45°).

This measure is only available if Head tracking is turned on.

Time the animal's head was moving away from the point

Description Reports the total amount of time for which the animal's head was moving away from the point.

Calculation method For each position of the animal's head, a vector is created between the current position and the next position. A second vector is then created between the current position and the point. The angle between these two vectors is calculated and the result is subtracted from 180° - see figure 4.

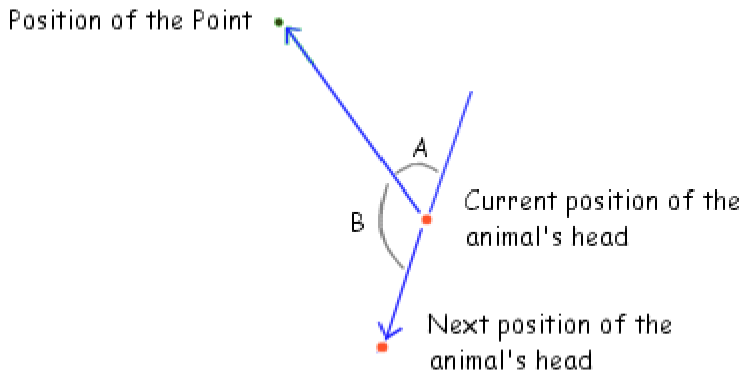


Figure 4. ANY-maze calculates the angle 'B' between the heading of the animal's head and the direct heading to the point; this is then subtracted from 180°, yielding angle 'A'.

The angle 'A' is compared to the critical angle for movement away from a point (see notes). If the angle is less than this critical angle, then the animal's head is deemed to be moving away from the point. If the animal's head is moving away from the point, then the time it took to move from the previous position to the current position is added to the total time moving away from the point.

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Seconds

Notes The *critical angle* used to define whether the animal's head is moving away from the point is specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle 'A' will be compared to 45°).

This measure is only available if Head tracking is turned on.

Average speed moving towards the point

Description Reports the average speed of the animal when it was moving towards the point.

Calculation method For each position of the animal, a vector is created between the current position and the next position. A second vector is then created between the current position and the point. The angle between these two vectors is calculated - see figure 1.

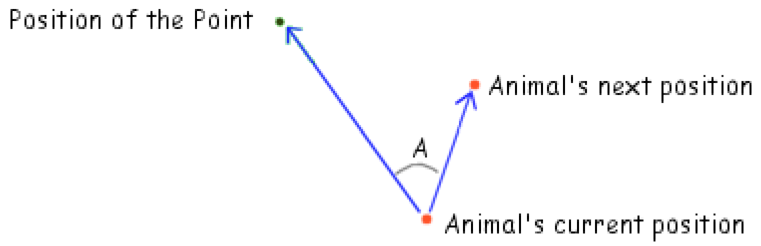


Figure 1. ANY-maze calculates the angle 'A' between the animal's heading and a direct heading to the point.

The absolute angle between the two vectors, 'A', is compared to the critical angle for movement towards a point (see notes). If the angle is less than this critical angle, then the animal is deemed to be moving towards the point. If the animal is moving towards the point, then the time taken to move from the current position to the next position is added to the total time moving towards the point and the distance from the current position to the next position is added to the total distance moving towards the point. At the end of the test the total distance moving towards the point is divided by the total time moving towards the point, yielding the average speed moving towards the point.

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units Metres per second

Notes The *critical angle* used to define whether the animal is moving towards the point is specified in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle 'A' will be compared to 45°).

Initial heading error to the point

Description Reports the angle between the animal's heading at the start of the test and a direct heading to the point.

Calculation method Calculation of this measure depends on the settings in the Analysis options > Heading error sub-element of the protocol. Specifically, there are two options for how the animal's heading at the start of the test is determined - one uses a specific time delay, the other a specific distance. In the first case, the animal's heading is taken to be the vector from the its first position in the test to the first position detected after the specified time interval has elapsed. In the second case, the heading is taken to be the vector from the animal's first position in the test to the first position that's more than the specified distance from it. In both cases, positions that are detected while the animal is considered to be immobile (if immobility detection is switched on) are ignored - thus in the first case, the animal must be *mobile* for the period that is specified.

Having determined the animal's initial heading, the system then calculates the vector from the first position in the test to the point. The angle between this vector and the animal's heading vector is the initial heading error.

Analysis across time This measure cannot be analysed across time.

Units Degrees

Notes None

Average absolute heading error to the point

Description Reports the average absolute angle between the animal's heading and a direct heading to the point.

Calculation method The animal's heading is defined as the vector that joins the position with the next position in time. The heading to the point is defined as the vector that joins the animal's position to the point, and the heading error is defined as the angle between the two vectors. This angle is calculated for every position of the animal, and the angle's absolute value is summed and then divided by the number of positions.

If immobility is being detected in a test, then all positions when the animal is deemed to be immobile are ignored in the calculation of the average heading error. If immobility is not being detected, then all positions are used - with the caveat that a position must be at least a minimum distance from the previous position for it to be considered. The minimum distance used is based on the size of the animal.

Analysis across time This measure can be analysed across time. The result is based on just those positions that fall within the time period.

Units Degrees

Notes None

Time the animal's head was oriented towards the point

Description	Reports the total amount of time for which the animal's head was oriented towards the point.
Calculation method	A vector is created between the centre point of the animal and the animal's head - this vector defines the direction in which the animal is oriented. A second vector is created between the position of the animal's head and the position of the point. The angle between these two vectors is calculated - see figure 5.

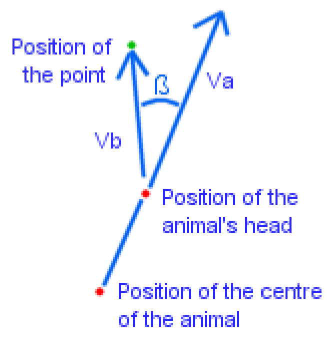


Figure 1. ANY-maze calculates the angle beta between the two vectors Va (the vector which defines the animal's orientation) and Vb (the vector from the animal to the point).

The absolute angle between the two vectors (*beta*) is compared to the critical angle for movement away from/towards a point (see notes). If the angle is less than this critical angle, then the animal is deemed to be oriented towards the point. If the animal is oriented towards the point, then the time between the previous position of the animal and its current position is added to the total time oriented towards the point (irrespective of whether it was oriented towards the point at the previous position).

Analysis across time	This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.
Units	Seconds
Notes	The *critical angle* used to define whether the animal is oriented towards the point is the same angle as is used to determine whether the animal is moving towards or away from a point, and is defined in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle *beta* will

be compared to 45°).

This measure is only available if Head tracking is turned on.

Time the animal's head was oriented away from the point

Description Reports the total amount of time for which the animal's head was oriented away from the point.

Calculation method A vector is created between the centre point of the animal and the animal's head - this vector defines the direction in which the animal is oriented. A second vector is created between the position of the animal's head and the position of the point. The angle between these two vectors is calculated - see figure 6.

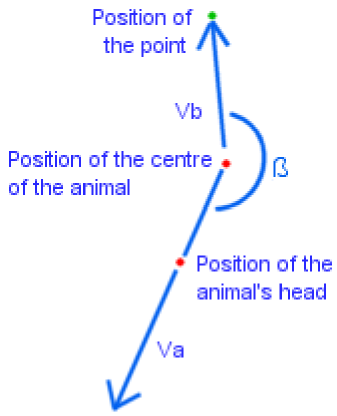


Figure 1. ANY-maze calculates the angle beta between the two vectors Va (the vector which defines the animal's orientation) and Vb (the vector from the animal to the point).

The angle between the two vectors (*beta*) is subtracted from 180° and the absolute value of the result is calculated; this is compared to the critical angle for movement towards/away from a point (see notes). If the angle is less than this critical angle, then the animal is deemed to be oriented away from the point. If the animal is oriented away from the point, then the time between the previous position of the animal and its current position is added to the total time oriented away from the point (irrespective of whether it was orientated away from the point at the previous position).

Analysis across time This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.

Units	Seconds
Notes	The *critical angle* used to define whether the animal's head is oriented away from the point is the same angle as is used to determine whether the animal is moving towards or away from a point, and is defined in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle *abs(180-beta)* will be compared to 45°). This measure is only available if Head tracking is turned on.

Number of times the animal's head was oriented towards the point

Description	Reports the count of occasions when the animal's head was oriented towards the point.
Calculation method	A vector is created between the centre point of the animal and the animal's head - this vector defines the direction in which the animal is oriented. A second vector is created between the position of the animal's head and the position of the point. The angle between these two vectors is calculated - see figure 5.

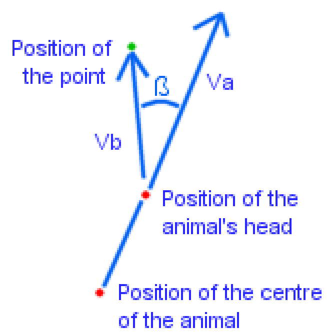


Figure 1. ANY-maze calculates the angle beta between the two vectors Va (the vector which defines the animal's orientation) and Vb (the vector from the animal to the point).

The absolute angle between the two vectors (*beta*) is compared to the critical angle for movement away from/towards a point (see notes). If the angle is less than this critical angle, then the animal is deemed to be oriented towards the point. If the animal is now oriented towards the point when previously it

	wasn't, then the count of occasions when the animal's head was oriented towards the point is increased by one.
Analysis across time	This measure can be analysed across time. The result is calculated using just those positions which fall within the time period.
Units	Seconds
Notes	The *critical angle* used to define whether the animal is oriented towards the point is the same angle as is used to determine whether the animal is moving towards or away from a point, and is defined in the protocol's Analysis options > Movement towards and away from zones and points sub-element. In fact, the value entered in the analysis options is twice the critical angle, as this is more intuitive. The default critical angle is 90° (i.e. by default the angle *beta* will be compared to 45°). This measure is only available if Head tracking is turned on.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Sequences

A Sequence is defined by a series of movements within the apparatus which the animal can perform. Sequences can be used to detect such things as rotations around a water-maze, centre crossings in an open field, or visits in a certain order to the arms of a radial maze.

For more information about sequences, refer to the topics below.

- An introduction to sequences
- Setting up a sequence
- Editing a sequence
- Deleting a sequence
- Sequence measures

An introduction to sequences

ANY-maze has been designed to be extensible, so if your experiments need other types of sequences beyond those described here, please let us know. Where possible we'll add them to the system - for free.

Introduction

ANY-maze includes many measures that relate to the entire apparatus or to individual zones but, while these are generally very useful, what they don't do is report information about movements *between* different parts of the apparatus.

For example, in a place preference box, you might want to know how often the animal goes from the left side of the box through the tunnel and into the right side of the box. Although at first this just appears to be the number of entries into the right side, it is in fact slightly different, because an entry into the right side would be counted both in this situation and also in the situation where the animal goes from the right side into the tunnel and then *back* into the right side again - see figure 1.

To address the need to measure these types of movements *between* different parts of the apparatus, ANY-maze uses sequences. Essentially, a sequence consists of a number of *steps* which the animal must complete in order to perform the sequence. An animal completes a step simply by entering a particular part of the apparatus, so, in the above example, the steps would be 'Enter (or just be in) the left side' then 'Enter the tunnel' then 'Enter the right side'. If the animal completes these steps in the right order, and without going anywhere else, then it would perform the sequence.

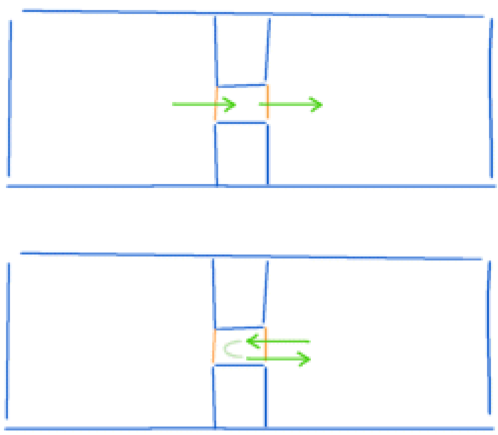


Figure 1. Both of the situations shown would generate an entry into the right side of the place preference box, but just the first one would satisfy a 'Left side > tunnel > Right side' sequence.

Types of sequences

Simple sequences

The sequence described above is an example of a simple sequence. Essentially, to complete the sequence the animal simply has to move through a certain number of steps in the correct order; in this case, Left side > tunnel > Right side.

Bi-directional sequences

The example described above will only be completed if the animal moves from the left side, through the tunnel and into the right side. But you might also be interested in movement from the right side, through the tunnel and into the left side - indeed you might not care which side the animal starts in and which side he ends in; what you want to know is how often he changes sides. In this case, you could specify that the sequence is bi-directional. ANY-maze will then consider the sequence to be performed whether the animal goes through the steps forwards or backwards.

Sequences which can start at any step

In the place preference box sequence, we would always want the animal to start at one end or the other of the sequence, but this might not always be the case. For example, in a water-maze, the sequence shown in figure 2 (below) could be used to detect rotations around the maze. Here, it wouldn't matter where the animal started in the sequence, provided it worked through all the steps in the right order.

As you may notice, there's an implicit requirement for this sequence to be 'circular' - in other words, for the last step to lead back to the first one - so in this case, you would want the sequence to be completed when the animal returns to the starting step. However, this may not always be the case. For example, in a Y-maze, you might be interested in how often the animal moves through all three arms in a certain order (for example, moving from arm A to B to C), but in this case you wouldn’t require the animal to return to the first step in order to complete the sequence; rather, the sequence would be completed when the animal enters the last step.

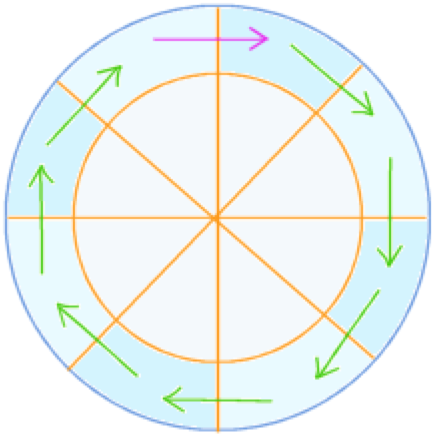


Figure 2. Using a sequence to detect rotations around a water-maze. The purple arrow represents the implicit link between the last and first steps that ANY-maze can add to 'close' a sequence which can start at any step.

You can mix sequences which can start at any step with bi-directional sequences, which is what you would probably want to do in this example - otherwise, the sequence would only count clockwise rotations around the maze.

Sequences which are not broken by moves that are not in any step

In the water-maze example shown above, the animal must move from step to step, and any move to a position which is not in a step will break the sequence – in the case of detecting water-maze rotations, this is exactly what we want. But consider the Y-maze example shown in figure 3; here, we wish to detect moves between the arms, but to get from one arm to another the animal *must* move through the centre of the maze, and this is not part of any arm (and therefore not part of any step). So if we wish to detect moves from arm A to arm B to arm C, the animal will have to actually go from A to centre to B to centre to C, and thus will never be able to complete the sequence.

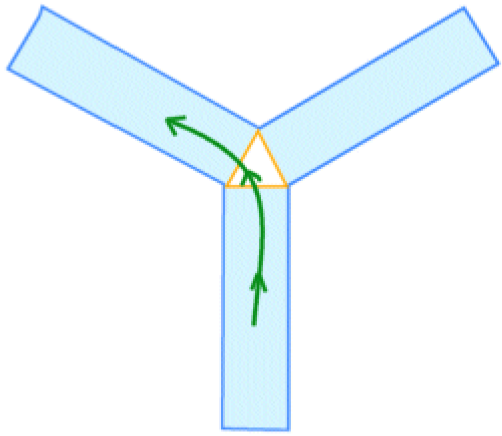


Figure 3. In the Y-maze, the animal will necessarily pass through the centre in order to move from one arm to another. In this case, in order to detect a sequence of movements between the arms, it is necessary to define that the sequence is not broken by moves that are not in any step (i.e. moves into the centre).

To address this, you can specify that a sequence is not broken by moves that are not in any step. Thus in the example, any move to the centre will not break the sequence. It is important to understand that a move to the centre is still a move, just that the sequence processing waits to see what happens next; so A to centre to B to centre to C is just like A to B to C. In the same way, a move from A to centre and then back to A is considered to be a move from A to A. This is important because, for example, the animal might move from B to A to A to C, and this would mean that it has NOT performed the sequence B to A to C.

Non-return sequences

A *non-return sequence* is one in which the steps don't actually have to be performed in order; what's important is that no step is repeated. For example, in a radial arm maze, you might want to see whether the animal visits certain arms just once each, without returning to an arm it's already visited - see figure 3.

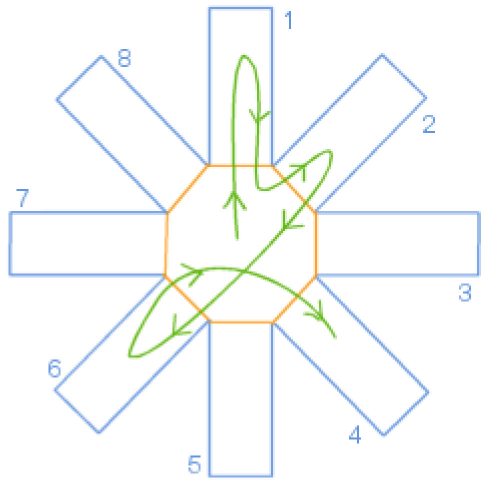


Figure 4. A non-return sequence of 'Arm 1, Arm 2, Arm 4, Arm 6' would be satisfied in this example, despite the fact that the animal actually visited the arms in a different order.

Defining a sequence

Sequences are defined in ANY-maze as a series of steps, where each step defines a part of the apparatus that the animal must enter to complete that step. An important aspect of steps is that they can either be defined by selecting one or more areas in the apparatus map, or they can be defined by specifying a zone.

The ability to define a step by specifying a zone is very useful when you want to include a movable zone in a sequence. For example, in the place preference box described above, we used a sequence to detect movement from the left side through the tunnel and into the right side. But perhaps you would be more interested to know how often the animal moves from a drug-paired side through the tunnel and into a non-drug-paired side. In this case the actual side which is drug-paired (left or right) would probably differ from animal to animal, but defining the steps of the sequence using zones, rather than directly selecting areas of the apparatus map, would automatically take this into account.

On the other hand, in the water-maze rotations example, the areas which the animal would have to enter to complete each step would be of little interest in their own right and you'd be unlikely to have defined them as individual zones. In this case, then, you'd want to define the steps by directly selecting areas in the apparatus map.

Note that when deciding whether to use zones or areas for a sequence step, you'll need to consider whether you'll want to compare sequence and zone measures. This is because a sequence step that uses an *area* of the apparatus will simply use the animal's centre point to determine entry to that area, whereas a step that uses a *zone* will use the zone's entry settings.

To see how this could make a difference, consider the following example: imagine you have set up a sequence in which the animal must move from area A to area B. Also imagine you have created a zone for area A, and you've specified that the animal is in the zone when its *head* is in the zone. Now, you compare the latency to enter Zone A with the latency to start the sequence. Despite the fact that to start the sequence the animal must enter area A, you'll find that the latency to enter Zone A is less than the latency to start the sequence! The reason will be because when the animal's head enters the area, a zone entry will be scored, but the sequence will only begin when the animal's centre point enters the area (which will happen later, and in fact may not even happen at all!). If you set up the sequence so that the first step required the animal to enter *Zone* A (rather than area A), then the latencies would agree.

See also:

- Sequence measures

Setting up a sequence

Introduction

For a general introduction to sequences, see An introduction to sequences.

To add a sequence to a protocol, click the *Add item* button in the ribbon bar and select *New Sequence* from the menu which appears.

The New Sequence menu option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the sequence name.
2. Specify the characteristics of the sequence such as whether it can be performed in either direction and whether it can start at any step.
3. Add steps to the sequence.

What next?

After completing these steps, you should consider whether you want to include any keys in this protocol.

See also:

- Adding elements to a protocol
- Editing a sequence
- Deleting a sequence

Sequence name

In brief

Enter a name for the sequence in the *Sequence name* field on the sequence's settings page. You must make an entry, but it can be anything you like.

Details

Sequence names should uniquely and clearly identify a sequence, because they'll be used to label results. For example, ANY-maze will report 'Number of [SequenceName] sequences', so if you called the sequence 'Sequence 1' then the result would be 'Number of Sequence 1 sequences' - which would be a rather ambiguous result; something like 'Number of Rotation sequences' would be much clearer.

To avoid having very long result names, you should try to keep sequence names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying the characteristics of a sequence

In brief

There are four characteristics which you can apply to a sequence by checking appropriate buttons on the sequence's settings page:

1. *Sequence can be performed in either direction:* The steps of the sequence can be performed either from the first step to the last or from the last to the first.
2. *Sequence can start at any step:* The sequence is treated as a *circular* series of steps, where the last step leads back to the first. You can also optionally specify whether the animal must explicitly return to the starting step to complete the sequence. For example, in a sequence with four steps in which the animal must return to the first step, then the sequence could be completed by moving from step 1 > 2 > 3 > 4 > 1, or by moving from steps 3 > 4 > 1 > 2 > 3. You can specify that a sequence is both a sequence which can start at any step and a sequence which can be performed in either direction.
3. *Sequence is not broken by moves to positions which are not in any step:* Ordinarily, if the animal moves to a position which is not inside any of the sequence steps, then the sequence is broken. For example if sequence is 1 > 2 > 3 and the animal moves 1 > 2 > ? > 3 (where '?' means a position that's not in step 1, 2 or 3) then the sequence would normally be broken. You can use this option to change this, so that in the example the sequence would still be scored.
4. *The steps can be performed in any order but no step can be repeated.* This type of sequence allows you to detect single visits to a number of different areas or zones.

Details

Sequence can be performed in either direction

Imagine a situation where you want to detect crossings between the two sides of a place preference box where the sides are connected with a tunnel. In this situation, you could use a sequence of 'Left side > Tunnel > Right side' to detect such an event. However, you'd be equally interested in a sequence of 'Right side > Tunnel > Left side', and it's this reversal of the sequence's direction which is controlled by this characteristic.

Sequence can start at any step

Imagine a situation where you want to detect rotations around a water-maze. This could be done by defining a sequence like the one shown in figure 1. Here the animal would have to move from one step to another to complete the sequence, but it wouldn't actually matter where the animal started the sequence, provided it went through all the steps. For example, a sequence which ran through steps 3,4,5,6,7,8,1,2 would be just as valid as one which ran from 1 to 8.

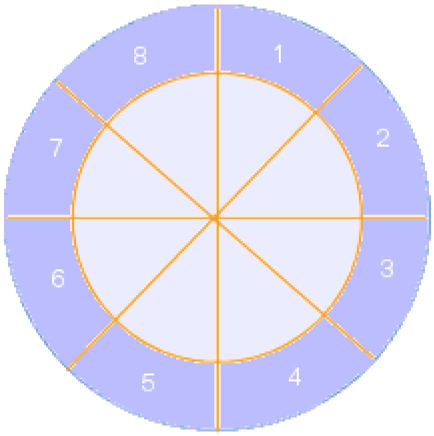


Figure 1. By moving through the numbered steps, the animal will complete a rotation around the water-maze, but it won't matter which step it starts in.

In fact, things are a little more complicated because if the animal starts, for example, in step 1 and works its way round to step 8, it won't actually have completed an entire rotation - see figure 2.

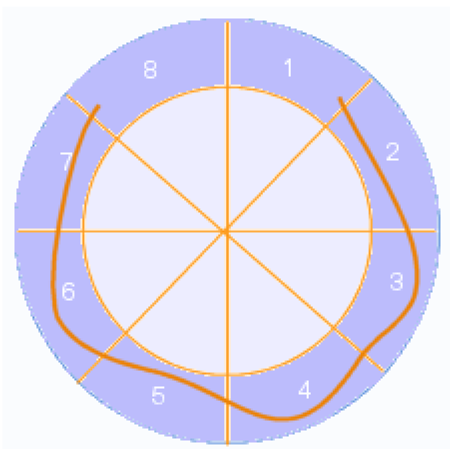


Figure 2. In fact, just moving into the correct area for each step won't necessarily complete a sequence. Here the animals track moves through all the steps from 1 to 8 but the animal hasn't actually completed a rotation.

This could normally be resolved by specifying the first step again at the end of the sequence, but in a sequence which can start at any step you don't actually know which will be the *first* step. ANY-maze resolves this by providing an option to *close* a '*Start at any step*' sequence for you, by requiring the animal to return to the starting step. Thus in the water-maze example, if the animal started at step 5, it would need to proceed as follows: 5 > 6 > 7 > 8 > 1 > 2 > 3 > 4 > 5 to complete the sequence; ANY-maze would automatically have *closed* the sequence by adding step 5 onto the end - see figure 3.

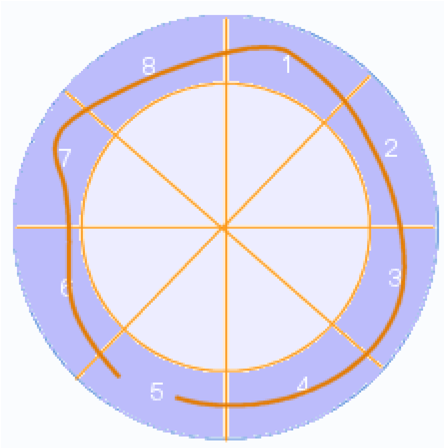


Figure 3. By requiring the animal to return to the first step, a complete rotation is created.

If you wish, you can specify that the sequence ends when the animal enters the final step (rather than having to 'close' the sequence). In this case, after completing the sequence ANY-maze will wait until the animal moves to another step before it starts checking for the next sequence. For example, consider a Y-maze sequence with three steps A,B,C; in this case, the animal might perform the following steps BCABCA; here ANY-maze would consider BCA as completing a sequence and would then wait for the next move (so the move from A to B) before starting the next sequence - which would then be BCA again. Contrast this to what *would* happen if ANY-maze did not wait for a move; in that case, the first sequence would end after BCA and the system would then start the next sequence at A and this would complete after the moves to B and C - in other words the move to A would be both the end step of one sequence and the start step of the following sequence.

In summary, when you specify a sequence that can start at any step, you need to choose whether the sequence requires the animal to return to the start step (in which case the move to the start step will both end one sequence and start the next one), or whether the sequence ends when the animal moves to the last step (in which case, the system waits for a move to another step before starting the next sequence).

By the way, it's often useful to mix the *Sequence can start at any step* characteristic with the *bi-directional* characteristic - in the water-maze example, this would count a rotation whether it was performed clockwise or anti-clockwise.

Sequence is not broken by moves to positions which are not in any step

Sequences which are not defined as *The steps can be performed in any order but no step can be repeated* require the animal to move through a certain sequence of steps, for example 1 > 2 > 3 > 4. So, when at step 2 the animal must move on to step 3, otherwise it will break the sequence.

But the steps are defined as areas in the apparatus, and there may be areas which are not in any of the steps at all - so what if the animal moves into one of these areas; does that break the sequence? Usually the answer is yes - a move to an area which is not part of any step does break the sequence - but you can use this option to change this.

A good example of when this is useful is the Y-maze. Typically, the Y-maze is defined something like figure 4, i.e. with an area in the centre which is not part of any arm. So, if you wanted to know if the animal moved from arm A > B > C you would have a problem, because to get from A > B it would have to pass through the centre and that's not part of any step, so moving there would break the sequence. By using this option you could resolve this, so a move from A > centre > B > centre > C would be scored as a sequence completion.

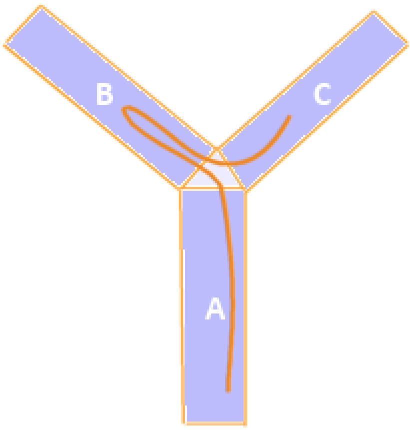


Figure 4. To score the sequence A>B>C, it would be necessary to select the option Sequence is not broken by moves to positions which are not in any step, *because to move from step to step the animal must pass through the centre, which is not part of any of the steps.*

A final point to understand is that the move into 'nowhere' is registered by the sequence analysis, such that a move from A into nowhere and then back into A, *would* break the sequence because once the animal has left A it must move into B, even if it passes through 'nowhere' in order to arrive there.

The steps can be performed in any order but no step can be repeated

Sequences which have this characteristic are called *non-return sequences* in ANY-maze. As this name implies, the animal can't return to a step it has already completed. This is, in fact, the only condition of such a sequence. This means that the animal doesn't have to proceed through the steps in order, nor will moving into an area which isn't part of a step *break* the sequence. The only thing which will *break* the sequence is returning to the area of a step which has already been visited.

This characteristic can't be used with a bi-directional sequence or a sequence which can start at any step.

Setting up a sequence step

Introduction

All sequences should include at least two *steps* (but can't include more than 31). Steps are added to a sequence by ensuring that the sequence element is selected in the protocol list and then clicking the *Add item* button and selecting *New Sequence Step* from the menu which appears. New steps are always added to the end of the sequence.

The New Sequence Step menu option will be disabled if a sequence is not selected in the protocol list.

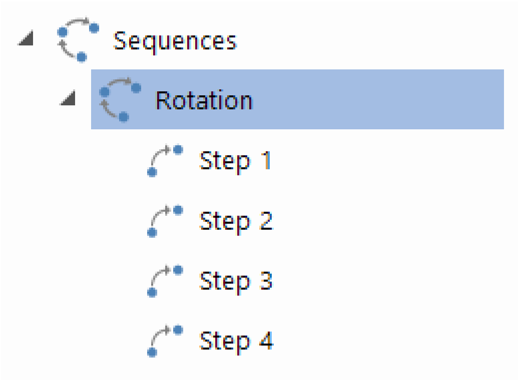


Figure 1. Part of the protocol list showing the steps of a 'Rotation' sequence.

After adding a step, you need to specify which part of the apparatus the animal needs to enter in order to complete the step. There are two ways of doing this; by selecting areas in the apparatus map or by specifying a particular zone.

Defining a step by selecting the areas in the apparatus map

A sequence step represents an area (or areas) of the apparatus which the animal must enter in order to complete the step. You can specify the area simply by clicking on the relevant parts of the apparatus map - the area(s) you select will be shaded in blue. To deselect an area, just click it a second time. Of course, if you have more than one piece of apparatus in your protocol, then you will need to specify the areas in each piece of apparatus separately - the Protocol page's Image pane will be showing all the apparatus maps, so this is a straightforward process.

There are no restrictions on selecting areas, so:

- One area can appear in more than one step
- The areas which constitute a step don't have to be contiguous (i.e. touching each other), although they usually will be.

Important: ANY-maze will consider that the animal has entered the area you select (and thus completed the step), when its *centre point* enters the area.

Defining a step by specifying a zone

Although it's often simplest to specify the areas of the apparatus map which constitute a step by selecting them in the apparatus map (see previous section), this isn't always possible or necessarily desirable.

For example, in a place preference box, you might want to count how often the animal moves from a 'Drug-paired side' through a 'Tunnel' and into a 'Non-drug-paired side'. The problem would be that the physical side of the apparatus which is drug-paired would differ from animal to animal. The solution to this is to specify the step using a zone. In other words, for the first step, the animal would have to move into the 'Drug-paired side zone', wherever that happens to be for the animal being tested, then into the 'Tunnel' and then into the 'Non-drug-paired side zone'. Indeed, in this example the 'Tunnel' would probably be a *hidden* zone (i.e. the animal would disappear from the camera's view while in the tunnel) so this step would also have to specified using a zone rather than selecting areas in the apparatus map.

Another situation when it's better to use a zone to define a step is when you want to compare sequence and zone measures. This is because ANY-maze will consider that the animal has entered the zone (and thus completed the step) based on the zone's entry settings, rather than simply on the centre point (which is what it does for steps defined using areas). For example, imagine you have set up a sequence in which the animal must move from area A to area B. Also imagine you have created a zone for area A, and you have specified that the animal is in the zone when its head is in the zone. Now you compare the latency to enter Zone A with the latency to start the sequence. Despite the fact that to start the sequence the animal must enter area A, you'll find that the latency to enter Zone A is less than the latency to start the sequence! The reason will be because when the animal's head enters the area, a zone entry will be scored, but the sequence won't begin until the animal's centre point enters the area (which will happen later, or may not even happen at all!). If you set up the sequence so that the first step required the animal to enter Zone A (rather than area A), then the latencies would agree.

You can mix how you define steps within a single sequence so, for example, some steps could be specified by selecting areas in the apparatus map while others could be specified by specifying zones.

Editing a step

You can edit a step at any time, whether before, during or after an experiment has been performed. To edit a step, you simply have to select it in the protocol list - its settings and areas will be displayed in the Settings and Image panes, respectively.

Deleting a step

You can delete a step at any time by selecting the step in the protocol list and clicking the ❌ *Remove item* button in the ribbon bar.

Deleting a step will remove it from the sequence, and the steps *below* it will simply be *moved up*. This means that the remaining steps will be renumbered, which can make it look like the step hasn't been deleted. For example, if you delete *Step 1* in a three step sequence then the old *Step 2* will be renumbered *Step 1* and the old *Step 3* will be renumbered *Step 2*. Thus, as there will still be a *Step 1,* it will appear that *Step 1* wasn't deleted.

If you delete *all* the steps in a sequence, then the sequence will continue to exist but all of its results will be either zero or undefined.

Editing a sequence

Introduction

You can edit absolutely everything related to a sequence at any time, whether before, during or after an experiment has been performed.

See also:

- Editing the elements of a protocol
- Setting up a sequence
- Deleting a sequence

Deleting a sequence

Introduction

You can delete a sequence, or sequence step, at any time whether before, during or after an experiment has been performed.

To delete a sequence or step, simply select it in the protocol list and then click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting sequence.

See also:

- Deleting protocol elements
- Editing a sequence

Sequence measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a sequence:

- Number of sequences
- Total time performing sequences
- Latency to start of first sequence
- Latency to completion of first sequence
- Average time to complete the sequence
- Maximum duration of a sequence
- Minimum duration of a sequence
- Total distance travelled during sequences
- Average distance travelled per sequence
- Maximum distance travelled during a sequence
- Minimum distance travelled during a sequence
- Average speed during the sequence

Number of sequences

Description	Reports the number of times the sequence was performed.
Calculation method	Each time a sequence ends, the number of sequences is incremented by one.
Analysis across time	This measure can be analysed across time. A sequence is considered to occur in the time period in which it **ends**. For example, if a sequence starts at time 15 seconds and continues until time 45 seconds, then (for 30 second time periods) it would be counted in the time period 30-60 seconds because this is the period in which it ended.
Units	None
Notes	None

Total time performing sequences

Description	Reports the total amount of time that the animal was performing sequences during the test.
Calculation method	Sums the time taken to complete each sequence.
Analysis across time	This measure can be analysed across time. A sequence is considered to occur in the time period in which it **ends**, therefore the time performing the sequence is ALL attributed to the time period in which it ends. In an extreme case, this can yield a result in which the time performing a sequence in a time period is *longer* than the period itself. For example, if a sequence started at time 5 seconds and ended at time 55 seconds, then (for 30 second time periods), its duration - 50 seconds - would be counted in the time period 30-60 seconds because this is the period in which it ended, but the duration would actually be longer than the time period. This apparent strange way of calculating results for periods is done to ensure that the result for Average time to complete the sequence is correct.
Units	Seconds
Notes	None

Latency to start of first sequence

Description	Reports the length of time which elapsed before the first sequence in the test started.
Calculation method	Time at which first sequence starts.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	This measure is affected by the option to *Use the test duration as the latency for events which don't occur* in the Analysis element.

Latency to completion of first sequence

Description	Reports the length of time which elapsed before the first sequence in the test ended.
Calculation method	Time at which the sequence first ended.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	This measure is affected by the option to *Use the test duration as the latency for*

events which don't occur in the Analysis element.

Average time to complete the sequence

Description	Reports the average amount of time taken to complete the sequences.
Calculation method	Calculated by dividing the Total time performing sequences by the Number of sequences. If there are no sequences in the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result is Total time performing sequences in the period divided by the Number of sequences in the period. If no sequences were performed during a period, then the result is undefined.
Units	Seconds
Notes	This measure is affected by the option to *Use zero as the result for undefined averages* in the Analysis options element.

Maximum duration of a sequence

Description	Reports the maximum amount of time taken to complete a single sequence.
Calculation method	The duration of each sequence is calculated when the sequence ends and the longest one is found. If no sequences were performed during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result for a time period is the duration of the longest sequence which **ended** during the time period. If no sequence ended during the time period, then the result is undefined.
Units	Seconds
Notes	None

Minimum duration of a sequence

Description	Reports the minimum amount of time taken to complete a single sequence.
Calculation method	The duration of each sequence is calculated when the sequence ends and the shortest one is found. If no sequences were performed during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result for a time period is the duration of the shortest sequence which **ended** during the time period. If no sequence ended during the time period, then the result is undefined.
Units	Seconds

Notes None

Total distance travelled during sequences

Description	Reports the total distance travelled by the animal while performing sequences in the test.
Calculation method	Sums the distance travelled during each sequence in the test.
Analysis across time	This measure can be analysed across time. The result for a time period is the sum of the distance travelled for all the sequences that **ended** during the time period.
Units	Metres
Notes	None

Average distance travelled per sequence

Description	Reports the average distance travelled by the animal while performing each of the sequences in the test.
Calculation method	Calculated by dividing the Total distance travelled during sequences by the Number of sequences. If there were no sequences performed in the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result for a time period is the Total distance travelled during sequences which **ended** during the time period divided by the Number of sequences which **ended** during the time period. If there were no sequences which ended during the time period, then the result is undefined.
Units	Metres
Notes	This measure is affected by the option to *Use zero as the result for undefined averages* in the Analysis options element.

Maximum distance travelled during a sequence

Description	Reports the maximum distance travelled by the animal while performing a single sequence.
Calculation method	The distance travelled while performing each sequence is calculated and the greatest distance is found. If no sequences were performed in the test, then the result is undefined.

Analysis across time	This measure can be analysed across time. The result for a time period is the greatest distance travelled during any sequence which **ended** during the time period. If no sequence ended during the time period, then the result is undefined.
Units	Metres
Notes	None

Minimum distance travelled during a sequence

Description	Reports the minimum distance travelled by the animal while performing a single sequence.
Calculation method	The distance travelled while performing each sequence is calculated and the smallest distance is found. If no sequences were performed in the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result for a time period is the smallest distance travelled during any sequence which **ended** during the time period. If no sequence ended during the time period, then the result is undefined.
Units	Metres
Notes	None

Average speed during the sequence

Description	Reports the average speed of the animal while performing the sequences in the test.
Calculation method	Calculated by dividing Total distance travelled during sequences by the Total time performing sequences. If no sequences were performed in the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result for a time period is the Total distance travelled during sequences which **ended** during the period divided by the Total time performing sequences which **ended** during the period. If no sequence ended during the time period, then the result is undefined.
Units	Metres per second
Notes	This measure is affected by the option to *Use zero as the result for undefined averages* in the Analysis options element.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Animal colour and size

Overview

ANY-maze detects animals by looking at the contrast between them and the background of the apparatus. Thus, for ANY-maze to track successfully, the animals must be either lighter or darker than the background (or both). That said, even a very small level of contrast is usually sufficient for ANY-maze to track well - see figure 1.

You must specify the animal's 'colour' in order to track it. You do this using the options on the *Animal colour* element of the protocol.

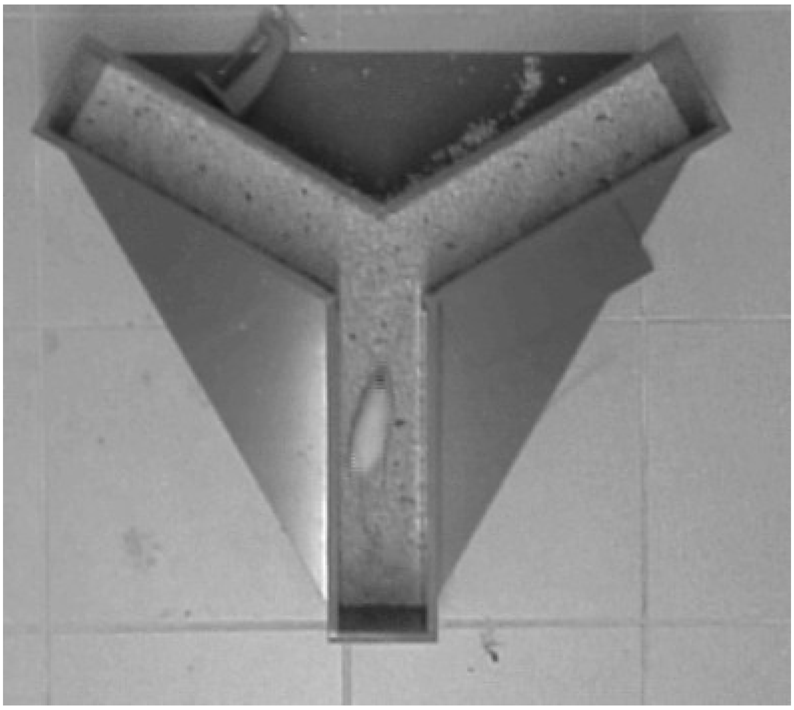


Figure 1. There's very little contrast between this white mouse and the sawdust in the Y-maze, but by telling ANY-maze that the animal is lighter than the background the system is still able to track accurately.

When your protocol includes a hidden zone, you will also need to specify the animal's approximate length.

- Specifying the animal's colour

- Specifying the animal's length
- Adjusting the contrast sensitivity

Specifying the animal's colour

Normally, it is easy to specify whether the animals are lighter or darker than the background, but there are a number of situations which require special attention.

- There's very little difference between the animals and the background
- The animals are not a uniform colour
- The background is not a uniform colour
- Some animals are lighter and some are darker than the background
- All of the animal is either lighter or darker than the background, but some parts contrast much more than others

There's very little difference between the animals and the background

In this situation, if the animals are light but are very similar in colour to the background (for example, a white mouse walking on sawdust), then you should specify that the animals are lighter than the background. You may notice that there's a shadow around the animal, and so you could specify that they're darker than the background - however, tracking shadows is generally a bad idea and will give poor results.

If the animals are dark but are very similar to the colour of the background, then you may need to experiment to see which setting gives the best tracking. Start by specifying that they're darker, and if this doesn't work very well then try the selecting the option *The background and/or the animals are not uniform*.

The animals are not a uniform colour

If you wish to track non-uniform animals, such as *Lister hooded rats* (which are white with black heads), then obviously they won't be either lighter or darker than the apparatus background. In this case, you can do one of three things:

- Select the *Animals are lighter* option. ANY-maze will then track just the light part of the animal.
- Select the *Animals are darker* option. ANY-maze will then track just the dark part of the animal.
- Select the *Animals are not uniform* option. ANY-maze will then attempt to track the entire animal. How well this will work will depend on how *balanced* the difference between the light and dark areas of the animal and the background are. For example, if the background is mid-grey and you're using a Lister hooded rat, then the intensity of

the difference between the white part of the animal and the background will be similar to the intensity of the difference between the black part of the animal and the background - in this case ANY-maze should track the entire animal. On the other hand, if a Lister hooded rat is on an light grey background, then the difference between the black part of the animal and the background will probably be much greater than the difference between the white part and the background - in this case the system will probably just track the dark part of the animal, as it will present a much bigger 'signal' than the light part. In general, we recommend that you avoid using the *Animals are not uniform* option in this type of situation, but in some cases it will work well.

The background is not a uniform colour

This situation is similar to the one described above (where the animals are not uniform), but in this case you can't use either of the *Animals are lighter* or the *Animals are darker* options because if you do, the system will only track the animal in some parts of the apparatus!

The only choice in this situation is to choose the option for *The background and/or the animals are not uniform*. See figure 1 for an example of when this option is required.

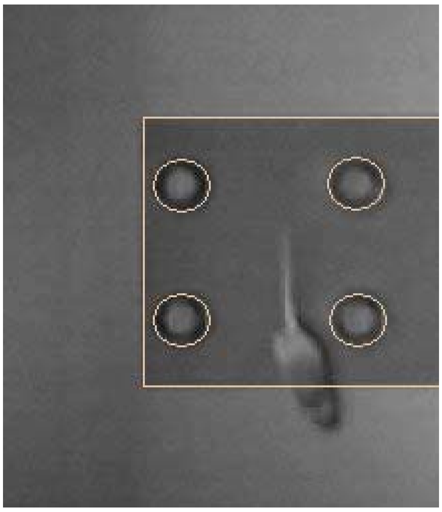


Figure 2. In this hole board apparatus, the animal is lighter than the background in the centre of the apparatus but is darker than the background around the edge. However, by using the option for a non-uniform background, ANY-maze tracks very well.

Some animals are lighter and some are darker than the background

If you're working with different strains of animals, then some may be lighter and some darker than the background of the apparatus. In this case, you should specify the option *Some are lighter and some are darker* and then choose whether ANY-maze should automatically determine the animal's colour or whether you will enter it manually.

- If there is good contrast between the animal and the apparatus background, and provided there are no moving reflections, then ANY-maze can generally determine the colour of the target animal for itself. This is obviously desirable, as it saves you from having to specify the colour of each animal manually.
- If there isn't very good contrast between the animals and the apparatus background, or if there are moving reflections in the apparatus (for example, reflections from the surface of the water in a water-maze), then ANY-maze will not be able to detect the animals' colour reliably. In this case, you should choose the option to *Specify the animal colour manually.* Doing this will cause ANY-maze to include a column in the Animals spreadsheet, where you will be able to specify the appropriate setting for each animal individually.

Adjusting the contrast sensitivity

All of the animal is either lighter or darker than the background, but some parts contrast much more than others

Imagine you have a 'white' rat on a brilliant white background - in this case, the animal will almost certainly be darker than the background and selecting the option *The animal is darker than the apparatus background* will work correctly. However, imagine that the animal now has a black *skull cap* fitted - this will offer very high contrast to the background, whereas the animal's body only offers low contrast. In this case, ANY-maze will tend to consider the high-contrast skull cap as the target to track and will ignore the low-contrast body of the animal - clearly, this is not what you want.

To overcome this, you can increase the *Contrast sensitivity* so that ANY-maze considers a wider range of contrasts as being part of the animal. There are 5 levels of contrast sensitivity, and you should use the lowest level that gives the correct results.

Note that increasing the sensitivity unnecessarily will usually make the tracking worse.

Specifying the animal's length

When a protocol includes a hidden zone, the *Animal colour* protocol element will change its name to *Animal colour and size,* and will include a field for you to enter the approximate length of the animal.

The reason ANY-maze needs to know the animal length is so that it can determine when the animal becomes visible if it is hidden at the very start of the test - otherwise, it would have no idea how large a target it should be looking for.

The length you enter can be quite approximate, but should exclude the animal's tail. For mice a value of around 60mm is usually about right, and for rats around 180mm - although this will depend on the age and strain of the animals.

Tracking the animal's head

Overview

Normally, ANY-maze tracks the centre of the animal, but in the right conditions, it can also track the animal's head - see figure 1 below.

To switch head tracking on, select the option *Yes, I want to track the animal's head* in the *Tracking the animal's head and tail* element of the protocol.

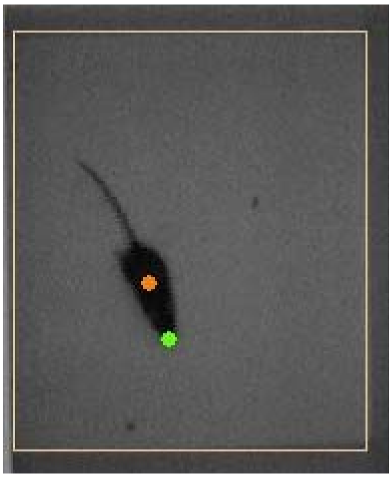


Figure 1. In this open field, ANY-maze is tracking both the animal's centre (the orange dot) and the animal's head (the green dot).

- Conditions required for head tracking to work
- Indicating the position of the animal's head during tests
- Head tracking results
- Tracking the animal's head after a test has been performed

Conditions required for head tracking to work

Head tracking requires a moderate level of contrast between the animal and the background of the apparatus; it is also most reliable when the background is a consistent colour throughout (as in the figure above).

The reason for these prerequisites is that ANY-maze determines the position of the animal's head by analysing the animal's shape, and the only way it is able to determine the shape reliably is if there's both moderate and consistent contrast between the animal and the apparatus background.

By the way, when tracking an animal's head, you might notice that sometimes the green dot (the position of the head) gets *left behind* for a moment when the animal moves (despite the fact that the orange dot continues to track the animal). This occurs when the system can't determine where the animal's head is, so it simply assumes the head position is wherever it was when it was last detected. Although this can look slightly odd while tracking, it doesn't actually make much difference to the results, provided the system doesn't lose the head for more than about a second.

Indicating the position of the animal's head during tests

By default, ANY-maze doesn't indicate the position of the animal's head while tracking (i.e. the green dot shown in the figure above is not displayed), but you can alter this by selecting the appropriate option in the What to display while testing page. You can also change the colour of the marker using the Options > Appearance page.

Head tracking results

If you set the system to track the animal's head, then ANY-maze will automatically include a range of results for the head position in both the apparatus measures and in the zone measures.

Tracking the animal's head *after* a test has been performed

If you wish to, you can turn head tracking on *after* a test has been performed. ANY-maze will simply reanalyse the animal's track from the data it recorded during the test, and the results will then be available. Similarly, you can switch head tracking off after a test has been performed - this will simply remove the *head measures* from the list of available results (if you turn it back on, they'll come back).

See also:

- What to display while testing
- Changing ANY-maze colours
- Apparatus measures
- Zone measures

Immobility detection

Introduction

ANY-maze can automatically detect periods when the animal is immobile and include analysis of mobility / immobility in test results.

You can specify whether you want it to do this using the options in the *Immobility detection* element of the Protocol.

- Determining when an animal is mobile
- Defining immobility
- The difference between mobility and activity
- Mobility and hidden zones

Determining when an animal is immobile

Contrary to expectations, determining when an animal is immobile isn't very obvious. For example, if an animal is walking along, stops for 10 seconds and then starts walking again then you'd probably say it was immobile for 10 seconds. But what if it stopped for 1 second? OK, maybe you'd still say it was immobile, but what if stopped for 1/4 of a second - you'd probably say that it never really stopped. So the first issue is how long it has to stop for before you consider it to be immobile.

A second issue is related to what *immobile* actually means. For example, if an animal lies down and goes to sleep then you'd probably say it's immobile, but what if it sits in one place and scratches itself. It's not changing location but it's definitely moving, so is it immobile or not?

These two issues are addressed in ANY-maze by allowing you to choose how you want mobility/immobility to be defined.

Defining immobility

There are two aspects to defining immobility; firstly, you need to decide how motionless the animal has to be in order to be considered immobile. For example, does it have to be completely immobile, perhaps asleep, or does it simply have to remain in the same place (even if it's moving its body, for example scratching or moving its tail). You specify this level of immobility by defining the *immobility sensitivity* on a scale from 50 to 100. In fact, these values are just the percentage of the animal which has to remain in the same place for the animal to be considered immobile. So 100% means the animal must be absolutely immobile, while 50% means that it can actually move quite a lot (grooming, stretching, etc.) and still be considered immobile. In general, a value of around 70% is a good level for most situations.

The second aspect to defining immobility is to specify how long the animal must *remain* in an immobile state before you want ANY-maze to actually consider it to be immobile. For example, setting 2 seconds will mean that the animal can be technically immobile for say 1 second, without ANY-maze actually counting it as immobile.

The exact values you use for both the immobility sensitivity and the immobile time will largely depend on why you're interested in immobility. For example, you might want to use immobility to end a test if the animal is continuously immobile for 5 minutes, e.g. it's gone to sleep. In this case, you'd probably want to use quite a high immobility sensitivity, as any movement (such as grooming) would suggest that the animal is still awake, and you could also choose quite a high immobile time, say 5 seconds, as you'd only actually stop the test if the animal remains immobile for 5 minutes.

On the other hand, if you want to know how often the animal stops moving around in an open field, then you might use a low value for the *immobility sensitivity* and a time value of perhaps 2 seconds. In other words, if the animal stops changing its location (even if it continues to move its body) for a period of 2 seconds or more, then you'd say it's immobile.

The actual time value you enter can be expressed in milliseconds, seconds, or minutes by using units of ms, s and min respectively; if you don't enter any units, then seconds will be assumed. You can also enter decimal values, for example '1.5s'. The minimum value you can enter is 200ms and the maximum is 1 minute.

The difference between *mobility* and *activity*

As well as *mobility,* ANY-maze can also analyse *activity*. In this sense, *mobility* is determined by the movement of the animal around the apparatus - when it's moving, it's mobile (subject to the conditions described above) whereas *activity* is determined by whether the animal is doing something. For example, if the animal is feeding then it probably won't be mobile (i.e. it'll be staying in the same place) but you would probably still consider that it's active - i.e. it is doing something. Effectively, then, mobility is one form of activity, but not the only one.

You will actually be able to choose whether other behaviours, such as feeding, should count as an activity when you add them to the protocol - see An introduction to using keys to score behaviours for more details.

Mobility and hidden zones

When the animal is in a hidden zone (i.e. you can't see it), ANY-maze will consider that it's mobile. This may seem odd, but essentially any decision as to whether the animal is mobile or not is arbitrary, as we can't actually know.

This means that results such as 'total time mobile' will include the period that the animal was in the hidden zone. If this doesn't suit your needs, then you can simply use a calculation to deduct the time in the hidden zone from the time mobile.

Freezing detection

Introduction

ANY-maze can automatically detect periods when the animal is freezing, and include analysis of them in the test results.

You can specify whether you want it to do this using the options in the *Freezing detection* element of the protocol.

- Determining when an animal is freezing
- Setting the freezing thresholds
- Setting the minimum freeze duration
- The difference between freezing detection and immobility detection

Determining when an animal is freezing

Freezing detection in ANY-maze is performed on an apparatus-wide basis. Essentially, ANY-maze looks for movement anywhere in the apparatus, and when it can't find any it considers the animal to be freezing. Of course, the system takes into account that the video image will include some noise (the individual pixels tend to flicker a little) and that even when freezing, an animal will still breathe and therefore it will move very slightly.

The mechanics of freezing detection require that the system finds a sequence of images in which there is no movement before it will actually score the animal as 'freezing'. This means that the moment when it concludes that the animal is freezing will always be a short time after the freezing actually began. ANY-maze automatically adjusts for this delay (which varies, but is typically around 200ms) such that the start of a freezing bout will be scored at the true start, and not at the 'detection' start.

Setting the freezing thresholds

To actually detect freezing, ANY-maze calculates a *freezing score*. This is a unit-less value whose exact calculation method is rather complex, but what's important is that the more (non-noise) movement there is in the image, the higher this score will be. So, to determine that the animal is freezing, ANY-maze requires that this score goes below a certain value, but the question is 'what value?'.

The answer to this is that you can specify the value yourself. By default, ANY-maze will use a value of 30 but you could reduce this if you want the animal to be *really* frozen, or you could raise it if you'd like to be a little less strict about what constitutes freezing.

In fact, you specify two *threshold* values; one which determines when the animal should be considered to be freezing (this is the one value described above), and another to determine when the animal should be considered as having *stopped freezing*. To understand this, consider the following sequence of freezing scores recorded at 10Hz:

120, 52, 40, 30, 28, 31, 27, 20, 18, 26, 28, 43, 60, 76

Assuming the freezing *on* threshold is 30, then after 0.3 seconds we see the start of a freezing episode, but 0.2 seconds later the freezing score goes to 31 before dropping back to less than 30 and staying there for a further half a second. So, is this two freezing episodes or just one? Well, if the freezing *off* threshold is also 30 then this would be scored as two episodes, but *you* would probably score it as one. To address this, you can set the *off* threshold to be slightly higher than the *on* threshold (40 is typically used); this means that *once* the freezing score goes below the *on* threshold, it must rise to above the *off* threshold before freezing will be considered to have ended. Thus by using two thresholds, ANY-maze introduces some hysteresis into the freezing detection, avoiding very rapid on/off switching which could otherwise occur.

Of course, you don't have to use this feature - if you set the *on* and *off* thresholds to the same value then no hysteresis will be used.

By the way, one of the options in the What to display while testing page of the protocol is the *Freezing score;* this isn't particularly useful, except that you can use it to get an idea of the kind of scores you might want to use as your freezing *on* and *off* thresholds. To do this, just set the score to be shown and then test some animals. As they're tested, you can see when you think they freeze and see what score ANY-maze is giving them; you can then use these scores as your thresholds to automatically detect freezing bouts.

Another place where ANY-maze displays the freezing score is in the Test data report, where you can see every score recorded throughout the test.

Like almost everything in the protocol, freezing thresholds can be changed at any time, including after a test has been performed. If you do this, then the new thresholds will be applied to any performed tests and their freezing results will be updated accordingly - you don't need to do anything to make this happen; it's automatic.

Setting the minimum freeze duration

Using the freezing score thresholds described above is a good way to prevent very rapid switching between the freezing and the non-freezing state but it won't make any difference if the animal moves, freezes for a very short time and then moves again, as the freezing score will drop and then rise smoothly. This is 'correct' in that the animal really did freeze for just a moment.

What this means, however, is that even an extremely short period of freezing will be scored, even though it may not be 'freezing' in the behavioural sense, it may just be that the animal didn't move for a tenth of a second. To address this you can specify a minimum 'freeze' duration, such that the animal must freeze for at least this long before ANY-maze will count it as actually being in a freezing episode.

The default value used is 250ms, but you can change this to 0ms (so that even the shortest detectable 'freeze' will be scored) all the way up to 6000ms (in which case the animal has to freeze for 6 complete seconds before ANY-maze will consider it to be 'frozen').

Note that ANY-maze will score a freezing episode as starting at the time the animal started to freeze, not at the time at which it had frozen for this minimum duration. For example, if the animal froze at time 10s and then moved again at 12s, and the minimum freeze duration was 500ms, then at time 10.5s, ANY-maze will decide that this really is a freezing episode. Since it knows the episode started at time 10s, it will report it as lasting for 2 seconds.

The difference between freezing detection and immobility detection

Immobility detection uses the ANY-maze tracking system to determine how much of the animal is moving. This works very well for determining whether the animal is staying in the same place, or even for determining if it is not moving its body very much, but it lacks the precision required to determine whether the animal is freezing.

To address this, the freezing detection uses a different approach which is completely unrelated to the tracking of the animal. This is based on analysis of movement throughout the apparatus, and is able to detect even the smallest movement while remaining broadly unaffected by image noise - making it ideal for detecting true 'freezing' of the animal.

Tracking options

Introduction

ANY-maze includes a couple of tracking options which are designed to address some specific situations which can arise in certain apparatus and/or tests. These are set using the *Tracking options* element of the protocol.

- Tracking when the animal will already be in the apparatus at the start of the test
- Tracking when the apparatus is inside an enclosure (Legacy option)

Tracking when the animal will already be in the apparatus at the start of the test

Before choosing to use this option, you should check whether including an accustomisation period in your tests would be a better solution.

In most experiments, you will put the animal into the apparatus at the start of the test - this is the case for the plusmaze, water-maze, etc. However, you might encounter a situation in which you want to start the test with the animal already in the apparatus. For example, you might want an animal to be in the apparatus for 24 hours before you actually perform the test.

As you may know, ANY-maze works by comparing images of the apparatus at different times, and it uses the difference in the images to detect the animal. At the very beginning of a test, the system needs to have an image of the apparatus with no animal in it at all - this means it can quickly determine the animal's position and start tracking without any delay (it can in fact start tracking even if the animal is present, but this can lead to inaccuracies in the first few seconds of the test).

So, the system needs to have an image of the apparatus with no animal in it, and normally it captures this image for itself before you start a test. However, if the animal will already be in the apparatus, then obviously this isn't going to work and so a different method must be adopted - that's what this option is for.

If you switch this option on, then before you can begin a test ANY-maze will ask you to 'capture' an image of the empty apparatus. You do this by clicking the button which will be shown above the apparatus image in the *Tests pane*. The important point about this is that you can capture this image at *any time* before the test - and after you've captured the image, you can exit ANY-maze or even switch your computer off.

So in the case of the example I gave above, in which the animal will be in the apparatus for 24 hours before you perform the test, you would simply capture the image *before* putting the animal in the apparatus. You could then switch off the computer for the 24 hour period, and then switch it on again when you want to run the test.

Actually performing a test in this situation is just the same as normal, except that the 'Auto-start' feature is usually unnecessary as the animal is already in the apparatus. So all you have to do is click *Start* and the test (and the tracking) will begin. By the way, you can use auto-start if you want to - this is useful if you need to make some kind of alteration to the apparatus at the start of the test. For example, if you need to open a door in the apparatus then you could use auto-start so that when you walk away from the apparatus after opening the door, the test would start automatically.

Retaining background images from one test to the next

As explained above, when manually 'capturing' images of the empty apparatus, you have to click the button before each test, and you have to do this when there is no animal in the apparatus. This is usually not a problem for the first test in an experiment, but what if you intend to run a series of tests on the same animal, and you don't intend to remove the animal from the apparatus between one test and the next?

Clearly, in this case you won't be able to 'capture' a new empty-apparatus image before each test, so instead you can select the option to *Retain the background picture from one test to the next.* This will cause ANY-maze to automatically use the background image from one test as the 'captured' background image for the next test, just as if you had 'captured' the image yourself.

There are a few points to note about this:

- For the first test in the experiment, you will still have to capture the background image yourself, as there is no 'previous' test whose background image can be retained.
- The background image that is saved is the background image from the *end* of the previous test, which is not the same as the background image you captured manually before the start of the first test. This is advantageous, as it means that should the background change during the test (for example, the animal digs some holes in sawdust bedding) then the background image that is saved will reflect these changes.
- The background image is saved in exactly the same way as it would be if you had manually captured it. This means that you can exit ANY-maze between one test and the next, and the background will still be retained.
- You can 'over-ride' the automatically retained background image by clicking the button in the usual way.

Tracking when the apparatus is inside an enclosure

This feature is a legacy option which was removed from ANY-maze in version 5.1. It will still appear if you open a file that was created in an earlier version and which used this function.

If your apparatus is inside an enclosure such as a soundproof box, then ANY-maze will need to know this. The main reason is related to the changes in light levels which will probably occur when you open the box to put the animal inside, although there are other technical considerations as well, including the fact that the auto-start feature will work in a different way as you are unlikely to actually appear inside the box.

Note that you don't need to switch this option on if the apparatus isn't entirely enclosed. For example, if you have a light-dark box in an open fronted cupboard (i.e. one with no doors), then you *don't* need to switch this option on. If, however, the box is inside a cupboard which does have doors, and you close the doors during tests, then you should turn this option on.

Although most of the issues related to tracking inside an enclosed box are addressed by setting this option on, there are a few other things you should be aware of and these are listed in Testing do's and don'ts

Keys

You can use keys to record behaviours that ANY-maze can't score automatically, such as grooming. To do this, you simply define a specific key on the computer's keyboard to correspond to the behaviour and then press the key when the behaviour occurs.

For more information about keys, refer to the topics below.

- An introduction to using keys to score behaviours
- Testing which keys can be used together
- Setting up a key
- Editing a key
- Deleting a key
- Key measures

An introduction to using keys to score behaviours

Introduction

Although ANY-maze can automatically score many measures, there may be some specific behaviours that you would like to score which the system can't detect - grooming is a good example. Of course, you could simply revert to a manual system to score these behaviours but a much better method is to use keys.

- Scoring behaviours with keys
- Working with multiple apparatus
- Key results
- Keys and activity
- Scoring large numbers of behaviours

Scoring behaviours with keys

Keys are keys on your computer's keyboard, and you can use them to record behaviours during a test. For example, you might choose to define the 'G' key for grooming.

In this case, you would simply observe the animal on the computer's screen during the test and whenever it starts grooming, you would press down the 'G' key, releasing it only when the behaviour ended. As you only have to press a key to score a behaviour, it's quite easy to simultaneously score two or more.

If you want it to, ANY-maze can indicate the active behaviours while it tracks - see figure 1.

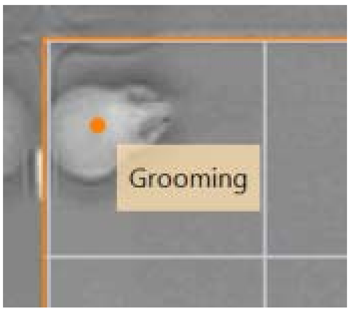


Figure 1. ANY-maze can indicate which behaviours are currently occurring while it tracks.

Working with multiple apparatus

If a protocol includes more than one piece of apparatus, then you will need to define one key for each behaviour in each apparatus. For example, with two plusmazes you might define the 'G' key for grooming on Plusmaze 1 and the 'L' key for grooming on Plusmaze 2. Of course, you could also score other behaviours too, perhaps rearing using the 'R' and 'P' keys.

Key results

For any behaviour that you score, ANY-maze can report a range of different results including the total number of presses (i.e. the number of episodes of the behaviour) and the total time pressed (i.e. the total duration of the behaviour). What's more, the system can report these values for the apparatus as a whole and also for individual zones - so, for example, you could easily get the result for 'Time spent grooming in the drug-paired side zone' as well as 'Total time spent grooming'.

Keys and activity

In some apparatus, you may be interested in how much time the animal is *active* for. In this case, you would probably consider that the animal is active if it's *mobile,* but what if it's standing in one place and grooming? You would probably still consider that it's active.

This is the essential difference between *mobility* and *activity* in ANY-maze - mobility just refers to the movement of the animal, whereas activity can also include other behaviours scored using keys.

When you set up a key, you can choose whether or not the behaviour it represents should be counted as an *activity* or not. For example, 'Grooming' probably would be, but 'Sleeping' probably wouldn't.

Scoring large numbers of behaviours

If you want to score more than a few behaviours, then it can become quite difficult to keep track of what's happening and press all the correct keys at the right times. For example, imagine you want to score grooming, sniffing, rearing, feeding and drinking; you would need to define five different keys. Then you would watch the animal and press the keys whenever one, or more, of the behaviours occurs. The problem is that the animal might be feeding, pause and start sniffing for a second, then immediately start grooming before quickly starting to feed again. Even with none of these behaviours occurring simultaneously, it would still be hard to register them all accurately - and if you did want to score behaviours which could occur simultaneously, then the problem would just get worse.

To address this, ANY-maze allows you to score the same test *more than once.* So in the example, you might score feeding and drinking while watching the 'live' test, then you could watch a video of the test and score grooming and rearing, and finally you'd watch the test video again and score sniffing. This way, you can focus on just a couple of behaviours each time, and it would be much easier to score them accurately. Note that you do need to have recorded a video of the test in order to do this, but you can simply have ANY-maze record the video automatically as the 'live' test is run.

This ability to score the same test multiple times is described in detail in the Adding additional scoring to a performed test topic.

See also:

- Testing which keys can be used together
- Adding additional scoring to a performed test
- Key measures

Testing multiple keys

If the multiple key test window appears to stop working, then close it and reopen it again. This can occur if you press the tab key.

Introduction

If you need to use more than one key to record behaviours in your protocol, you should test that the keys you want to use can all be used together. Because of the way computer keyboards work, some combination of keys won't be registered if all the keys are pressed down at the same time.

To test which keys can be used together, ANY-maze includes the *Multiple key test window,* which can be accessed by selecting the *Keys* element in the protocol list and clicking the *Multiple key test window link shown in the Settings pane.*

Using the multiple key test window

To test a combination of keys, you should perform the following steps:

1. Press and hold down each key one after the other, and check that as you do this, each key is shown as pressed in the window. For example, to test R H and L, press the R key and hold it down, then press the H key and hold it down too, finally press the L key - in the end, all three keys should be held down.
2. Hold down all the keys, then release and press each one in turn checking that each one is shown as released and pressed in the window as you do this. For example, hold down the R, H and L keys. Then release the R key and check it's shown as released, then press it again and check it's shown as pressed. Next release the H key and check it's shown as released, then press it again and check it's shown as pressed. Finally, repeat the same process for the L key.
3. Press all the keys together and check that they're all shown as pressed in the window.

While testing keys, your computer may beep - this is normal and you can ignore it.

Setting up a key

Introduction

For a general introduction to keys, see An introduction to using keys to score other behaviours.

To add a key to a protocol, click the *Add item* button in the ribbon bar and select *New Key* from the menu which appears.

The New Key menu option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the key's name.
2. Specify key strokes for the key
3. Specify the key's type, i.e. how it should actually work
4. Specify whether the Ctrl key must be pressed for the key to work
5. Specify whether the behaviour represented by the key should be counted as an activity.

What next?

After completing these steps, you should consider whether you want to include any On/off inputs in this protocol.

See also:

- Adding elements to a protocol
- Editing a key
- Deleting a key

Key name

In brief

Enter a name for the key in the *Key name* field on the key's settings page. You must make an entry, but it can be anything you like.

Details

Key names should specify the behaviour that the key will be used to score. For example, 'Grooming' or 'Rearing'.

The name of the key will be included in the names of the key's results so, to avoid having very long result names, you should try to keep key names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying key strokes for the key

In brief

You should use the fields on the key's settings page to specify a unique key stroke for each piece of apparatus in your protocol. For example, if you have two plusmazes, then for a 'Grooming' key you must specify one key stroke for Plusmaze 1, perhaps 'G', and another for Plusmaze 2, perhaps 'L'. You can use the letter keys from 'A' - 'Z', the number keys from '0' - '9' and the numbers '0' - '9' on the numeric keypad.

*While testing, ANY-maze will **always** intercept the key strokes that you specify here. This means that you won't be able to type into other fields during a test - or if you do, the key strokes you have specified here won't work. To overcome this, you can choose to use the Ctrl key together with the specified key strokes; thus, for example, to indicate that a the animal is grooming you would press Ctrl+G, which would allow you to type the letter G into fields during the test.*

Details

Key strokes must all be unique, both between different apparatus and between different keys - ANY-maze will check this for you and complain if you try to redefine a key.

You can use any of the keys 'A' - 'Z', the number keys from '0' - '9' and the numbers '0' - '9' on the numeric keypad. ANY-maze doesn't differentiate between upper and lower case, so 'A' and 'a' are the same key; however, '2' on the main part of the keyboard *is* different to '2' on the numeric keypad. Therefore ,there are 46 different key strokes available (26 for the letters and 20 for the numbers).

When you are including more than one key in your protocol or you have more than one piece of apparatus, then you will naturally have to define more than one key stroke. Although it might be tempting to use *logical* key strokes such as 'G' for grooming and 'R' for rearing, you should consider how easy it will be to use the keys. In this example 'R' and 'G' are close together on a QWERTY keyboard, and choosing 'A' and 'L' might be easier.

Clearly, if you define more than one key in your protocol or you have more than one piece of apparatus, then during a test, you may need to press down multiple keys at the same time. Unfortunately, this can be problematic because computer keyboards are not designed to register lots of keys all pressed down together. For example, on many keyboards if you press down both the E and R keys and then press T, the computer won't register the key press. To help you resolve this problem, ANY-maze includes a *Multiple Key test* facility which you can use to find out which keys on your keyboard can be pressed down at the same time - see the Testing multiple keys topic for full details.

See also:

- Specifying how the key should actually work
- Specifying whether the Ctrl key must be pressed for the key to work
- Adding additional scoring to a performed test

Specifying a key's type

In brief

You should use the drop-down list on the key's settings page to specify a key's type - i.e. how the key should actually work. There are three options:

Simple	The activity is registered as starting when the key is pressed and as ending when the key is released.
Toggle	The activity is registered as starting the first time the key is pressed and as ending the second time the key is pressed.
Radio	*Radio* keys work like *Toggle* keys, except that only one *Radio* key can be active at the same time. This means that registering the start of a *Radio* key behaviour will automatically register the end of any other currently active *Radio* key behaviours.

Details

If you include any *Toggle* or *Radio* keys in a protocol, then you will probably find it useful if ANY-maze indicates during tests which keys it considers to be active (otherwise it's easy to lose track of whether a key is on or off). You can choose to have active keys indicated using the options on the What to display while testing element of the protocol.

Specifying that the Ctrl key must be pressed for a key to be registered

In brief

You can use the check box on the key's settings page to specify that a key should only be registered if both the key itself AND the 'Ctrl' key are pressed down together.

Details

When you add a key to a protocol, ANY-maze will automatically register key presses during a test as relating to the relevant behaviour - this is true no matter what you happen to be doing in the system. In other words, if the key 'G' is set to register *Grooming,* then whenever you press G on the keyboard ANY-maze will register a grooming bout.

This sounds sensible enough, but imagine that during a test, you want to add some notes to the Animal's details report - you type in '*The animal began the test....*' but when you press the 'g' in 'began' ANY-maze will register a grooming bout (and the letter 'g' won't appear).

Of course, if you're planning to score grooming using keys, you probably won't have time to type in any notes - you'll be too busy watching the animal. Nevertheless, there might be situations when this inability to use behaviour-recording keys becomes more problematic, and in this case you can simply specify that the Ctrl key must also be pressed in order for the behaviour to be registered.

In the earlier example, this would mean that typing in 'began' would work OK, because to register grooming you would have to type 'Ctrl+G'.

Specifying whether the behaviour represented by the key should be counted as an activity.

In brief

If you want ANY-maze to consider the behaviour which a key represents to be an activity, then check the box titled *This behaviour should be counted as an activity* shown on the key's settings page.

Details

In ANY-maze, an animal is defined as being active if it's mobile (assuming mobility is being detected) **or** it's performing some other behaviour which has been defined to be an activity.

For example, if an animal is stationary but 'Grooming', and 'Grooming' has been defined as an activity, then the animal will be considered to be active. If it stops grooming and remains stationary, then it will then be considered to be inactive.

If you have chosen NOT to detect mobility (see Detecting immobility), then activity will ONLY be defined by the keys which are defined as activities. In other words, if the animal is performing none of the 'activity' behaviours, then it will be considered to be inactive - even if it's moving about.

Editing a key

Introduction

You can edit absolutely everything related to a key at any time, whether before, during or after an experiment has been performed. However, you should take care when altering the key stroke - see details below.

Details

If you change the key stroke for a key, then ANY-maze will no longer recognise the key in tests you've already performed. This is because the system records the *actual key you press* during the test, so if you change the keystroke for a particular behaviour, the system will no longer recognise the key presses it has recorded as relating to that behaviour.

The system will warn you about this issue if you edit a key.

See also:

- Editing the elements of a protocol
- Setting up a key
- Deleting a key

Deleting a key

Introduction

You can delete a key at any time, whether before, during or after an experiment has been performed.

To delete a key, simply select it in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting keys.

See also:

- Deleting protocol elements
- Editing a key

Key measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a key:

- Number of presses
- Time pressed
- Latency to first press
- Latency to first release
- Distance travelled before 1st press
- Longest press
- Shortest press
- Average press duration
- Press frequency
- List of activation durations

Each measure is available for the apparatus as a whole (i.e. irrespective of where the behaviour occurred) and also for each defined zone. For example, if a protocol includes two zones called 'Left' and 'Right' and two keys have been defined, one for 'Grooming' and one for 'Rearing' then the following sets of measures will be available:

- All the measures listed above for grooming anywhere in the apparatus
- All the measures listed above for rearing anywhere in the apparatus
- All the measures listed above for grooming in the left zone
- All the measures listed above for rearing in the left zone
- All the measures listed above for grooming in the right zone
- All the measures listed above for rearing in the right zone

Number of presses

Description	Reports the number of times the key was pressed down, i.e. the number of times the animal started to exhibit the key's behaviour.

Calculation method	Counts the number of key presses.
Analysis in zones	Counts the number of times the key was pressed when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of times the key was pressed down during the time period.
Units	None
Notes	None

Time pressed

Description	Reports the total amount of time the key was pressed down, i.e. the total amount of time that the animal was exhibiting the key's behaviour.
Calculation method	Sums the amount of time for which the key was pressed.
Analysis in zones	Sums the amount of time for which the key was pressed in the zone. For a particular zone, it's possible for the *Time pressed* to be non-zero while the Number of presses is zero. This can occur if the animal enters the zone when the key is pressed. In this case, the time the key is pressed will be registered but the key press itself won't be.
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of time that the key was pressed during the period. For a particular time period, it's possible for the *Time pressed* during the period to be non-zero while the Number of presses for the period is zero. This can occur if the key is already pressed at the start of the period. In this case, the time the key is pressed will be registered but the key press itself won't be.
Units	Seconds
Notes	None

Latency to first press

Description	Reports the amount of time that elapsed in the test before the key was pressed for the first time, i.e. the amount of time that elapsed before the key's behaviour started for the first time.
Calculation method	The value of the test clock at the first key press.
Analysis in zones	The value of the test clock at the first key press that occurred within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds

Notes	None

Latency to first release

Description	Reports the amount of time that elapsed in the test before the key was released for the first time, i.e. the amount of time that elapsed before the key's behaviour ended for the first time.
Calculation method	The value of the test clock at the first key release.
Analysis in zones	The value of the test clock at the first key release that occurred within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Distance travelled before 1st press

Description	Reports the distance the animal had travelled in the apparatus up to the moment the key was first pressed.
Calculation method	The accumulated total distance travelled is noted at the time of the first key press.
Analysis in zones	The measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Metres
Notes	None

Longest press

Description	Reports the duration of the longest period for which the key was continuously pressed down, i.e. the longest period for which the animal continuously exhibited the key's behaviour.
Calculation method	The duration of each key press is calculated and the largest value is found.
Analysis in zones	The longest period for which the key was continuously pressed in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest period for which the key was continuously pressed during the period.
Units	Seconds
Notes	None

Shortest press

Description	Reports the duration of the shortest period for which the key was continuously pressed down, i.e. the shortest period for which the animal continuously exhibited the key's behaviour.
Calculation method	The duration of each key press is calculated and the smallest value is found.
Analysis in zones	The shortest period for which the key was continuously pressed in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest period for which the key was continuously pressed during the period.
Units	Seconds
Notes	None

Average press duration

Description	Reports the average duration for which the key was held down, i.e. the average duration of the episodes of the key's behaviour.
Calculation method	Calculated by dividing the Time pressed by the Number of presses.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Press frequency

Description	Reports the frequency with which the key was pressed, i.e. the frequency with which the animal exhibited the key's behaviour.
Calculation method	Calculated by dividing the Number of presses by the *Test duration*.
Analysis in zones	Calculated by dividing the Number of presses in the zone by the *Total time in the zone*.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Number of presses which occurred during the period divided by the period's duration.
Units	Hertz
Notes	None

List of activation durations

Description	Reports a comma-separated list of the duration of each activation of the key. For example, if key was activated three times, for 10s on the first occasion and for 20s on the second and third, then the list would show '10.0, 20.0, 20.0'.
Calculation method	The duration of each activation is calculated and added to the list. If the key is active at the end of the test, then the last duration in the list will be from the time the key was activated to the time of the test end.
Analysis in zones	The measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	All the durations in the list are reported in seconds.
Notes	When included on the Data page, this measure will show all the activation durations in a single cell. If the spreadsheet is saved in CSV format and then opened in (for example) Microsoft Excel, then the durations will be listed in individual cells. The length of the list is limited to 8192 characters, but usually at last 1,000 activations will be listed before the limit is reached.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures

- Event measures
- Virtual switch measures

Electrophysiology synchronisation

Introduction

It's common to track animals using ANY-maze while simultaneously recording neuronal activity using an electrophysiology system. At the end of such a test, you will usually want to merge the results from ANY-maze and the electrophysiology system in order to analyse neuronal activity when specific behaviours were occurring. The challenge here is aligning the time-base of the results from the two systems, and it is this that the Electrophysiology synchronisation feature in ANY-maze is designed to address.

To perform synchronisation you should select the *Electrophysiology synchronisation* element in the protocol list, it is the first item in the *Inputs and outputs* group.

Electrophysiology synchronisation generates TTL signals and is therefore only available if you are using a ***protocol mode*** *which supports inputs and outputs, such as the 'Video tracking mode with input/output'. If the protocol does not include an element for Electrophysiology synchronisation the most likely reason is that the protocol mode is not set correctly.*

- Test synchronisation
- Frame synchronisation
- Frame synchronisation options
- Position stored synchronisation
- Specifying the ports to use for the synchronisation signals
- Accessing the results of a test
- Timing accuracy for electrophysiology synchronisation

Test synchronisation

The simplest form of synchronisation is *Test synchronisation,* which activates a TTL output when the test starts and deactivates it at the test end.

Test results in ANY-maze are all based on 'Test time', which starts from zero at the moment the test begins. If the electrophysiology recording system is setup to record the changes in state of the synchronisation output from ANY-maze, then the time (in the electrophysiology recording system's time-base) that the output changes from inactive to active is ANY-maze's time zero. Thus, to determine when something recorded by ANY-maze occurred in ephys time, one simply adds the time at which the Test synchronisation output was activated to the ANY-maze time of the event. For example, if the animal moved into a zone at ANY-maze time 18.342 seconds, and the ephys system recorded the Test synchronisation signal activation at time 2min 12.723s then the zone change occurred at 2min 31.065s in 'ephys time'.

To setup Test synchronisation you need to check the appropriate option on the *Electrophysiology synchronisation* settings and then select the physical port that will output the signal.

An alternative (or addition) to the method of Test synchronisation described above, is for the electrophysiology system to output a TTL signal at a time of its choosing and for this signal to start the test in ANY-maze. Although this will work, there is a latency of between 4 to 8 milliseconds from the test control switch changing state and the test actually starting in ANY-maze. Therefore, even if the ephys system does trigger the test start, it's still a good idea to output the Test synchronisation signal as its latency (from the test starting in ANY-maze to the physical output changing state) is typically a little under 1ms.

Frame synchronisation

Frame synchronisation is only available if you are using a protocol mode *that supports video tracking.*

As you probably know, cameras (or any other video source) provide multiple images every second and these images are what create the video. The images are called *frames* and the number of images provided per second is called the camera's *frame rate*. The most common frame rate is 30 frames per second (fps), although you can alter the frame rate of most cameras from within ANY-maze, and some cameras can provide substantially higher frame rates, for example, 87 fps.

The frame rate is important, because ANY-maze only actually knows where the animal is / what it is doing, when a frame is captured by the camera. So for a camera that is capturing frames at 30fps, ANY-maze knows the animal's location at 33ms, 66ms, 99ms, etc.

Frame synchronisation generates a pulse on a TTL output whenever a new frame is captured by ANY-maze. This TTL pulse can be recorded by the electrophysiology system and allows you to associate recorded data to the frames. The pulse length is nominally 3 milliseconds, although this is not strictly enforced.

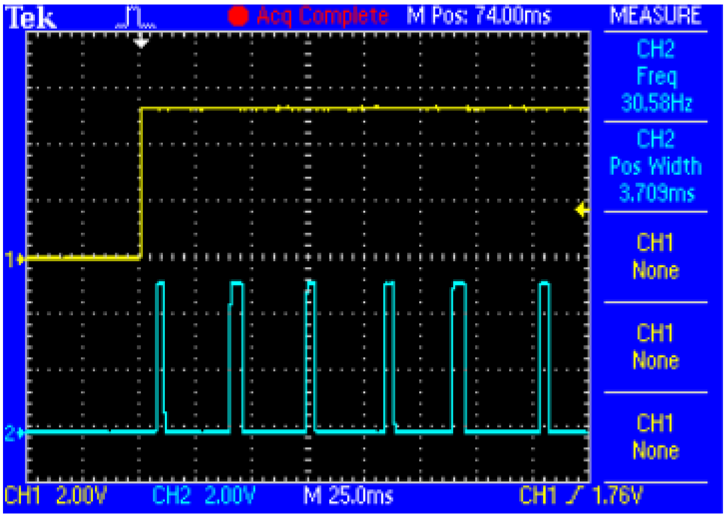


Figure 1. An oscilloscope trace showing a Test synchronisation signal as channel 1 (yellow) and a Frame synchronisation signal as channel 2 (blue).

There are a few things to be aware of when using Frame synchronisation:

- Although a camera may be providing 30 frame per second, this does not mean the first frame in the test will arrive at time 33ms, because the test start is not synchronised with frame capture. So, for example, the test might start 10ms after a frame was last captured, which would mean the first frame in the test would arrive at time 23ms, the second at time 56ms (i.e. 33ms later), etc. This can be seen in figure 1, where the first frame (first blue pulse) arrives a little after the test start (yellow trace going high).
- Although a camera will generate frames at a strict frequency, the software may not 'capture' them immediately. The path from the camera, through the physical connection to the computer (usually USB), into Window's video processing sub-system and finally into ANY-maze, is complex and ANY-maze won't necessarily receive frames at exactly 33ms intervals. Over time, the average inter-frame interval will indeed be 33ms, but the individual frame times may differ from this by as much as ±4ms (see figure 2). This can also be seen in figure 1 (above), where the oscilloscope reports channel 2 frequency as 30.58Hz, whereas the camera is generating the frames at exactly 30Hz.

Frame time	Inter frame interval (ms)
16.177	
16.209	32
16.244	35
16.276	32
16.310	34
16.342	32
16.377	35
16.409	32
16.444	35
16.476	32
16.511	35
Average inter frame interval	**33.4**

Figure 2. Some real frame times captured by ANY-maze. As can be seen, the inter-frame interval, which in theory is 33.333ms varies a few milliseconds either side of 33ms, although the average is correct.

Frame synchronisation options

When ANY-maze receives a frame it will analyse it to determine where the animal is and, to some extent, what it is doing. If the system fails to identify the animal in the frame then it simply won't store any information about the frame in the test results. So, for example, the results might have a location recorded for the animal at time 33ms (the first frame in the test), but no location recorded for the next frame (at time 66ms) and then another location recorded at time 99ms - so the results will appear to have a gap.

Furthermore, ANY-maze can be set to not record every position of the animal. For example, if you are testing an animal in a plusmaze, with a camera that provides 30fps, then ANY-maze will know where the animal is every 33ms. But animals on a plusmaze generally don't move very quickly, and in 33ms an animal might only move a few millimetres. Recording the animal's position, for example, 10 times per second (rather than 30 times) will still give good spatial resolution but will reduce the amount of data saved (and hence the size of the experiment file) by two thirds. In fact, except for situations where the animal moves very quickly, recording all 30 positions per second is usually unnecessary and so ANY-maze will by default, record at most 10 positions per second, although, of course, this can be adjusted.

When trying to synchronise with an electrophysiology system, the fact that ANY-maze doesn't, by default, record every position of the animal, and the fact that in some frames ANY-maze may not determine the animal's location, can make it complicated to associate data in the ANY-maze results, with the frames signals generated by 'Frame synchronisation'. To address this there are two Frame synchronisation options:

Record every position of the animal

This option means ANY-maze will include the animal's position in the results for every frame in which it determines the position. This is, in fact, identical to setting the Recording frequency to *Record every position of the animal*.

Record frames in which the animal was not tracked

This option will cause ANY-maze to include an entry in the test's results for frames in which the animal's location was not determined. Thus there will be an entry in the results for every frame captured from the camera, even if for some of them there is no x,y location reported - see figure 3.

Time	Centre position X	Centre position Y
0:00:16.177	275	213
0:00:16.209	276	218
0:00:16.244	#N/A	#N/A
0:00:16.276	281	237
0:00:16.310	282	238
0:00:16.342	283	241
0:00:16.377	#N/A	#N/A
0:00:16.409	#N/A	#N/A
0:00:16.444	#N/A	#N/A
0:00:16.476	270	205
0:00:16.511	275	204

Figure 3. In this test the camera was running at 30 frames per second. At time 16.244s ANY-maze captured a frame but failed to detect the animal, hence the x,y coordinates are shown as #N/A (not available). This occurred again at time 16.377s and continued for three frames.

Generally speaking you will want to use both these options when using Frame synchronisation.

A final point to be aware of, is that ANY-maze may not actually capture some frames. This can occur if it takes so long to analyse one frame that by the time ANY-maze is ready to capture the next frame, an intervening frame has been 'dropped'. For example, ANY-maze captures a frame at time 33ms and then takes 70ms to analyse the frame. When it is ready to capture another frame, at time 103ms, it will receive the frame timed at 99ms, so the frame at time 66ms will have been 'dropped'. This situation is rare, as ANY-maze usually only takes a few milliseconds to analyse a frame, however, if you are using very large frames, or you're using an older (slower) computer, or the computer is doing other things as well as running ANY-maze, or (most commonly) a combination of these, then ANY-maze may drop some frames. In this situation there will be 'missing' frames in the results, even if you have switched on both the options above.

Position stored synchronisation

Frame synchronisation is only available if you are using a protocol mode *that supports video tracking.*

Position synchronisation generates a pulse (the pulse length is nominally 3 milliseconds) on a TTL output whenever a position is stored in the experiment's results. This means that electrophysiology recording system will have a note of every frame which was stored. An important point to understand about this option is that the pulse is generated *after* ANY-maze has analysed the frame and determined the animal's location (if you think about it, you'll realise that this is inevitable) and NOT at the time the frame was captured. However, in the ANY-maze results, the position is time-stamped at the frame capture time (which is when the animal was actually at the specified position). This means that the ephys system may record a pulse at time 100ms, but in the ANY-maze results, there will be no position recorded at time 100ms, rather the position will be recorded a few milliseconds earlier, perhaps at time 96ms. If you want to address this, you can choose to use both Frame synchronisation and Position stored synchronisation, thus when a Position stored synchronisation pulse occurs you can look back for the immediately prior Frame synchronisation pulse and this should be timed at the same time as the position within the results.

Specifying the ports to use for the synchronisation signals

When you choose to use any of synchronisation options listed above, the protocol will automatically update to include items which you can use to specify the physical output ports for the relevant signals - see figure 4.

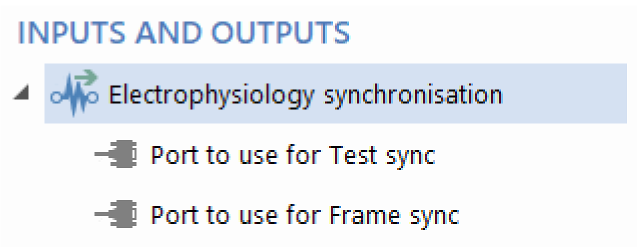


Figure 4. In this protocol the user has chosen to use Test and Frame synchronisation. The Electrophysiology synchronisation element therefore includes two sub-items, one to choose the port, or ports, to use for Test synchronisation, and the other to choose the port/ports for Frame synchronisation.

Selecting one of these items will cause the electrophysiology synchronisation ports page to be displayed, where you will need to select the physical output port to use for the relevant signal, for each of the apparatus in your experiment (remember, if you wish, you can setup multiple apparatus to run in parallel).

You can choose any TTL output port on any of the I/O devices that you have included in the protocol. If the ports list does not include any ports, or doesn't include the port you want to use, then it probably means you have not included the relevant I/O device in the protocol. Alternatively it might be that you have included the I/O device, but you have not configured the ports you want to use as TTL outputs. To resolve this you should alter the device's configuration - exactly how you do this is device dependent, so you should refer to the specific device's help topics.

Accessing the results of a test

Throughout this help topic I have been talking about synchronising the results of a test between ANY-maze and an electrophysiology system - but what results am I talking about?

In ANY-maze, when we refer to results, what we are usually talking about are the results of the ANY-maze measures such as *Total distance travelled* or *Time in the centre zone*. These are indeed the results of the test (and are what are reported on the Results page), but ANY-maze generates them by analysing the raw results that are recorded during the test, such as the animal's x,y coordinates in each frame. In fact in ANY-maze we usually call this the 'Test data' and it can be accessed on the Test data report - see figure 5.

Time	Centre position X	Centre position Y	Speed	In NW	In SW	In NE	In SE	In Island
0:00:00.000	#N/A	#N/A	#N/A	0	1	0	0	0
0:00:00.500	225	356	#N/A	0	1	0	0	0
0:00:00.840	224	359	#N/A	0	1	0	0	0
0:00:01.570	227	374	0.068m/s	0	1	0	0	0
0:00:01.840	231	374	0.068m/s	0	1	0	0	0
0:00:02.010	238	378	0.068m/s	0	1	0	0	0
0:00:02.260	250	383	0.141m/s	0	0	0	1	0
0:00:02.590	269	381	0.233m/s	0	0	0	1	0
0:00:02.760	279	378	0.233m/s	0	0	0	1	0
0:00:02.920	291	375	0.270m/s	0	0	0	1	0
0:00:03.260	315	365	0.309m/s	0	0	0	1	0
0:00:03.430	328	358	0.333m/s	0	0	0	1	0
0:00:03.590	342	348	0.430m/s	0	0	0	1	0
0:00:03.760	356	337	0.404m/s	0	0	0	1	0
0:00:03.930	366	325	0.357m/s	0	0	0	1	0
0:00:04.100	376	313	0.357m/s	0	0	0	1	0

Figure 5. An example of the Test data report, showing the animal's position, speed and whether or not it was in certain zones.

As can be seen in the figure, the Test data report is organised as a spreadsheet, and if you want to merge the ANY-maze data from the test with the electrophysiology data, you will usually want to export this spreadsheet and merge the data in another program, such as Excel. Full details about the Test data report, including how to copy or export it, can be found here.

In fact ANY-maze also includes two options for exporting all the test data for an entire experiment. These options are available on the File page (under Export) and are described in full here.

Timing accuracy for electrophysiology synchronisation

A common concern when synchronising an electrophysiology system with ANY-maze is the accuracy of the timing of the signals that ANY-maze generates.

The signals generated by ANY-maze will be always be a little delayed, because there is necessarily some latency from the moment that ANY-maze recognises that a signal needs to be generated to the moment that the physical hardware changes the state of the relevant output port. This delay depends on a number of factors, such as what else the computer is doing, what hardware is actually generating the signal, etc. but it will usually be no more than 3ms (and is typically around 1ms). Of course in the electrophysiology world this still seems like quite a long delay and is made worse by the fact that it's not uniform. However, the delay needs to be viewed in the context of what we're actually trying to do - which is synchronise data from a system which typically only has a sampling rate of 30sps (i.e. the frame rate of the camera). Sure, some cameras might provide 100 frame per second (i.e. 100sps) but in any case, a delay of a few milliseconds when synchronising to data that has a sampling time of 33ms (or perhaps just 10ms) is not very relevant.

In fact, you might look at this information and think that it would be a benefit to use a camera that captures 100 frames per second, because then you will be sampling the animal's position at 100sps which will be 'more accurate' than 30sps. However, it's worth considering how you'll be using the position data. For example, you might want to look at the electrophysiology results at the moment the animal moves onto the open arm of a plusmaze. If you use a 100fps camera then, yes, you will get told when the animal moves onto the open arm with a timing resolution of 10ms, but you need to step back from that for a moment and consider what 'moving onto the open arm' actually means. In ANY-maze a zone entry can be determined in a number of ways, but in the case of the plusmaze the most common method is to use the percentage of the animal's body that is in the zone, with a value of 80% typically being used for an open arm entry (this equates well with the traditional 4-paws-in-the-arm rule). But this is quite arbitrary and you could use 75% or 85% and if you did so, then the time when the animal enters the arm would almost certainly change by tens or hundreds of milliseconds. The important point here is that yes, at time (say) 9s you can see that you animal is clearly not in the open arm and at time (say) 12s you can see that it clearly is in the open arm, but the precise time when it entered the arm is hard to pin down and trying to do this to the nearest 10ms, or even 33ms, is really quite arbitrary. Realistically you can probably attribute the time of something like an arm entry to a period of 250ms at best. So my point is, that worrying too much about capturing lots of frames per second, or about having very precise timing for synchronisation signals, is all a bit pointless when the resolution of the data you are actually interested in, is in reality only 250ms.

Having said all that, there is a situation in ANY-maze when the timing of behaviour can be viewed as quite precise, and that's when the animal's behaviour is being detected by I/O devices. For example, you might be interested in the time when an animal pressed a lever. The hardware to detect the lever will typically generate a signal which ANY-maze will detect and store in the results within 5ms, so in these cases you would want the synchronisation between ANY-maze and the electrophysiology system to be as tight as possible - but with a typical value of 1ms, it is typically good enough for this task.

On/off inputs

As the name implies, *On/off inputs* are inputs which can have two states, either on or off. For example, push button switches and photobeams are both 'on/off' inputs. The *On/off inputs* element of the protocol is used to detect activation of these types of inputs.

For more information about on/off inputs refer to the topics below.

- An introduction to on/off inputs
- Setting up an on/off input
- Editing an on/off input
- Deleting an on/off input
- On/off input measures

An introduction to on/off inputs

Introduction

Although ANY-maze can use the video image of your apparatus to determine the location of an animal, there may be situations when you wish to use some type of electronic input to detect things which the analysis of the video image simply can't provide.

For example, your apparatus might include a lever which the animal can press to receive a reward (perhaps a food pellet). Clearly, video tracking will not be able to determine when the animal presses the lever, so to overcome this you could connect a switch to the lever and use the switch to detect the presses - this is exactly the kind of thing that ANY-maze *On/off inputs* are designed to do. Of course, in this example, you'd not only want to register the lever press, you'd also want it to cause a pellet to be dispensed. This could be done by using an output switch to control the pellet dispenser, and a procedure to instruct ANY-maze to dispense a pellet when the lever is pressed.

On/off inputs aren't limited to detecting lever presses - they can also be used to detect activation of other types of inputs. For example, a touch switch connected to the spout of a water bottle could be used to detect when the animal licks the spout, or a photobeam placed across a hole could be used to detect nose-pokes.

The full list of supported on/off inputs is:

- Physical switches, such as push button switches and microswitches
- Digital (TTL/CMOS) inputs
- Touch switches
- Photobeams
- Photobeam arrays

Connecting the device to ANY-maze

Clearly, if ANY-maze is going to sense the state of a switch, photobeam, etc. then the device will have to be physically connected to your computer. Connecting these types of inputs is actually quite simple, and is done using a special piece of equipment called an *Input/Output device* (usually abbreviated to 'I/O device') which acts as an interface between your PC and the actual switch, photobeam, or whatever.

You'll find full details about the I/O devices which can be used with ANY-maze in the I/O devices supported by ANY-maze topic - suffice to say that there are a range of devices available to suit most situations and budgets.

Using on/off inputs

Having connected an input - such as micro-switch - to your computer, you then need to add **two** elements to your protocol in order to use it: an I/O Device element, for the physical I/O device to which the switch is connected, and an *On/off input* for the switch itself.

In fact, most I/O devices support multiple inputs, so you only need to set up the I/O device element once to make all of its inputs available within the protocol. For example, the ANY-maze Interface device has 16 inputs, so you could connect 16 switches to it - but you would only need to add the device itself to the protocol once. The process of adding an I/O device to your protocol is described in detail in the Setting up an I/O device topic.

It doesn't matter what type of On/off input you want to include in your protocol (it can be any of those listed above). In all cases, you simply add an 'On/off input' element. Doing so will mean that ANY-maze can actually detect when the device is on or off, and the system will then be able to:

- Report a range of measures for the input, including such things as: the number of activations, the total time active, the activation frequency, etc. These measures will all be available both for the apparatus as a whole and also for each zone individually. See On/off input measures for full details.
- Use the on/off input in a procedure so as to perform some action when the input is activated. For example, you could set up a procedure which would detect when the animal activates a lever twice within a period of 5 seconds and perform some action because of this - perhaps to dispense a food pellet, activate a shocker or play a sound.

Using multiple physical inputs with a single 'On/off input' item

In some situations, you might find that you want to use more than one physical input to control a single logical 'On/off input' in ANY-maze. This is best explained using an example: Imagine you have a box divided into two parts with a pair of swing doors between them. The animal can push through the doors to get from one side to the other. As there are two doors, you connect two micro-switches, one to each door, to detect the door being opened. Of course, from your point of view it doesn't actually matter which switch is activated, so you'd include an 'On/off input' which would be active when either switch is activated.

This type of situation requires that multiple inputs are *logically ORed,* i.e. the overall input is active if Switch 1 is active OR Switch 2 is active.

ANY-maze allows you to use multiple inputs in this way - you simply have to specify which physical inputs you want to OR together.

In fact, ANY-maze will also allow you to AND inputs together. As you'd expect, this simply means that the input will only be considered active if ALL the selected inputs are active.

See also:

- An introduction to I/O devices
- An introduction to Procedures
- Setting up an On/off Input

Setting up an on/off input

Introduction

For a general introduction to on/off inputs, see An introduction to on/off inputs.

To add an on/off input to a protocol, click the *Add item* button in the ribbon bar and select *New input item > New on/off input* from the menu which appears.

The New input item *menu option is only included if the* protocol mode *includes input/output.*

The New on/off input *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the input's name.
2. Optionally specify an index value for the input.
3. Specify what triggers the input (i.e. makes it active).
4. Specify whether the physical port used by the input is always the same, or whether it depends on the location of a movable zone.
5. Specify the physical port (or ports) used by the input.

What next?

After completing these steps, you should consider whether you want to include any rotary encoders in the protocol.

See also:

- Adding elements to a protocol
- Editing an on/off input
- Deleting an on/off input

On/off input name

In brief

Enter a name for the on/off input in the *On/off input name* field on the On/off input's settings page. You must make an entry, but it can be anything you like.

Details

Input names should specify the thing that the input is being used to detect. For example, if the input comes from a photobeam positioned across a 'nose-poke' hole then a good name would be 'Nose poke', rather than something like 'Photobeam 1'.

The name of the input will be included in the names of the input's results so, to avoid having very long result names, you should try to keep input names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying an index value for an on/off input

In brief

You can optionally specify an index value for an input. This allows ANY-maze to analyse sequences of activations.

Details

Generally, you will not need to specify an index value for an input, but doing so is useful in two situations:

- When inputs are in some sort of logical order, giving them index values allows you to view the sequence of activations that occur during a test. For example, you might have a series of photobeams in a corridor; in this case if you give the beams increasing index values you would be able to see the animal's progress up the corridor, either by including a chart plotting the indices and/or by including them in the Test data report.
- When working with the RAPC apparatus, the index allows ANY-maze to analyse the sequence of door openings. In this case, you should give each input an increasing value from 1 to 12; where the doors in the chamber closest to the start box have values 1, 2 and 3; the doors in the next chamber have values 4, 5 and 6, and so on.

Specifying what actually triggers an input switch

In brief

You should use the buttons on the On/off input's settings page to specify what will trigger the input, i.e. what will make the input active.

Details

In general, you will want an input to be triggered (i.e. made active) when its physical input is ON. However, the input might normally be ON and you would like the input to be triggered when it goes OFF.

Obviously, there's a question here of when a physical input is ON and when it's OFF, and this definition depends on the how the physical I/O device has been configured.

As well as simply requiring a physical input to be ON or OFF, you can also require that it remains in this state for a certain period before the switch will be considered active. This can be useful if you don't want to count very short activations of an input as true activations, or if you want to apply a longer debounce interval to an input (by default, ANY-maze debounces all inputs using a 20ms debounce interval).

The duration value that you enter can use any of the following units: h (hours), min (minutes), s (seconds) or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of one and a half minutes could be entered as '1.5min' or as '1min 30s'. If you don't specify any units, then seconds will be assumed. You can enter any value from 1ms to 1 hour, or you can enter 0, which is useful if you don't want the switch to be debounced at all.

If you are using multiple physical inputs, then the settings you make will apply to all the inputs individually. For example, if you specify that the input is triggered by the input line switching OFF and you have chosen to AND 4 physical inputs, then the switch will only be considered active when all 4 inputs lines are all OFF.

Specifying whether the physical port used by an on/off input is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port used by an input will always be the same - it's simply the port on the interface device to which the switch, photobeam, etc. actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Y-maze, you may have a lever (connected to a switch) mounted at the end of each arm of the maze. Now you want to create an element called 'Goal arm lever' (let's imagine that the animal has to press the lever to achieve the goal of the test), but you have set the 'Goal arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests. Thus the 'Goal arm lever' will be the lever in whichever arm happens to be the goal arm in a particular test, which necessarily means it won't always be the same physical lever (and therefore 'Goal arm lever' won't always be connected to the same physical port).

Details

If you do have an on/off input whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the *On/off input* element's page of the protocol.

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing an on/off input

Introduction

You can edit absolutely everything related to an on/off input at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up an on/off input
- Deleting an on/off input

Deleting an on/off input

Introduction

You can delete an on/off input at any time, whether before, during or after an experiment has been performed.

To delete an input, simply select it in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting on/off inputs.

See also:

- Deleting protocol elements
- Setting up an on/off input
- Editing an on/off input

On/off input measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for an on/off input:

- Number of activations
- Time active
- Latency to first activation
- Latency to first deactivation
- Longest activation
- Shortest activation
- Average activation duration
- Frequency of activations

Each measure is available for the apparatus as whole (i.e. irrespective of where the animal was when the input activation occurred) and also for each defined zone.

Number of activations

Description	Reports the number of times the input was activated.
Calculation method	Counts the number of activations.
Analysis in zones	Counts the number of activations when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of activations during the time period.
Units	None
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Time active

Description	Reports the total amount of time that the input was active.
Calculation method	Sums the amount of time for which the input was active.

Analysis in zones	Sums the amount of time for which the input was active while the animal was in the zone. For a particular zone, it's possible for the *Time active* to be non-zero while the Number of activations is zero. This can occur if the animal enters the zone when the input is active. In this case, the time the input is active will be registered but the activation itself won't be.
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of time that the input was active during the period. For a particular time period, it's possible for the *Time active* during the period to be non-zero while the Number of activations for the period is zero. This can occur if the input is already active at the start of the period. In this case, the time the input is active will be registered but the input activation itself won't be.
Units	Seconds
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Latency to first activation

Description	Reports the amount of time that elapsed in the test before the input was activated for the first time.
Calculation method	The value of the test clock at the first activation of the input.
Analysis in zones	The value of the test clock at the first activation of the input that occurred while the animal was within the zone.
Analysis across time	This measure can be analysed across time. The result for a time segment or time period is the latency to the first activation within the time segment or period. If the input is not activated within the time segment or period then the result is #N/A (Not available) unless the analysis option to *Use the test duration as the latency for events which don't occur* is set, in which case the result will be the duration of the time segment or period. Note that an input which is already active at the start of a time segment or time period will NOT be reported as having a latency of zero; this measure reports the latency to the first activation DURING the time segment or period.
Units	Seconds
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Latency to first deactivation

Description	Reports the amount of time that elapsed in the test before the input was

	deactivated for the first time.
Calculation method	The value of the test clock at the first input deactivation.
Analysis in zones	The value of the test clock at the first input deactivation that occurred while the animal was within the zone.
Analysis across time	This measure can be analysed across time. The result for a time segment or time period is the latency to the first deactivation within the time segment or period. If the input is not deactivated within the time segment or period then the result is #N/A (Not available) unless the analysis option to *Use the test duration as the latency for events which don't occur* is set, in which case the result will be the duration of the time segment or period.
Units	Seconds
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Longest activation

Description	Reports the duration of the longest period for which the input was continuously active.
Calculation method	The duration of each input activation is calculated and the largest value is found.
Analysis in zones	The longest period for which the input was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest period for which the input was continuously active during the period.
Units	Seconds
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Shortest activation

Description	Reports the duration of the shortest period for which the input was continuously active.
Calculation method	The duration of each input activation is calculated and the smallest value is found.
Analysis in zones	The shortest period for which the input was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest period for which the input was continuously active during the period.

Units	Seconds
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Average activation duration

Description	Reports the average duration for which the input was active.
Calculation method	Calculated by dividing the Time active by the Number of activations.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

Frequency of activations

Description	Reports the frequency with which the input was activated.
Calculation method	Calculated by dividing the Number of activations by the *Test duration*.
Analysis in zones	Calculated by dividing the Number of activations in the zone by the *Total time in the zone*.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Number of activations which occurred during the period divided by the period's duration.
Units	Hertz
Notes	*Activation* is defined when you create the input; see Setting up an on/off input.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- Signal measures
- Sensor measures

- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Pellet dispenser measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Selecting I/O ports

Introduction

Having included an I/O element in a protocol, you will need to specify the actual physical *port* which the device is connected to. For example, this might be an *ANY-maze interface General Purpose I/O port.*

Choosing ports

Every I/O element in a protocol has one or more *Port to use* sub-element. In most cases, there will just be a single sub-element; however, if you have specified that the ports the element uses can vary depending on the location of a movable zone, then there will be one sub-element for each position that the zone can adopt.

The *Port to use* page itself consists of one drop-down list for each apparatus item defined in the protocol. Thus, if, for example, you have defined four open fields (numbered 1 to 4) and each one has a light over it which can be switched on and off, you will have one *output switch* element (probably called 'Light') and this will have a *Port to use* sub-element showing four drop-down lists; one where you can choose the physical port that the light for Open field 1 connects to; one where you can choose the physical port that the light for Open field 2 connects to; etc.

The default entry for a port is '- None -' (because ANY-maze has no idea which port the device is connected to); if you leave the port set to '- None -' then the input or output whose port you are setting up simply won't work. This isn't *wrong,* but it's not of any practical use.

What ports are listed?

The actual drop-down list of ports is filled in with one entry for each port of the relevant type provided by each of the I/O Devices you have included in the protocol and which are *currently connected* to your computer (so the ports of I/O devices that are not connected will NOT be listed, even if you have included the device in the protocol).

Using multiple ports

Not all port types support this option.

For some types of input and output, the port's list includes an entry for '- Multiple ports -'. This allows you to choose more than one physical port for the input or output element to use. For example, in a plusmaze you may have placed a lamp over each of the two closed arms, and you may have created a *switch* element called 'Closed arm lights' to control them. Now you need to specify which physical ports the lights are connected to - but you've connected one of them to 'Output 1' of your AMi device and the other one to 'Output 2'. So, in this case, you would specify that the element uses multiple ports and then select both 'Output 1' and 'Output 2' in the window that appears. Thus, when you *activate* the 'Closed arm lights' element, both ports will be activated and both lamps will switch on.

When you choose a port on this page, ANY-maze will automatically check that the port is not already being used by some other element. For example, if you specified that the 'Lamp' element was connected to 'AMi output 1' and then you tried to specify that the 'Shocker' element was also connected to this port, the software would complain, because the port is already being used.

Situations where one port is used by more than one element

In fact, there are circumstances where the same port *could* be used by more than one element. In the first case, you may have specified that the port that is used for a certain element depends on the location of a specific zone. For example, imagine a Y maze which has lamps over the three arms. These lamps connect to 'Output 1', 'Output 2' and 'Output 3' on your AMi device. Now you want to create an element called 'Start arm lamp' which will control the lamp over the arm where the animal will start the test. If the animal always starts in the same arm, then this would be very easy - you'd just look to see which lamp is over that arm, see which output it connects to (let's say 'Output 2'), and then specify that the port for the element is that port. But what if the start arm varies - for example, during the first trial it's arm 'A', but for the next trial it's arm 'B'? In this case, you can't look to see which port the lamp over the arm connects to, because you don't know which the 'start' arm will be. This problem is resolved by specifying that the ports the element uses depend on the location of the start arm. This means that ANY-maze will, in this example, provide three 'Port' sub-elements, one for each of the three possible positions of the start arm. These elements will be labelled 'Port to use when Start arm is in position A', 'Port to use when Start arm is in position B', etc. So then you *can* define the port. BUT, let's now imagine that you designate another arm in your Y-maze as the 'goal' arm, and you want to create an I/O element called 'Goal arm lamp' (which might turn the goal arm lamp on after a certain duration). Like for the 'start' arm, this 'goal' arm can vary its position between different tests - that's no problem; you simply specify that the port used for the 'Goal arm lamp' depends on the location of the goal arm. And, like for the 'start arm', you'll then be given three port sub-elements labelled 'Port to use when the goal arm is in position A', 'Port to use when the goal arm is in position B', etc. and you'll select the appropriate ports. But hang on a minute - the port to use for the goal arm light when the goal arm is in position A will be the same as the port to use for the start arm light when the 'START arm' is in position A (because there's only one lamp over the arm). This seems to mean that the same port will be used by two elements at the same time, but if you think about it, you'll realise that this would only happen if the goal arm and the start arm are both in position A <u>in the same test</u>, which one assumes they wouldn't be.

So, the upshot of all this is that you *can* use the same port in more than one element when the port is shared between elements whose ports depend on the positions of a variable position zone. It's then up to you to satisfy yourself that elements won't try to use the same port at the same time. ANY-maze will advise you should you get in to this situation, and ask you to confirm your selection.

Output switch aliasing

In fact, there is another situation in which the same port can be used by more than one element, and this creates so-called 'Port aliases' - but this ONLY applies to 'switch' outputs.

Essentially 'port aliases' allow you to give two 'uses' to the same port. For example, imagine in our Y-maze example, above, that in an earlier stage of our experiment, we simply wanted all the lamps to be switched on. So far, we have defined a 'Start arm light' and a 'Goal arm light', both of which use different ports depending on the position of different zones. So when we want all the lamps switched on, we *could* turn on the 'Start arm light' and the 'Goal arm light', but this would still leave one arm with the light off, and which arm that is would be hard to determine, as it would depend on the locations of both the start and goal zones. It would be much easier to create a new 'switch' element and call it 'All lights'; we'd then specify that this element's ports are always the same, and that they are 'Output 1', 'Output 2' and 'Output 3'. Now, activating this element would switch all the lights on. Of course, this element would be using the same ports as the 'Start arm light' and the 'Goal arm light', but we'd be saying that this is just an 'alias' for the ports; in other words, another (easier) way of talking about the same ports.

Of course, if we do alias ports in this way then we could create apparent conflicts in what the port is supposed to do; for example, the 'Start arm light' element could be active while the 'All lights' element is inactive - should the light then be on or off? ANY-maze uses a simple rule about this; if ANY alias element specifies that a port should be active, then it is active (for the technically inclined, the state of the elements is logically ORed).

Multiple inputs window

In brief

Use this window to choose which physical inputs will be used by an on/off input, and to specify how the inputs should be combined - either ANDed or ORed.

For a description of why you might want to use multiple inputs, see the introduction to on/off inputs.

Details

You can specify that any number of physical inputs should be combined to create a single ANY-maze 'On/off input'. You can even combine inputs from different physical devices.

You can't choose physical inputs which are already being used by another on/off input (or for other apparatus), but ANY-maze will warn you if you try to do this.

Having chosen the physical inputs you wish to use, you should then specify how they should be combined: either logically ANDed or logically ORed. ANDing inputs means that they must *all* be active for the overall input to be considered active; ORing inputs means that if *any of them* are active, the overall input will be considered active. Note that what constitutes 'active' in this sense depends on the *trigger* setting made in the on/off input's settings page.

Multiple outputs window

In brief

Use this window to choose which physical ports will be controlled by an *Output switch* protocol element.

Details

You can specify that any number of physical ports should be controlled by a single ANY-maze output switch. You can even use ports on different physical devices.

You can't choose ports which are already being used by another output switch (or for other apparatus), but ANY-maze will warn you if you try to do this.

Rotary encoders

Rotary encoders are devices which detect rotations around an axis, and they can be used to determine such things as rotations of a running-wheel or rotations of the animal (by tethering it to the encoder).

For more information about rotary encoders in ANY-maze, refer to the topics below.

- An introduction to rotary encoders
- Setting up a rotary encoder
- Editing a rotary encoder
- Deleting a rotary encoder
- Rotary encoder measures

An introduction to rotary encoders

Introduction

As the name implies, a rotary encoder is a device that senses rotations - specifically, rotations of an axle. These can be useful in at least two ways: to detect the rotations of a running wheel, and to detect rotations of the animal (by tethering the animal to the encoder's axle).

In fact, ANY-maze video tracking can detect rotations of the animal, so it might seem that using an encoder to do this is unnecessary - and in some respects it is. But if the only behaviour you are interested in is rotations, then using an encoder can provide a cheaper and easier alternative than using a camera - this is particularly true if you want to automate a lot of cages for simultaneous testing.

Incidentally, an encoder can sense both the direction and extent of rotations - thus it's possible to detect both clockwise and anti-clockwise rotations as well as partial rotations.

Physically connecting a rotary encoder to ANY-maze

Rotary encoders are supported by the ANY-maze interface (AMi) and full details of the types of encoders that can be used and how they connect to AMi can be found here.

See also:

- The ANY-maze interface
- Setting up a rotary encoder
- Rotary encoder measures

Setting up a rotary encoder

Introduction

For a general introduction to rotary encoders, see An introduction to rotary encoders.

To add a rotary encoder to a protocol, click the *Add item* button in the ribbon bar and select *New input item* > *New rotary encoder* from the menu which appears.

The New input item *menu option is only included if the* protocol mode *includes input/output.*

The New rotary encoder *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the rotary encoder's name.
2. Specify whether ANY-maze should try to detect the encoder's presence, and what it should do if the encoder is not present
3. Specify whether the physical port used by the rotary encoder is always the same, or whether it depends on the location of a movable zone.
4. Specify the physical port used by the encoder.

What next?

After completing these steps, you should consider whether you want to include any signal inputs in the protocol.

See also:

- Adding elements to a protocol
- Editing a rotary encoder
- Deleting a rotary encoder

Rotary encoder name

In brief

Enter a name for the rotary encoder in the *Rotary encoder name* field on the rotary encoder's settings page. You must make an entry, but it can be anything you like.

Details

Rotary encoder names should specify the thing that the encoder will be sensing. For example, 'Running wheel' or 'Animal'.

The name you use will be included in the names of the encoder's results so, to avoid having very long result names, you should try to keep rotary encoder names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying whether ANY-maze should detect the presence of a rotary encoder

In brief

ANY-maze is able to detect the presence of some rotary encoders. In the cases where it can do this, you can specify whether a test should be allowed to start if the encoder is missing.

Details

Some rotary encoders can be removed from your test apparatus - for example, a running wheel might be removable so it can be washed. Clearly, this means that there's a chance that you might forget to put it back before you started a test; what would happen then? The answer is probably that the test would be wasted.

However, for some encoders, ANY-maze can detect whether the encoder, or the thing that's turning (for example, the wheel) is physically present, so it would make sense if ANY-maze could warn you that the wheel is missing and prevent you from starting the test until it has been correctly installed. That's what this setting allows you to do.

There are three options:

- *Don't try to detect the presence of this encoder*
- *If encoder isn't present, ignore it in tests*
- *If encoder isn't present, don't allow tests to start*

The first option is the default - in this case, ANY-maze won't care whether the encoder is present or not. If it is missing, then it will either have no results or the results will just be zero. This is also the option that is automatically used (irrespective of the option you choose) for encoders which don't support presence detection.

The second option will cause ANY-maze to report the encoder's results as undefined in any tests where the encoder is not present. Other than this, the tests will be performed normally.

The third option will prevent a test from being shown as *Ready...* until the encoder is installed correctly. Instead of the *Ready...* status, ANY-maze will report *Missing I/O...*, and the status shown on the tests page will tell you what the actual problem is, for example *Running wheel not present.*

Specifying whether the physical port used by a rotary encoder is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port used by a rotary encoder will always be the same - it's simply the port on the interface device to which the sensor actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Place preference box, you might have two running wheels, one in either side of the box. Now you decide to associate one side of the box with a drug, but you choose to balance this between animals, so for some animals the left side is the drug-associated side and for other animals it's the right side.

If you then wanted to see the distance the animal ran in the wheel in the 'drug-associated side', you would have a problem, because for some animals it would be the data from the encoder on the left side, and for others it would be the data from the encoder on the right side. And, as these two devices would be connected to two different physical ports, the physical port would not always be the same.

Details

If you have a rotary encoder whose physical port depends on the location of a movable zone, like in the Place preference box example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the rotary encoder's *settings page*. (Note that this drop-down list is only included if the protocol includes at least one movable zone).

In the case of the Place preference box example, you would specify that *The port varies depending on the location of the drug-associated zone.*

When you do this, the rotary encoder element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the example there would be one sub-element for 'Port to use when drug-associated zone is in left position' and a second one for 'Port to use when drug-associated zone is in right position'. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a rotary encoder

Introduction

You can edit absolutely everything related to a rotary encoder at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up an rotary encoder
- Deleting a rotary encoder

Deleting a rotary encoder

Introduction

You can delete a rotary encoder at any time, whether before, during or after an experiment has been performed.

To delete a rotary encoder, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting rotary encoders.

See also:

- Deleting protocol elements
- Setting up an rotary encoder
- Editing a rotary encoder

Rotary encoder measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a rotary encoder. Note that each measure is available for the apparatus as whole, and also for each defined zone.

- Number of rotations
- Time turning
- Distance
- Number of clockwise rotations
- Number of anti-clockwise rotations
- Number of reversals
- Number of half rotations
- Number of quarter rotations
- Degrees of clockwise rotation
- Degrees of anti-clockwise rotation
- Maximum RPM
- Minimum RPM
- Average RPM
- Average RPM while turning

Number of rotations

Description	Reports the number of complete rotations of the encoder.
Calculation method	Counts a rotation when an unbroken sequence of 'number of pulses per rotation' same direction pulses is received.
Analysis in zones	Counts a rotation when an unbroken sequence of 'number of pulses per rotation' same direction pulses is received and the animal is in the zone for the entire sequence.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of rotations that ENDED during the time period.
Units	None

Notes	None

Time turning

Description	Reports the time for which the encoder was actually turning.
Calculation method	Reports the time for which the instantaneous rotational velocity (IRV) was not zero. The IRV is calculated as follows: Starts with a single pulse of a certain direction. Then counts the number of consecutive same direction pulses until at least 200 milliseconds have elapsed. Uses the elapsed time and the number of pulses detected during it to calculate the rotational velocity. This value is averaged (using a moving average) over ten values, and this is the instantaneous rotational velocity.
Analysis in zones	The time the encoder was turning while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the time that the encoder was turning during the time period.
Units	Seconds
Notes	None

Distance (only applies to running wheels)

Description	Reports the distance the animal 'travelled'.
Calculation method	The number of rotations multiplied by the circumference of the wheel.
Analysis in zones	The distance the animal 'travelled' in the wheel while it was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of rotations that occurred during the time period multiplied by the circumference of the wheel.
Units	Metres
Notes	This result is only available if the device which includes the rotary encoder reports a circumference - this feature is only supported by some running wheels.

Number of clockwise rotations

Description	Reports the number of complete clockwise rotations of the encoder.
Calculation method	Counts a rotation when an unbroken sequence of 'number of pulses per rotation' clockwise pulses is received.

Analysis in zones	Counts a rotation when an unbroken sequence of 'number of pulses per rotation' clockwise pulses is received and the animal is in the zone for the entire sequence.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of clockwise rotations that ENDED during the time period.
Units	None
Notes	None

Number of anti-clockwise rotations

Description	Reports the number of complete anti-clockwise rotations of the encoder.
Calculation method	Counts a rotation when an unbroken sequence of 'number of pulses per rotation' anti-clockwise pulses is received.
Analysis in zones	Counts a rotation when an unbroken sequence of 'number of pulses per rotation' anti-clockwise pulses is received and the animal is in the zone for the entire sequence.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of anti-clockwise rotations that ENDED during the time period.
Units	None
Notes	None

Number of reversals

Description	Reports the number of times the direction of the encoder changed.
Calculation method	Counts the number of times a clockwise pulse was followed by an anti-clockwise pulse and vice versa.
Analysis in zones	Counts the number of times a clockwise pulse was followed by an anti-clockwise pulse and vice versa when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of reversals that occurred during the time period.
Units	None
Notes	None

Number of half rotations

Description	Reports the number of half rotations of the encoder.
Calculation method	Counts a half rotation when an unbroken sequence of 'number of pulses per rotation' divided by two, same direction pulses is received.
Analysis in zones	Counts a half rotation when an unbroken sequence of 'number of pulses per rotation' divided by two, same direction pulses is received and the animal is in the zone for the entire sequence.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of half rotations that ENDED during the time period.
Units	None
Notes	If the number of pulses per rotation is not exactly divisible by 2, then the result is rounded down. This can yield some inaccuracy. Consider when the number of pulses per rotation is 5, then the number of pulses per half rotation will be considered to be 5/2 = 2.5, which when rounded down = 2. So after 2 same direction pulses, a half rotation will be counted; thus after 2 complete rotations 5 half rotations will be counted, when the correct value is 4. For this reason, it is not recommended to use encoders with a number of pulses per rotation (PPR) that is not divisible by 2. Fortunately, many encoders use binary powers for their PPR, for example 16, 32, 64, etc.

Number of quarter rotations

Description	Reports the number of quarter rotations of the encoder.
Calculation method	Counts a quarter rotation when an unbroken sequence of 'number of pulses per rotation' divided by four, same direction pulses is received.
Analysis in zones	Counts a quarter rotation when an unbroken sequence of 'number of pulses per rotation' divided by four, same direction pulses is received and the animal is in the zone for the entire sequence.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of quarter rotations that ENDED during the time period.
Units	None
Notes	If the number of pulses per rotation is not exactly divisible by 4 then the result is rounded down. This can yield some inaccuracy. Consider when the number of pulses per rotation is 5, then the number of pulses per quarter rotation will be considered to be 5/4 = 1.25, which when rounded down = 1. So after 1 pulse, a quarter rotation will be counted; thus after 4 complete rotations 20 quarter rotations will be counted, when the correct value is 16. For this reason, it is not recommended to use encoders with a number of pulses per rotation that is not divisible by 4. Fortunately, many encoders use binary powers for

their Pulses Per Rotation, for example 16, 32, 64, etc.

Degrees of clockwise rotation

Description	Reports the number of degrees of clockwise rotation of the encoder.
Calculation method	For each clockwise pulse of the encoder, adds '360 / Number of Pulses Per Rotation' to the result.
Analysis in zones	The number of degrees of clockwise rotation of the encoder while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is updated for each clockwise pulse that occurred during the time period.
Units	Degrees
Notes	This result is useful because it provides fine resolution (compared to number of rotations), while being normalised for all encoders (unlike number of pulses).

Degrees of anti-clockwise rotation

Description	Reports the number of degrees of anti-clockwise rotation of the encoder.
Calculation method	For each anti-clockwise pulse of the encoder, adds '360 / Number of Pulses Per Rotation' to the result.
Analysis in zones	The number of degrees of anti-clockwise rotation of the encoder while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is updated for each anti-clockwise pulse that occurred during the time period.
Units	Degrees
Notes	This result is useful because it provides fine resolution (compared to number of rotations), while being normalised for all encoders (unlike number of pulses).

Maximum RPM

Description	Reports the maximum rotational velocity of the encoder in units of revolutions per minute.
Calculation method	Starts with a single pulse of a certain direction, then counts the number of consecutive same-direction pulses until at least 200 milliseconds have elapsed. Uses the elapsed time and the number of pulses detected during it to calculate the rotational velocity. This value is averaged (using a moving average) over

ten values, and this is the instantaneous rotational velocity. The highest instantaneous rotational velocity is the maximum RPM.

Analysis in zones	The highest instantaneous rotational velocity while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the maximum value determined during the time period.
Units	Revolutions per minute
Notes	A rotary encoder has an inherent maximum rotational velocity, above which the encoder will not be read accurately. This value is reported on the I/O page when an encoder is selected. In fact, strictly speaking, this is a limitation of the interface that reads the encoder and not of the encoder itself. For example, when reading a 32 PPR encoder with the ANY-maze interface, the maximum RPM is 416, which is approximately 7 revolutions per second (enough for most likely situations in behavioural research). Note that using an encoder with half as many pulses per rotation will double this value.

Minimum RPM

Description	Reports the minimum rotational velocity of the encoder in units of revolutions per minute.
Calculation method	Starts with a single pulse of a certain direction. Then counts the number of consecutive same direction pulses until at least 200 milliseconds have elapsed. Uses the elapsed time and the number of pulses detected during it to calculate the rotational velocity. This value is averaged (using a moving average) over ten values, and this is the instantaneous rotational velocity. The lowest instantaneous rotational velocity is the minimum RPM.
Analysis in zones	The lowest instantaneous rotational velocity while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the minimum value determined during the time period.
Units	Revolutions per minute
Notes	If the encoder stops turning, this value will be zero.

Average RPM

Description	Reports the average rotational velocity of the encoder in units of revolutions per minute.
Calculation method	Starts with a single pulse of a certain direction. Then counts the number of consecutive same direction pulses until at least 200 milliseconds have elapsed. Uses the elapsed time and the number of pulses detected during it to calculate

the rotational velocity. This value is averaged (using a moving average) over ten values, and this is the instantaneous rotational velocity. The instantaneous rotational velocity is averaged throughout the test to yield this measure (see notes).

Analysis in zones	The average of the instantaneous rotational velocity values reported while the animal was in the zone (see notes).
Analysis across time	This measure can be analysed across time. For any time period, the result is the average of the instantaneous rotational velocity values reported during the time period (see notes).
Units	Revolutions per minute
Notes	The averaging of the instantaneous rotational velocity (IRV) values uses a time-based averaging technique, such that the average is the sum of each IRV value multiplied by the time between two consecutive values divided by the sum of the time between all values. This is required because the IRV values are not reported at a fixed frequency.

Average RPM while turning

Description	Reports the average rotational velocity of the encoder for the time when it was turning, in units of revolutions per minute.
Calculation method	Starts with a single pulse of a certain direction. Then counts the number of consecutive same direction pulses until at least 200 milliseconds have elapsed. Uses the elapsed time and the number of pulses detected during it to calculate the rotational velocity. This value is averaged (using a moving average) over ten values, and this is the instantaneous rotational velocity (IRV). All non-zero IRV values are averaged to yield this measure (see notes).
Analysis in zones	The average of the non-zero instantaneous rotational velocity values reported while the animal was in the zone (see notes).
Analysis across time	This measure can be analysed across time. For any time period, the result is the average of the non-zero instantaneous rotational velocity values reported during the time period (see notes).
Units	Revolutions per minute
Notes	The averaging of the instantaneous rotational velocity (IRV) values uses a time-based averaging technique, such that the average is the sum of each non-zero IRV value multiplied by the time between two consecutive non-zero values divided by the sum of the time between all such values. This is required because the IRV values are not reported at a fixed frequency.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Movement detector measures
- Output switch measures
- Pellet dispenser measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Signals

Signals are inputs which have a continuously variable value - for example, the voltage generated by a blood pressure monitoring device would be a signal input.

For more information about signal inputs, refer to the topics below.

- An introduction to signals
- Setting up a signal
- Setting up a filtered signal
- Editing a signal or filtered signal
- Deleting a signal or filtered signal
- Signal measures

ANY-maze help topic H0213

An introduction to signals

Introduction

Signals are inputs which have a continuously variable value - this will typically be a voltage that's being output from some type of sensing device. For example, a blood pressure monitoring device might output a voltage proportional to pressure, or a gas analysis device might output a voltage proportional to the level of CO_2 at its sensor.

To get a signal into ANY-maze, you need to use an I/O device which can convert the analogue voltage into a digital value that the software can process - this could be, for example, an AMi-2 Analogue interface, which includes four analogue inputs for just this purpose.

Converting signal inputs to the real-world value they represent

As mentioned above, a signal input will typically be a voltage, which means that ANY-maze will understand its value as being a certain number of volts. For example, a blood pressure monitoring device might be outputting a value of 0.5V, which might change over time to a value of 0.72V; but this isn't really what we want to know - we need to know blood pressure, probably in units of mmHg.

The voltage output by sensing devices, such as our hypothetical blood pressure monitor, usually has a linear relationship to the real-world value being sensed. For example, a voltage of 0V might mean a pressure of 0mmHg and a voltage of 1V might mean a pressure of 250mmHg. Knowing this we could deduce that a voltage of 0.5V would equate to a pressure of 125mmHg. Clearly then, it would be desirable to have ANY-maze report the data from the blood pressure signal in mmHg by having it perform this conversion for us; and that's what it can do. You just have to specify two values, such as 0V and 1V with their equivalent real-world values, which would be 0mmHg and 250mmHg in our example. The only requirement for ANY-maze to perform this type of conversion is that the relationship is **linear.**

Note that you don't have to convert the signal's value - if you choose not to, then the signal's data will just be reported in volts.

Reporting data in relation to a baseline

In our blood pressure monitoring example, we have seen how the voltage that's output by the device can be input into ANY-maze and then converted into mmHg, but what information about blood pressure would ANY-maze report? Well, the answer is 'quite a lot' - for example, it could tell you the maximum, minimum and average blood pressure during the test; it could report on blood pressure across the duration of the test (using the Analysis across time functions); it could report blood pressure when the animal was in different zones; etc. (The full list can be found here). This is all quite useful, but the fact that you're measuring blood pressure suggests that you're interested in how it *changes,* and you're likely to want to look at these changes in relation to a baseline - which would probably be the animal's blood pressure during a period at the start of the test (perhaps before you do something, like giving it access to some new area of the apparatus).

ANY-maze allows you to optionally define a baseline for a signal as being: the signal's value until either a set duration has elapsed, or until something occurs in the test - a door open, a sound starts playing, the animal moves into a certain zone, etc.

Having specified what ends the baseline period, you then need to specify what constitutes a 'deviation from the baseline' - this could be a change of:

- A certain percentage
- A fixed number of mmHg (or whatever the real-world units are)
- A fixed number of volts
- A certain number of standard deviations of the values captured during the baseline period.

Whatever basis you use, ANY-maze will determine when the signal deviates from the baseline (and when it returns to it) and will then report more information based on this - including such things as: latency to first deviation above the baseline, latency to return to the baseline, integral of deviations above the baseline, etc. - the full list is here (and remember if what you want isn't in the list, we'll gladly add more measures for free - you just have to ask).

Filtering signals

In some cases, the data that a sensing device outputs might be a mix of different information. A classic example of this is the data captured by an EEG, which is a mix of different waves - Delta, Theta, Alpha, etc. In these cases, the individual data can be extracted from the 'raw' signal by filtering it. For example, Theta waves have a frequency of between 4Hz and 7Hz, so if we filtered a raw EEG signal so that only those frequencies were retained we'd then have the Theta signal. Of course, we could also filter the same raw EEG signal for frequencies of between 15Hz and 30Hz, and we'd then have the Beta signal too.

ANY-maze can do this type of filtering - you just have to specify that you want to add a new *filtered signal* to an existing signal, and then specify the type of filter to apply. A single input signal can have any number of filters applied to it - so in the example above, we could have one to extract the Theta waves and another to extract the Beta waves.

By the way, filtering can also be useful when you want to remove some sort of artefact from a signal. For example, in our hypothetical blood pressure monitor, imagine that the device had a tendency to pick up interference caused by the 50Hz or 60Hz alternating current of the electrical supply. This might make our 'raw' blood pressure data almost unusable, but if you applied a filter to look at just the data from 0Hz to 20Hz, then the mains interference would be 'filtered out' and you'd be left with the true blood pressure signal. Of course, this assumes that the blood pressure wouldn't itself change at a frequency above 20Hz, but that seems rather unlikely.

See also:

- Setting up a Signal input
- I/O devices supported by ANY-maze
- An introduction to I/O Devices

Setting up a signal

Introduction

For a general introduction to signals, see An introduction to signals.

To add a signal to a protocol, click the *Add item* button in the ribbon bar and select *New input item* > *New signal input* from the menu which appears.

The New input item *menu option is only included if the* protocol mode *includes input/output.*

The New signal input *menu option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the signal's name.
2. Optionally enter conversion parameters to convert the signal from volts to real-world units.
3. Optionally set up details of the signal's baseline.
4. Specify whether the physical port used by the signal is always the same, or whether it depends on the location of a movable zone.
5. Specify the physical port (or ports) used by the signal.

What next?

After completing these steps, you should consider whether you want to include any filtered signals derived from this signal.

See also:

- Adding elements to a protocol
- Editing a signal or filtered signal
- Deleting a signal or filtered signal

ANY-maze help topic H0215

Signal name

In brief

Enter a name for the Signal in the *Signal name* field on the Signal's settings page. You must make an entry, but it can be anything you like.

Details

Signal names should specify the thing that the signal is measuring. For example, if the signal originates from a blood pressure monitor, then a good name would be 'Blood pressure', rather than something like 'BP Monitor voltage'.

The name of the signal will be included in the names of the signal's results, so to avoid having very long result names, you should try to keep signal names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Setting up a signal's conversion data

In brief

The conversion fields on the Signal's settings page allow you to convert a signal from volts to the real-world units of whatever it is that the signal is measuring.

The conversion is a simple linear conversion, and so requires just two pairs of values, for example: 0V = 0mmHg, 1V = 250mmHg. You also need to specify what the converted units are (mmHg in this example).

You don't have to enter any conversion data - if you don't, then ANY-maze will just report the values it reads from the signal input port, which will typically be in Volts.

Details

A signal typically reads a voltage that's being output from some sort of sensing device. But the thing that's being 'sensed' is rarely a number of volts; usually it's something like a pressure, a temperature or a gas concentration. In all these cases, it would be helpful to you if the signal could convert the voltage back into the appropriate real-world value - perhaps mmHg, °C or %CO2, in the examples.

This is what the conversion fields do. Here you first need to specify what the *Output units* of the converted value are, and then provide two pairs of values. Each pair consists of a value in volts and a value in the signal's output units, for example: 0V = -20°C, 1V = 180°C. ANY-maze requires that there is a **linear** relationship between the voltage and the real-world units, so it is then very easy to perform the conversion - effectively it's the same as plotting a conversion line on a graph. Having done this, the software can just 'read-off' the output value of any voltage - see figure 1.

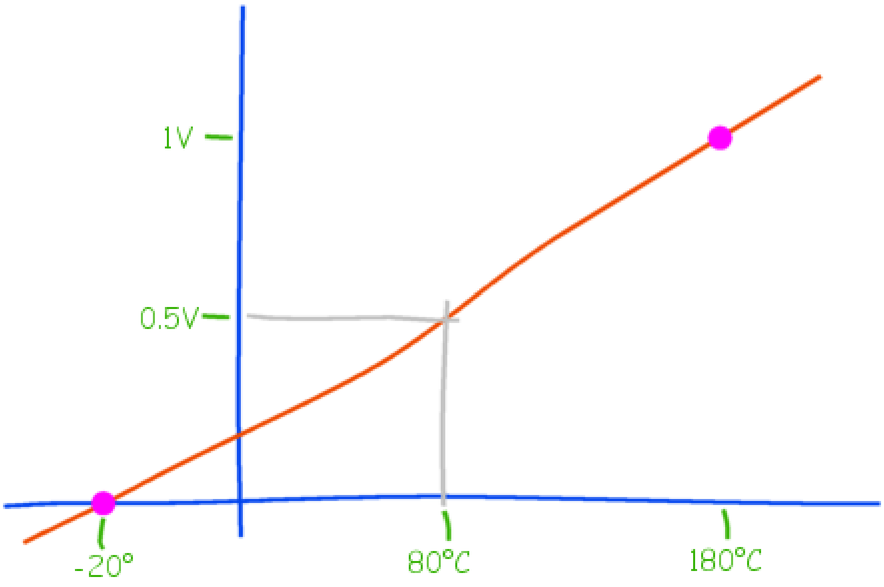


Figure 1. Conceptually, ANY-maze plots a graph using the two conversion points you provide and can then 'read-off' the output value of any voltage. Here, for example, a voltage of 0.5V is 'read-off' as a temperature of 80°C. Note that this requires that the relationship between voltage and the output units is linear.

Clearly, this process requires that you know two conversion points. In the majority of cases, these will be easy to find out from the documentation of the device which is generating the voltage. Often, the documentation will simply imply that 0V = 0° (let's assume our device is measuring a temperature) and so it will simply report something like '10mV/°C'. In this case you could use one conversion point of 0V = 0°C and a second one of 0.01V = 1°C.

There may be cases where you don't know this information, in which case you will need to work it out either using another sensor (perhaps a standard mercury thermometer) or by using some 'standard' (for example, melting ice is at 0°C and boiling water is at 100°C). If you do generate these conversion values yourself, try to use values towards the two ends of the range of values you will be measuring in your tests.

Limits

For the *Output units* you can enter anything you like up to a maximum of 6 characters.

For the conversion values in volts, you can enter any value from -99.0V to 99.0V.

For the output value, you can enter any value from -9999.0 to 9999.0.

Setting up a signal's baseline

In brief

A signal's baseline is generated from the values it has during a 'baseline period' at the start of a test. This period starts when the test starts and ends either after some specific period (for example, 30 seconds) or when something happens in the test (for example, the animal enters a specific zone).

If you chose to set up a baseline (it's optional), then you will also need to specify what is considered to be a *Deviation from the baseline*. This will be something like 'a change of more than 10% from the signal's average value during the baseline period'.

Armed with this information, ANY-maze will then report a range of measures relating to the signal and how it changes in comparison to the baseline during a test.

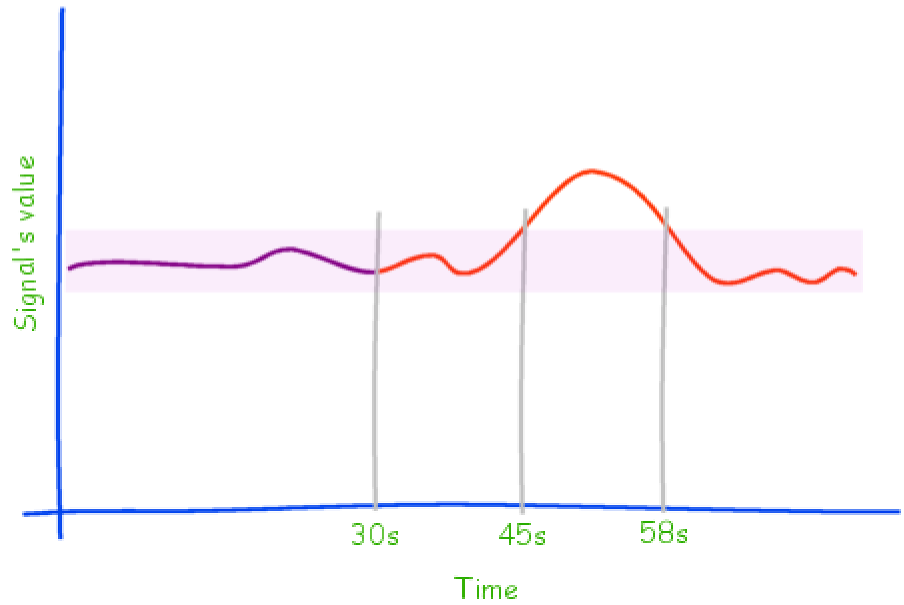


Figure 1. Here the baseline period is 30s, and 'deviation from the baseline' is defined as 2 standard deviations of the signal's value during the baseline period - this gives rise to the shaded rectangle, which shows the baseline region. At 45s, the signal deviates from the baseline and at 58s, it returns to it.

Details

The first part to setting up a baseline is to decide what will end the 'baseline period'. This is a the period during which the signal's value will be considered to be the baseline. This period can either end after a specific duration, or it can end when something happens in the test; for example, the animal enters a certain zone.

- To end the baseline after a certain period of time, you simply need to check the box *Baseline is signal's average value until the following time in the test,* and then enter the time in the relevant field. You can enter any value from 100ms to 10min. As is the case with most times in ANY-maze, you can use units of ms (milliseconds), s (seconds) or min (minutes) and you can mix units or use decimals, for example, you could enter '1min 30s' or '1.5min' or '90s'.
- To end the baseline when something occurs in the test, you will need to create a procedure which will determine when the specific situation arises and which will then use an Action to set a *Time marker*. You should then check the box *Baseline is signal's average value until the following time marker,* and choose the relevant time marker in the drop-down list.

Whichever method you choose, the baseline will cover a certain period at the start of the test, and ANY-maze will average the signal's value during this period and also calculate the standard deviation.

The second part to setting up a baseline is to specify what will be considered a 'deviation from the baseline'. Here, you will be defining how far from the baseline's average value the signal will need to go for a 'deviation' to have occurred. For example, if the baseline's value was 10 and the deviation was set to 10%, then if the signal went above 11 or below 9, it would be considered to have deviated. Note that a deviation above the baseline is termed a 'positive' deviation, while a deviation below the baseline is a 'negative deviation'.

You have the following options for specifying the deviation:

- A certain number of standard deviations. Here, the standard deviation is the standard deviation calculated for all of the signal's values during the baseline period. For example, if the baseline's average was 10 and the standard deviation was 0.65, and you specified that a deviation is 2 standard deviations, then ANY-maze would consider a positive deviation to have occurred if the signal goes above 11.3 and a negative deviation if it goes below 8.7.
- A certain percentage of the signal's average during the baseline period.
- A certain number of volts. Here, ANY-maze will convert the number of volts to whatever units you are using for the signal (if any). For example, if you specified 0.01 volts and the signal is being generated by a temperature sensor which generates a voltage of 10mV/°C, then 0.01 volts would equate to 1°C. If the signal's average during the baseline period was 24°C, then a deviation would occur if the signal went above 25°C or below 23°C.
- A certain number of whatever units the converted signal uses. For example, if the units

of the converted signal are mmHg and you enter 10, and the signal's average during the baseline period was 130mmHg, then a deviation would occur if the signal went above 140mmHg or below 120mmHg.

Baseline measures

If you set up a baseline for a signal, then ANY-maze will report a range of measures that would otherwise not be available. The full list can be found in Signal measures.

Specifying whether the physical port used by a signal is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port used by a signal will always be the same - it's simply the port on the interface device to which the signal actually connects - for example, an analogue input on an ANY-maze interface.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Place preference box you might have devices to measure the CO2 concentration in the two sides of the box. Now you decide to associate one side of the box with a drug, but you choose to balance this between animals, so for some animals the left side is the drug-associated side and for other animals it's the right side.

If you then wanted to see how CO2 concentrations change in the 'drug-associated side', you would have a problem, because for some animals it would be the signal from the gas analysis device connected to the left side, and for others it would be the signal from the gas analysis device connected to the right side. And as these two devices would be connected to a physical port (perhaps analogue input 1 and 2 of an ANY-maze interface), the physical port would not always be the same.

Details

If you do have a signal input whose physical port depends on the location of a movable zone, like in the Place preference box example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the signal's *settings page*. (Note that this drop-down list is only included if the protocol includes at least one movable zone).

In the case of the Place preference box example, you would specify that *The port varies depending on the location of the drug-associated zone.*

When you do this, the signal element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the example, there would be one sub-element for 'Port to use when drug-associated zone is in left position' and a second one for 'Port to use when drug-associated zone is in right position'. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Setting up a filtered signal's baseline

In brief

A filtered signal's baseline is generated from the values it has during a 'baseline period' at the start of a test. This period starts when the test starts and ends either after some specific period (for example, 30 seconds) or when something happens in the test (for example, the animal enters a specific zone).

If you chose to set up a baseline (it's optional), then you will also need to specify what is considered to be a *Deviation from the baseline*. This will be something like 'a change of more than 10% from the filtered signal's average value during the baseline period'.

Armed with this information, ANY-maze will then report a range of measures relating to the filtered signal and how it changes in comparison to the baseline during a test.

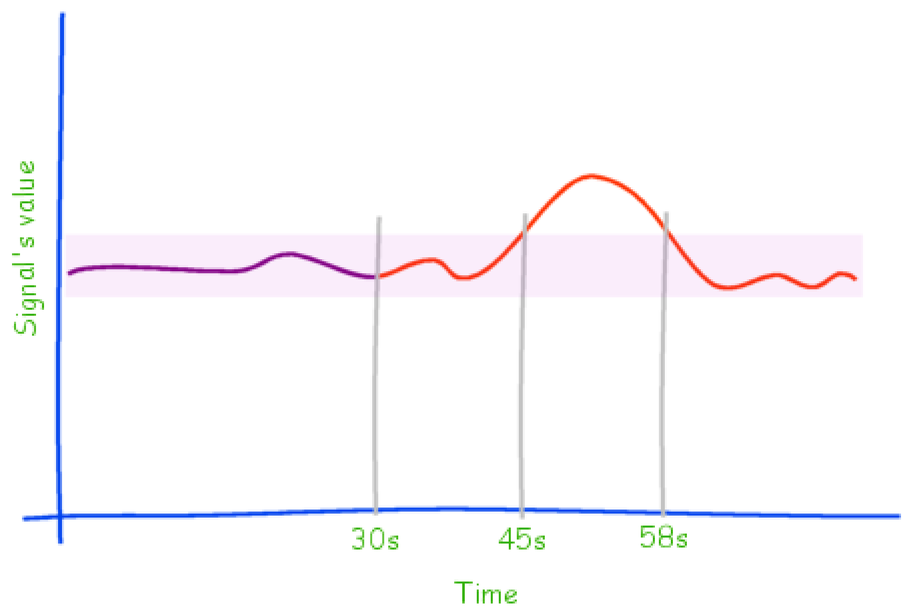


Figure 1. Here the baseline period is 30s, and 'deviation from the baseline' is defined as 2 standard deviations of the filtered signal's value during the baseline period - this gives rise to the shaded rectangle, which shows the baseline region. At 45s, the filtered signal deviates from the baseline and at 58s, it returns to it.

Details

The first part to setting up a baseline is to decide what will end the 'baseline period'. This is a the period during which the filtered signal's value will be considered to be the baseline. This period can either end after a specific duration, or it can end when something happens in the test; for example, the animal enters a certain zone.

- To end the baseline after a certain period of time, you simply need to check the box *Baseline is the signal's average value until the following time in the test,* and then enter the time in the relevant field. You can enter any value from 100ms to 10min. As is the case with most times in ANY-maze, you can use units of ms (milliseconds), s (seconds) or min (minutes) and you can mix units or use decimals, for example, you could enter '1min 30s' or '1.5min' or '90s'.
- To end the baseline when something occurs in the test, you will need to create a procedure which will determine when the specific situation arises and will then use an Action to set a *Time marker*. You should then check the box *Baseline is signal's average value until the following time marker,* and choose the relevant time marker in the drop-down list.

Whichever method you choose, the baseline will cover a certain period at the start of the test, and ANY-maze will average the filtered signal's value during this period and also calculate the standard deviation.

The second part to setting up a baseline is to specify what will be considered a 'deviation from the baseline'. Here, you will be defining how far from the baseline's average value the filtered signal will need to go for a 'deviation' to have occurred. For example, if the baseline's value was 10 and deviation was set to 10%, then if the filtered signal went above 11 or below 9, it would be considered to have deviated. Note that a deviation above the baseline is termed a 'positive' deviation, while a deviation below the baseline is a 'negative deviation'.

You have the following options for specifying the deviation:

- A certain number of standard deviations. Here, the standard deviation is the standard deviation calculated for all of the filtered signal's values during the baseline period. For example, if the baseline's average was 10 and the standard deviation was 0.65, and you specified that a deviation is 2 standard deviations, then ANY-maze would consider a positive deviation to have occurred if the filtered signal goes above 11.3 and a negative deviation if it goes below 8.7.
- A certain percentage of the filtered signal's average during the baseline period.
- A certain number of volts. Here, ANY-maze will convert the number of volts to whatever units you are using for the source signal (if any). For example, if you specified 0.01 volts and the source signal is being generated by a temperature sensor which generates a voltage of 10mV/°C, then 0.01 volts would equate to 1°C. If the filtered signal's average during the baseline period was 24°C, then a deviation would occur if the filtered signal went above 25°C or below 23°C.

- A certain number of whatever units the converted source signal uses. For example, if the units of the converted signal are mmHg and you enter 10, and the filtered signal's average during the baseline period was 130mmHg, then a deviation would occur if the filtered signal went above 140mmHg or below 120mmHg.

Baseline measures

If you set up a baseline for a filtered signal, then ANY-maze will report a range of measures that would otherwise not be available. The full list can be found in Signal measures.

Setting up a filtered signal

Introduction

A signal can be used as the source of any number of filtered signals. For example, a signal from an EEG device could have different filtered signals derived from it for the different EEG waves - Delta, Theta, Alpha, etc.

It's important to understand that you must first include the unfiltered 'source' signal in your protocol, and then add a filtered signal to it.

Adding a filtered signal

To add a filtered signal, first select the 'source' signal in the protocol list, then click the *Add item* button in the ribbon bar and select *New input item* > *New filtered signal* from the menu which appears.

You can add any number of filtered signals to a single 'source' filter.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the filtered signal's name.
2. Specify the filtered signal's filter characteristics.
3. Choose whether or not to integrate the filtered signal.
4. Optionally, set up details of the filtered signal's baseline - this is exactly the same as setting up an unfiltered signal's baseline.

What next?

After completing these steps, you should consider whether you want to include any sensors in the protocol.

See also:

- Setting up a signal
- Editing a signal or filtered signal
- Deleting a signal or filtered signal

Signal name

In brief

Enter a name for the Filtered signal in the *Filtered signal name* field on the Filtered signal's settings page. You must make an entry, but it can be anything you like.

Details

Filtered signal names should specify the thing that the signal is measuring. For example, if the filtered signal is extracting the theta waves from an EEG, then a good name would be 'Theta'; on the other hand, if the filter is removing mains hum from a blood pressure signal then a good name might be 'Blood pressure (cleaned)' (or better still, call the source signal something like 'Blood pressure raw' and then just call the filtered signal 'Blood pressure').

The name of the filtered signal will be included in the names of the filtered signal's results, so to avoid having very long result names, you should try to keep filtered signal names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying a filtered signal's filter characteristics

In brief

There are four filters available for generating a filtered signal from a source signal. These are: low pass, high pass, band pass and averaging. In the case of low and high pass filters, you just have to specify the filter's *cut off frequency,* which is the frequency above which (for high pass) or below which (for low pass) the signal will pass. For a band pass filter, you have to specify both a low and a high *cut off frequency* - the signal between these frequencies will pass. For high, low and band pass filters, there is also a *Transition width* setting; this is described in detail below, but you can usually just it leave blank - this will cause ANY-maze to use an automatic value.

The final filter type is an averaging filter - this just averages across a certain number of samples of the incoming signal; you have to specify how many samples this is.

Details

In order to understand the details of the filters which can be applied to a signal, you need to first understand what a signal actually is. Although signals were introduced in this help as 'inputs which have a continuously variable value', that's not exactly correct - or to be precise, the thing the signal is reporting on *does* have a continuously variable value, but the signal will only actually know its value at discrete (usually very short) moments in time. In other words, a signal will 'sample' the physical input at some set frequency (perhaps 100 times a second) and will then string all the values together to produce something which looks pretty much like the input - see figure 1.

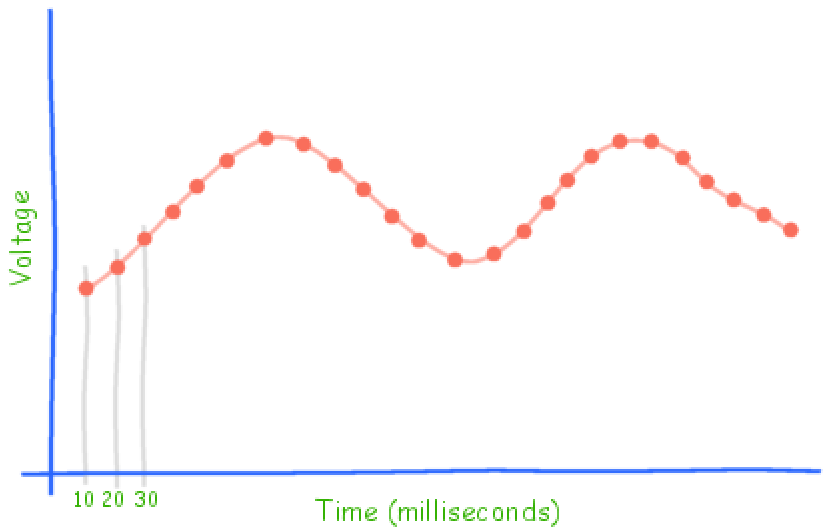


Figure 1. A signal is actually made up of lots of discrete values (the orange dots), which are 'sampled' at some set frequency by the signal's interface device. In this case, the dots are 10ms apart, so the signal is being sampled 100 times per second, i.e. at 100Hz.

While this may seem quite esoteric, it actually has an important repercussion - it means that a signal can't detect changes which are shorter than its sampling period. If you stop and think about it, this is self-evident. If the signal in figure 1 had a change which happened so quickly it fell between two of the dots, then it would be 'missed'. In fact, the highest frequency which can be successfully sampled by a signal is HALF the sampling frequency - this is called the *Nyquist frequency.* For example, if a signal is sampled at 100Hz then the Nyquist frequency is 50Hz - in other words with a sampling frequency of 100Hz, you won't be able to 'capture' any signal with a frequency higher than 50Hz.

And this is important for various reasons:

- You should always endeavour to use a sampling frequency which has a Nyquist frequency above (ideally well above) the highest frequency you expect to be present in the signal. This is true even if you're not interested in any high frequency components.
- You can't use a filter to remove frequencies above the Nyquist frequency.
- The filter parameters - which are what we're about to talk about - are limited by the Nyquist frequency. In this case, ANY-maze will check that you don't try to use inappropriate parameters and will warn you if you do.

Low pass filter

Let's start by considering a low pass filter, although many of the considerations I'll describe apply to high and band pass filters too.

To set up a low pass filter, you need to specify the *Low pass cut off frequency* in Hertz. This is the frequency below which the signal will pass and above which it will be blocked - see figure 2.

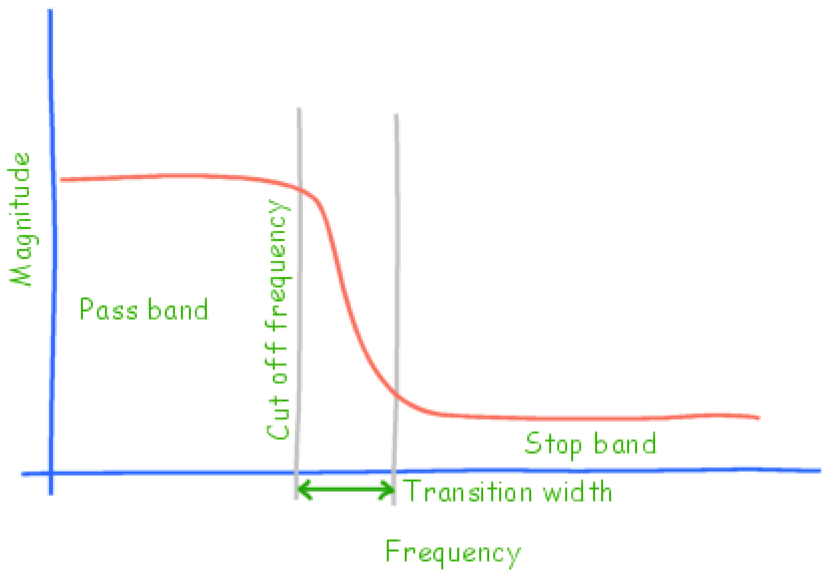


Figure 2. The frequency response of a low pass filter. Here you can see that the transition width represents how quickly the filter changes from the pass band to the stop band.

Transition width

You can also optionally specify the transition width. As can be seen in figure 2, this determines how quickly the filter transitions from the pass band to the stop band. It seems quite obvious that a narrow transition width is desirable; however, there's a trade-off, because a narrow transition width will generate a large 'delay' while a wide one will create a small 'delay'. The 'delay' is how far behind the source signal the filtered signal will be. For example, imagine that you have a source signal which is not oscillating at all, so it's at 0Hz. You then introduce a small burst of 5Hz oscillations. How long after the 5Hz signal actually starts would you expect the output of a 10Hz low-pass filter to show this 5Hz signal? The answer seems obvious - immediately! But in fact, digital filters (like those used in ANY-maze) introduce a delay.

In the above example, if we were to use the default transition width (which for a low pass filter is 10% of the cut off frequency, so 1Hz in this case) then the delay would be about 1s. If we changed the transition width to 0.2Hz, the delay would be about 5s, and if we changed it to 10Hz it would be about 0.1s.

The fact that this delay exists is often not actually very important. What it means is that we'll only know what the filtered signal looks like 1 second (using the default transition width in our example) after the signal enters the filter - but who cares?

Well, if you're analysing this data to generate results *after* the test has actually occurred (which is what ANY-maze will do) then the delay is almost meaningless - the only point to be aware of is that it will mean that you won't have any filtered signal data for the last 1s of the test. This is because the filter needs all the data from a 1s 'window' into the future in order to generate its output. So the filtered data for time 0s is created from the source data for time 0s-1s and the filtered data for time 0.1s is created from the source data for time 0.1s-1.1s, and so on. But if the test ends at time 10s, then the filtered data for time 9.5s (for example) needs the source data from time 9.5s - 10.5s - but we have no data after time 10s, so we can't create this output. So, other than the fact that we'll lose the last 'delay length' of the filtered signal, the delay doesn't really matter too much...

Except when we're using the filtered signal in real time. For example, imagine that you were using a filtered signal to monitor some aspect of an animal's physiology, and you wanted to switch on a light if the signal value increases by 10%. This would be simple to do using a procedure, but there'd be a problem - the signal would only actually change at a point 'delay' seconds after the animal's physiology changed, and so the light would come on 'delay' seconds *too late.* There's nothing you can do about this, but as filter delay will usually be quite small, it's generally not really a problem. However, if you decided to change the transition width to something very small, then you'd create a long delay and then this effect might become important.

You'll find that when you enter the cut-off frequency(s) and transition width for a filter, ANY-maze will calculate the delay and tell you what it will be. You can then make a decision as to whether to adjust the parameters so as to reduce the delay, based on the requirements of your experiment.

High pass

Setting up a high pass filter is almost identical to setting up a low pass filter, just that the cut off frequency will define the frequency *above which* the signal will pass. The filter will have a transition width and a delay, just like the low pass filter has (see above).

Band pass filter

A band pass filter passes the signal in a specific band of frequencies - see figure 3. Again, like the low pass filter, the band pass filter has a transition width and a delay.

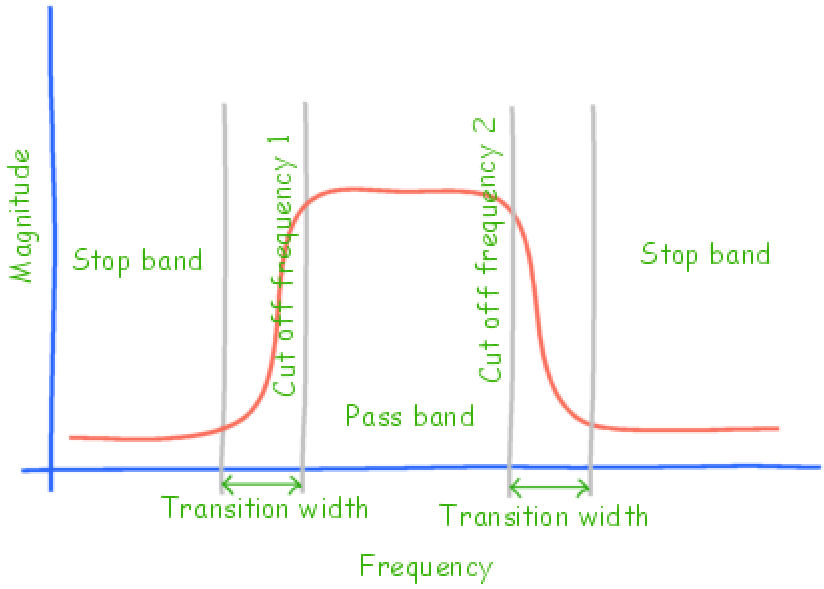


Figure 3. The frequency response of a band pass filter.

Averaging filter

An averaging filter simply averages a certain number of source 'samples' to create the filter output. For example, if you specify that an averaging filter should average 20 samples, then for each sample it will take the 10 samples before it and the 10 after it (including the sample itself) and average their values to produce the new, filtered, sample.

If you think about how this works for a moment, you'll realise that it means that the filter requires the ten (in our example) samples *after* the sample it's working on; this means it will have to wait for these samples to arrive, and so this filter will have a delay, just like the low, high and band pass filters. The same considerations relating to the delay (described above) apply to this filter too. Again, ANY-maze will calculate the delay and display it when you enter the filter parameters.

Charting filtered signals

It can often be very helpful to have ANY-maze plot a chart of the source and filtered signals. This makes it easy to see the effect the filter is having.

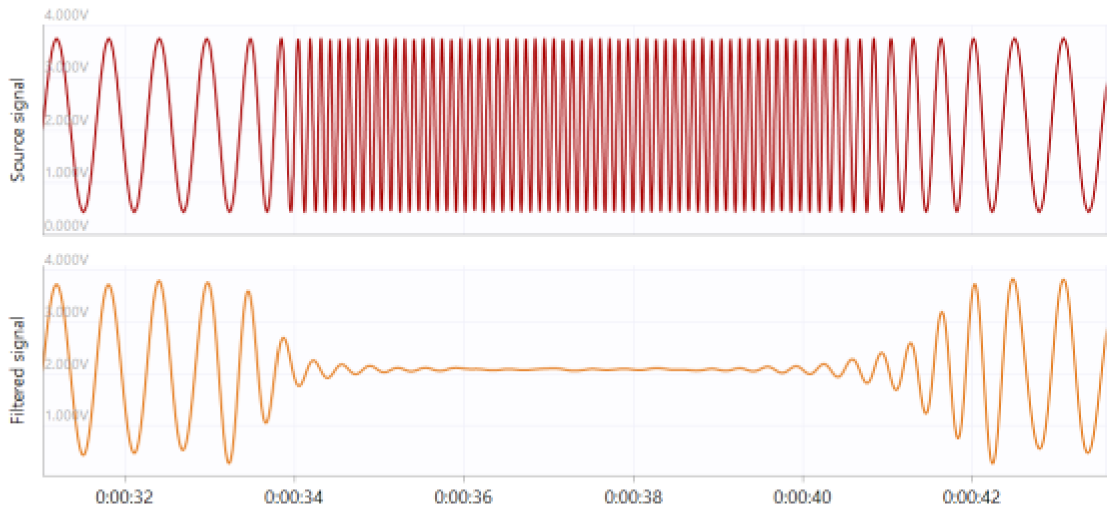


Figure 4. A source signal (top) and a low pass filtered signal (bottom). The filter had a cut off frequency of 3Hz and a transition width of 1Hz. The source signal starts at around 2.5Hz goes to about 9Hz and drops back to around 2Hz again.

Choosing whether or not to integrate a filtered signal

In brief

You can optionally integrate a filtered signal over a specific period. For example, if you integrate over 1s then the system will sum the area under the curve of the filtered signal for the 1s prior to the sample it is currently processing, and this will become the filter output. In fact, the value is divided by the number of samples, which makes it of the same order as the original signal, but this is effectively just dividing all the signal values by a constant and makes no real difference to the result.

Details

Integrating a signal is similar to applying a moving average to it. A couple of points to note, though:

- It is the *filtered signal* which is integrated, not the source signal - i.e. integration is applied after the source signal has been filtered.
- The integration period is entirely *before* the value being integrated.

Integrating a filtered signal adds a further delay to the signal. In this case, the delay is the same as the integration period and is cumulative with the filter delay. For example, if a filter adds a 1s delay to a signal and you then integrate over 2s, the total delay of the resulting filter signal (in comparison to the source signal) will be 3s.

Editing a signal or a filtered signal

Introduction

You can edit absolutely everything related to a signal or a filtered signal at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a signal
- Setting up a filtered signal
- Deleting a signal or filtered signal

Deleting a signal or filtered signal

Introduction

You can delete a signal or a filtered signal at any time, whether before, during or after an experiment has been performed.

To delete a signal or filtered signal, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Note that if you delete a signal, any filtered signals derived from it will be deleted too.

Restrictions

There are no restrictions on deleting signals or filtered signals.

See also:

- Deleting protocol elements
- Setting up a signal
- Setting up a filtered signal
- Editing a signal or filtered signal

Signal measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a signal or a filtered signal:

- Average
- Maximum
- Minimum
- Time of maximum
- Time of minimum
- Baseline
- Baseline deviation
- Baseline standard deviation
- Time of the end of the baseline period
- Time of first positive deviation from baseline
- Time of first return to baseline from positive deviation
- Time of first negative deviation from baseline
- Time of first return to baseline from negative deviation
- Integral above baseline
- Integral below baseline
- Average maximum for each visit to zone
- Average time to maximum for each visit to zone
- Average minimum for each visit to zone
- Average time to minimum for each visit to zone
- Average at zone entry
- Average at zone exit

Average

Description	The mean of the signal's value over the duration of the test.
Calculation method	All the signal's samples are summed across the duration of the test and the sum is divided by the count of samples.
Analysis in zones	This measure can be analysed in zones. It reports the mean of the signal's value while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It reports the mean of the signal's value during the time period.
Units	Same as the signal's units
Notes	None

Maximum

Description	The signal's maximum value over the duration of the test.
Calculation method	All the signal's samples are scanned and the largest value is found.
Analysis in zones	This measure can be analysed in zones. It reports the signal's maximum value reports while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It reports the signal's maximum value during the time period.
Units	Same as the signal's units
Notes	None

Minimum

Description	The signal's minimum value over the duration of the test.
Calculation method	All the signal's samples are scanned and the smallest value is found.
Analysis in zones	This measure can be analysed in zones. It reports the signal's minimum value reports while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It reports the signal's minimum value during the time period.
Units	Same as the signal's units
Notes	None

Time of maximum

Description	The time at which the signal's maximum value occurred
Calculation method	The test clock at the time the maximum value occurred
Analysis in zones	This measure can be analysed in zones. It reports the time at which the signal's maximum value occurred, while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It reports the time *in the time period* at which the signal's maximum value during the time period occurred.
Units	Seconds
Notes	None

Time of minimum

Description	The time at which the signal's minimum value occurred
Calculation method	The test clock at the time the minimum value occurred
Analysis in zones	This measure can be analysed in zones. It reports the time at which the signal's minimum value occurred, while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It reports the time *in the time period* at which the signal's minimum value during the time period occurred.
Units	Seconds
Notes	None

Baseline

Description	The signal's baseline value
Calculation method	The average of the signal's values during the baseline period. All the signal's samples are summed across the baseline period and the sum is divided by the count of samples.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Same as the signal's units
Notes	None

Baseline deviation

Description	The signal's baseline deviation value, as defined in the protocol. This is the amount the signal must deviate from the baseline value for it to count as a *deviation.* For example, if the protocol specifies that the signal's baseline deviation is 10% and the baseline value is 20, then the baseline deviation will be 2.
Calculation method	Depends on how the deviation is defined in the protocol.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Same as the signal's units
Notes	None

Baseline standard deviation

Description	The standard deviation of the signal's value during the baseline period.
Calculation method	The standard deviation of all of the signal's samples during the baseline period.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Same as the signal's units
Notes	None

Time of the end of the baseline period

Description	The time at which the baseline period ended
Calculation method	The test clock at the time the baseline period ended
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	If the baseline period is defined in the protocol as a set period, then this will be that period, but if the baseline period ends at some time marker (which could be different in different tests), then this will be the time of the time marker.

Time of first positive deviation from baseline

Description	The time at which the signal's value first deviated above the baseline value by more than the Baseline deviation.
Calculation method	The test clock at the time at which the deviation occurred.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure can be analysed across time. It is the time *within the period* at which the signal's value first (within the period) deviated above the baseline value by more than the baseline deviation.
Units	Seconds
Notes	None

Time of first return to baseline from positive deviation

Description	The time at which the signal's value first returned from being above the baseline value to be within the baseline deviation of it.
Calculation method	The test clock at the time at which the return to the baseline occurred. It is implicit that a positive deviation from the baseline must already have occurred.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure can be analysed across time. It is the time *within the period* at which the signal's value first (within the period) returned from being above the baseline value to be within the baseline deviation of it.
Units	Seconds
Notes	None

Time of first negative deviation from baseline

Description	The time at which the signal's value first deviated below the baseline value by more than the baseline deviation.
Calculation method	The test clock at the time at which the deviation occurred.
Analysis in zones	This value measure be analysed in zones.
Analysis across time	This measure can be analysed across time. It is the time *within the period* at which the signal's value first (within the period) deviated below the baseline value by more than the baseline deviation.
Units	Seconds
Notes	None

Time of first return to baseline from negative deviation

Description	The time at which the signal's value first returned from being below the baseline value to be within the baseline deviation of it.
Calculation method	The test clock at the time at which the return to the baseline occurred. It is implicit that a negative deviation from the baseline must already have occurred.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure can be analysed across time. It is the time *within the period* at which the signal's value first (within the period) returned from being below the baseline value to be within the baseline deviation of it.
Units	Seconds
Notes	None

Integral above baseline

Description	The sum of the area under the signal's line and above the baseline
Calculation method	Every sample after the baseline period is compared to the baseline; if it is above it, then the difference between the signal value and the baseline (divided by the sampling rate) is added to the integral.
Analysis in zones	This measure can be analysed in zones. It is the sum of the area under the signal's line and above the baseline for all the samples captured while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It is the sum of the area under the signal's line and above the baseline for all the samples captured during the time period.
Units	Same as the signal's units · seconds
Notes	None

Integral below baseline

Description	The sum of the area above the signal's line and below the baseline
Calculation method	Every sample after the baseline period is compared to the baseline; if it is below it, then the difference between the baseline and the signal value (divided by the sampling rate) is added to the integral.
Analysis in zones	This measure can be analysed in zones. It is the sum of the area above the

	signal's line and below the baseline for all the samples captured while the animal was in the zone.
Analysis across time	This measure can be analysed across time. It is the sum of the area above the signal's line and below the baseline for all the samples captured during the time period.
Units	Same as the signal's units · seconds
Notes	None

Average maximum for each visit to zone

Description	The average of the signal's maximum value on each of the animal's visits to the zone. This measure can **only** be reported for zones - it cannot be reported for the apparatus as a whole.
Calculation method	The maximum signal value on each of the animal's visits to the zone is found. These are summed and divided by the number of zone visits.
Analysis in zones	This measure can **only** be reported in zones.
Analysis across time	This measure can be analysed across time. For each time period, it is the average of the signal's maximum value on each of the animal's visits to the zone during the period.
Units	Same as the signal's units
Notes	None

Average time to maximum for each visit to zone

Description	The average time after entering a zone at which the signal reaches the maximum value during the visit to the zone. This measure can **only** be reported for zones - it cannot be reported for the apparatus as a whole.
Calculation method	For each visit to the zone, the maximum signal value is found. The value of the test clock when the animal entered the zone is subtracted from the value of the test clock at the time of this maximum, and the resulting time is added to a sum of times. This sum is then divided by the total number of zone visits.
Analysis in zones	This measure can **only** be reported in zones.
Analysis across time	This measure can be analysed across time. For each visit to the zone during the time period, the maximum signal value is found. The value of the test clock when the animal entered the zone (or when the time period started, whichever is later) is subtracted from the value of the test clock at the time of this maximum, and the resulting time is added to a sum of times. This sum is then

	divided by the total number of zone visits during the time period.
Units	Seconds
Notes	None

Average minimum for each visit to zone

Description	The average of the signal's minimum value on each of the animal's visits to the zone. This measure can **only** be reported for zones - it cannot be reported for the apparatus as a whole.
Calculation method	The minimum signal value on each of the animal's visits to the zone is found. These are summed and divided by the number of zone visits.
Analysis in zones	This measure can **only** be reported in zones.
Analysis across time	This measure can be analysed across time. For each time period, it is the average of the signal's minimum value on each of the animal's visits to the zone during the period.
Units	Same as the signal's units
Notes	None

Average time to minimum for each visit to zone

Description	The average time after entering a zone at which the signal reaches the minimum value during the visit to the zone. This measure can **only** be reported for zones - it cannot be reported for the apparatus as a whole.
Calculation method	For each visit to the zone, the minimum signal value is found. The value of the test clock when the animal entered the zone is subtracted from the value of the test clock at the time of this minimum, and the resulting time is added to a sum of times. This sum is then divided by the total number of zone visits.
Analysis in zones	This measure can **only** be reported in zones.
Analysis across time	This measure can be analysed across time. For each visit to the zone during a time period, the minimum signal value is found. The value of the test clock when the animal entered the zone (or when the time period started, whichever is later) is subtracted from the value of the test clock at the time of this minimum, and the resulting time is added to a sum of times. This sum is then divided by the total number of zone visits during the time period.
Units	Seconds
Notes	None

Average at zone entry

Description	The average of the signal's values at the moment the animal entered the zone. This measure can **only** be reported for zones - it cannot be reported for the apparatus as a whole.
Calculation method	Each time the animal enters the zone, the signal's value is added to a sum. This sum is then divided by the number of zone entries.
Analysis in zones	This measure can **only** be reported in zones.
Analysis across time	This measure can be analysed across time. For each zone entry during a time period, the signal's value at the moment the animal entered is added to a sum. This sum is then divided by the number of zone entries during the period.
Units	Same as the signal's units
Notes	None

Average at zone exit

Description	The average of the signal's values at the moment the animal exited the zone. This measure can **only** be reported for zones - it cannot be reported for the apparatus as a whole.
Calculation method	Each time the animal leaves the zone, the signal's value is added to a sum. This sum is then divided by the number of zone exists.
Analysis in zones	This measure can **only** be reported in zones.
Analysis across time	This measure can be analysed across time. For each zone exit during a time period, the signal's value at the moment the animal exited is added to a sum. This sum is then divided by the number of zone exists during the period.
Units	Same as the signal's units
Notes	None

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures

- On/off input measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Pellet dispenser measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

ANY-maze help topic H0227

Sensors

Sensors can be used in ANY-maze to register such things as the light level, humidity or temperature in your apparatus, either before a test starts or continuously throughout the test.

For more information about sensors, refer to the topics below.

- An introduction to sensors
- Setting up a sensor
- Editing a sensor
- Deleting a sensor
- Sensor measures

An introduction to sensors

Introduction

Sensors are used in ANY-maze to determine some physical value, for example, temperature or light level. ANY-maze currently supports four types of sensors:

- Temperature sensors - to either determine the ambient temperature or the temperature of some specific thing.
- Light sensors - to determine the light intensity in lux.
- Weight sensors - to determine the weight of something;, for example, the weight of food consumed by an animal.
- Humidity sensors - to determine the relative humidity of the air, usually in a cage or piece of apparatus.

Sensors can be read before a test starts, which would be appropriate to register the temperature of the water in a water-maze, or continuously throughout the experiment, which would be appropriate to measure an animal's food consumption across a 12 hour period.

Connecting a sensor to ANY-maze

Clearly, if ANY-maze is going to measure the value of a sensor, then the sensor device will have to be physically connected to your computer. In some cases, the sensor may be an integral part of a device you are using - this is the case, for example, with the temperature sensor in OPAD, which registers the temperature of the OPAD thermal elements. In other cases, you can connect a sensor using a special piece of equipment called an *Input/Output device* (usually abbreviated to 'I/O device') which acts as an interface between your PC and the actual sensor.

You'll find full details about the I/O devices which can be used with ANY-maze in the I/O devices supported by ANY-maze topic - but suffice to say that at present, only the ANY-maze interface and OPAD support sensors.

Using sensors

Having connected a sensor to your computer, you then need to add two elements to your protocol in order to use it: an I/O Device element, for the physical I/O device to which the sensor is connected, and a *Sensor input* for the sensor itself.

In fact, I/O devices usually support multiple sensors, so you only need to set up the I/O device element once to make all of its sensors available within the protocol. For example, an ANY-maze interface supports as many as 8 sensors - but you only need to include the device itself in the protocol once for all the sensors to be available. The process of adding an I/O device to your protocol is described in detail in the Setting up an I/O device topic.

It doesn't matter what type of sensor input you want to include in your protocol (it can be any of those listed above); in all cases, you simply add a 'Sensor' element.

As mentioned above, sensors can be set to be read once at the start of a test, or they can be read continuously throughout the test.

Reading a sensor at the start of a test causes the sensor's value (in degrees, lux, grams or %) to be included in the test results. This is useful as you can then analyse the result just like any other value; so, for example, you could read the light level in a plusmaze before every test in an experiment, and then check that there was no significant difference between the light levels for your treatment groups.

Reading a sensor throughout a test is useful if you expect the sensor's value to change. For example, if you place the animal's food on top of a weight sensor, then you can register the amount of food the animal eats during the experiment. Not only will you know how much the animal ate overall, but you will know *when* it ate and how much it ate on each occasion. For example, you might test the animal over 24 hours and then divide the test into 1 hour periods - you would then be able to see how much the animal ate during each hour of the test.

Another advantage of reading a sensor continuously is that it can be used in procedures. For example, you could create a procedure which would determine if the animal has eaten more than 10g of food and if it has, trigger some sort of action, which might end the test, activate a shocker, play a sound, etc.

Sensors can also be used by the ANY-maze watchdog to alert you if a sensor's value goes outside certain limits. This feature is independent of experiments, and a sensor that the watchdog is monitoring can simultaneously be used in an experiment if desired.

See also:

- The ANY-maze interface
- The AMi sensor ports
- Configuring AMi sensors
- The ANY-maze Watchdog
- Setting up a sensor
- Sensor measures
- An introduction to procedures

Setting up a sensor

Introduction

For a general introduction to sensors, see An introduction to sensors.

To add a sensor to a protocol, click the *Add item* button in the ribbon bar and select *New input item* > *New light sensor* from the menu which appears.

The New input item *menu option is only included if the* protocol mode *includes input/output.*

The New sensor *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the sensor's name
2. Specify when the sensor should be read
3. Specify the value that should be reported in the test results
4. Specify whether ANY-maze should try to detect the sensor's presence, and what it should do if the sensor is not present
5. Specify whether the physical port used by the sensor is always the same, or whether it depends on the location of a movable zone.
6. Specify the physical port used by the sensor

What next?

After completing these steps, you should consider whether you want to include any Output switches in this protocol.

See also:

- Adding elements to a protocol
- Editing a sensor
- Deleting a sensor

Sensor name

In brief

Enter a name for the sensor in the *Sensor name* field on the sensor's settings page. You must make an entry, but it can be anything you like.

Details

Sensor names should specify what the sensor is intended to measure - so this might be something like 'Water temperature' or 'Food consumption'.

The names of sensor results will include this name, so it's a good idea to try to use relatively short names so as to avoid making the names of results very long. For example, 'Water temp.' would be a better name than 'Water temperature'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying when to read a sensor

In brief

Use the options on the sensor's settings page to choose when a sensor should be read: at the start of a test, continuously throughout the test, and/or once at the end of the test.

These options, together with those described in the Specifying the sensor value recorded in results topic, influence the measures that will actually be reported for the sensor.

Details

Sensors can either be read once to register the sensor's value at the start of a test, or they can be read continuously throughout the test.

Reading the sensor's value at the start of tests

To read a sensor once, you should choose one of the following options:

- *Read at the start of the test*
- *Read before the test, when the sensor's key is pressed*

If you choose the first option, ANY-maze will automatically read the sensor's value at the very start of every test; you don't need to do anything, this will just happen.

If you choose the second option, you will need to press the sensor's key before each test - the sensor's value at the moment you press the key will be noted by ANY-maze and recorded in the test's results. Note that not all sensors include a 'key' - the AMi light sensor does, for example (it's the LUX key on the sensor) but the OPAD temperature sensor does not. If you choose this option for a sensor which does not have a key, then ANY-maze will display a message to tell you that the option is inappropriate.

When you select the option to read the sensor when its key is pressed, you should make sure you read the sensor when the test status is shown as *Ready...*; trying to read the sensor once the test has started won't work. If you want to repeat a reading, simply press the sensor's key again - the value that will be recorded in the test results is the value registered from the *last* key press before the test starts. If you fail to press the key before the test starts (for example, you forget) then the result will be undefined.

When a sensor is only configured to be read at the start of tests, ANY-maze will include just one measure for the sensor - *Initial value*.

When using the ANY-maze-interface light or temperature sensors, you may want to set the sensors to have their values shown in the ANY-maze status bar (and possibly have ANY-maze speak the *value too), as you can then easily confirm the value that has been registered - see the* Configuring AMi sensors *topic for more details.*

Reading a sensor continuously throughout tests

To read a sensor continuously throughout a test, you should select the option *Read continuously during the test*. In this case, ANY-maze will note the sensor's value each time it's reported by the sensor hardware. This frequency varies depending on the sensors being used, but is typically somewhere between 0.2Hz and 10Hz.

In this case, ANY-maze will include various measures for the sensor such as its average, maximum and minimum values.

Reading a sensor at the end of a test when the sensor's key is pressed

In some circumstances you may want to read a sensor *after* a test has ended. For example, if you use a weight sensor to determine how much food the animal consumes during a test, then you will probably find that while the animal is actually eating, the sensor's value doesn't change. This is because the animal will be causing the weight to fluctuate, but as soon as the animal stops eating the weight will stabilize and the sensor will report the new value. But what happens if the animal is eating at the very end of a test? In this case, the test will end before the sensor sends a final 'stable' reading. To address this, you could specify that the sensor should be read at the end of the test when its key is pressed. So at the end of the test, you would remove the animal from the apparatus and then press the sensor's key; as the animal would no longer be eating, the sensor would have a stable value and this would be the value you would record.

Note that not all sensors include a 'key'. If you choose this option for a sensor which does not have a key, then ANY-maze will display a message to tell you that the option is inappropriate.

See also:

- Specifying the sensor value recorded in results
- Sensor measures
- Configuring AMi sensors

Specifying the sensor value reported in results

In brief

Use the options on the sensor's settings page to choose the value that should be reported for the sensor in the test results. The options are essentially the absolute value of the sensor, or the change in the sensor's value during the test.

Details

There are three options for the value which a sensor should report in the test results. They are:

- *The actual value read from the sensor*
- *The increase in the sensor's value (relative to the start value)*
- *The decrease in the sensor's value (relative to the start value)*

The first option is the most commonly used. For example, if you wish to record the temperature of the water in a water-maze, then you would use this option. So the results would report the actual temperature read by the sensor.

The other options are both similar, and report the change in the sensor's value during the test. Which you use will depend on what the sensor is actually sensing. For example, if you are using a weight sensor to determine the weight of faeces excreted by an animal during the test, then you would specify that the sensor's value should be the increase during the test, relative to the start value. Thus the sensor might be reading 50g at the start of the test (being the weight of the receptacle that will hold the faeces) and 58g at the end of the test - so the sensor's result would be 8g.

On the other hand, if you wish to record the weight of food consumed by the animal during a test, then you would specify that the sensor's value is the decrease during the test, relative to the start value. In this case, the weight of food might be 100g at the start of the test and 94g at the end, so the result (how much food the animal ate) would be 6g.

Note that for either of the 'difference' values to be reported, you must have either specified that the sensor is to be read continuously throughout the test or that it should be read at the end of a test when the sensor's key is pressed. If neither of these options is specified, then the sensor will simply not report any results at all.

See also:

- Specifying when to read a sensor
- Sensor measures

ANY-maze help topic H0233

Specifying whether ANY-maze should detect the presence of a sensor

In brief

ANY-maze is able to detect the presence of some sensors. In the cases where it can do this, you can specify whether a test should be allowed to start if the sensor is missing.

Details

Some sensors can be removed from your test apparatus - a good example is the water-bottle in the OPAD cage. Clearly you need to be able to remove this to clean and fill it, but what would happen if you forgot to put it back before you started a test? The answer probably is that the test would be wasted.

However, ANY-maze can detect whether the water bottle is physically present (when it's missing, the weight sensor registers a much lower value than normal) so it would make sense if ANY-maze could warn you that the bottle is missing and prevent you from starting the test until it has been correctly installed. That's what this setting allows you to do.

There are three options:

- *Don't try to detect the presence of this sensor*
- *If sensor isn't present, ignore it in tests*
- *If sensor isn't present, don't allow tests to start*

The first option is the default - in this case, ANY-maze won't care whether the sensor is present or not. If it is missing then it will either have no results, or the results will just be zero. This is also the option that is automatically used (irrespective of the option you choose) for sensors which don't support presence detection.

The second option will cause ANY-maze to report the senor's results as undefined in any tests where the sensor is not present. Other than this, the tests will be performed normally.

The third option will prevent a test from being shown as *Ready...* until the sensor is installed correctly. Instead of the *Ready...* status, ANY-maze will report *Missing I/O...* and the status shown on the tests page will tell you what the actual problem is, for example *Water bottle not present.*

Specifying whether the physical port used by a sensor is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port used by a sensor will always be the same - it's simply the port on the interface device to which the sensor actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Place preference box, you might have devices to measure the weight of liquid the animal drinks in the two side of the box. Now you decide to associate one side of the box with a drug, but you choose to balance this between animals; so for some animals the left side is the drug-associated side, and for other animals it's the right side.

If you then wanted to see the liquid consumed in the 'drug-associated side', you would have a problem, because for some animals it would be the data from the sensor on the left side and for others it would be the data from the sensor on the right side. And, as these two devices would be connected to two different physical ports, the physical port would not always be the same.

Details

If you have a sensor input whose physical port depends on the location of a movable zone, like in the Place preference box example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the sensor's *settings page*. (Note that this drop-down list is only included if the protocol includes at least one movable zone).

In the case of the Place preference box example, you would specify that *The port varies depending on the location of the drug-associated zone.*

When you do this, the sensor element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the example there would be one sub-element for 'Port to use when drug-associated zone is in left position' and a second one for 'Port to use when drug-associated zone is in right position'. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a sensor

Introduction

You can edit everything related to a sensor at any time, whether before, during or after an experiment has been performed - there are no restrictions.

Note, however, that if you alter a sensor that was being read only at the start of a test to be read continuously throughout the test, only the tests performed after the change will actually include the data for the entire test.

See also:

- Editing the elements of a protocol
- Setting up a sensor
- Specifying when to read a sensor
- Deleting a sensor

Deleting a sensor

Introduction

You can delete a sensor at any time, whether before, during or after an experiment has been performed.

To delete a sensor, simply select it in the protocol list and click the ❌ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting sensors.

See also:

- Deleting protocol elements
- Setting up a sensor
- Editing a sensor

Sensor measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a sensor:

- Initial value
- Average value
- Maximum value
- Minimum value
- Change

Initial value

Description	The sensor's value at the start of the test.
Calculation method	Depends on how the sensor is configured; either the last value reported by the sensor prior to the start of the test (this will typically be read less than 1 second before the start of the test), or the value read from the sensor when the sensor's key was pressed before the test started.
Analysis in zones	This value is not available within zones.
Analysis across time	This measure cannot be analysed across time.
Units	Depends on the sensor (lux, degrees, grams or %)
Notes	This is the only measure that is reported when a sensor is set to be read at the test start. When a sensor is set to report the change in its value during a test, this measure is NOT reported (as it would always be 0).

Average value

Description	Reports the average value reported by the sensor.
Calculation method	Simple average of the values reported by the sensor during the test.
Analysis in zones	Average of the values reported while the animal was in the zone.
Analysis across time	This measure can be analysed across time. The result is the average of the values reported during the time period.

Units	Depends on the sensor (lux, degrees, grams or %)
Notes	This measure is only reported when a sensor is set to be read continuously throughout a test.

Maximum value

Description	Reports the maximum value reported by the sensor.
Calculation method	The largest value reported by the sensor during the test.
Analysis in zones	The largest value reported while the animal was in the zone.
Analysis across time	This measure can be analysed across time. The result is the largest of the values reported during the time period.
Units	Depends on the sensor (lux, degrees, grams or %)
Notes	This measure is only reported when a sensor is set to be read continuously throughout a test.

Minimum value

Description	Reports the minimum value reported by the sensor.
Calculation method	The smallest value reported by the sensor during the test.
Analysis in zones	The smallest value reported while the animal was in the zone.
Analysis across time	This measure can be analysed across time. The result is the smallest of the values reported during the time period.
Units	Depends on the sensor (lux, degrees, grams or %)
Notes	This measure is only reported when a sensor is set to be read continuously throughout a test.

Change

Description	Reports the change in the sensor's value relative to the value at the start of the test.
Calculation method	The first value recorded in the test is subtracted from the last value recorded in the test.
Analysis in zones	This value is not available within zones.
Analysis across time	This measure can be analysed across time (see notes). The result is the value at the start of the time period subtracted from the last value recorded in the time

	period.
Units	Depends on the sensor (lux, degrees, grams or %)
Notes	This measure is only reported when a sensor is set to be read continuously throughout a test or is set to be read after the test when the sensor's key is pressed. If the sensor is not set to be read continuously, then analysis across time is not possible.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Pellet dispenser measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Movement detectors

Movement detectors use arrays of photobeams to detect movement of the animal - as it moves, it breaks the beams (for this reason they're also known as *Beam Break Systems*).

Movement detectors can't provide the same depth of information about the animal's movements as video tracking does, but they can generate reliable movement 'counts' in circumstances in which video tracking can't be used.

For more information about movement detectors in ANY-maze, refer to the topics below.

- An introduction to movement detectors
- Setting up a movement detector
- Editing a movement detector
- Deleting a movement detector
- Movement detector measures

An introduction to movement detectors

Introduction

As the name implies, a movement detector is a device that detects movement of the animal. It does this using an array of photobeams - as the animal moves, it breaks the beams and so the movement is detected.

Unlike video tracking, a movement detector can't determine such things as the distance the animal has moved or the speed at which it is moving, but it can provide movement 'counts', which can be compared to determine the relative amount of movement between animals, or groups of animals.

Although the depth of information that a movement detector can generate is far less than that provided by video tracking, they do have some advantages: They are not dependent on the contrast between the animal and the background; they're unaffected by lighting conditions, including changes in lighting during a test; and they're more *lightweight*, requiring far less processing power and generating much smaller experiment files than video tracking.

Physically connecting a movement detector to ANY-maze

Movement detectors are implemented as *Photobeam arrays,* and these are supported by the ANY-maze interface (AMi) and the AMi-2 Digital interface. These support standalone photobeam arrays, which are available in a range of sizes - full details can be found here.

See also:

- The ANY-maze interface
- Setting up a movement detector
- Movement detector measures

Setting up a movement detector

Introduction

For a general introduction to movement detectors, see An introduction to movement detectors.

To add a movement detector to a protocol, click the *Add item* button in the ribbon bar and select *New input item* > *New movement detector* from the menu which appears.

The New input item *menu option is only included if the* protocol mode *includes input/output.*

The New movement detector *option is only included when the* protocol mode *excludes video tracking (i.e. it is a TakeNote or Input/Output only mode).*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the movement detector's name.
2. Enter the movement bout timeout value.
3. Specify the physical port used by the detector.

What next?

After completing these steps, you should consider whether you want to include any Output switches in this protocol.

See also:

- Adding elements to a protocol
- Editing a movement detector
- Deleting a movement detector

Movement detector name

In brief

Enter a name for the movement detector in the *Movement detector name* field on the movement detector's settings page. You must make an entry, but it can be anything you like.

Details

Movement detector names should specify what the detector will be measuring; usually this is the movement of the animal, so a name such as 'Animal movement' is a good choice. Alternatively, you may prefer to use a name such a 'Beam breaks' as this reflects what the detector is actually reporting, and is also the name traditionally given to the data from these types of devices.

The name you use will be included in the names of the detector's results, so to avoid having very long result names, you should try to keep movement detector names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Movement bout timeout

In brief

The *Movement bout timeout* field on the movement detector's settings page is used to specify the maximum period of time between beam breaks for which the breaks will still be considered to be part of the same bout (if this sounds confusing, you'll find full details below).

You must make an entry, in milliseconds - if you're not sure what to enter, then the default value of 500ms is usually good.

Details

The *Movement bout timeout* is used to determine the duration of bouts of movement, which are then summed to generate the result for the *Time moving* measure. To understand what this value is, we first need to think a little about how a beam-break movement detector works:

As the animal moves, it will break photobeams with each beam break occurring at some discrete point in time. So for example, we might see a break at time 1.2s, another at time 1.4s and another at time 1.5s. Probably, you would conclude from this data that the animal was moving continuously because, of course, it takes some time to move from one beam to the next. However if the data showed a break at time 1.2s and another at time 20.5s, you would probably conclude that the animal moved a little bit, then stopped, then moved a little bit more. So the time between the breaks indicates whether the animal is continuing to move, or stopping and starting - a short inter-break interval suggesting continuous movement. Therefore, to determine how long the animal is moving for, we need to choose some sort of threshold value - when the time between beam breaks is below this threshold, we can consider them to all be part of the same movement bout; when above, they can be considered to be separate bouts. This value is the *Movement bout timeout,* and can be set for each movement detector that you include in the protocol. The default value is 500ms (half a second).

One thing to be aware of is that ANY-maze will consider that the animal continues to move for this duration *after* a beam break. This makes sense if you consider what the threshold is actually doing - we're saying that two beam breaks separated by less than this time should be considered to all be part of the same movement, which implies that the animal didn't stop moving after the first beam break; but if we see no other beam break before this time, then we should consider that it has THEN stopped.

This has an important repercussion. Consider the following data: beam break at time 1.2s and another break at time 1.6s; how long was the animal moving for? You might say 1.6 - 1.2 = 0.4s. But that means you are considering that the animal stopped immediately after the break at 1.6s, whereas the rule described above says that the animals should be considered to have continued to move for a certain time AFTER the last break. Using the default *Movement bout timeout* value of 0.5s, we would actually say that, in this example, the animal moved for 0.9s. That's to say it started moving at 1.2s, was still moving at time 1.6s but stopped 0.5s after that (because we didn't see any further beam breaks).

Another way to look at this is to consider how long the animal was moving in a test where we have 2 beam breaks, one at time 10.4s and one at 30.5s? In this case the two breaks are clearly not part of the same movement, but the animal did move, so it must have spent at least some time moving, but how long? Using the default timeout value, ANY-maze would say that each movement lasted 0.5s, so the total time moving in the test would be reported as 1s.

Limits

You can enter any period between 10 milliseconds and 10 seconds.

Editing a movement detector

Introduction

You can edit absolutely everything related to a movement detector at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a movement detector
- Deleting a movement detector

Deleting a movement detector

Introduction

You can delete a movement detector at any time, whether before, during or after an experiment has been performed.

To delete a movement detector, simply select it in the protocol list and click the ❌ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting movement detectors.

See also:

- Deleting protocol elements
- Setting up a movement detector
- Editing a movement detector

Movement detector measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a movement detector. Note that these measures are ONLY available for the apparatus as whole.

- Count
- Time moving
- Latency to first movement

Count

Description	Reports the number of times beams in the movement detector's photobeam array were broken.
Calculation method	Counts breaks reported by the photobeam array. Note that ANY-maze does NOT count repeated breaks of the same beam - this is to avoid counting small oscillatory movements of the animal's body as *movement* (i.e. as in changes of location). For example, consider an animal that moves across the area covered by a photobeam array: we would see beam 1 broken, then beam 2, then beam 3, etc. Now consider an animal that is sitting in one place and grooming; its leg might keep breaking the same beam again and again; this would not be counted as movement.
Analysis in zones	Not available
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of counts that occurred during the time period.
Units	None
Notes	None

Time moving

Description	Reports the amount of time that the animal was moving.
Calculation method	Sums the durations of all the individual bouts of movement.
Analysis in zones	Not available

Analysis across time	This measure can be analysed across time. For any time period, the result is the duration of the movement within the time period.
Units	Seconds
Notes	Movements are detected by a beam being broken. This is necessarily an instantaneous event, but it is deemed to mean that the animal is moving and will continue to do so for a certain period (the movement detector's time-out). If another beam is broken within this period, then the animal is deemed to still be moving and so the movement bout extends to include this break. This continues until no break occurs within the timeout period, when the bout is then deemed to end. This mechanism is described in detail in this topic.

Latency to first beam break

Description	Reports the latency to the first beam being broken in the test - i.e. the first bout of movement.
Calculation method	The time up to the first beam break *during* the test is measured.
Analysis in zones	Not available
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	Beams that are already broken at the test start have no effect on this measure.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Output switch measures
- Pellet dispenser measures
- Syringe pump measures

- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Output switches

As the name implies, *output switches* are outputs that can switch things on and off. For example, an output switch might control a shocker or a lamp, or it might send a TTL signal to an electrophysiology recording device. The *Output switches* element of the protocol is used to control these types of outputs.

For more information about Output switches, refer to the topics below.

- An introduction to Output switches
- Setting up an Output switch
- Editing an Output switch
- Deleting an Output switch
- Output switch actions
- Output switch measures

An introduction to output switches

Introduction

Output switches allow ANY-maze to switch things on and off during an experiment. For example, you could use an output switch to turn on a 'cue light' in an operant chamber or to switch on (and off) a shocker in a fear conditioning experiment.

Connecting the switch

As you may know, a protocol can include I/O devices which are physical devices, such as the ANY-maze interface, that provide the electronics necessary to connect inputs and outputs to your PC.

An output switch is always part of some I/O device, and it is this device which actually connects to the thing being 'switched'. Let's take the ANY-maze interface as an example, and let's assume we want to control two lamps in our experiment. For simplicity, we'll assume the lamps already have a mechanical switch which you can manually use to turn them on and off. So, what we need to do is replace this manual switch with some sort of 'computer controlled switch'. Fortunately, there's an electronic device that does exactly that; it's called a relay, and the ANY-maze interface contains eight of them. So in our case, we would simply need to remove the manual switches from our lamp circuit and in their place connect two of the ANY-maze interface relays - voila! we would have two lamps that could be switched on and off by ANY-maze.

To set this up in our protocol, we would *have* to include the ANY-maze interface as an I/O device so that its output switches would then be available within our protocol. We would then need to include TWO output switches in the protocol, one for each of the lamps. It is the output switches which we would then be able to use to independently control the lamps during the tests.

Of course, this description glosses over the detail of exactly how we would connect the lamps to the ANY-maze interface, but that is described in the ANY-maze interface output switches topic.

The ANY-maze interface is not the only I/O device supported by ANY-maze - a full list can be found here.

Types of output switches

A relay output is one type of output switch, but there are others. For example, there are also Digital I/O (DIO) outputs - these provide a digital on/off signal which compatible electronics will understand. This type of signal is quite common on animal shockers (where it is often referred to as a TTL signal) and so by using an I/O device that supports this type of output, you could set up ANY-maze so it could control a shocker. A full description of the different types of output switches can be found here.

Using the switch

An *Output switch* protocol element represents a physical switch. When you set up the switch, you give it a name and tell ANY-maze whether it should be on or off at the start of a test, and which I/O device it is part of. But that's not very useful, if it just means that ANY-maze will turn the switch on (or off) at the start of every test - what you will probably want to do is to control the switch during the test; perhaps switching it on in response to something the animal does.

Controlling a switch during a test is the job of a procedure. Procedures can be set up to detect when something happens in a test (for example, the animal enters a certain zone), and to then take an action because of it - which includes turning an output switch on or off.

Summary

Output switches are a little more complicated than other protocol elements because they usually require various other elements to be set up too in order to make them useful:

- You need to set up the I/O device that the switch is physically part of
- You need to set up the switch itself
- You will usually want to set up at least one procedure to control the switch during the test

See also:

- An introduction to I/O devices
- Setting up an output switch
- Editing an output switch
- Deleting an output switch

Setting up an output switch

Introduction

For a general introduction to output switches, see An introduction to output switches.

To add an output switch to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New output switch* from the menu which appears.

The New output item *menu option is only included if the* protocol mode *includes input/output.*

The New output switch *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the output switch's name.
2. Specify the effect of activating the switch.
3. Set the switch activation options
4. Specify whether the physical port used by the switch is always the same, or whether it depends on the location of a movable zone.
5. Specify the physical port (or ports) used by the switch.

What next?

After completing these steps, you should consider whether you want to include any speakers in this protocol.

See also:

- Adding elements to a protocol
- Editing an output switch
- Deleting an output switch

Output switch name

In brief

Enter a name for the output switch in the *Switch name* field on the output switch's settings page. You must make an entry, but it can be anything you like.

Details

Output switch names should specify the thing that the switch is controlling, for example, 'Shocker' or 'Lamp'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Setting the switch activation options

In brief

You can use the option on the output switch's settings page to specify when the switch should be activated.

As well as these options, you can also control the switch using a procedure, something which is described in more detail below.

Specifying when the switch should be activated

Clearly, if you've included an output switch in your protocol it's because you want to turn something on and/or off during the tests - but when should this thing turn on? In some cases, the answer may simply be 'when the test starts'; but in other cases you may want to do more sophisticated things, perhaps turning it on and off as the animal moves into different zones.

Activating the switch at the start of the test

For the simple case of activating the switch at the start of the test, you can use the option here (in the protocol's Output switch settings). This doesn't mean you can't also use a procedure (perhaps to switch it off again); it just means that when the test starts, the switch will be activated.

Activating the switch based on more complex rules

In this case, you should use a procedure to activate (and/or deactivate) the switch. Full details are beyond the scope of this help topic, as it rather depends on what you want to do, but as an example, the procedure in figure 1 will wait until the animal enters the 'Target zone' and will then activate the 'Stimulus' switch output. More details can be found in the Introduction to procedures.

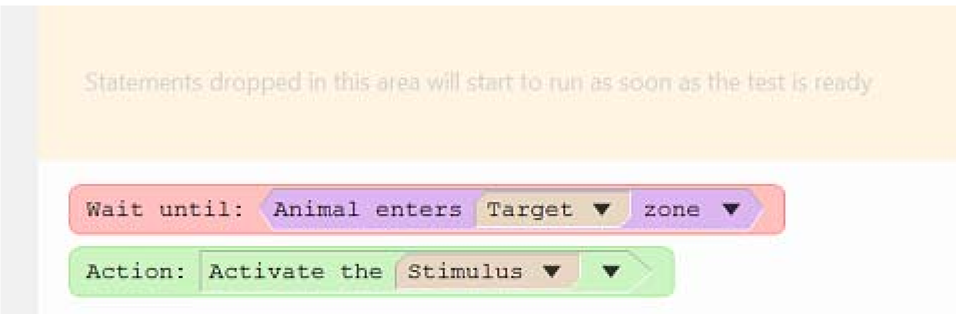


Figure 1. This simple procedure will activate the 'Stimulus' switch when the

animal enters the 'Target' zone.

Activating the switch BEFORE the test starts (legacy option)

In versions of ANY-maze prior to 5.20, there was an option to activate the switch *before* the test began (i.e. when the test was 'Ready'). If you used this option, and you open your experiment or protocol in version 5.20 (or later), then you will find that the option is still displayed and it will still work in the same way as it did previously. However, if you turn the option off, then it will disappear from the settings and you won't be able to turn it back on again.

In version 5.20 (and later), if you want to activate a switch before the test starts then the best method is to create a procedure to do this; as can be seen in figure 2, this is very simple.

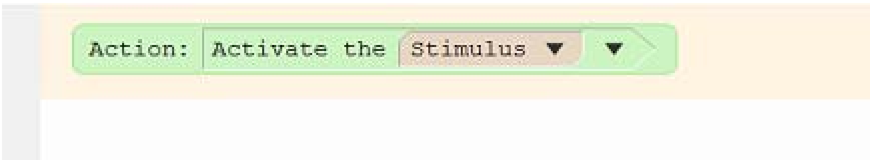


Figure 2. This one-statement procedure will activate the 'Stimulus' switch before the test starts (because statements in the orange-shaded area are performed as soon as the test is ready)

The key point here is that statements in the orange-shaded area at the top of the procedure editor are performed as soon as the test is *ready*. So, if you add an Action statement (to activate a switch output) to this area, then the switch will be activated *before* the test begins.

Preventing the switch being reset at the end of the test (legacy option)

When a test ends, ANY-maze will turn the switch off and reset it to the values entered on the switch's settings page. However, there are circumstances when you might not want this to occur, for example:

- You might be planning to run tests back-to-back, and whatever the state was at the end of one test, you want to continue in the next one. So if a light was on at the end of one test, you may want it to still be on at the start of the next one.
- You might have used a procedure to alter some aspects of a *pulse* output switch; for example, the procedure may have changed the pulse frequency during the test. If the switch is reset at the end of the test, then the frequency will be reset to whatever the standard frequency is (as defined in the switch's settings in the protocol), but what if you want to retain the frequency set by the procedure (so it's used in the next test)?

In versions of ANY-maze prior to version 6, there was an option to *not* reset the switch at the end of the test. However, this would not necessarily work in the way that you'd expect it to:

- The state would be retained from one *test* to the next, but not from one *trial* to the next.

For example, imagine that your test schedule is such that you will test animal 1 in its first trial, then animal 2 in its first trial and then animal 1 in its second trial, etc. What would happen is that after animal 1's first trial the state of the outputs (i.e. the open/closed state of the doors) would remain the same and so this would be the state that would apply to animal **2's** first trial... which is probably not what you want.

- Even if the above didn't apply (for example, in your schedule you might test animal 1 four times in a row), then if you exited ANY-maze after trial 2 and then restarted it again, the state of the outputs would be lost.

So, not resetting the outputs at the end of a test doesn't provide a way to achieve our goal, and therefore this option has been removed. If you have an older experiment file in which this option was selected, however, it will still appear in the settings for backward compatibility.

The best way to ensure that an output switch's state is retained from one trial to the next (even if you close the experiment, exit ANY-maze and turn off your computer between tests) is to use a procedure; this is described in detail here.

See also:

- An introduction to procedures

Specifying the effect of activating a switch

In brief

You should use buttons on the output switch's settings page to specify the effect of activating an output switch. For example, this might be to turn the switch ON, it might be to turn it OFF, or it might be to just *pulse* it ON for a short period.

Details

In general, the effect of activating a switch will be to turn it ON. However, there might be situations when you would like a switch activation to turn the switch OFF, or turn it ON but only for a short period.

Of course, there's a question here of what turning a switch ON actually does - in ANY-maze, this depends on the output device you're using; for example:

- For devices which use a relay for their outputs, then turning the output ON will activate the relay. For some devices, even this is ambiguous because some relays have a *common* line (COM) and two *outputs,* one of which is *normally open* (NO) and one of which is *normally closed* (NC). To make the relay behave as expected, you would simply connect the device between COM and NO - i.e. when the relay is activated, the device will be turned on.
- The GPIO output ports of the RTV-24 are *active high* - that's to say that activating the output will drive the output line to a TTL high level.
- The ANY-maze TTL cable outputs are *active low* - that's to say that activating the output will drive the output line to a TTL low level.

Having said all that, many I/O devices include configuration options which can be used to alter the 'active' state of their outputs.

Pulsing a switch on or off

As well as being able to turn an output ON or OFF, activating an output switch can also be set to turn the output ON (or OFF) for just a specified period of time. Effectively, this allows you to *pulse* an output, and this can be useful if you want to apply a short duration shock, flash a light, make a speaker beep, etc.

For example, imagine you might want to administer a half-second shock to an animal when a certain situation occurs. In this case, you would simply create an output switch whose action is to activate your shocker for 0.5s.

The value you enter can use any of the units h (hours), min (minutes), s (seconds) or ms (milliseconds). You can mix units, and also specify decimal values. So, for example, the period of one and a half minutes could be entered as '1.5min' or as '1min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 10 milliseconds to 1 hour.

Generating a pulse train

As the name implies, a *pulse train* is a series of pulses at an output. In this case, the pulses continue while the output is active, as shown in figure 1.

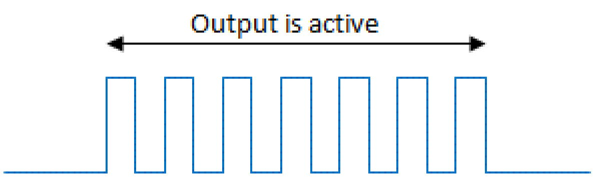


Figure 1. While the switch is active, it outputs a series of pulses.

There are two settings you must make if you set a switch to output a pulse train when active: the frequency and the duty cycle of the pulses. The frequency is simply the frequency of the pulses - for example, if you specify 50Hz then there will be 50 pulses per second, so each one will last 20ms. The duty cycle specifies how long within each pulse's period the output will be high and (by implication) how long it will be low. For example, if you specify a duty cycle of 25% then, in the case where the pulse lasts 20ms, the pulse will be high for 25% of the time and low for 75%, so it will be high for 5ms and low for 15ms - see figure 2.

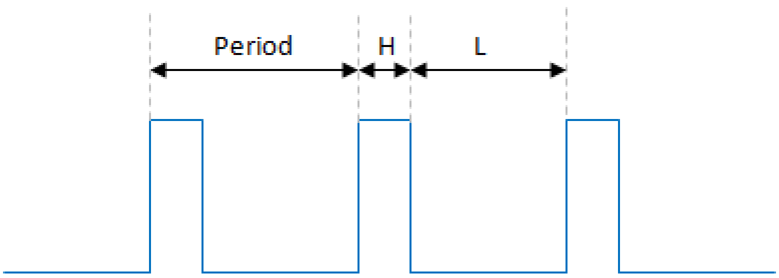


Figure 2. The frequency you specify defines the period of each pulse (Period = 1/Frequency). The duty cycle defines the proportion of the period that the output

is high (H) and the proportion for which it is low (L). In the above figure the duty cycle is 25%, so H is 25% of period and L is 75%.

There are some limitations regarding frequency and duty cycle:

- The highest frequency you can specify is 100Hz, i.e. the shortest period of each pulse is 10ms.
- The shortest high or low pulse duration is 5ms. For example, setting a duty cycle of 10% for a 50Hz pulse train would, in theory, mean a period of 20ms with the high pulse lasting 10% of this, so 2ms. However, because of the 5ms limitation this would *automatically* be adjusted by the system to be 5ms, with the low pulse lasting 15ms to ensure the frequency is correct.

Note that these two limitations mean that a 100Hz pulse train will always have a duty cycle of 50%, no matter what you specify, because the high and low periods will be fixed at the minimum 5ms duration.

You can optionally specify that the pulse train should last a certain duration; for example, entering 5s will cause the switch to output a pulse train of the specified frequency and duty cycle for 5 seconds, after which the output will automatically become inactive. If you don't specify a duration, then the pulse train will continue until such time as it is explicitly switched off - usually by an action.

Note that if you specify that the default state of a switch is 'On' and the switch is set up to output a pulse train, then the pulse train will begin as soon as a test is *Ready* to be performed.

Accuracy of pulse trains

The accuracy of the frequency and duty cycle of a pulse train will depend to some extent on the physical device that is being switched. For example, if you are trying to switch mechanical relays, then they will probably not support a frequency of more than a few Hertz (indeed, it would be best not to try to switch a mechanical relay at a high frequency). Solid state relays will switch more quickly and logic outputs, such as the TTL outputs on an ANY-maze interface, will switch very quickly.

However, accuracy is not only defined by the speed with which the hardware reacts but also by how the software works. Trying to generate millisecond accurate timing under the Windows operating system is not easy, as Windows does not guarantee that an application will always be running; Windows switches between the applications very quickly, so it may be that at the exact time when a pulse should be switched on or off, ANY-maze might not actually be running.

Using an ANY-maze interface, the accuracy of a 50Hz pulse train with a 25% duty cycle was measured as:

- Frequency: 49.9Hz ±1Hz
- On duration: 5.1ms ±1.5ms
- Off duration: 14.9ms ±1.5ms

Specifying whether the physical port used by an output switch is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port used by a switch will always be the same - it's simply the port on the interface device to which the device being switched actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Y-maze, you may have a lamp (connected to a switch) mounted at the end of each arm of the maze. Now you want to create an element called 'Goal arm light' (let's imagine that the light comes on if the animal spends a certain amount of time in this arm), but you have set the 'Goal arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests. Thus the 'Goal arm light' will be the lamp in whichever arm happens to be the goal arm in a particular test, which necessarily means it won't always be the same physical lamp (and therefore, the 'Goal arm light' won't always be connected to the same physical port).

Details

If you do have an output switch whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the switch element's page of the protocol.

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing an output switch

Introduction

You can edit absolutely everything related to an output switch at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up an output switch
- Deleting an output switch

Deleting an output switch

Introduction

You can delete an output switch at any time, whether before, during or after an experiment has been performed.

To delete an output switch, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting output switches.

See also:

- Deleting protocol elements
- Setting up an output switch
- Editing an output switch

Output switch actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect an output switch - these are detailed below.

Actions

Activate the switch	Sets the switch to its active state. If a switch is already active, this action has no effect. If the switch is set to 'pulse' on or off, then it will automatically deactivate after the appropriate time. If the switch is set to output a pulse train, this will start the pulses.
Deactivate the switch	Sets the switch to its inactive state. If the switch is already inactive, this action has no effect. If the switch is set to 'pulse' on or off but the pulse duration has not yet passed, then this action *will* deactivate the switch (and cancel the pulse timing). If the switch is set to output a pulse train, this will end the pulses immediately, even if this is in the middle of a pulse.
Set switch frequency	Changes the frequency of the pulses of a *pulse train* to the value specified as the action's parameter. The parameter is in Hertz. If the switch is not set to output a pulse train, this action will have no effect. This action will **not** activate (or deactivate) the switch. If the switch is active, then the pulse train frequency will change on-the-fly and the change will be asynchronous, i.e. it will **not** occur at the end of a cycle.
Set duty cycle	Changes the duty cycle of a *pulse train* to the value specified as the action's parameter. The parameter is in percent and represents the amount of a single cycle for which the output will be ON. If the switch is not set to output a pulse train, this action will have no effect. This action will **not** activate (or deactivate) the switch. If the switch is active, then the duty cycle will change on-the-fly and the change will be asynchronous, i.e. it will **not** occur at the end of a cycle.
Set switch duration	Changes the duration for which the switch will be active to the value specified as the action's parameter. The duration can be any time value, and can include units of ms (milliseconds), s (seconds), min or m (minutes), h (hours) and d (days); if no units are specified

seconds are assumed. If the switch is not set to pulse on, pulse off or output a pulse train, this action will have no effect. This action notes the new duration, but it will only be used the **next time** the switch is activated. This means that if the switch is active when the duration is set, the switch will continue to use the *old* duration until that duration passes (or the switch is deactivated).

Output switch measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for an Output switch:

- Number of activations
- Time active
- Latency to first activation
- Latency to first deactivation
- Longest activation
- Shortest activation
- Average activation duration
- Frequency of activations

Each measure is available for the apparatus as whole (i.e. irrespective of where the animal was when the output switch activation occurred), and also for each defined zone.

Number of activations

Description	Reports the number of times the switch was activated.
Calculation method	Counts the number of activations.
Analysis in zones	Counts the number of activations when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of activations during the time period.
Units	None
Notes	None

Time active

Description	Reports the total amount of time that the switch was active.
Calculation method	Sums the amount of time for which the switch was active.

Analysis in zones	Sums the amount of time for which the switch was active while the animal was in the zone. For a particular zone, it's possible for the *Time active* to be non-zero while the Number of activations is zero. This can occur if the animal enters the zone when the switch is active. In this case, the time the switch is active will be registered but the activation itself won't be.
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of time that the switch was active during the period. For a particular time period, it's possible for the *Time active* during the period to be non-zero while the Number of activations for the period is zero. This can occur if the switch is already active at the start of the period. In this case, the time the switch is active will be registered but the activation itself won't be.
Units	Seconds
Notes	None

Latency to first activation

Description	Reports the amount of time that elapsed in the test before the switch was activated for the first time.
Calculation method	The value of the test clock at the first activation of the switch.
Analysis in zones	The value of the test clock at the first activation of the switch that occurred while the animal was within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Latency to first deactivation

Description	Reports the amount of time that elapsed in the test before the switch was deactivated for the first time.
Calculation method	The value of the test clock at the first switch deactivation.
Analysis in zones	The value of the test clock at the first switch deactivation that occurred while the animal was within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Longest activation

Description	Reports the duration of the longest period for which the switch was continuously active.
Calculation method	The duration of each switch activation is calculated and the largest value is found.
Analysis in zones	The longest period for which the switch was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest period for which the switch was continuously active during the period.
Units	Seconds
Notes	None

Shortest activation

Description	Reports the duration of the shortest period for which the switch was continuously active.
Calculation method	The duration of each switch activation is calculated and the smallest value is found.
Analysis in zones	The shortest period for which the switch was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest period for which the switch was continuously active during the period.
Units	Seconds
Notes	None

Average activation duration

Description	Reports the average duration for which the switch was active.
Calculation method	Calculated by dividing the Time active by the Number of activations.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Frequency of activations

Description	Reports the frequency with which the switch was activated.
Calculation method	Calculated by dividing the Number of activations by the *Test duration.*
Analysis in zones	Calculated by dividing the Number of activations in the zone by the *Total time in the zone.*
Analysis across time	This measure can be analysed across time. For any time period, the result is the Number of activations which occurred during the period divided by the period's duration.
Units	Hertz
Notes	None

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Pellet dispenser measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Speakers

Speakers can be used in ANY-maze to play white noise, tones, or most sound files at any volume and for any duration.

For more information about speakers, refer to the topics below.

- An introduction to speakers
- Setting up a speaker
- Editing a speaker
- Deleting a speaker
- Speaker actions

An introduction to speakers

Introduction

Using speakers connected to an I/O device or soundcard, ANY-maze can play various sounds, including white noise, tones, and (for some devices) sound files. Additionally, both the volume and duration of playback can be controlled.

Altering playback during a test

In a simple scenario, ANY-maze can be set up to play a sound throughout a test, but its real power comes from the fact that you can control audio playback dynamically as a test progresses. What's more, as this ability is based on procedures, playback can change depending on what the animal is actually doing.

For example, you could set up a speaker to play a different pitch tone when the animal is in each zone of your apparatus and you could alter the volume of the tone so it becomes louder the longer the animal remains in the zone. Not very useful perhaps, but it serves to illustrate the flexibility available.

Connecting a speaker

For a speaker to be used by ANY-maze, it must be connected to some sort of I/O device such as the ANY-maze interface (AMi). This means that you not only need to include a *Speaker* in your protocol, but you must also include the I/O device the speaker is connected to.

Note that you can also use any Soundcards connected to your computer as I/O devices, giving you access to their speakers.

To allow you to use a soundcard as an I/O device, you'll need to enable it using the Set up devices *option on the* ***I/O page****.*

See also:

- An introduction to I/O devices
- Setting up a speaker
- Editing a speaker
- Deleting a speaker
- Using soundcards as I/O audio devices

ANY-maze help topic H0260

Setting up a speaker

Introduction

For a general introduction to speakers, see An introduction to speakers.

To add a speaker to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New speaker* from the menu which appears.

The New output item *menu option is only included if the* protocol mode *includes input/output.*

The New speaker *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the speaker's name
2. Specify what sound the speaker should play
3. Specify the volume of the sound
4. Specify how long the sound should play for
5. Specify whether the speaker should start playing at the start of the test
6. Specify whether the speaker should be reset at the end of the test (legacy option)
7. Specify whether the physical port used by the speaker is always the same or whether it depends on the location of a movable zone
8. Specify the physical port used by the speaker

What next?

After completing these steps, you should consider whether you want to include any shockers in this protocol.

See also:

- Adding elements to a protocol
- Editing a speaker
- Deleting a speaker

ANY-maze help topic H0261

Speaker name

In brief

Enter a name for the speaker in the *Speaker name* field on the speaker's settings page. You must make an entry, but it can be anything you like.

Details

Speaker names should specify what the speaker is intended to do, so for example it might be called 'Audio cue' or 'Background noise'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying what the speaker should play

In brief

Speakers can play white noise, tones (a sound at a fixed frequency), and most sound files. You can specify what the speaker should play by choosing an option from the drop-down list on the speaker's settings page.

Details

White noise	White noise is sound which contains random frequencies of equal power - a good example of something close to white noise is the sound a TV makes when it's not tuned to any station. White noise is useful because it tends to mask other sounds, thus creating a more controlled environment. For example, if your lab has an air conditioning unit that switches on and off automatically, then it might be on (and making sound) during some of your tests and off during others - thus there would be an environmental difference between the tests. By playing white noise during your experiment, you could mask the sound made by the air conditioner so that the environment would be more closely equal in all the tests.
Tones	A *tone* in ANY-maze is a sound of a fixed frequency (the signal is simply a fixed frequency sine wave). The list of sounds that can be played by a speaker includes a number of 'standard' frequencies, ranging from 500Hz up to 4500Hz, which are useful for creating cues. You can also create a tone at a precise frequency of your choosing, by selecting the option to *Specify a tone...* This will cause the Specify an audio tone window to open, where you can specify a tone, either as a certain frequency (in Hertz), or as a musical note.
Sound files	ANY-maze can play most sound files that can be played in Windows Media Player - this includes Wave (.wav) files, MP3 files and Windows media audio (.wma) files, although *not* MIDI files. Note that the .wma format supports digital rights management (DRM), and ANY-maze cannot play DRM protected files. If you want to play a file that's currently on a CD, then you will need to 'rip' the CD first and then play the resulting sound file. To play a sound file, you should choose the option in the sounds list to *Browse for a sound file...* - this will cause the Choose a sound file to play

window to be displayed, where you can choose the file you want to play. Note that ANY-maze will usually convert the file you choose to a format it can play, and the converted file will be placed in the ANY-maze\Sounds folder. The list of sounds shown on the speaker's settings page automatically includes all the files in this folder, so having browsed to a file once, you will find it is automatically included in the sounds list thereafter.

Playing a random sound You also have the option to play a random sound. If you select this option in the sounds list, the Play a random sound window will open. There you can either choose to play a sound file, chosen randomly from a selection you make, or to play a random tone, again chosen randomly from a series of tones you specify.

Sound quality

Different I/O devices will provide different audio quality. For example, sounds played through the ANY-maze interface (AMi) are not very high quality, whereas sounds played by the ANY-maze cage are significantly better.

Specifically, AMi uses 8-bit audio at 11,025Hz (for comparison, CDs use 16-bit audio at 44,100Hz), which means that:

- White noise sounds a little 'compressed' because AMi can't play frequencies above 5500Hz (in fact, any frequency above 4000Hz is not reproduced very well).
- Tones, especially those below 3000Hz, play well, but those above this frequency sound ragged.
- Sound files usually need to be resampled - this tends to add some noise, which manifests as hiss or crackle.

Nevertheless, despite these shortcomings, the sound quality is acceptable for the intended use.

It is worth mentioning that the quality of the speakers you use will also make a substantial difference to the perceived quality of the sound, particularly at lower frequencies. Most small, cheap speakers are unable to play sounds below about 300Hz, so trying to play a tone at 200Hz will create no audible output or, at best, a low volume vibration.

A note about MIDI files

As mentioned above, ANY-maze cannot play MIDI files, nor can it convert them to a format it can play. However, there are commercially available products that can convert MIDI files to wave (*.wav) files.

In fact, playing converted MIDI files provides the best sound quality on the ANY-maze interface, as the wave file can be *generated* as an 8-bit, mono, 11025Hz WAV file, thus avoiding the noise problems associated with resampling files that were originally recorded at higher frequencies and/or resolutions.

Specifying an audio tone

Introduction

ANY-maze can play audio tones in the range 65Hz to 32000Hz. You can use the *Audio tone* window to specify the exact frequency of a tone you wish to play, either by entering the frequency itself or by specifying the tone as a musical note.

Although ANY-maze can play tones from 65Hz to 32000Hz, your audio device is unlikely to have such a wide range. For example, the ANY-maze interface can only play tones up to 4500Hz, so if you specify a value above this, it will be played at 4500Hz.

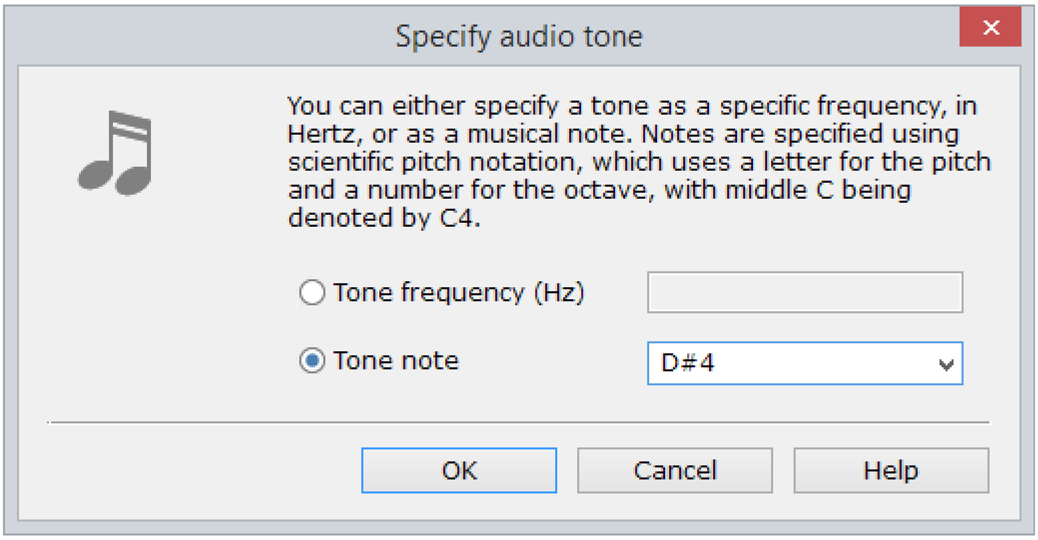


Figure 1. The 'Specify audio tone' window.

Specifying a tone as a frequency

To specify a tone of a certain frequency, simply enter the frequency into the frequency field. The value you enter will automatically be *clipped* to the range 65-32000Hz, which is the range of frequencies that ANY-maze can play.

Specifying a tone as a note

As an alternative to specifying the actual frequency of a tone, you can specify it as a note instead - for example, you could specify the tone as *Middle C*, which has a frequency of 261.63Hz

ANY-maze uses *Scientific pitch notation* to name notes. This system uses a letter (and sometimes the *accidental* '#', meaning sharp) to specify the note's pitch and a number to specify the note's octave, where *middle C* is denoted by C4. Thus, for example, the note G#3 is *G sharp* in the octave below middle C. For further details, refer to Scientific pitch notation.

ANY-maze can play all notes from C2 to B7.

Browsing for a sound file

Introduction

ANY-maze can play most sound files that can also be played in Windows Media Player - this includes Wave (.wav) files, MP3 files and Windows media audio (.wma) files, although it cannot play MIDI files (see note below). Also note that the .wma format supports digital rights management (DRM), and ANY-maze cannot play files that use DRM.

File conversion

ANY-maze will determine the format used by your audio device and, if necessary, convert the sound file you choose to that format. For example, the ANY-maze interface uses an audio format of 8-bits 11025Hz. This is a standard, low quality format, but most sound files use higher frequencies and are generally 16-bit, so to play them ANY-maze has to convert the files to this format. This conversion occurs when you select a file in this window. The converted file is placed in the ANY-maze\Sounds folder. Note that conversion reduces the quality of the sound and generally introduces some noise in the form of hiss.

MIDI files

As mentioned above, ANY-maze cannot play MIDI files, nor can it convert them to a format it can play - however, there are commercially available products that can convert MIDI files to wave (*.wav) files.

In fact, playing converted MIDI files provides the best sound quality on the ANY-maze interface, as the wave file can be *generated* as an 8-bit, mono, 11025Hz WAV file, thus avoiding the noise problems (mentioned above) associated with converting files that were originally recorded at higher frequencies and/or resolutions.

Specifying random sounds to be played

Introduction

There are two types of random sound that ANY-maze can play - sound files and tones. In both cases, you can specify a selection of sounds from which the system will choose one, at random, that it will play.

When sounds are changed randomly during a test, using an Action, the system will ensure that the sound does always change, even though this implies that the choice is not entirely random.

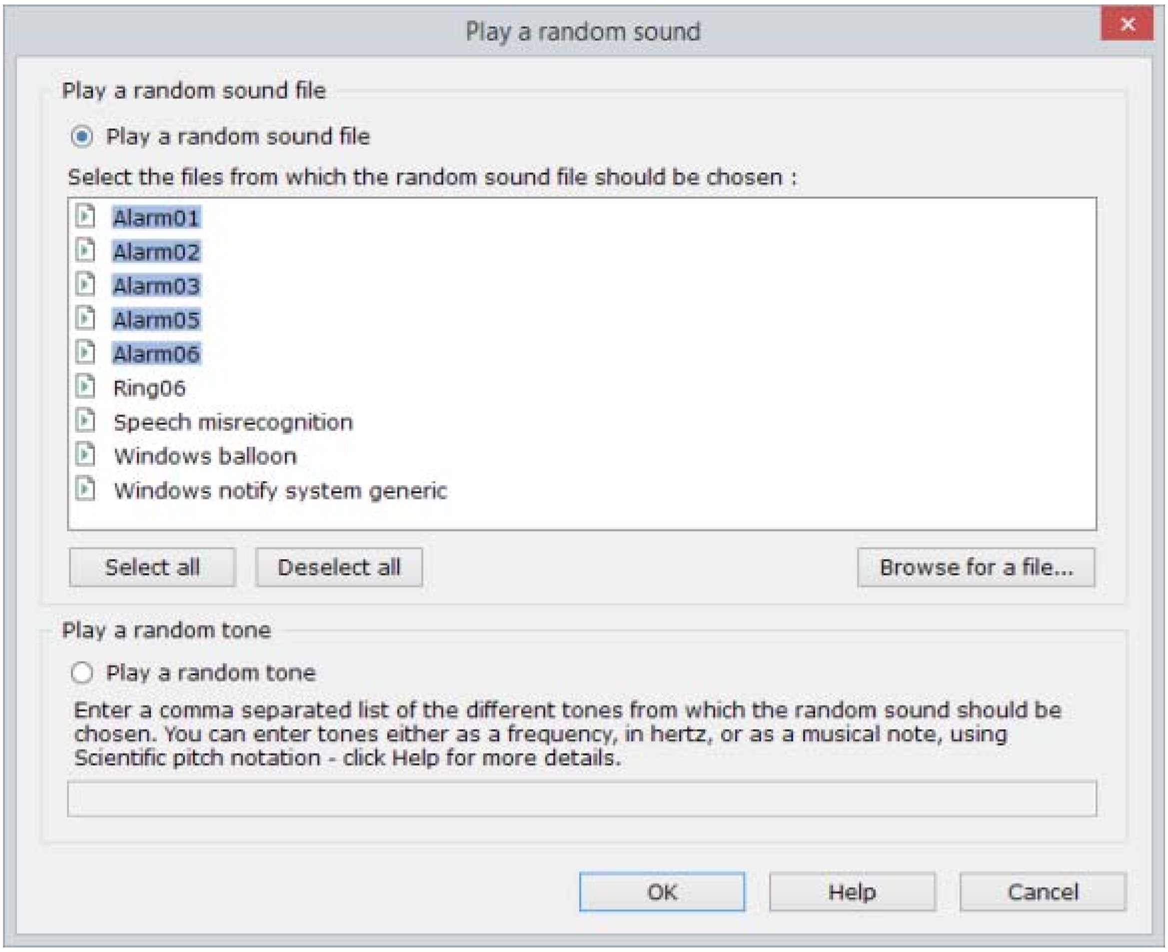


Figure 1. The Play a random sound window.

Playing a random sound file

To play a random sound file, first select the button *Play a random sound file* and then select all the possible sound files which could be played - the system will choose one from those you select.

The list of sound files will show all the files in the ANY-maze\Sounds folder. If you want to add more files to the list, just select the *Browse for a file* button.

Playing a random tone

To play a random tone, first select the button 'Play a random tone' and then enter the different possible tones as a comma-separated list - the system will choose one from those you enter. You can specify tones either as a frequency (in Hertz), or as a note, using Scientific pitch notation. For example, you could enter:

1000, 1500, 2000, C4, C5, G#6

This would mean that the system would choose randomly from:

- A tone of 1000Hz
- A tone of 1500Hz
- A tone of 2000Hz
- The note C in the 4th octave (this is middle C)
- The note C in the 5th octave (one octave above middle C)
- The note G sharp in the 6th octave (2 octaves above middle C)

You can enter any frequency from 65Hz through to 32000Hz, and any note from C2 to B7. If you enter something ANY-maze can't play (or doesn't understand), then a message will be displayed explaining what the problem is.

Specifying the volume of the sound

In brief

You can specify the volume of the sound played by a speaker by dragging the volume slider on the speaker's settings page.

Details

Volume is shown as a percentage scale, with 0% being silent (the speaker will actually be muted) and 100% being the maximum volume available.

The actual volume of a sound depends on a number of factors other than the *volume setting*. These include the volume of the source signal (for example, a sound file can be recorded with a low or high volume) and the speaker being used.

It's also important to understand that the volume of a tone played by a speaker depends on the frequency of the tone, because speakers (especially cheap ones) don't have a flat frequency response. This means, for example, that a 500Hz tone played at 80% volume may not sound as loud as a 3000Hz tone played at the same volume.

Volume in decibels

You may be wondering why ANY-maze uses a percentage as the volume scale rather than using decibels; after all, in papers, it's common for researchers to report the volume of, for example, a cue tone, in decibels, making it easy for others to replicate their conditions.

The answer is that ANY-maze simply can't know what the final volume (in decibels) will be.

- In most cases, ANY-maze is simply requesting the I/O device to play the sound and it doesn't know how loud the device will play it.
- The final volume is very dependent on the speaker used. So if ANY-maze was to play two identical tones at the same volume through two identical audio devices, the final volume would almost certainly be different if two *different* speakers were used.
- When playing a sound file, the volume is very dependent on the recording.
- The final volume will depend on the acoustics of the location where the sound is being played.

Of course, that doesn't help you if you want to play a 1KHz tone at 80 decibels. In this case, the answer is to use a 'Sound level meter', which will allow you to measure the volume at the location where the animal will be. So then you could install the meter and adjust the volume in ANY-maze until you get the level you want, 80dB in our example. This might occur with the volume at 93%, for example. But remember, if you change the speaker, or you alter the apparatus (which might change it's acoustics) then 93% will probably no longer equate to 80dB and you'll need to use the sound level meter again to check the volume.

Sound level meters are quite easy to find (a search of Google will almost certainly list lots of them) and not too expensive, but beware of very cheap devices - they'll probably be quite inaccurate.

Specifying how long a sound should play for

In brief

You can specify how long a speaker should play a sound for by selecting one of the three options:

- Play for a certain duration
- Play continuously
- Play until the end of the sound file

Details

Tones and white noise can be played for a specific duration or can simply be played continuously.

Files can be played once, repeated continuously or played for a specific duration. Note that if you specify that a sound file should be played for a duration that is longer than the file, then the file will repeat until this duration has elapsed.

When specifying a duration to play the sound for, the value you enter can use any of the units d (days), h (hours), min (minutes), s (seconds) or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of one and a half minutes could be entered as '1.5min' or as '1min 30s'. If you don't specify any units then seconds will be assumed.

You can enter any value from 1 millisecond to 1 day (although it is unlikely that the hardware will be able to accurately play a sound for just 1 millisecond).

Specifying whether the speaker should be playing at the start and/or end of the test

In brief

You can use the buttons on the speaker's settings page to specify whether the speaker should start playing at the moment the test starts, or whether it should be silent when the test starts.

Details

If you want the speaker to start playing when the test starts, then select the *Start playing when the test starts* option.

If you want the speaker to be silent at the start of the test and to only start playing *during* the test, then you should not select the above option, but rather use a procedure to switch the speaker on at the appropriate moment.

Note that you can also use procedures to alter what the speaker is playing and to change its volume, while the test is running.

Starting the speaker playing BEFORE the test starts (legacy option)

In versions of ANY-maze prior to 5.20, there was an option to start the speaker playing *before* the test began (i.e. when the test was 'Ready'). If you used this option, and you open your experiment or protocol in version 5.20 (or later), then you will find that the option is still displayed and it will still work in the same way as it did previously. However, if you turn the option off, then it will disappear from the settings and you won't be able to turn it back on again.

In version 5.20 (and later), if you want to start a speaker playing before the test starts then the best method is to create a procedure to do this; as can be seen in figure 2, this is very simple.

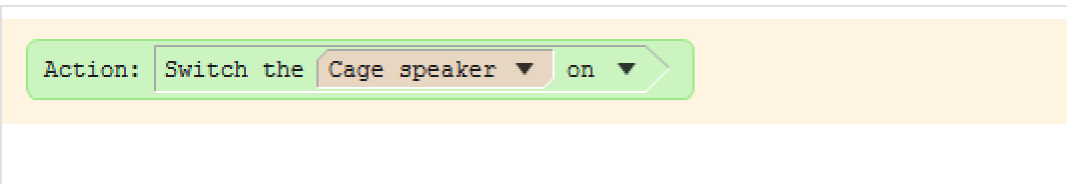


Figure 2. This one-statement procedure will start the 'Cage speaker' playing before the test starts (because statements in the orange-shaded area are performed as soon as the test is ready)

The key point here is that statements in the orange-shaded area at the top of the procedure editor are performed as soon as the test is *ready*. So, if you add an Action statement (to start a speaker playing) to this area, then the speaker will start playing *before* the test begins.

Specifying whether the speaker should be reset at the end of the test (legacy option)

In brief

At the end of a test, ANY-maze will stop a speaker playing, and will reset it to the frequency and volume specified on the speaker's settings page. However, there may be circunstances under which you would like what the speaker plays to roll over from one test to the next.

For example, during a test you might use a procedure to dynamically alter what the speaker is playing depending on what the animal is doing, thus at the end of the test the speaker may be playing something different to the default options specified in the protocol. Now, you may want to retain whatever it is the speaker is now playing so that this continues to play in the next test - in this case, you *won't* want to reset the speaker at the end of the test.

Prior to ANY-maze version 6, the speaker's settings page included an option to specify that the speaker should *not* be reset at the end of a test. This option has now been removed, although if you have an experiment file created with an older version of ANY-maze with this option selected, then it will still remain for backward compatibility.

However, using the option to *not* reset the switch at the end of the test would not necessarily work in the way that you'd expect it to:

- The state would be retained from one *test* to the next, but not from one *trial* to the next. For example, imagine that your test schedule is such that you will test animal 1 in its first trial, then animal 2 in its first trial and then animal 1 in its second trial, etc. What would happen is that after animal 1's first trial the state of the speakers would remain the same, and so this would be the state that would apply to animal **2's** first trial... which is probably not what you want.
- Even if the above didn't apply (for example, in your schedule you might test animal 1 four times in a row), then if you exited ANY-maze after trial 2 and then restarted it again, the state of the speakers would be lost.

So for this reason, this option has been removed, and the best way to ensure that a speaker's state is retained from one trial to the next is to use a procedure; this is described in detail here.

Specifying whether the physical port used by a speaker is always the same

In brief

In the majority of situations, the physical port used by a speaker will always be the same - it's simply the port on the interface device to which the speaker actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Y-maze, you may have a speaker mounted on the end of each arm of the maze.

Now you want to create an element called 'Aversive arm speaker' (let's imagine that the animal has to leave the arm when the speaker plays a tone), but you have set the 'Aversive arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests.

Thus the 'Aversive arm speaker' will be the speaker in whichever arm happens to be the aversive arm in a particular test, which necessarily means it won't always be the same physical speaker, and therefore the 'Aversive arm speaker' won't always be connected to the same physical port.

Details

If you do have a speaker whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the speaker element's page of the protocol. (Note that this list is only displayed when the protocol includes one or more movable zones).

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a speaker

Introduction

You can edit absolutely everything related to a speaker at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a speaker
- Deleting a speaker

Deleting a speaker

Introduction

You can delete a speaker at any time, whether before, during or after an experiment has been performed.

To delete a speaker, simply select it in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting speakers.

See also:

- Deleting protocol elements
- Setting up a speaker
- Editing a speaker

Speaker actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a speaker - these are detailed below.

Actions

Switch the speaker on	Start the speaker playing. The speaker will play the current source at the current volume for the current duration (if set). If the speaker is already playing, this action will have no effect.
Switch the speaker off	Stops the speaker playing. If the speaker was set to play for a set duration, and that duration has not passed yet, then the speaker will still stop playing and the duration timer will be reset. If the speaker is not playing, this action will have no effect.
Set speaker volume	Changes the speaker volume to the volume specified in the action's parameter. The volume is set as a percentage (see this topic for full details about specifying speaker volume). If the speaker is playing, the volume will change immediately, otherwise the new volume will be used next time the speaker starts playing.
Make speaker source a tone	Changes the speaker source to a tone of the frequency specified in the action's parameter. The frequency is specified in hertz. If the speaker is already playing a tone, this action will change the tone's frequency immediately. If the speaker is playing white noise or a sound file, then the speaker will change to play the specified tone. If the speaker is not playing, then the next time it starts playing, it will play a tone of the specified frequency.
Make speaker source white noise	Changes the speaker source to white noise. If the speaker is already playing white noise this action has no effect. If the speaker is playing a tone or a sound file, it will change to play white noise. If the speaker is not playing then, the next time it starts playing, it will play white noise.
Make speaker source a sound file (played repeatedly)	Changes the speaker source to the sound file specified in the action's parameter. The parameter is the full path and file name of the file and can be any valid Windows file path. If the speaker is playing a sound file, it will change to play the file specified in the

action's parameter. If the same file is specified as the one which is playing, the file will restart from the beginning. If the speaker is playing a tone or white noise, it will change to play the sound file. If the speaker is not playing, then the next time it starts playing, it will play the sound file. The file will play repeatedly.

Make speaker source a sound file (played once)

Changes the speaker source to the sound file specified in the action's parameter. The parameter is the full path and file name of the file and can be any valid Windows file path. If the speaker is playing a sound file, it will change to play the file specified in the action's parameter. If the same file is specified as the one which is playing, the file will restart from the beginning. If the speaker is playing a tone or white noise, it will change to play the sound file. If the speaker is not playing, then the next time it starts playing, it will play the sound file. The file will play just once, i.e. when playback reaches the end of the file, the speaker will switch off.

Set speaker playback duration

Sets the duration for which the speaker will play to the duration specified in the action's parameter. The duration can be any time value and can include units of ms (milliseconds), s (seconds), min or m (minutes), h (hours) or d (days); if no units are specified seconds are assumed. If the parameter is 0, the speaker will have no duration and will play continuously until explicitly switched off. If the speaker is playing, the playback duration will be reset to the new duration starting from the moment the duration is changed. For example, if a speaker is playing a tone for 10s, and 8s after it starts playing this action is used to set the duration to 5s, the speaker will play for **another** 5s from now. If a speaker is playing a sound file just once, and the playback duration is longer than the sound file duration, the speaker will stop playing at the end of the file, i.e. the duration will have no effect. If the speaker is not playing when this action is performed, then the speaker will play for the specified duration the next time it starts playing.

Shockers

As the name implies, shockers are used to administer a shock to an animal. This will usually be a foot shock being used as an aversive stimuli, but it doesn't have to be.

For more information about shockers, refer to the topics below.

- An introduction to shockers
- Setting up a shocker
- Editing a shocker
- Deleting a shocker
- Shocker actions

An introduction to shockers

Introduction

Shockers are usually stand-alone devices, such as the San Diego Instruments shocker, but they are sometimes built into another device, such as the shocker built into the Ugo Basile Fear Conditioning Box. As far as ANY-maze is concerned, it doesn't actually matter which type of shocker you have, as they are controlled in the same way.

An important characteristic of shockers is the fact they can be controlled and *adjusted* by procedures during a test - for example, you could alter the shock current depending on where the animal is in the apparatus.

Connecting a shocker

How you connect a shocker will depend on the device itself, but they typically simply connect via USB.

As with all inputs and outputs supported by ANY-maze, you must add to the protocol the I/O device the shocker is part of, before you add the shocker output itself. For example, if you wish to use the shocker in the Ugo Basile Fear Conditioning Box, then you will first need to add the FC box as an I/O device and then you will be able to add the shocker output that it contains (the box contains other outputs too, such as a sound output and a 'cue' light).

Using a shocker

When you add a shocker to your protocol, you will be able to specify the duration and intensity (i.e. current) of the shock - but this won't actually cause a shock to be delivered. To do that, you will need to use a procedure, which will determine when in the test a shock should be administered. In fact, a procedure can not only cause a shock to occur, but it can also alter its intensity and/or duration. Thus, for example, you could administer different intensity shocks as the test progresses.

See also:

- An introduction to I/O devices
- An introduction to procedures
- Setting up a shocker
- Editing a shocker
- Deleting a shocker
- Shocker actions

Setting up a shocker

Introduction

For a general introduction to shockers, see An introduction to shockers.

To add a shocker to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New shocker* from the menu which appears.

The New output item menu option is only included if the protocol mode *includes input/output.*

The New shocker option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the shocker's name
2. Specify the default duration that a shock should last
3. Specify the default intensity (current) of the shock
4. Specify the absolute maximum shock duration and intensity
5. Specify whether the physical port used by the shocker is always the same, or whether it depends on the location of a movable zone
6. Specify the shocker's input/output port

What next?

After completing these steps, you should consider whether you want to include any Pellet dispensers in this protocol.

See also:

- Adding elements to a protocol
- Editing a shocker
- Deleting a shocker

ANY-maze help topic H0277

Shocker name

In brief

Enter a name for the shocker in the *Shocker name* field on the shocker's settings page. You must make an entry, but it can be anything you like.

Details

It's quite rare to have more than one shocker in an experiment, so usually you will just call the shocker something like 'Shock'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying a shocker's default duration

In brief

You should use the *Shock duration* field on the shocker's settings page to specify the *default* duration of the shock that the shocker will administer.

Details

When a shocker is activated by a procedure, it will administer a shock of the duration you specify here. So, for example, if you are setting up a Fear Conditioning protocol in which animals will receive 0.5s shocks, then this is the value that you will enter here.

Note, however, that procedures can also alter the shock duration to some other value, for example 1s. This is why the duration you enter in this field is the *default* duration; it is used by default when the shocker is activated, but it can be changed.

Limits

You can enter any value you like from 0.05s to 10.0s. You can specify the duration using units of s (for seconds) or ms (for milliseconds) and you can enter decimal values. For example, 200ms and 0.2s are both valid entries, and actually specify the same duration.

Specifying a shocker's default intensity

You are advised to read the Important Information at the end of this topic.

In brief

You should use the *Shock intensity* field on the shocker's settings page to specify the *default* intensity (current) of the shock that the shocker will administer.

Details

When a shocker is activated by a procedure, it will administer a shock of the intensity you specify here. So, for example, if you are setting up a Fear Conditioning protocol in which animals will receive a shock of 1mA, this is the value that you will enter here.

Note, however, that procedures can also alter the shock intensity to some other value, for example 1.5mA. This is why the intensity that you enter in this field is the *default* intensity; it is used by default when the shocker is activated, but it can be changed.

Limits

You can enter any value you like from 0.01mA to 10mA. You don't need to enter the units (as the value must be in milliamps) and you can enter decimal values. For example 0.01 and 3.23 are both valid entries.

IMPORTANT INFORMATION

The shocker you are using may not be able to administer the level of shock you specify. For example, the San Diego Instruments shocker can administer a maximum shock of 5mA. If you specify an intensity that is above the shocker's maximum, then the maximum will be used.

Also note that the shock level actually administered will be rounded down to the nearest 'step' that the shocker supports. For example, a shocker might support 256 different levels from 0mA to 5mA, in which case its step will be 0.0196mA. So in this example, if you specify a shock of 2.5mA, what would actually be delivered would be a shock of 2.49mA, because that's as close to 2.5mA as the shocker can get.

You can find out the maximum shock intensity and the shock 'step' for a shocker by switching to the I/O page, selecting the device which contains the shocker, and then choosing *Shocker* in *Device ports*. The *Specification* section of the information panel will show the required details.

Specifying the absolute maximum shock duration and intensity

Introduction

As you may know, to activate a shocker you need to create a procedure which will determine that some situation has arisen and switch the shocker on, i.e. administer a shock.

In fact, there are three actions which a procedure can perform on a shocker: administer a shock, set the shock duration and set the shock intensity, and it is this ability to alter the shock duration and intensity which are important here. Consider the following situation: you create a procedure which will increase the shock intensity by 1mA each time the animal does something - for example, enters a zone. Clearly then, if the animal enters the zone 20 times the shock intensity is going to end up increasing by 20mA!

In fact, your shocker is unlikely to actually be able to administer a 20mA shock, but the same thing applies to the shock duration, so you could create a similar procedure which would result in the shock duration becoming 20 seconds, or even longer.

Clearly what's needed are some limits which will prevent the shock intensity and duration from increasing *ad infinitum;* this is what the *Maximum permitted shock duration* and *Maximum permitted shock intensity* fields are for.

Details

The *Maximum permitted shock duration* and *Maximum permitted shock intensity* fields are automatically set to default values. If, in your protocol, you don't intend to change the shock duration and/or intensity using a procedure, then you can just leave the defaults as they are. Otherwise, you should consider what maxima for duration and intensity would be appropriate in your experiment and enter them here.

You must make an entry in both fields.

The absolute maximum duration you can enter is 60s and the absolute maximum intensity is 10mA. If your shocker doesn't support the intensity you enter, then the shock will actually be limited to the shocker's maximum rather than the value that you specify.

Specifying whether the physical port used by a shocker is always the same

In brief

In the majority of situations, the physical port used by a shocker will always be the same. However, you may encounter situations in which the port depends on the location of a movable zone.

For example, in a Y-maze, you may have shock floor in the base of each arm of the maze. Now you want to create an element called 'Aversive arm shocker' (let's imagine that after the animal is in the aversive arm for 10s, it will receive a shock), but you have set the 'Aversive arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests.

Thus the 'Aversive arm shocker' will be the shocker for whichever arm happens to be the aversive arm in a particular test, which necessarily means it won't always be the same physical shocker and therefore the 'Aversive arm shocker' won't always be connected to the same physical port.

Details

If you do have a shocker whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the shocker element's page of the protocol. (Note that this list is only displayed when the protocol includes one or more movable zones).

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a shocker

Introduction

You can edit absolutely everything related to a shocker at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a shocker
- Deleting a shocker

Deleting a shocker

Introduction

You can delete a shocker at any time, whether before, during or after an experiment has been performed.

To delete a shocker, simply select it in the protocol list and click the ❌ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting shockers.

See also:

- Deleting protocol elements
- Setting up a shocker
- Editing a shocker

Shocker actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a shocker - these are detailed below.

Actions

Activate the shocker	Switches the shocker on so that it administers a shock at the current intensity and for the current duration. The shocker automatically switches off (i.e. becomes inactive) at the end of the shock duration. If you activate a shocker which is already active, then the current shock will be abandoned and a new shock of the current intensity and duration will start. WARNING: This means that if you were to activate a shocker once every 2 seconds, for example, for a duration of 3s, then the shocker would effectively be on all the time.
Set shocker intensity	Set the intensity of the shocks that the shocker will administer to the value of the action's parameter. The parameter has units of mA (milliamps). If you set the shock intensity of a shocker which is active, the shock intensity will only change the **next time** the shocker becomes active; in other words, the shock being delivered will NOT change. If the specified shock intensity is greater than the shock intensity limit, the intensity will be set to the limit.
Set shocker duration	Sets the duration of the shocks that the shocker will administer to the value of the action's parameter. You can specify the duration using units of ms (milliseconds) or s (seconds); if you don't specify any units, seconds will be assumed. If you set the shock duration of a shocker which is active, the duration will only change the **next time** the shocker becomes active; in other words, the shock being delivered will NOT change. If the specified shock duration is greater than the shock duration limit, the duration will be set to the limit.

Pellet dispensers

As the name implies, *pellet dispensers* are outputs that can be connected to a pellet dispenser in order to dispense pellets during a test. The *Pellet dispensers* element of the protocol is used to control these types of outputs.

For more information about pellet dispensers, refer to the topics below.

- An introduction to Pellet dispensers
- Setting up a Pellet dispenser
- Editing a Pellet dispenser
- Deleting a Pellet dispenser
- Pellet dispenser actions
- Pellet dispenser measures

An introduction to Pellet dispensers

Introduction

Pellet dispenser outputs allow ANY-maze to control a physical pellet dispenser during an experiment, dispensing pellets based on certain conditions during a test. For example, ANY-maze could dispense a pellet when the animal presses a lever.

Pellet dispensers supported by ANY-maze.

Only one *specific* pellet dispenser is support by ANY-maze, and that is the Ugo Basile pellet dispenser. This has feedback built into the dispenser, allowing ANY-maze to confirm that a pellet was actually dispensed and to determine certain error conditions, such as the pellets having run out. Most other pellet dispensers can be controlled by setting them up as 'generic' pellet dispensers.

Connecting the pellet dispenser

Pellet dispensers can be connected to ANY-maze in a number of ways:

- The Ugo Basile pellet dispenser must be connected to an AMi-2 Digital interface as described here
- If the pellet dispenser's input is a 5V TTL input, then it can be controlled using the AMi-2 Digital interface as described here
- If the pellet dispenser requires a higher voltage input, then it can be controlled using the AMi-2 Relay interface

In all cases, you simply need to configure the relevant output port of the AMi-2 device as a 'Pellet dispenser'; for more details on how to do this, see Configuring an AMi-2 Digital interface or Configuring an AMi-2 Relay interface.

To set up pellet dispensers in our protocol, we would *have* to include the relevant AMi-2 device as an I/O device. Then, its outputs that are configured as pellet dispensers would be available within our protocol.

Of course, this description glosses over the detail of exactly how we would connect the pellet dispensers to the AMi-2 interface, but that is described in the AMi-2 Digital interface pellet dispenser ports and AMi-2 Relay interface ports topics.

These AMi-2 interfaces are not the only I/O devices supported by ANY-maze - a full list can be found here.

Using the pellet dispenser

A *Pellet dispenser* protocol element represents a physical pellet dispenser. When you set up the output in an ANY-maze protocol, you give it a name and tell ANY-maze which I/O device it is part of. But that's not very useful; what you really want to do is to control the pellet dispenser during a test, probably dispensing a pellet in response to something the animal does.

Dispensing pellets during a test is the job of a procedure. Procedures can be set up to detect when something happens in a test (for example, the animal enters a certain zone), and to then take an action because of it - which includes dispensing a pellet using a pellet dispenser that has previously been set up as part of the protocol.

Summary

Pellet dispensers are a little more complicated than some other protocol elements because they usually require various other elements to be set up too in order to make them useful:

- You need to set up the I/O device that the pellet dispenser is physically connected to
- You need to set up the pellet dispenser itself
- You will usually want to set up at least one procedure to control the pellet dispenser during the test

See also:

- An introduction to I/O devices
- Setting up a pellet dispenser
- Editing a pellet dispenser
- Deleting a pellet dispenser

Setting up a pellet dispenser

Introduction

For a general introduction to pellet dispensers, see An introduction to pellet dispensers.

To add a pellet dispenser to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New pellet dispenser* from the menu which appears.

The New output item *menu option is only included if the* protocol mode *includes input/output.*

The New pellet dispenser *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the pellet dispenser's name.
2. Specify whether the physical port used by the pellet dispenser is always the same, or whether it depends on the location of a movable zone.
3. Specify the physical port (or ports) used by the pellet dispenser.

What next?

After completing these steps, you should consider whether you want to include any laser controllers in this protocol.

See also:

- Adding elements to a protocol
- Editing a pellet dispenser
- Deleting a pellet dispenser

Pellet dispenser name

In brief

Enter a name for the pellet dispenser in the *Pellet dispenser name* field on the pellet dispenser's settings page. You must make an entry, but it can be anything you like.

Details

Pellet dispenser names should be named to make them easily recognizable, for example, 'Reward'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying whether the physical port used by a pellet dispenser is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port used by a pellet dispenser will always be the same - it's simply the port on the interface device to which the pellet dispenser actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Y-maze, you may have a pellet dispenser mounted at the end of each arm of the maze. Now you want to create an element called 'Goal arm reward' (let's imagine that a pellet is dispensed if the animal spends a certain amount of time in the 'Goal arm'), but you have set the 'Goal arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests. Thus the 'Goal arm reward' will be the pellet dispenser in whichever arm happens to be the goal arm in a particular test, which necessarily means it won't always be the same physical pellet dispenser (and therefore, the 'Goal arm reward' won't always be connected to the same physical port).

Details

If you do have a pellet dispenser whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the pellet dispenser element's page of the protocol.

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a pellet dispenser

Introduction

You can edit absolutely everything related to a pellet dispenser at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a pellet dispenser
- Deleting a pellet dispenser

Deleting a pellet dispenser

Introduction

You can delete a pellet dispenser at any time, whether before, during or after an experiment has been performed.

To delete a pellet dispenser, simply select it in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting pellet dispensers.

See also:

- Deleting protocol elements
- Setting up a pellet dispenser
- Editing a pellet dispenser

Pellet dispenser actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a pellet dispenser - these are detailed below.

Actions

Dispense pellet	Dispenses a pellet, based on the settings in the configuration of the port. These settings determine how long the output will be active for, and whether the output is 'active-low'.

Pellet dispensers will deactivate automatically.

Pellet dispenser measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a pellet dispenser:

- Number of pellets dispensed
- Latency to first pellet dispensed

Each measure is available for the apparatus as whole (i.e. irrespective of where the animal was when the pellet dispenser activation occurred), and also for each defined zone.

Number of pellets dispensed

Description	Reports the number of pellets that were dispensed. For Ugo Basile pellet dispensers, feedback from the device ensures that this is the actual count of pellets dispensed; for all other pellet dispensers, this is the count of the number of times that ANY-maze asked the device to dispense a pellet (since there is no feedback, ANY-maze must assume that this was succesful!).
Calculation method	Counts the number of pellets dispensed.
Analysis in zones	Counts the number of pellets dispensed when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of pellets dispensed during the time period.
Units	None
Notes	None

Latency to first pellet dispensed

Description	Reports the amount of time that elapsed in the test before the first pellet was dispensed.
Calculation method	The value of the test clock when the first pellet was dispensed.
Analysis in zones	The value of the test clock at the time the first pellet was dispensed while the animal was within the zone.
Analysis across time	This measure can be analysed across time. The value is the time since the start

of the time segment or time period that the first pellet was dispensed in the time segment or time period. If no pellet was dispensed in the time segment or time period then the result is #N/A (Not Available), unless the analysis option to *Use the test duration as the latency for events which don't occur* is set, in which case the result will be the duration of the time segment or period.

Units	Seconds
Notes	None

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Laser controllers

As the name implies, *laser controllers* are used to control lasers (typically optogenetic lasers) during a test. This includes setting the laser's pulse frequency, duty cycle, intensity and, of course, actually switching the laser on and off.

For more information about laser controllers, refer to the topics below.

- An introduction to laser controllers
- Setting up a laser controller
- Editing a laser controller
- Deleting a laser controller
- Laser controller actions
- Laser controller measures

An introduction to laser controllers

Introduction

Laser controllers allow ANY-maze to switch lasers on and off during an experiment. They can also be used to adjust the intensity of the laser, and to adjust the properties of the laser signal - for example, setting the signal to a pulse with a specific frequency and duty cycle.

Connecting a laser controller

In order for ANY-maze to control a laser, you will need an AMi-2 Optogenetics interface, as this is currently the only laser controller ANY-maze supports. This device connects to your PC via USB and connects to a *trigger* input on your laser, usually using a BNC cable - full details can be found here. The interface can be connected to up to four lasers, each of which can be controlled independently.

As with all inputs and outputs supported by ANY-maze, you must add the physical I/O device, in this case the *AMi-2 Optogenetics interface* to the protocol before you can actually use it in a laser controller element.

Using a laser controller

When you add a laser controller to your protocol, you will be able to specify such things as the frequency and duty cycle of the pulse train you want the laser to output, how long the pulse train should remain on for, the intensity of the laser and whether or not the laser should be on at the start of the test.

Having the laser start when the test starts can be useful, but the real power of laser controllers comes from the fact that you can control them *during* a test using procedures. For example, you might create a procedure to detect when the animal moves into a certain zone and when it does, switch the laser on. In fact, procedures can do more than just switch the laser on and off; they can also change its characteristics, such as frequency, duty cycle and intensity. So, for example, you could have different zones and pulse the laser at a different frequency depending on which one the animal enters.

See also:

- An introduction to I/O devices
- Setting up a laser controller
- Editing a laser controller

- Deleting a laser controller
- An introduction to procedures
- Laser controller actions

Setting up a laser controller

Introduction

For a general introduction to laser controllers, see An introduction to laser controllers.

To add a laser controller to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New laser controller* from the menu which appears.

The New output item *menu option is only included if the* protocol mode *includes input/output.*

The New laser controller *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the laser controller's name
2. Specify the effect of activating the laser controller
3. Optionally, specify the laser controller's intensity
4. Set the laser controller's activation and reset options
5. Specify whether the physical port used by the controller is always the same, or whether it depends on the location of a movable zone
6. Specify the physical port (or ports) used by the controller

What next?

After completing these steps, you should consider whether you want to include any syringe pumps in this protocol.

See also:

- Adding elements to a protocol
- Editing a laser controller
- Deleting a laser controller

Laser controller name

In brief

Enter a name for the laser controller in the *Laser controller name* field on the laser controller's settings page. You must make an entry, but it can be anything you like.

Details

Laser controller names should specify the thing that is being controlled, for example 'Optogenetic laser'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying the effect of activating a laser controller

In brief

You should use the options on the laser controller's settings page to specify the effect of activating a laser. You will probably want the laser to pulse with some frequency and duty cycle, but you can also just switch the laser on without pulsing it, or even specify any arbitrary sequence of pulses for the laser using an external 'pulse train file'.

- Pulsing the laser on and off
- Turning the laser on (without pulsing it) for a specific duration
- Turning the laser on continuously
- Using a pulse train file to output an arbitrary sequence of pulses
- Adjusting these setting during a test

In this topic, it is assumed that the laser is triggered (switched on) using a high-level signal. Some lasers are 'active low' (they are triggered by a low level signal); if your laser works in this way, you can adjust the active state using the Optogenetics interface configuration window.

Pulsing the laser on and off

In many situations, for example, optogenetics, you will want the laser to pulse on and off at some specific frequency whenever it is activated. This is achieved by selecting the option to *Output a pulse train...*

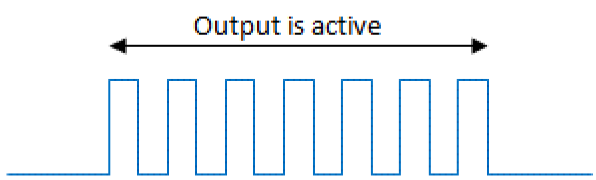


Figure 1. When set to 'Output a pulse train', the laser controller outputs a series of pulses whenever it is activated.

There are two settings you must make if you set a laser to output a pulse train when active: the frequency and the duty cycle of the pulses. The frequency is simply the frequency of the pulses - for example if you specify 10Hz, then there will be 10 pulses per second, so each one will last 100ms. The duty cycle specifies how long within each pulse's period the output will be high and (by implication) how long it will be low. For example, if you specify a duty cycle of 25% then, in the case where the pulse lasts 100ms, the pulse will be high for 25% of the time and low for 75%, so it will be high for 25ms and low for 75ms - see figure 2.

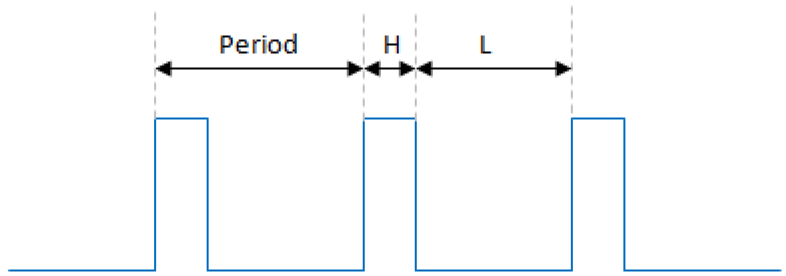


Figure 2. The frequency you specify defines the period of each pulse (Period = 1/Frequency). The duty cycle defines the proportion of the period that the output is high (H) and the proportion for which it is low (L). In the above figure the duty cycle is 25%, so H is 25% of the period and L is 75%.

The frequency you specify should be in Hertz, and can be any value from 0.01Hz to 10,000Hz. The duty cycle should be in %, and can be any value from 1% to 99%. (The shortest on or off pulse is therefore 1μs).

As an alternative to specifying a frequency and duty cycle, you can just specify the time on and time off. For example, you could specify 40ms on and 10ms off. This would be identical to specifying a frequency of 20Hz and a duty cycle of 80%.

Whether you specify a frequency and duty cycle, or a time on and time off, is just a matter of preference - it makes no functional difference to how the laser will behave.

If you do define the pulses using time on and time off, then you can specify any value from 0.1ms to 100,000ms. This actually means that the pulses you can define are a little more limited than using the frequency and duty cycle technique; so, if you need an on or off period smaller than 0.1ms, you should define them by specifying the frequency and duty cycle instead.

Setting a duration for the pulse train

As well as specify the characteristics of the pulse train, you can also optionally specify how long it should be output for. For example, you might want your laser to output pulses at 20Hz with a 50% duty cycle, but just for 3 seconds. In this case, you can specify that the duration is 3s and the laser will automatically switch off 3 seconds after it is activated. If you don't specify a duration, then the laser will continue to output pulses after it has been activated, until it is explicitly deactivated by a procedure or the end of the test.

If you do specify a duration, it can be any period from 0.1ms to 100s, and you can use any of the units: min (minutes), s (seconds) or ms (milliseconds). You can mix units and also specify decimal values.

Turning the laser on (without pulsing it) for a specific duration

In some circumstances, you may want a laser to switch on *without pulsing* for some specific duration whenever it is activated. In this case, you should select the option *Turn output on for the following duration* and then enter the duration. For example, if you want the laser to come on for 500ms then you can simply select this option and enter '500ms' as the duration. Then, whenever the laser is activated, it will switch on (it won't pulse) and after 500ms it will automatically switch off. The duration can be any period from 0.1ms to 100s, and you can be use any of the units: min (minutes), s (seconds) or ms (milliseconds). You can mix units and also specify decimal values.

Turning the laser on continuously

The final option is to turn the laser on continuously. Selecting this option will mean that whenever the laser is activated, it will turn on and stay on until it is explicitly switched off by a procedure or the end of the test.

Using a pulse train file to output an arbitrary sequence of pulses

If the sequence of pulses that you wish to use for the laser is more complex than just a simple frequency and duty cycle, then you can specify this sequence in an external text file. This file will be loaded into the AMi-2 Optogenetics interface just before the test starts, and can then be used at any point during the test. For more details on this, see Playing an arbitrary sequence of pulses using an external file.

If this is what you want the laser to do, then you must specify the full path to the file containing the sequence of pulses. You can type the file path, or use the button to open the Choose pulse train file window where you can navigate to the file on your computer.

There are two options available to you when using an external pulse train file; the first will play the file just once, and then deactivate the laser; the second will continue to repeat the pulse sequence contained in the file until either the laser controller is deactivated, the laser's source is changed to a pulse train or continuous signal, or the test ends.

Adjusting these setting during a test

Imagine you have some apparatus where you have two zones (A and B) and you would like your laser to output 20Hz pulses at 50% duty cycle for 2s, whenever the animal enters zone A, and 10Hz pulses, also at 50% duty cycle for 2s, whenever it enters zone B. But what should you specify as the laser characteristics here? And how will you change them based on the zone the animal has entered?

The answer is that you should simply specify one of the sets of characteristics here, so perhaps 20Hz, 50% duty cycle, active for 2s and then use a procedure to adjust these when the animal enters a zone. For example, the procedure below could be used for the test I described above:

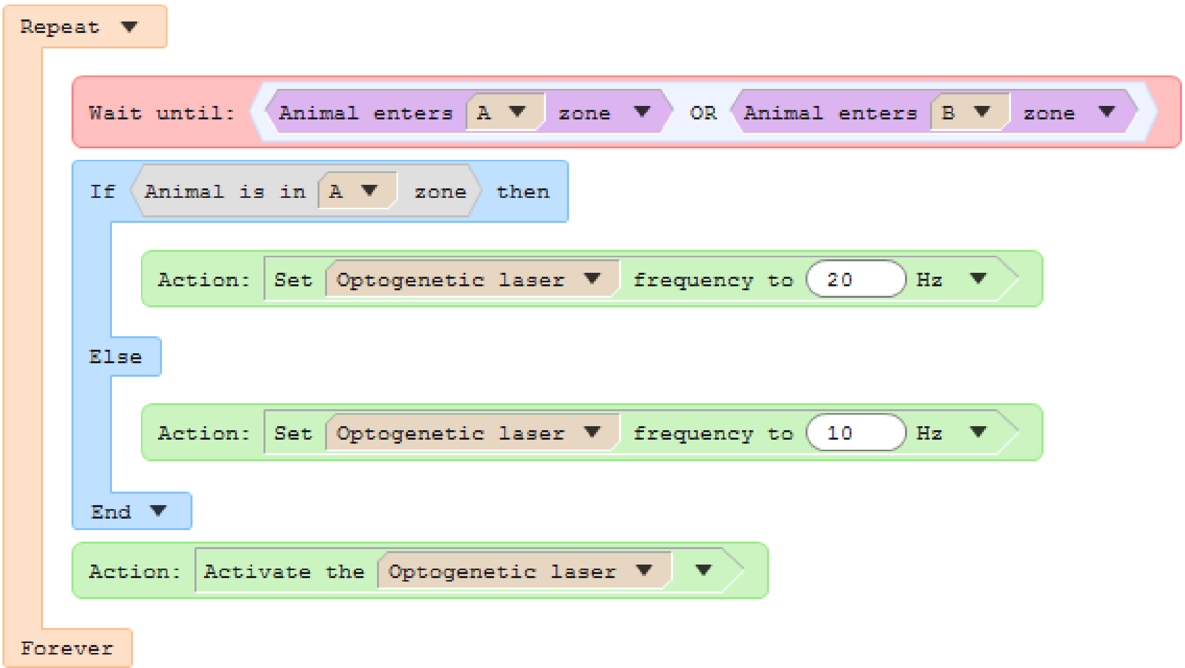


Figure 3. This procedure will alter the frequency of a laser controller's pulses depending on which zone the animal enters. Note that in this example, the duty cycle and duration are the same irrespective of the zone, but they could also have been adjusted based on the zone, if that was required.

All aspects of a laser controller can be adjusted by a procedure's Actions - you'll find a full list of the available actions here. If you're new to procedures, you can learn more about them in the Introduction to procedures.

Playing an arbitrary sequence of pulses using an external file

In brief

ANY-maze's laser controller settings are quite comprehensive, but it might be that rather than using a pulse train or a continuous signal, you want to use an arbitrary sequence of pulses. For example, you may have recorded an animal's neural activity from a previous test, and you may wish to pulse the laser on or off at the exact times that the animal's neural activity happened in this previous test. ANY-maze allows you to do this by specifying the sequence of pulses in a *'pulse train file'*.

- How it works
- Creating a pulse train file
- Using the pulse train file in a test

How it works

ANY-maze allows you to specify, for each laser controller, a 'pulse train file' containing a sequence of pulses for that laser. You can use a different file for each laser, or the same one for all of them - it's up to you. The pulse train file can contain an entirely arbitrary sequence of pulses, down to millisecond resolution, and can even adjust the intensity (voltage) of the laser controller (if your lasers support this feature).

All pulse train files that you specify are loaded into the AMi-2 Optogenetics interface just before the start of a test. The pulse sequence contained in the file can be played from the moment the test starts (using the laser controller's settings page, or alternatively you can turn it on at any point during the test using a procedure. The important thing to remember is that once the file has been downloaded to the hardware, it will stay there for the duration of the test - if you're using a procedure, you can switch between this pulse train file, a continuous signal and a basic pulse train as many times as you like during the test. So, for example, you could play this pulse train file when the animal enters a specific zone, and switch to a standard pulse train on entry to other zones. Alternatively, you could play the pulse train file whenever the animal is feeding, and turn it off at other times.

Creating a pulse train file

Creating a pulse train file is easy - it's just a simple text file. Full details of the contents of a pulse train file are contained in the Pulse train file formats topic, but briefly, the file can contain:

Durations	Specifies the duration that the laser is off, followed by the duration that it is on (together with an optional voltage/intensity).

Pulse times and widths	Specifies the time that a pulse should be turned on, together with the duration to turn it on for (together with an optional voltage/intensity).
Pulse on/off times	Specifies the time that a pulse should be turned on, followed by the time that the pulse should be turned off. Again, the voltage/intensity can be optionally specified.

Using the pulse train file in a test

Once you've created a pulse train file, you need to use the laser controller's settings page to link the pulse file with a given laser controller. To do this, simply use the to open the Choose pulse train file window. This will allow you to navigate to the file that you've created.

On the laser controller's settings page, you'll also specify whether you'll play the pulse sequence just once (and then turn the laser off), or whether you'll repeat the contents of the file. If you choose to play it repeatedly, then the pulse sequence will play from start to finish, then repeat again from the start, and so on until either the laser is turned off, or the source of the laser controller changes (to a pulse train or a continuous signal), or the test finishes.

Specifying a pulse train file will simply ensure that the file is loaded into the AMi-2 Optogenetics interface at the start of the test, making it available for use during the test. To actually *use* the file, you can do one of the following:

- Specify *Turn on when the test starts* on the laser controller's setup page to start the pulse sequence playing as soon as the test starts.
- Use a procedure to specify that the laser controller's source should be a pulse train file - using one of the Set laser controller output to pulse train file... actions. Once you've set the source, use the Activate laser controller action to actually start it playing.

See also:

- Specifying the effect of activating a laser controller
- Pulse train file formats

The Choose pulse train file window

Overview

When you use the *Browse* button on the laser controller setup page to load a pulse train file, the *Choose pulse train file* window will be displayed. Here you can navigate around your computer or network to find the file you want to open.

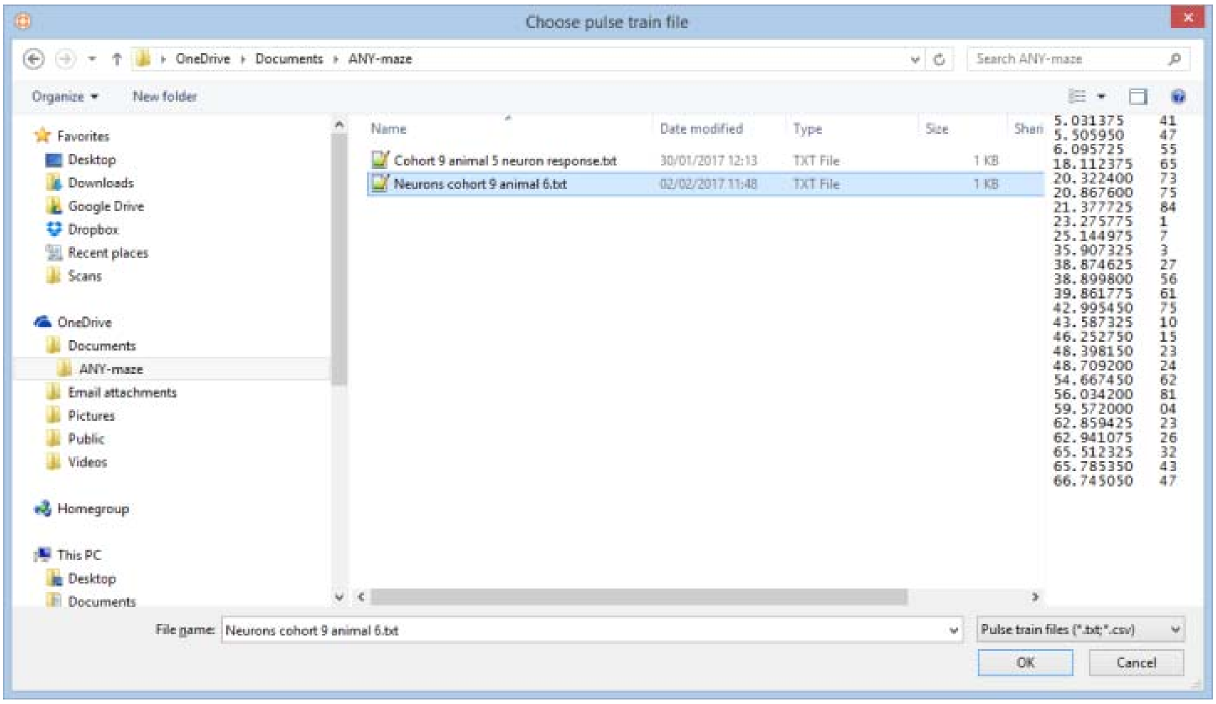


Figure 1. The Choose pulse train file *window.*

Details

You will probably be familiar with this window already, as it's based on the standard 'Open file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list	The files list shows the folders and files stored in the current location. You can double click a folder to open it so you can access the files inside. If you double click a .txt or .csv file, then ANY-maze will try to open it as a pulse train file.
File name	You can use this field to type in the location and name of the file you want to open.
Type of file	You can use this list to choose what types of files are shown in the *Files list*. By default, the list shows text (.txt) and comma-separated (.csv) files, but you can change it to show all files, irrespective of their type. Note however that ANY-maze can only open valid pulse train files.

See also:

- Setting up a laser controller
- Specifying the effect of activating a laser controller
- Pulse train file formats

Pulse train file formats

In brief

ANY-maze has the capability of generating pulse trains for laser controllers, based on the contents of a file. This means that rather than a repetitive pulse of a given frequency and duty cycle, you can explicitly tell the laser how long you want the laser to be on for, and how long to turn off for, to the nearest millisecond. This can be used to generate arbitrary sequences of pulse trains and even laser intensities (if your lasers support this feature).

For more details of how to use a pulse train file within the ANY-maze software, please see Playing an arbitrary sequence of pulses using an external file. This topic describes the formats of the files that can be used.

Creating a pulse train file

A pulse train file is simply a text file, so creating one is easy. First of all, open Windows Explorer and navigate to the folder in which you want to store the file. Them right-click anywhere in the folder and select 'New -> Text Document' from the menu that appears. The file will be created in that folder, and the name of the file will be highlighted (to start with, it will be 'New Text Document.txt'). Type in the name that you want to give the file. You can then edit the file in Windows Notepad by simply double-clicking on the file, and it will open for editing.

Alternatively, you can create a file from within Microsoft Excel; simply use the 'New' option to create a 'Blank workbook'. Each of the file formats described below has two or three 'columns', so you just need to enter the data in these columns and then save the data in 'comma-separated' (.csv) format.

File formats

There are three basic file formats that ANY-maze recognizes. It's entirely up to you which one of these you choose; just use whichever one seems easiest to enter the pulse sequence that you need. If you're starting with some data exported from another application (for example, NeuroExplorer) then you'll find that this influences the format you choose.

All of the file formats recognized by ANY-maze must contain information on the first line of the file which indicates the type of data in the file. The rest of the file will consist of a number of lines of 'comma-separated' numeric values; that is, there will be two or three columns of data that are separated by commas. Spaces in the file are ignored. Within the file, there must be one line for each set of data (so each line of the file will have two or three numeric values, separated by commas). File names should end in .txt or .csv (which is the default file extension if you're saving a comma-separated file from Microsoft Excel).

The file formats described below contain some examples which should allow you to create your own files, but if you have any questions about these file formats, then please contact ANY-maze technical support who will be happy to help.

Durations

This file format contains two or three columns, and each line of the file will contain:

- Duration off, in ms
- Duration on, in ms
- Optionally, a voltage/intensity, in Volts

The first line of this file defines the file format, and must contain the following text:

```
Duration off, Duration on
```

or

```
Duration off, Duration on, Voltage
```

Note that this text is not case-sensitive, and any spaces are ignored.

If you want the laser controller to turn on immediately when the file starts playing, you need to enter 0 for the first 'duration off'. Likewise, if you want a period at the end of the file where the laser is off, then enter 0 for the 'duration on' in the last line of the file.

The following simple file will turn the laser on for 1ms, then off for 1ms, then turn it on for 2ms, then off for 2ms, and so on up to 10ms.

```
Duration off, Duration on
0, 1
1, 2
2, 3
3, 4
4, 5
5, 6
6, 7
7, 8
8, 9
9, 10
10, 0
```

Note that the first line of the file contains 0 for the 'time off', so the laser will turn on (for 1ms) immediately when the laser controller is activated. At the end of the file, the last line shows that the laser is turned off for 10ms (then on for 0ms, which means that this 0 is ignored). If this file were to be repeated, then the laser would be off for 10ms at the end of the file, before turning on for 1ms as it starts to play again.

As with all pulse train files recognised by ANY-maze, you can optionally change the voltage. To do this, just add a value for the voltage at the end of the relevant line. If the voltage is entered as 0, or is not specified, then it will be left as it was when the laser was previously used (i.e. the voltage will not be changed).

A simple example of this file format, with voltages, is as follows:

```
Duration off, Duration on, voltage
5,   5,   5.0
```

```
    5,    5,    2.5
```

This file will leave the laser off for 5ms, then pulse it on for 5ms at full intensity (5.0V). After turning it off for another 5ms, it will pulse it on for 5ms at half intensity (2.5V). This file would generate a simple pulse of 100Hz, but with the laser intensity alternating between half- and full-voltage.

ANY-maze will check the pulse train file when it is loaded, to ensure that the data is valid. When you select the file on the laser controller's settings page, you'll be warned if the values contained in the file are invalid. All durations must be a positive number of milliseconds, and voltages (if specified) must be between 0.02 and 5.0V (or 0, meaning "don't change the voltage").

Pulse times and widths

This second file format simply lists a number of 'pulse times' at which to pulse the laser, together with the width of each pulse. Pulse times are specified in seconds, and pulse widths in ms. So the columns of the file are:

- Time (after activating the laser controller), in seconds
- Width of the pulse, in ms
- Optionally, a voltage/intensity, in Volts

The first line of this file defines the file format, and must contain the following text:

```
    Pulse time, Width
```

or

```
    Pulse time, Width, Voltage
```

Note that this text is not case-sensitive, and any spaces are ignored.

The following file has been created from the data contained in a file exported from NeuroExplorer. The NeuroExplorer output has created a list of times (after the start of the test) that neural spikes were recorded - these times have been extracted and pasted into Microsoft Excel; pulse widths of 5ms have been added as a second column of data, and the result has been saved as a comma-separated (.csv) file:

```
    Pulse time, width, voltage
    1.535050,  5
    2.401675,  5
    3.404325,  5
    4.584225,  5
    5.031375,  5
    5.505950,  5
    6.095725,  5
    18.112375, 5
```

This file will pulse the laser on at 1.535 seconds after the laser is activated, for 5ms; then at 2.401 seconds, for 5ms; etc. Note that within ANY-maze, all pulse train files have 1ms resolution, so the extra sub-millisecond data in the above file will just be ignored; this example has been created in a few simple steps from a NeuroExplorer output file, which is why it is specified in more detail than is necessary for ANY-maze.

Again, this file format can optionally specify the voltage. If the above file was to gradually increase the voltage of the laser for each pulse, it could be written as follows:

```
Pulse time, width, voltage
1.535050,  5,  3.5
2.401675,  5,  3.7
3.404325,  5,  3.8
4.584225,  5,  4.0
5.031375,  5,  4.2
5.505950,  5,  4.4
6.095725,  5,  4.6
18.112375, 5,  4.8
```

Note that all voltages specified in these files must be within the 0.02V to 5.0V range allowed for an AMi-2 Optogenetics interface laser; if the file contains voltages outside this range, then playing the pulse file will fail. (Remember that 0.0 is a valid value, and means "don't change the voltage"). If any of the voltages are outside this range, then when you load the file into the laser controller's settings page, you'll be warned that the file is invalid and you won't be allowed to select it.

ANY-maze will check the pulse train file when it is loaded, to ensure that the data is valid. When you select the file on the laser controller's settings page, you'll be warned if the values contained in the file are invalid. The pulse times and widths in this file cannot be negative, and the pulses defined cannot overlap. In addition, voltages (if specified) must be between 0.02 and 5.0V, or 0 (meaning "don't change the voltage").

Pulse on/off times

The final format of file recognized by ANY-maze is similar to the previous format, but rather than defining the pulse width, it defines the time to turn the pulse on, and the time to turn it off. The columns of the file are:

- Time (after activating the laser controller) to pulse the laser on, in seconds
- Time (after activating the laser controller) to pulse the laser off, in seconds
- Optionally, a voltage/intensity, in Volts

The first line of this file defines the file format, and must contain the following text:

```
Pulse on, Pulse off
```

or

```
Pulse on, Pulse off, Voltage
```

Note that this text is not case-sensitive, and any spaces are ignored.

The following file will simply pulse the laser on for 10ms after 1 second; for 20ms after 2 seconds; and so on until 8 seconds have passed:

```
Pulse on, Pulse off
1.0,  1.010
2.0,  2.020
3.0,  3.030
4.0,  4.040
5.0,  5.050
6.0,  6.060
7.0,  7.070
8.0,  8.080
```

Again, an optional voltage can be specified for each pulse, if required:

```
Pulse on, Pulse off, Voltage
```

```
    1.0,  1.010,   5.0
    2.0,  2.020,   4.5
    3.0,  3.030,   4.0
    4.0,  4.040,   3.5
    5.0,  5.050,   3.0
    6.0,  6.060,   2.5
    7.0,  7.070,   2.0
    8.0,  8.080,   1.5
```

This simple example will pulse the laser as before, but will reduce the intensity of the laser each time it does it.

ANY-maze will check the pulse train file when it is loaded, to ensure that the data is valid. When you select the file on the laser controller's settings page, you'll be warned if the values contained in the file are invalid. The pulse times in this file cannot be negative, and must be consecutive (i.e. all 'pulse off' times must be after the corresponding 'pulse on' time). In addition, voltages (if specified) must be between 0.02 and 5.0V, or 0 (meaning "don't change the voltage").

Note that it's possible to create a set of pulse train data that can be interpreted as more than one of these formats; for this reason, it's important to ensure that the titles on the first line of the file are correct.

Specifying the intensity of the laser

In brief

In some lasers, the trigger input can be used not just to switch the laser on and off, but also to adjust its intensity. This works by altering the voltage at the trigger input, for example, a voltage of 5V might mean 'switch on at 100% intensity', while a voltage of 2.5V might mean 'switch on at 50% intensity', and so on.

If your laser works in this way, you can optionally specify the intensity when you set up the laser controller in the protocol.

Details

The laser intensity will default to 5V (i.e. the TTL high value), but you can alter it to any value down to 0.02V. The exact relationship between the voltage and the intensity of the laser will depend on your device, and should be detailed in the laser's manual.

If you set an intensity other than 5V for a laser which DOES NOT include the ability to adjust the intensity, the laser may not switch on when you activate the output. For example, if you specify an intensity of 1V then when you activate the laser the output will be set to 1V, which is below the minimum voltage for a TTL high value and therefore the laser will probably not consider this to be a 'trigger' signal and therefore won't switch on. So, if your laser doesn't support intensity adjustments, simply leave this value set to 5V.

Adjusting the intensity during tests

As with all the laser controller parameters, you can adjust the intensity during a test using a procedure. In this case, your procedure would need to include the action to Set laser controller intensity - perhaps altering it based on the zone the animal is in, or how many times it has pressed a lever.

Setting the laser controller activation and reset options

In brief

You should use the buttons on the laser controller's settings page to specify when the laser should be activated.

As well as these options, you can also control the laser using a procedure, something which is described in more detail below.

Specifying when the laser should be activated

Clearly, if you've included a laser controller in your protocol, it's because you want to turn it on and off during the tests - but when should it turn on? In some cases, the answer may simply be 'when the test starts'; but in other cases you may want to do more sophisticated things, perhaps turning it on and off as the animal moves into different zones.

Activating the laser at the start of the test

For the simple case of activating the laser at the start of the test, you can use the option here (in the protocol's laser controller settings). This doesn't mean you can't also use a procedure (perhaps to switch it off again); it just means that when the start starts, the laser will be activated.

Activating the laser based on more complex rules

In this case, you should use a procedure to activate (and/or deactivate) the laser. Full details are beyond the scope of this help topic, as it rather depends on what you want to do; but, for example, the procedure in figure 1 will wait until the animal enters the 'Target zone' and will then activate the 'Optogenetic laser'. More details can be found in the Introduction to procedures.

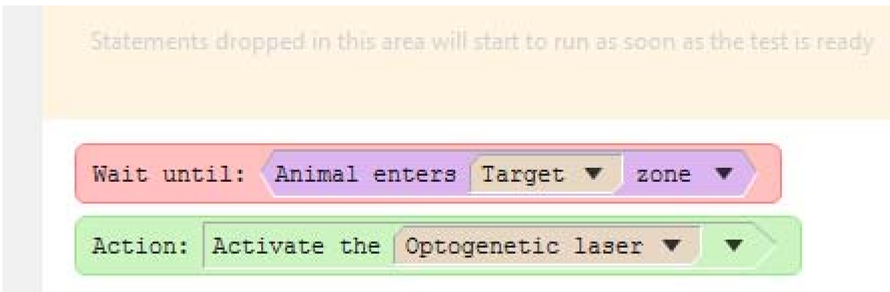


Figure 1. This simple procedure will activate the 'Optogenetic laser' when the

animal enters the 'Target' zone.

Activating the laser BEFORE the test starts

If you want to activate a laser *before* the test starts, then the best method is to create a procedure to do this; as can be seen in figure 2, this is very simple.

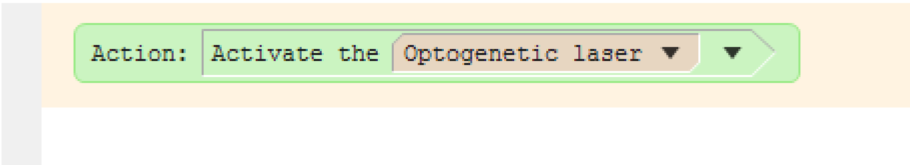


Figure 2. This one-statement procedure will activate the 'Optogenetic laser' before the test starts (because statements in the orange-shaded area are performed as soon as the test is ready).

The key point here is that statements in the orange-shaded area at the top of the procedure editor are performed as soon as the test is *ready*. So, if you add an Action statement (to activate a laser controller) to this area, then the laser will be activated before the test begins.

Resetting a laser controller at the end of a test

When a test ends, ANY-maze will deactivate the laser and reset it to the values entered on the laser's settings page. However, there are circumstances when you might not want this to occur, for example:

- You might be planning to run tests back-to-back, and whatever the laser's state was at the end of one test, you want to continue in the next one. So if it's on at the end of one test, then if it's not reset, it'll still be on at the start of the next one.
- You might have used a procedure to alter some parameters of the laser; for example, the procedure may have changed the pulse frequency during the test. If the laser controller is reset at the end of the test, then the frequency will be reset to whatever the default frequency is (as defined in the laser controller's settings in the protocol), but what if you want to retain the frequency set by the procedure (so it's used in the next test)?

In versions of ANY-maze prior to version 6, there was an option to *not* reset the laser at the end of the test. However, this would not necessarily work in the way that you'd expect it to:

- The state would be retained from one *test* to the next, but not from one *trial* to the next. For example, imagine that your test schedule is such that you will test animal 1 in its first trial, then animal 2 in its first trial and then animal 1 in its second trial, etc. What would happen is that after animal 1's first trial the state of the laser (e.g. its pulse frequency) would remain the same and so this would be the state that would apply to animal **2's** first trial... which is probably not what you want.

- Even if the above didn't apply (for example, in your schedule you might test animal 1 four times in a row), then if you exited ANY-maze after trial 2 and then restarted it again, the state of the laser would be lost.

So, not resetting the outputs at the end of a test doesn't provide a way to achieve our goal, and therefore this option has been removed. If you have an older experiment file in which this option was selected, however, it will still appear in the settings for backward compatibility.

The best way to ensure that a laser's state is retained from one trial to the next (even if you close the experiment, exit ANY-maze and turn off your computer between tests) is to use a procedure; this is described in detail here.

See also:

- An introduction to procedures

Specifying whether the physical port used by a laser controller is always the same

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations (especially in the case of optogenetics), the physical port used by a laser controller will always be the same - it's simply the port on the interface device to which the laser actually connects.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Y-maze, you may have three lasers, each mounted in one of the arms of the maze. Now you want to create an element called 'Goal arm laser' (let's imagine that the laser comes on if the animal spends a certain amount of time in this arm), but you have set the 'Goal arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests. Thus the 'Goal arm laser' will be the laser in whichever arm happens to be the goal arm in a particular test, which necessarily means it won't always be the same physical laser (and therefore 'Goal arm laser' won't always be connected to the same physical port).

Details

If you do have a laser controller whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the laser controller element's page of the protocol.

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

ANY-maze help topic H0305

Editing a laser controller

Introduction

You can edit absolutely everything related to a laser controller at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a laser controller
- Deleting a laser controller

Deleting a laser controller

Introduction

You can delete a laser controller at any time, whether before, during or after an experiment has been performed.

To delete a laser controller, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting laser controllers.

See also:

- Deleting protocol elements
- Setting up a laser controller
- Editing a laser controller

Laser controller actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a laser controller - these are detailed below.

Actions

Activate laser controller	Sets the laser controller to its active state. If the controller is set to output a pulse train, this will start the pulses. If the laser controller is set to output for a given duration, then it will automatically deactivate after the appropriate time. If the controller is already active, and it is set to output for a given duration, this action will start timing the duration again. For example, if the output should remain active for 5s and it has already been active for 4s when it is activated again, then it will remain active for ANOTHER 5s.
Deactivate laser controller	Sets the laser controller to its inactive state. If the controller is already inactive, this action has no effect. If the laser controller is set to output for a specific duration, but this duration has not yet passed, then this action *will* deactivate the controller (and cancel the duration timing). If the controller is set to output a pulse train, this will end the pulses immediately, even if this is in the middle of a pulse.
Set laser controller output to pulse train	Changes the output of the laser controller to be a pulse train rather than a continuous output. This action will **not** activate (or deactivate) the laser controller. If the controller is currently active, either outputting a continuous signal or a pulse file, then the output will immediately switch to a pulse train.
Set laser controller output to continuous	Changes the output of the laser controller to be a continuous (DC) output rather than a pulse train. This action will **not** activate (or deactivate) the laser controller. If the controller is currently active, either outputting a pulse train or a pulse file, then the output will immediately switch to a continuous (DC) output.
Set laser controller output to pulse train file (played once)	Changes the output of the laser controller to be the pulse sequence defined in the pulse train file specified

in the laser controller's settings. This action will **not** activate (or deactivate) the laser controller. If the controller is active and outputting a pulse train or a continuous (DC) output, then the output will immediately switch to playing the pulse sequence specified in the pulse train file. This pulse sequence will be played once, and then the laser will be turned off.

If no pulse train file has been specified for this sequence, then a procedure error will be logged, and this action will have no effect.

Set laser controller output to pulse train file (played continuously) Changes the output of the laser controller to be the pulse sequence defined in the pulse train file specified in the laser controller's settings. This action will **not** activate (or deactivate) the laser controller. If the controller is active and outputting a pulse train or a continuous (DC) output, then the output will immediately switch to playing the pulse sequence specified in the pulse train file. After the pulse sequence has been played, the sequence will be started again immediately, and will repeat until the laser is turned off, the laser source is changed again, or the test ends.

If no pulse train file has been specified for this sequence, then a procedure error will be logged, and this action will have no effect.

Set laser controller frequency Changes the frequency of the pulses of a *pulse train* to the value specified as the action's parameter. The parameter is in Hertz. If the laser controller is not set to output a pulse train, this action will have no effect. This action will **not** activate (or deactivate) the laser controller. If the controller is active and outputting a pulse train, then the pulse train frequency will change on-the-fly.

Set laser controller duty cycle Changes the duty cycle of a *pulse train* to the value specified as the action's parameter. The parameter is in percent, and represents the amount of a single cycle for which the laser controller will be ON. If the controller is not set to output a pulse train, this action will have no effect. This action will **not** activate (or deactivate) the laser controller. If the controller is active and outputting a pulse train, then the duty cycle will change on-the-fly.

Set laser controller duration Changes the duration for which the laser controller will be active to the value specified as the action's parameter. The duration can be any time value, and can include units of ms (milliseconds), s (seconds), min or m (minutes), h (hours) and d

	(days); if no units are specified, seconds are assumed. If the laser controller is already in the middle of an existing output duration when a new duration is set, the new duration will be applied immediately and will effectively re-start. For example, let's say that you have already performed an action to set the output duration to 4 seconds. After 2 seconds have elapsed, you set the duration to 3 seconds; the laser controller will start to output for 3 seconds, thus giving you 5 seconds of output in total.
Set laser controller intensity	Changes the intensity of the laser controller's output. This value is in Volts. This action will **not** activate (or deactivate) the laser controller. If the controller is currently active, then the intensity will change on-the-fly.

Laser controller measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a laser controller:

- Number of activations
- Time active
- Latency to first activation
- Latency to first deactivation
- Longest activation
- Shortest activation
- Average activation duration
- Frequency of activations

Each measure is available for the apparatus as whole (i.e. irrespective of where the animal was when the laser controller activation occurred), and also for each defined zone.

Number of activations

Description	Reports the number of times the laser controller was activated.
Calculation method	Counts the number of activations.
Analysis in zones	Counts the number of activations when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of activations during the time period.
Units	None
Notes	None

Time active

Description	Reports the total amount of time that the laser controller was active.
Calculation method	Sums the amount of time for which the laser controller was active.

Analysis in zones	Sums the amount of time for which the laser controller was active while the animal was in the zone. For a particular zone, it's possible for the *Time active* to be non-zero while the Number of activations is zero. This can occur if the animal enters the zone when the controller is active. In this case, the time the controller is active will be registered but the activation itself won't be.
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of time that the laser controller was active during the period. For a particular time period, it's possible for the *Time active* during the period to be non-zero while the Number of activations for the period is zero. This can occur if the controller is already active at the start of the period. In this case, the time the controller is active will be registered but the activation itself won't be.
Units	Seconds
Notes	None

Latency to first activation

Description	Reports the amount of time that elapsed in the test before the laser controller was activated for the first time.
Calculation method	The value of the test clock at the first activation of the laser controller.
Analysis in zones	The value of the test clock at the first activation of the laser controller that occurred while the animal was within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Latency to first deactivation

Description	Reports the amount of time that elapsed in the test before the laser controller was deactivated for the first time.
Calculation method	The value of the test clock at the first laser controller deactivation.
Analysis in zones	The value of the test clock at the first laser controller deactivation that occurred while the animal was within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds

Notes	None

Longest activation

Description	Reports the duration of the longest period for which the laser controller was continuously active.
Calculation method	The duration of each laser controller activation is calculated, and the largest value is found.
Analysis in zones	The longest period for which the laser controller was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest period for which the laser controller was continuously active during the period.
Units	Seconds
Notes	None

Shortest activation

Description	Reports the duration of the shortest period for which the laser controller was continuously active.
Calculation method	The duration of each laser controller activation is calculated, and the smallest value is found.
Analysis in zones	The shortest period for which the laser controller was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest period for which the laser controller was continuously active during the period.
Units	Seconds
Notes	None

Average activation duration

Description	Reports the average duration for which the laser controller was active.
Calculation method	Calculated by dividing the Time active by the Number of activations.
Analysis in zones	This measure cannot be analysed in zones.

Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Frequency of activations

Description	Reports the frequency with which the laser controller was activated.
Calculation method	Calculated by dividing the Number of activations by the *Test duration.*
Analysis in zones	Calculated by dividing the Number of activations in the zone by the *Total time in the zone.*
Analysis across time	This measure can be analysed across time. For any time period, the result is the Number of activations which occurred during the period, divided by the period's duration.
Units	Hertz
Notes	None

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Switch output measures
- Pellet dispenser measures
- Syringe pump measures
- OPAD measures
- Procedure measures

- Event measures
- Virtual switch measures

Syringe pumps

Syringe pumps provide a way in which a drug can be administered to an animal depending on the situation in a test. For example, a small quantity of a drug could be administered whenever the animal presses a lever, or whenever it enters a specific zone. The volume, flow rate and direction of the pump can all be adjusted dynamically by the system.

For more information about syringe pumps, refer to the topics below.

- An introduction to syringe pumps in ANY-maze
- Syringe pumps supported by ANY-maze
- Setting up a syringe pump
- Editing a syringe pump
- Deleting a syringe pump
- Syringe pump actions
- Syringe pump measures

ANY-maze help topic H0310

An introduction to syringe pumps

Introduction

Many syringe pumps include the ability to be controlled by a computer - just connect the pump to the computer, usually with a USB cable, and the computer can send commands to the pump to tell it to start, stop, change direction, etc.

ANY-maze is able to take advantage of this facility in a range of pumps from various manufacturers - you just need to connect the pump to your ANY-maze computer, tell it what size syringe you'll be using, and the system can then control the pump during your tests. This means, for example, that the pump can be set to start and stop, or that the rate of infusion can be changed, all based on what's actually happening in the test.

Setting up a syringe pump

The first step in setting up a syringe pump for use in ANY-maze is to physically connect it; this is described in detail here.

As with all inputs and outputs supported by ANY-maze, you must then add to the protocol the I/O device the syringe pump is part of, before you add the pump itself. In fact, this will usually just be the pump, unless the pump is part of a pump chain (where multiple pumps are connected one to another), where only the first pump in the chain needs to be set up as an I/O device.

Using a syringe pump

When you add a syringe pump to your protocol you will be able to specify the pump's flow rate, an optional target volume, and whether or not the pump should be on at the start of the test. Having the pump start when the test starts can be useful, but the real power of syringe pumps comes from the fact that you can control them during a test using procedures. For example, you might create a procedure to detect when the animal moves into a certain zone, and when it does this switch the pump on so as to infuse a small volume. In fact, procedures can do more than just switch the pump on and off; they can also change the infusion rate and alter the target volume to be infused. So, for example, you could have different zones and infuse a different volume depending on which one the animal enters.

See also:

- An introduction to I/O devices
- Setting up a syringe pump

- Editing a syringe pump
- Deleting a syringe pump
- An introduction to procedures
- Syringe pump actions

Setting up a syringe pump

Introduction

For a general introduction to syringe pumps, see An introduction to syringe pumps.

To add a syringe pump to a protocol, click the *Add item* button in the ribbon bar and select *New output item > New syringe pump* from the menu which appears.

The New output item menu option is only included if the protocol mode *includes input/output.*

The New syringe pump option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the syringe pump's name
2. Specify the syringe you will be using in the pump
3. Specify whether the pump should infuse or withdraw
4. Specify the flow rate
5. Optionally specify the target volume
6. Choose whether the pump should start when the test starts
7. Specify the syringe pump's input/output port

What next?

After completing these steps, you should consider whether you want to include any analogue outputs in this protocol.

See also:

- Adding elements to a protocol
- Editing a syringe pump
- Deleting a syringe pump

Syringe pump name

In brief

Enter a name for the syringe pump in the *Syringe pump name* field on the syringe pump's settings page. You must make an entry, but it can be anything you like.

Details

It's quite rare to have more than one syringe pump in an experiment, so usually you will just call the syringe pump something like 'Pump'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying the syringe being used in a pump

In brief

In order to function, a syringe pump needs to be told the inner diameter of the syringe installed in it. ANY-maze includes an extensive library of syringes and so you just need to use the drop-down lists on the syringe pump's settings page to choose the syringe you're using and the system will set the inner diameter appropriately.

Details

ANY-maze includes a library of over 120 syringes from the major manufacturers, so in most cases you will simply need to select the syringe series and volume, and ANY-maze will automatically set the inner diameter.

However, if the syringe you're using is not listed, then you can select the option to *Specify syringe diameter manually* (it's the last option in the *syringe type* drop-down list) and then enter the *inner diameter,* in millimetres, into the appropriate field.

Limits

The inner diameter field accepts any values from 0.01mm to 50.0mm, inclusive.

Specifying whether the pump should infuse or withdraw

In brief

Many syringe pumps can both infuse and withdraw. If this is the case with your pump, you should specify the default direction you want it to move in - this can be changed during a test using a procedure.

Details

If you set an 'infuse-only' pump to withdraw, then ANY-maze will warn you and you'll have to change your selection.

Setting a syringe pump's flow rate

In brief

You should use the *Flow rate* field on the syringe pump's settings page to specify the default flow rate for the pump. You need to enter the rate and select the units from the drop-down list to the right of the field. Note that the flow rate can be changed during a test using a procedure.

Details

If you select a flow rate which is beyond the capabilities of the pump you are using, ANY-maze will warn you and you will have to change your entry.

Setting a syringe pump's target volume

In brief

You can use the *Target volume* field on the syringe pump's settings page to specify the volume you would like the pump to infuse or withdraw. After the pump starts, ANY-maze will stop it automatically when this volume is reached.

If you don't want to set a target volume, then leave this field blank - once started, the pump will continue to run until explicitly stopped by a procedure or it stalls (reaches the limit of its travel).

Details

If you do enter a target volume then you need to enter the volume and then select the units from the drop-down list to the right of the field.

Note that the target volume can be changed during a test using a procedure.

If you select a target volume which would be dispensed extremely quickly (based on the selected flow rate), then ANY-maze will ask you to confirm that your entry is correct.

Choosing whether a syringe pump should start when the test starts

In brief

You can optionally set a syringe pump to start when a test starts. This option can be very useful combined with a target volume, as the pump will switch on at the start of the test, infuse (or withdraw) the target volume, and then stop.

If you don't select this option, the only way to start the pump during the test will be using a procedure. If you don't do this either, the pump simply won't ever switch on and nothing will be infused (or withdrawn).

Editing a syringe pump

Introduction

You can edit absolutely everything related to a syringe pump at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a syringe pump
- Deleting a syringe pump

Deleting a syringe pump

Introduction

You can delete a syringe pump at any time, whether before, during or after an experiment has been performed.

To delete a syringe pump, simply select it in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting syringe pumps.

See also:

- Deleting protocol elements
- Setting up a syringe pump
- Editing a syringe pump

Syringe pump actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a syringe pump - these are detailed below.

Actions

Start the syringe pump	Starts the syringe pump in the current direction at the current flow rate. The pump will continue to run until either: a) the target volume is infused (or withdrawn), assuming a target volume has been set b) the pump is stopped by an action c) the test ends d) the pump stalls (reaches the limit of its travel). If the syringe pump is already running, this action has no effect. Starting a pump **does not** affect the current volume infused (or withdrawn). For example, if a pump has a target volume of 3ml and has so far infused 1ml when it is stopped, then when it is started again, it will continue to run until a further 2ml have been infused. If a pump which has stopped because the target volume was reached is started again, it will immediately stop (because the target has still been reached). To start the pump in these circumstances, you must either set the target volume again (using a *Set target volume* action) or clear the target volume, so there is no target (using a *Clear target volume* action).
Stop the syringe pump	Stops the syringe pump. If the syringe pump is already stopped, this action has no effect. Stopping a pump **does not** clear the current volume infused (or withdrawn). For example, if a pump has a target volume of 3ml and has so far infused 1ml when it is stopped, then when it is started again, it will continue to run until a further 2ml have been infused.
Set direction to infuse	Sets the direction of syringe pump to infuse (liquid will come out of the syringe). If the pump is currently set to infuse, this action will have no effect. If the pump is running when the direction is changed, then it will, typically, stop for a moment, change direction

and then start again in the new direction. If the pump is not running, then this action sets the direction it will move in when next started. If the pump has a target volume set, then this action will reset the current volume (the amount withdrawn so far) to zero, and the pump will run in the new direction until the target volume has been infused.

Set direction to withdraw	Sets the direction of syringe pump to withdraw (liquid will be drawn into the syringe). If the pump is currently set to withdraw, this action will have no effect. Not all syringe pumps can withdraw; if you perform this action on a syringe pump which doesn't support it, then it will have no effect. If the pump is running when the direction is changed, then it will, typically, stop for a moment, change direction and then start again in the new direction. If the pump is not running, then this action sets the direction it will move in when next started. If the pump has a target volume set, then this action will reset the current volume (the amount infused so far) to zero, and the pump will run in the new direction until the target volume has been withdrawn.
Reverse direction	Switches the direction of the pump so if it was infusing, it will start to withdraw and vice versa. Not all syringe pumps can withdraw; if you perform this action on a syringe pump which doesn't have a withdraw mode, it will have no effect. If the pump is running when the direction is changed, then it will, typically, stop for a moment, change direction and then start again in the new direction. If the pump is not running, then this action sets the direction it will move in when next started. If the pump has a target volume set, then this action will reset the current volume (the amount infused or withdrawn so far) to zero, and the pump will run in the new direction until the target volume has been infused or withdrawn.
Set flow rate	Changes the flow rate to the rate specified in the action's parameter. The flow rate is always specified in units of µl/min; if you want a rate in some other units, then you will need to convert it - for example, a rate of 3ml/min would be entered as 3000. Any target volume is unaffected by this action; in other words, the current volume infused (or withdrawn) will be retained and the pump will stop when the total volume infused (or withdrawn) reaches the target. If an action sets a flow rate that is outside a pumps limits (e.g. it would require the pump to move faster than its maximum speed), then the flow rate will become the pump's maximum speed. If the pump is running, then ANY-maze will attempt to change the flow rate without stopping it; however, if the pump does not allow this, then it will be stopped briefly, the rate will be changed and then it will start again. If the pump is not

	running, then the rate will be used the next time it starts.
Set target volume	Sets a new target volume for the pump to the volume specified in the action's parameter. The volume is always specified in units of µl; if you want a volume in some other units, then you will need to convert it - for example, a volume of 3ml would be entered as 3000. This action always resets the current volume (infused or withdrawn so far) back to zero, and then infuses (or withdraws) the entire target volume specified. If the pump is running, this action will simply change the target volume and the pump will continue to run without any interruption. If the pump is not running, then this volume becomes the target volume which will be infused or withdrawn the next time it starts.
Update target volume	Changes the target volume for the pump to the volume specified in the action's parameter. The volume is always specified in units of µl; if you want a volume in some other units, then you will need to convert it - for example, a volume of 3ml would be entered as 3000. This action does not alter the current volume (infused or withdrawn so far). For example, a pump has a target volume of 5ml and has so far infused 3ml when the target volume is updated to 10ml; the pump will continue to run until a further 7ml are infused, and it will then stop. Note that if the updated target volume has already been infused, then the pump will stop as soon as the target volume is updated. For example, a pump has a target volume of 5ml and has so far infused 3ml when the target volume is updated to 2ml - it has already infused more than 2ml, so it will immediately stop. If the pump is not running, then the adjusted target volume becomes the target volume which will be infused or withdrawn the next time it starts.
Clear target volume	Sets the pump to have no target volume, so it will then run continuously until either stopped by an action, or the test ends, or it stalls (reaches the limit of its travel).

Syringe pump measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a syringe pump:

- Volume infused
- Volume withdrawn

Volume infused

Description	The total volume infused by the syringe pump.
Calculation method	The sum of the amounts infused each time the pump was run in the infuse direction during the test. The amount infused is calculated by multiplying the infusion rate by the amount of time the pump was running.
Analysis in zones	The volume infused while the animal was in the zone.
Analysis across time	This measure can be analysed across time. The result for a time period is the total volume infused during the time period.
Units	Microlitres
Notes	None.

Volume withdrawn

Description	The total volume withdrawn by the syringe pump.
Calculation method	The sum of the amounts withdrawn each time the pump was run in the withdraw direction during the test. The amount withdrawn is calculated by multiplying the withdrawal rate by the amount of time the pump was running.
Analysis in zones	The volume withdrawn while the animal was in the zone.
Analysis across time	This measure can be analysed across time. The result for a time period is the total volume withdrawn during the time period.
Units	Microlitres
Notes	Not all syringe pumps can be run in the withdraw direction. For those that can't, the result of this measure will always be zero.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- OPAD measures
- Output switch measures
- Pellet dispenser measures
- Laser controller measures
- Procedure measures
- Event measures
- Virtual switch measures

Analogue outputs

Analogue outputs allow ANY-maze to generate a voltage which can vary during the test. This voltage can depend on the animal's location in the apparatus, or it can be set by a procedure. You can then use the voltage to control some other equipment.

For more information about analogue outputs, refer to the topics below.

- An introduction to analogue outputs
- Setting up an analogue output
- Editing an analogue output
- Deleting an analogue output
- Analogue output actions

An introduction to analogue outputs

Introduction

An analogue output is simply a voltage which can change based on conditions in a test. This in itself isn't really of any use, but if you have some equipment which can be *controlled* by a voltage *input,* then it means you can now control that equipment from ANY-maze. For example, imagine you have a laser which you're using in your tests, and the laser's power can be adjusted by applying a voltage to a connector on it - perhaps 0V means 'off', 1V means 'on at full power', and any voltage in-between means 'on at a proportional power'. If you connect this laser to an analogue output, you could then have ANY-maze adjust the laser's power depending on the situation in a test.

There are really two aspects to analogue outputs in ANY-maze: the generation of the output, and how it is used.

Generating the output

Clearly, to generate an analogue output, you will need some sort of I/O device which includes this type of output - the ANY-maze interface, for example, has two analogue output ports. But that's only half the story - you also need some way to set the output during a test, and ANY-maze provides two options for how this can be done:

- The output can simply track the animal's X or Y position in the apparatus. For example, if an output is tracking the X position, then, as the animal moves to the left, so the output's voltage will go down and as the animal moves right, so the output's voltage will go up. If you use two outputs, you can use one to track the animal's X position and one to track Y.
- The second way to set the output is by using an action in a procedure. As you may know, procedures are a series of instructions which ANY-maze performs while a test is running, so you could create a procedure which would determine what's happening in the test and alter the analogue output voltage accordingly.

  For example, imagine you have a foot shocker which has a voltage input to set the shock level, where 0.1V means 1mA shock, 0.2V means 2mA shock, and so on. Now you want to alter the shock the animal will receive based on which corner of a cage it is in. You could do this by creating four zones, one in each corner, and then writing a procedure so that each time the animal enters one of the corner zones, the analogue output would be adjusted, thus causing the shock level to change. Of course, you'd probably also want a second procedure, which would control when the shocks are actually administered.

How an analogue output is used

In the previous section, I've given a couple of examples of equipment which might have a voltage *input* to which an analogue output could be connected: a laser and a shocker. Doubtless there is other equipment which can be controlled via a voltage input, and if you're wondering whether a piece of equipment in your lab can be controlled in this way, just contact ANY-maze technical support and we'll be happy to advise you.

There's a second way an analogue output could be used, and that's to control some device you build yourself. If you're thinking of doing this, then you probably already have a good idea about what an analogue output is (it's just a DAC - a digital to analogue converter) and how you will use it, but, again, if you have questions, just contact us and we'll be happy to help.

Setting up an analogue output

Introduction

For a general introduction to analogue outputs, see An introduction to analogue outputs.

To add an analogue output to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New analogue output* from the menu which appears.

The New output item *menu option is only included if the* protocol mode *includes input/output.*

The New analogue output *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the analogue output's name
2. Specify how the output will be set
3. If the output is set based on the animal's position, then specify what the output is set relative to
4. Specify whether the physical port used by the analogue output is always the same, or whether it depends on the location of a movable zone
5. Specify the physical port used by the analogue output

What next?

After completing these steps, you should consider whether you want to include any temperature controllers in the protocol.

See also:

- Adding elements to a protocol
- Editing an analogue output
- Deleting an analogue output

Setting the name of an analogue output

In brief

Enter a name for the analogue output in the *Analogue output name* field on the analogue output's settings page. You must make an entry, but it can be anything you like.

Details

Analogue output names should specify what the output is intended to do, so for example it might be called 'Laser power', 'Shock intensity' or 'Animal X position'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying how an analogue output will be set

In brief

An analogue output can either be explicitly set to some voltage by an action within a procedure, or it can simply track the animal's position (so the voltage is directly proportional to the animal's X or Y coordinate).

Details

There are three options for how an analogue voltage will be set:

Set by an action	If you select this option, then you'll be able to use an action in a procedure to explicitly set the voltage to some value. This provides enormous flexibility, as the voltage can be based on essentially anything and can be updated under any circumstances you choose.
Track animal's X position	Using this option is very simple - just select it and the output will automatically track the animal's X position throughout the test. If you use this option, you will also need to specify what the output voltage is relative to, either the apparatus or the video picture. This will affect exactly what voltage is output - the relationship of position to voltage is explained in the topic which explains these options.
Track animal's Y position	This is essentially identical to tracking the animal's X position, except that the voltage is tracking Y - the same considerations mentioned above apply here too.

If you want to track both X and Y, then you will need to set up two analogue outputs, one for each.

Specifying what the animal's position should be output relative to

In brief

If you choose to set an analogue output to output the animal's X or Y position, then you will also need to specify what the voltage should be relative to: the size of the apparatus, or the size of the video picture.

Details

These options are easiest understood through an example. Imagine you are tracking animals in the video picture shown in figure 1, below.

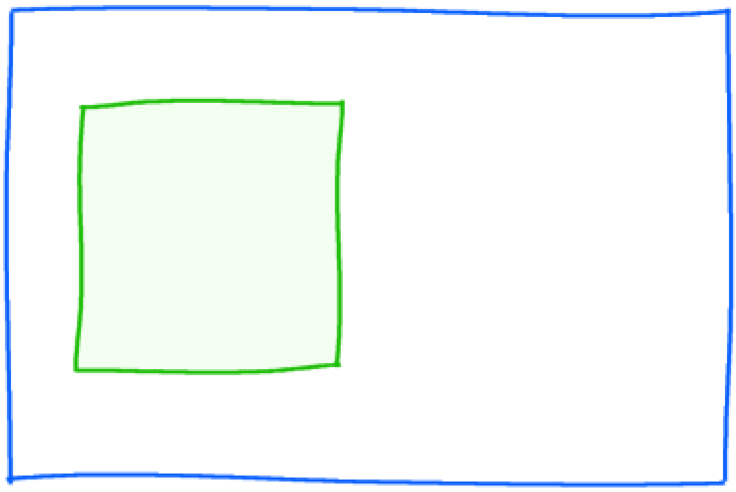


Figure 1. A 640x480 pixel video picture (the blue outline) with a 40cm x 40cm open field in it (the green shaded area).

Here, the apparatus is 40cm x 40cm and the picture showing it is 640x480 pixels. Now we want to create an analogue output which will track the animal's X position within the apparatus, so what voltage should actually be output? Well, the answer depends:

If you choose to output the voltage relative to the *size of the apparatus,* then ANY-maze will output the minimum voltage when the animal is on the far left of the apparatus and the maximum voltage when it is on the far right. As the animal moves left to right, so the voltage will increase linearly, in direct proportion to how far across the apparatus it is.

For example, if the analogue output's range was 0V - 4.095V, and the animal was 10cm from the left edge of the apparatus in figure 1, then the output would be 1.024V.

The same thing applies if the output is tracking the animal's Y position rather than X, the only difference is that the minimum voltage is output when the animal is at the bottom edge of the apparatus (closest to the bottom of the screen) and the maximum is output when it is at the top edge.

The other option is to output the voltage relative to the *size of the video picture*. This is very similar to the previous option, but the voltage range is spread across the image's width, rather than the apparatus width. Note, however, that the mid-point voltage is still output when the animal is at the centre of the apparatus. For example, with the animal 10cm from the left edge of the apparatus in figure 1, we would have the voltage range of 0-4.095V spread across 640 pixels, so each pixel would equate to a voltage step of 0.0064V. The centre of the apparatus would be a voltage of 2.048V. A position 10cm from the left edge of the apparatus would be around 60 pixels left of the centre of the apparatus, so the output would be 2.048 - (0.0064 x 60) = 1.664V.

As you might imagine, the first option is usually the simplest, but the second option has the advantage that the voltage output is normalised by the image width, i.e. with an image that's 640 pixels wide and an analogue output with a range of 0-4.095V, one pixel is always 0.0064V.

Specifying whether the physical port used by an analogue output is always the same

In brief

In the majority of situations, the physical port used by an analogue output will always be the same - it's simply the port on the interface device which outputs the analogue signal.

However, you may encounter situations in which the port depends on the location of a movable zone. For example, in a Y-maze you may have three shockers connected to the floor of each arm of the maze, and the shockers' intensity might be controlled by an analogue output.

Now you want to create an analogue output called 'Known arm shock intensity' (because you want to alter the shock intensity in the 'Known arm' during the test), but you have set the 'Known arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests.

Thus the 'Known arm shock intensity' will be the analogue output which connects to the shocker/floor in whichever arm happens to be the 'Known arm' in a particular test, which necessarily means it won't always be the same physical shocker, and therefore the 'Known arm shock intensity' analogue output won't always be connected to the same physical port.

Details

If you do have an analogue output whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the analogue output's setting in the protocol. (Note that this drop-down list is only displayed when the protocol includes one or more movable zones).

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing an analogue output

Introduction

You can edit absolutely everything related to an analogue output at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up an analogue output
- Deleting an analogue output

Deleting an analogue output

Introduction

You can delete an analogue output at any time, whether before, during or after an experiment has been performed.

To delete an analogue output, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting analogue outputs.

See also:

- Deleting protocol elements
- Setting up an analogue output
- Editing an analogue output

Analogue output actions

Introduction

As you may know, a procedure can include *actions,* some of which can alter an analogue output - these are detailed below.

Actions

Set analogue output level	Changes the voltage at the analogue output to some new value specified as a parameter to the action. The parameter is in Volts. If the parameter falls outside the analogue output's range, the output will be clipped to be within range. For example, if an analogue output with a range of 0V - 4.095V is set to 5.65V, then the output will simply be set to 4.095V. Note that when an analogue output changes value, it reports its new value back to the procedure, so in this example, after setting the analogue output to 5.65V the *analogue output*'s *value* variable would actually be 4.095, not 5.65. When *set,* an analogue output will change voltage immediately; it won't 'ramp' from the old value to the new.

Temperature controllers

As the name implies, temperature controllers are used to control the temperature of something - a good example is the thermal elements in the OPAD cage.

For more information about temperature controllers, refer to the topics below.

- An introduction to temperature controllers
- Setting up a temperature controller
- Editing a temperature controller
- Deleting a temperature controller
- Temperature controller actions

An introduction to temperature controllers

Introduction

A temperature controller is usually part of some I/O device such as the OPAD cage, and is used to heat something up or cool it down. For example, in the OPAD cage, the temperature controller is used to adjust the temperature of the thermal stimulus elements which touch the animal's cheeks while it is licking the bottle spout.

An important characteristic of temperature controllers is the fact they can be adjusted by procedures during a test - thus the temperature can be adjusted based on what is happening in the test.

Connecting a temperature controller

Temperature controllers are not available as 'standalone' devices; they are always part of some I/O device, such as the OPAD cage. Therefore, connecting them to your PC is very simple - you just need to connect the device they're part of (usually using a USB cable).

As with all inputs or outputs supported by ANY-maze, you must add to the protocol the I/O device that the temperature controller is part of, before you add the controller itself.

Using a temperature controller

When you add a temperature controller to your protocol, you will be able to specify the temperature the controller should have at the start of the test. Of course this is useful, but the real power of temperature controllers comes from the fact that you can change their temperature during the test. This is done using a procedure, which can detect that something happens in the test and alter the temperature accordingly. For example, you might create a procedure to detect when the animal has been in contact with the thermal elements in the OPAD cage for a certain amount of time and then ramp the temperature from 20°C to 40°C over 30 seconds.

See also:

- An introduction to I/O devices
- Setting up a temperature controller
- Editing a temperature controller
- Deleting a temperature controller
- An introduction to procedures
- Temperature controller actions

Setting up temperature controllers

Introduction

For a general introduction to temperature controllers, see An introduction to temperature controllers.

To add a temperature controller to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New temperature controller* from the menu which appears.

The New output item menu option is only included if the protocol mode *includes input/output.*

The New temperature controller option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the temperature controller's name
2. Specify the temperature controller's start temperature
3. Optionally set the temperature controller to ramp to a different temperature during the test (not available for the OPAD temperature controller)
4. Specify the maximum and minimum temperatures that the controller can assume
5. Specify whether the physical port used by the temperature controller is always the same, or whether it depends on the location of a movable zone
6. Specify the temperature controller's input/output port
7. For the OPAD 'thermal element' temperature controller, optionally set up a temperature cycle

What next?

After completing these steps, you should consider whether you want to include any lighting controllers in this protocol.

See also:

- Adding elements to a protocol
- Editing a temperature controller
- Deleting a temperature controller

Temperature controller name

In brief

Enter a name for the temperature controller in the *Temperature controller name* field on the temperature controller's settings page. You must make an entry, but it can be anything you like.

Details

It's recommended that temperature controller names are based on what the temperature controller will be heating/cooling. So this might be something like 'Thermal stimulus' or 'Floor plate'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying a temperature controller's start temperature

In brief

You should use *Temperature at start of test* field on the temperature controller's settings page to specify the temperature of the controller at the start of the test. You can also optionally specify that the test shouldn't start until the controller is at this temperature.

Details

When ANY-maze prepares to perform a test, it will set the temperature controller to the temperature that you specify here.

In some experiments you may not want the start temperature to be the same in all trials - for example, you might want to alternate the start temperature between 37°C and 45°C in each trial. In this case you can override the start temperature specified here using a procedure - as in figure 1, below.

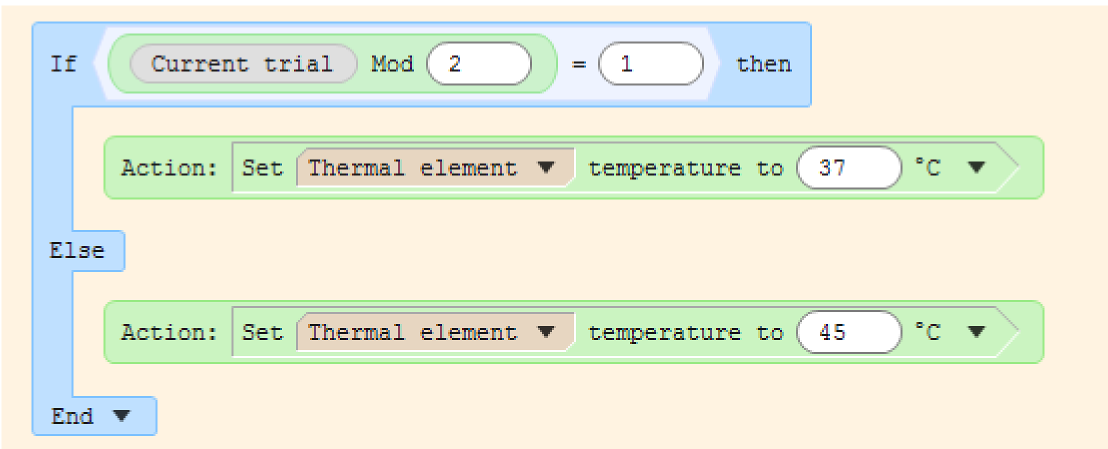


Figure 1. This procedure will alternate the start temperature of the 'Thermal element' between 37°C and 45°C in each trial. Note that the statements are all in the shaded part at the top of the procedure, which means they run as soon as the test is ready.

If you use a procedure to set the start temperature in this way, then the value you enter here will still be used as the *default* start temperature, i.e. the temperature that will be used if the procedure doesn't explicitly set it to some other value.

How quickly the actual thing being heated or cooled will take to reach the start temperature will depend on the device, but you can optionally specify that ANY-maze should prevent the test from starting until this temperature is reached. If you do this, then rather than displaying *Ready...* as the test status, ANY-maze will display something like *Waiting for thermal stimulus to reach start temp. of 60°C : X°C* where, in this example, 'Thermal stimulus' is the name of the temperature controller and X is the current temperature. ANY-maze will actually wait until the temperature has not only reached the start temperature but has stabilized (remained at the start temperature for a few seconds).

Note that ANY-maze can't know whether the temperature you specify is actually achievable, so if you specify, for example, that the start temperature is 70°C, but the device can't reach this temperature, then your test will never be ready to start.

If you choose *not* to wait for the device to be at the start temperature, then ANY-maze will set the temperature controller to the start temperature when it prepares the test, but you will be able to start the test immediately, even if it has not reached the correct temperature yet.

Setting a temperature controller to ramp to a different temperature during the test

The ramp settings described in this topic are not available for the OPAD temperature controller - they are replaced by the ***OPAD temperature cycle*** *settings.*

In brief

You can use *Ramp* fields on the temperature controller's settings page to specify that the temperature of the controller should ramp from the start temperature to some other temperature over the programmed duration of the test. The temperature will start ramping as soon as the test begins.

Details

If you select this option, then the temperature controller will 'ramp' from whatever temperature it has at the start of the test (which you can specify using the start temperature field) to the value you specify. The ramp will be calculated so that the temperature at the *programmed* end of the test will be the value you specify, and the ramp will be linear. For example, if you specify that the start temperature is 20°C and the end temperature is 30°C, and the test is programmed to last 100 seconds, then the temperature will change 0.1°C each second.

There are a few things to be aware of:

- The test may not last for the programmed duration; for example, it might be ended by an procedure before this - in which case, the temperature will never reach the value you set.
- The device may not be able to reach the temperature you specify, in which case it will just get as hot or cold as it can.
- While the temperature will in theory ramp in a linear way, the device will actually 'step' the temperature. The size of the steps will depend on the device, so for example in a device which can only set the temperature to a resolution of 1°C, then the steps will be 1°C and so, in the case of ramping from 20°C to 30°C during a 100 second test, the device will actually change temperature by 1°C every 10 seconds.

Specifying the maximum and minimum temperatures that a temperature controller can assume

In brief

Temperature controllers will always have some physical limit to the temperatures that they can achieve, but these might be temperatures that are much hotter or colder than you would like an animal to be exposed to. So to safeguard the animals in your tests, you must specify the maximum and minimum temperatures a temperature controller can ever assume. No matter what happens in the test, ANY-maze will ensure that the temperature controller is never set to a temperature outside the range you enter.

Details

Depending on the actual design of a temperature controller, it's possible that the device's temperature could reach quite high (or low) temperatures. For example, a temperature controller might be physically capable of cooling something to below freezing; however, you would probably never want to expose an animal to such extreme cold. To ensure this doesn't happen, you could specify that the temperature controller's minimum temperature is, for example, 5°C - this would mean that ANY-maze would never set it to a value below this.

You may be wondering why this is necessary; after all, if you specify the temperature at the start of the test and you specify the 'ramp' temperature at the end of the test, then you could simply make sure you don't set either value to below 5°C. This is true, but the temperature can also be set by procedures and you could create a procedure which would take the temperature to some unknown value - this is best explained using an example:

Imagine you have an apparatus where the animal can press a lever and receive a reward, but each time it does this the floor it is standing on gets colder by 1 degree. At the start of the test, the floor is at 25°C - now the animal presses the lever 30 times. In theory, the floor would go down to -5°C (and the device might even be able to achieve this temperature) - but you would probably not want to freeze your animal's feet! So if you had set a limit of 5°C, then it wouldn't matter how many times the animal presses the lever; once the temperature gets down to 5°C, ANY-maze wouldn't reduce it any further.

Specifying whether the physical port used by a temperature controller is always the same

In brief

In the majority of situations, the physical port used by a temperature controller will always be the same. However, you may encounter situations in which the port depends on the location of a movable zone.

For example, in a Y-maze, you may have a hot/cold plate mounted in the floor of each arm of the maze.

Now you want to create an element called 'Aversive arm temperature controller' (let's imagine that the animal is expected to leave the arm when the temperature of the plate gets above a certain value), but you have set the 'Aversive arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests.

Thus, the 'Aversive arm temperature controller' will be the temperature controller for whichever arm happens to be the aversive arm in a particular test, which necessarily means it won't always be the same physical temperature controller and therefore the 'Aversive arm temperature controller' won't always be connected to the same physical port.

Details

If you do have a temperature controller whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the temperature controller element's page of the protocol. (Note that this list is only displayed when the protocol includes one or more movable zones).

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Setting up OPAD cage temperature cycles

The settings described in this topic are only available for the OPAD temperature controller.

In brief

When setting up the temperature controller in an OPAD cage, you will find that the temperature controller element in the protocol list includes a sub-element titled 'OPAD temperature cycle', which can be used to program a 'temperature cycle' - see figure 1.

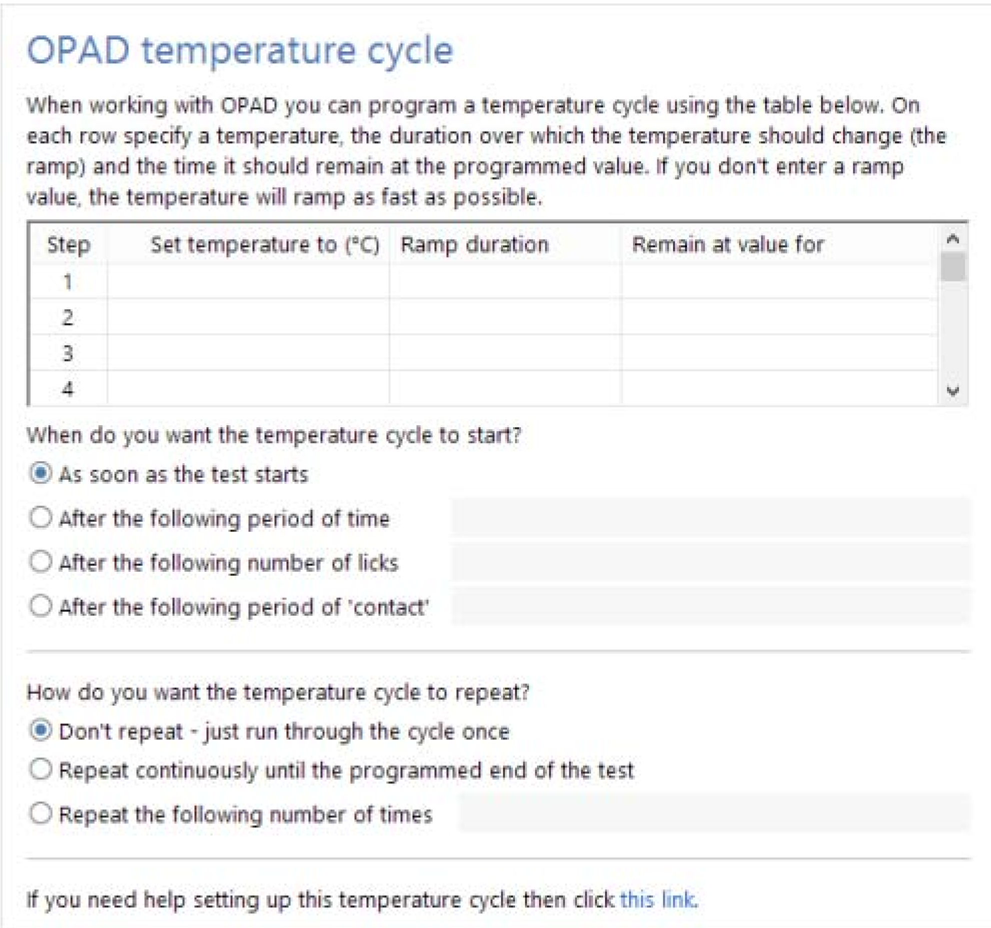


Figure 1. The OPAD temperature cycle settings page.

A temperature cycle is simply a series of temperatures that the controller should assume. For example, you might want the temperature controller to be at 30°C for 1 minute, then heat up to 50°C over the course of 30 seconds, remain at 50°C for 1 minute, and then cool back down to 30°C, again over the course of 30 seconds. You might want this to happen just once, or you might want it to repeat again and again, throughout the test. It is this sort of 'cycle' which you can set up using this option.

Details

A temperature cycle is set up using a simple table with three columns: *Temperature, Ramp duration, Remain at value for* (see figure 1, above). For example, entering 50, 30s, 1min would ramp the temperature from whatever value it has now to 50°C over the course of 30s, and then remain at that temperature for 1 minute. After the 1 minute, the next step (or row in the table) would be performed - this might be 10, 5min, 1min - which would ramp the temperature down (from 50°C) to 10°C over the course of 5 minutes and then remain at 10°C for 1 minute.

Clearly, the temperature at the start of one step is whatever the temperature was at the end of the previous step, but what about the temperature at the start of the first step? This will be whatever value you specified as the temperature controller's start temperature.

An important thing to understand about temperature controllers is that they can't alter the temperature of something instantly - so entering a value of 0 for ramp duration does NOT mean that the temperature will jump immediately to the new temperature - it will just cause the temperature to change as fast as possible. In fact, if you want the temperature to change as fast as possible, then you should just leave out the ramp value entirely.

Specifying when the cycle should start

Unless you specify otherwise, the temperature cycle will start as soon as the test begins, so the temperature will immediately start to change from the 'Start temperature' to whatever temperature is specified in the first step. However, you can alter this:

- You can specify that the temperature cycle should only begin after a certain period - for example, 1 minute (which you would enter as 1min).
- You can specify that the cycle should begin after the animal has licked the bottle spout a certain number of times - for example, after 100 licks.
- You can specify that the cycle should begin after the animal has been in contact with the thermal stimulus elements for a certain period - for example, 20 seconds.

Repeating the temperature cycle

Normally, a temperature cycle will be performed just once, and the temperature will end up at whatever the value is specified in the last step. However, if you wish, you can specify that the cycle should repeat, either a certain number of times, or continuously (until the test ends). Note that when a cycle repeats the criteria specified for when it should start are NOT repeated - for example, if you specify that the cycle should start after 100 licks and then you set it to repeat, then the FIRST cycle will only start once the animal has licked 100 times, but then the cycle will just keep repeating continuously, irrespective of whether the animal licks the bottle spout or not.

Editing a temperature controller

Introduction

You can edit absolutely everything related to a temperature controller at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a temperature controller
- Deleting a temperature controller

Deleting a temperature controller

Introduction

You can delete a temperature controller at any time, whether before, during or after an experiment has been performed.

To delete a temperature controller, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting temperature controllers.

See also:

- Deleting protocol elements
- Setting up a temperature controller
- Editing a temperature controller

ANY-maze help topic H0343

Temperature controller actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a temperature controller - these are detailed below.

Actions

Set temperature	Sets the temperature of the temperature controller to the value specified in the action's parameter. The parameter is in units of degrees Celsius. The temperature will begin to change immediately from the current temperature to the new temperature, using the current ramp duration. If the temperature was already changing when the action occurs (for example, as a result of a previous action), then the old change is discarded and the temperature will ramp from whatever the temperature is NOW to the new temperature, over the current ramp duration. If a temperature is passed which is outside the temperature controller's range, then the temperature will be adjusted to be the maximum or minimum, as appropriate, and the temperature controller will start to adjust to THAT temperature over the current ramp duration.
Set ramp duration	Sets the duration over which the temperature controller will change from one temperature to another. The duration can be specified with units of s (seconds), min or m (minutes), or h (hours); if no units are given, then seconds will be assumed. If the ramp duration is too short, for example, trying to change from 10°C to 70°C in 1s, then the controller will ramp as fast as it can. If the duration specified is outside the durations the controller supports, then it will be set to the maximum or minimum duration, as appropriate. For example, the maximum ramp duration supported by OPAD is 2 hours; setting it to 3h will cause it to be set to 2h instead. If you want to create a specific slope, for example 1°C/min, then you can use the temperature controller's *Thermal element temperature* variable in your procedure to determine what the temperature is now, and then set the ramp duration to achieve your target temperature with the desired slope.

If you set the ramp duration while the controller is ramping, the new duration will only be used the **next time** you change the temperature.

Lighting controllers

As the name implies, lighting controllers are used to control the brightness of lights; effectively, they are like a dimmer switch. A good example is the cage lights in an ANY-maze cage.

For more information about lighting controllers, refer to the topics below.

- An introduction to lighting controllers
- Setting up a lighting controller
- Editing a lighting controller
- Deleting a lighting controller
- Lighting controller actions

An introduction to lighting controllers

Introduction

A lighting controller is usually part of some I/O device such as the ANY-maze cage, and is used to control the brightness of one or more lights. For example, in the ANY-maze cage, the lighting controller is used to adjust the brightness of the cage lights.

An important characteristic of lighting controllers is the fact they can be adjusted by procedures during a test - thus the light level can be adjusted based on what the animal is doing in the test.

Connecting a lighting controller

Lighting controllers are currently not available as 'standalone' devices; they are always part of some I/O device such as the ANY-maze cage. Therefore connecting them to your PC is very simple - you just need to connect the device they're part of (usually using a USB cable).

As with all inputs and outputs supported by ANY-maze, you must add to the protocol the I/O device the lighting controller is part of, before you add the controller itself.

Using a lighting controller

When you add a lighting controller to your protocol, you will be able to specify the light level at the start of the test. Of course this is useful, but the real power of lighting controllers comes from the fact that you can change the light level during a test. This is done using procedures, which can detect when something happens in a test and alter the lighting because of it. For example, you might create a procedure to detect when the animal moves into a certain zone, and make the cage lights much brighter.

See also:

- An introduction to I/O devices
- An introduction to procedures
- Setting up a lighting controller
- Editing a lighting controller
- Deleting a lighting controller
- Lighting controller actions

Setting up a lighting controller

Introduction

For a general introduction to lighting controllers, see An introduction to lighting controllers.

To add a lighting controller to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New lighting controller* from the menu which appears.

The New output item menu option is only included if the protocol mode *includes input/output.*

The New lighting controller option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the lighting controller's name
2. Specify the light level at the start of the test
3. Specify whether the physical port used by the lighting controller is always the same, or whether it depends on the location of a movable zone
4. Specify the lighting controller's input/output port

What next?

After completing these steps, you should consider how you want ANY-maze to schedule tests.

See also:

- Adding elements to a protocol
- Editing a lighting controller
- Deleting a lighting controller

Lighting controller name

In brief

Enter a name for the lighting controller in the *Lighting controller name* field on the lighting controller's settings page. You must make an entry, but it can be anything you like.

Details

It's recommended that lighting controller names are based on what the lights being controller actually do - for example, 'Cage lights' or 'Cue light' would be good names.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying a lighting controller's start level

In brief

You should use the *Light level at start of test* field on the lighting controller's settings page to specify the light level at the start of the test, in lux.

Here's a very approximate guide to lux values:

- 0.01 - 5 Very dim (full moon on a cloudless night is around 0.5 lux)
- 5 - 20 Dim (street lighting is usually around 5 - 20 lux)
- 20 - 200 Normal (a living room is typically around 100 lux)
- 200 - 1000 Light (typical office lighting is around 250 lux)
- >1000 Bright (TV studio lighting is around 1000 lux)

Details

When ANY-maze prepares to perform a test, it will set the lighting controller to the level that you specify here.

Note that ANY-maze can't know whether the level you specify is actually achievable, so you could specify for example that the start light level should be 2000 lux, but if the device can't get this bright, then it will simply be set to its brightest level.

Specifying whether the physical port used by a lighting controller is always the same

In brief

In the majority of situations, the physical port used by a lighting controller will always be the same. However, you may encounter situations in which the port depends on the location of a movable zone.

For example, in a Y-maze, you may have a light mounted over each arm of the maze. Now you want to create an element called 'Aversive arm lighting controller' (let's imagine that the lighting in the aversive arm will be 1000 lux, while in the other arms it will be 20 lux), but you have set the 'Aversive arm' to be a movable zone - i.e. it isn't always the same physical arm of the maze, but changes between tests.

Thus the 'Aversive arm lighting controller' will be the lighting controller for whichever arm happens to be the aversive arm in a particular test, which necessarily means it won't always be the same physical lighting controller and therefore the 'Aversive arm lighting controller' won't always be connected to the same physical port.

Details

If you do have a lighting controller whose physical port depends on the location of a movable zone, like in the Y-maze example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the lighting controller element's page of the protocol. (Note that this list is only displayed when the protocol includes one or more movable zones).

When you do this, the element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the Y-maze example, there will be three 'Port to use' sub-elements, one for each arm of the maze. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a lighting controller

Introduction

You can edit absolutely everything related to a lighting controller at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a lighting controller
- Deleting a lighting controller

Deleting a lighting controller

Introduction

You can delete a lighting controller at any time, whether before, during or after an experiment has been performed.

To delete a lighting controller, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting lighting controllers.

See also:

- Deleting protocol elements
- Setting up a lighting controller
- Editing a lighting controller

ANY-maze help topic H0353

Lighting controller actions

Introduction

As you may know, a procedure can include *actions,* some of which can affect a lighting controller - these are detailed below.

Actions

Switch the lighting controller on	Turns the lights on at the current light level. If a ramp duration has been set, then the lights will ramp to the specified level over the specified duration, otherwise they'll come on instantly. If the lights are already on, this action will have no effect.
Switch the lighting controller off	Turns the lights off. If a ramp duration has been set, then the lights will ramp from their current level to off over the specified duration, otherwise they'll switch off instantly. If the lights are already off, this action will have no effect.
Set light level	Changes the light level of the lights to the value specified in the action's parameter. The parameter is in lux. If the lights are on, then if there is no ramp duration set, they will change instantly to the new level; otherwise, if there is a ramp duration set, they will change from the current level to the new level over the specified duration. If the lights are off, then the level will be noted and next time they are switched on, they will be at this level. This action will not switch the lights on.
Set ramp duration	Sets the duration over which the lighting controller will change from one light level to another. The duration can be specified with units of ms (milliseconds), s (seconds), min or m (minutes), or h (hours); if no units are given, seconds will be assumed. To set no ramp, so changes to light levels are instant, use a value of 0. If you want to create a specific slope, for example 10lux/s, then you can use the lighting controller's *light level* variable in your procedure to determine what the light level is now, and then set the ramp duration to achieve your target light level with the desired slope. If you set the ramp duration while the controller is ramping, the new duration will only be used the **next time** you change the light level.

ANY-maze help topic H0354

Selectors

Selectors allow ANY-maze to control multiple output ports on an I/O device (such as the AMi-2 Digital interface). The selector can be defined so that a single output value represents a given combination of these output ports being active. When the selector is set to that value, all the relevant physical ports will be changed.

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Selectors will not appear in the protocol list unless the Ugo Basile Operon protocol mode is selected.

For more information about selectors, refer to the topics below.

- An introduction to selectors
- Setting up a selector
- Editing a selector
- Deleting a selector
- Selector actions

An introduction to selectors

Introduction

A selector in ANY-maze is an output port that aggregates a number of TTL outputs (and optionally, a TTL input used as for 'feedback'). Changing the selector value will cause multiple outputs to be changed at the same time.

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Selectors will not appear in the protocol list unless the Ugo Basile Operon protocol mode is selected.

Physically connecting a selector to ANY-maze

Selectors are supported by the AMi-2 Digital interface, and full details of how to configure a selector can be found here.

See also:

- The AMi-2 Digital interface
- Setting up a selector
- Editing a selector
- Deleting a selector
- Selector actions

Setting up a selector

Introduction

For a general introduction to selectors, see An introduction to selectors.

To add a selector to a protocol, click the *Add item* button in the ribbon bar and select *New output item* > *New selector* from the menu which appears.

The New selector *menu option is only included if the* protocol mode *allows this type of port to be added. Currently, there are no protocol modes that support this.*

The New selector *option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the selector's name
2. Enter names for the possible values of the selector
3. Specify whether the physical port used by the selector is always the same, or whether it depends on the location of a movable zone
4. Specify the physical port used by the selector

What next?

After completing these steps, you should consider how you want ANY-maze to schedule tests.

After completing these steps, you should review the First stage that ANY-maze has automatically included in this protocol and also enter the duration of the tests in the stage. Then you should consider whether you want to add any further stages.

See also:

- Adding elements to a protocol
- Editing a selector
- Deleting a selector

ANY-maze help topic H0357

Setting the name of a selector

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

In brief

Enter a name for the selector in the *Selector name* field on the selector's settings page. You must make an entry, but it can be anything you like.

Details

Selector names should specify what the selector is intended to do, so for example it might be called 'Light selector' or 'Olfactory valve'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Setting names for the different values of a selector

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

In brief

Each value that a selector can take can have a text value associated with it. For example, a selector which represents a bank of different coloured LEDs could have text values of "Red", "Green", "Orange" etc. to make it clearer which LED is which. This makes it easier to refer to the selector's values from within a procedure.

Details

The apparatus you're using will determine whether you can enter your own text values for a selector, and whether you can edit existing values. For example, the names of the LEDs on a Ugo Basile Operon are pre-defined, and cannot be edited; however, the values associated with the odours in the Operon's Olfactory add-on *can* be entered yourself, because the odour being used depends on what you put in each of the pumps.

If the apparatus allows it, then you can enter some text for each possible value of a selector. For example, if you have a 4-port selector where each port is connected to a different colour LED, you might define their values as follows:

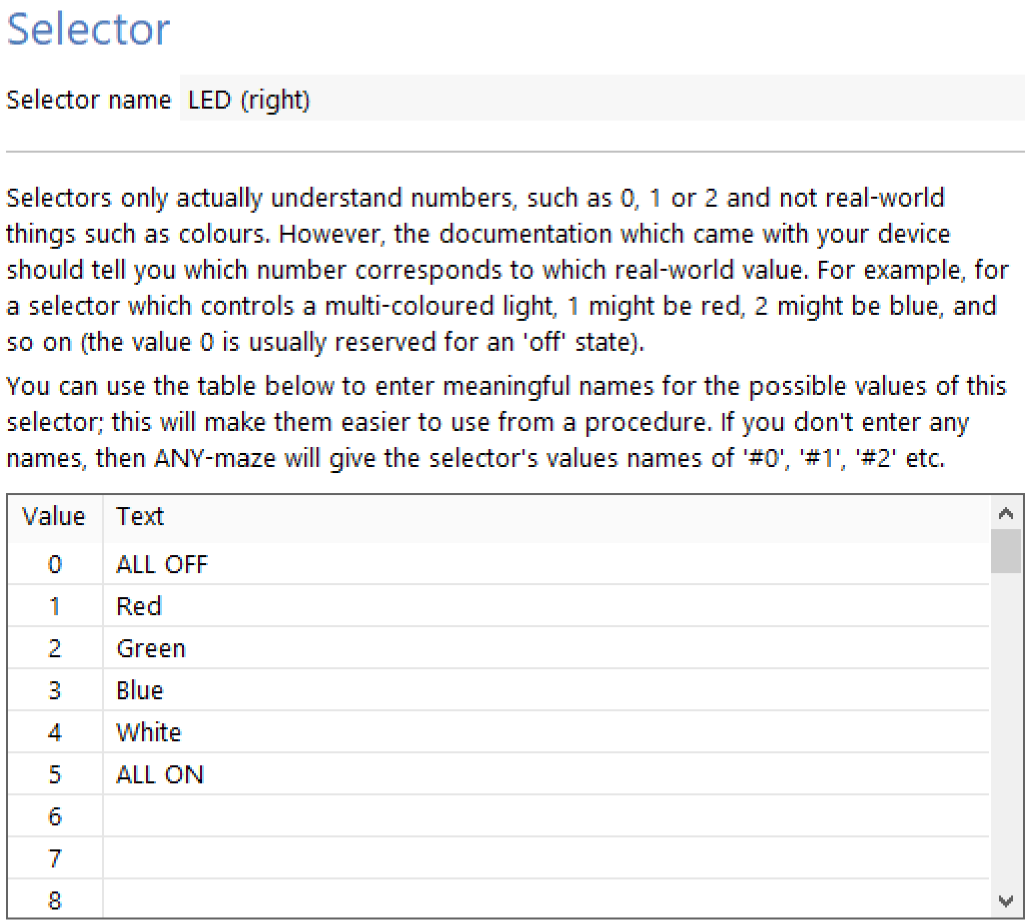


Figure 1. A selector which has been given appropriate names for the LEDs that will be activated when the selector's value is changed.

Note that in the example above, although there are only four output ports, there are actually six values defined. The values from 1 to 4 will activate single LEDs; the value of 0 will turn off all LEDs, and the last value will turn on all LEDs simultaneously. The definition of which values will activate which ports is specified in the selector's configuration).

The names of the selector's values should reflect what the selector will do when each specific value is selected.

Limits

You can enter anything you like for each selector value, up to a maximum of 47 characters. Names must be unique within this selector's values.

The value names may not be editable if the apparatus has pre-defined names for the selector's values - for example, the LEDs on a Ugo Basile Operon.

ANY-maze help topic H0359

Specifying whether the physical port used by a selector is always the same

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

The option described in this topic is only shown when a protocol includes at least one movable zone.

In brief

In the majority of situations, the physical port(s) used by a selector will always be the same - it's simply the set of ports on the interface device to which the selector actually connects.

However, you may encounter situations in which the selector's port depends on the location of a movable zone. For example, in a Place preference box, you might have two selectors, one in either side of the box. Now you decide to associate one side of the box with a drug, but you choose to balance this between animals, so for some animals the left side is the drug-associated side and for other animals it's the right side.

If you wanted to set the value of the selector in the 'drug-associated side', you would have a problem, because for some animals it would be the selector on the left side, and for others it would be the selector on the right. Since these two devices would be connected to two different physical ports, the physical port would not always be the same.

Details

If you have a selector whose physical port depends on the location of a movable zone, like in the Place preference box example above, then you simply need to specify this by selecting the appropriate option from the drop-down list in the selector's *settings page*. (Note that this drop-down list is only included if the protocol includes at least one movable zone).

In the case of the Place preference box example, you would specify that *The port varies depending on the location of the drug-associated zone.*

When you do this, the selector element will automatically be given one 'Port to use' sub-element for each location of the movable zone you chose - so in the example there would be one sub-element for 'Port to use when drug-associated zone is in left position' and a second one for 'Port to use when drug-associated zone is in right position'. You'll then be able to use these sub-elements to specify which port is used for each of the possible locations of the zone.

Editing a selector

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Introduction

You can edit absolutely everything related to a selector at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a selector
- Deleting a selector

Deleting a selector

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Introduction

You can delete a selector at any time, whether before, during or after an experiment has been performed.

To delete a selector, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting selectors.

See also:

- Deleting protocol elements
- Setting up a selector
- Editing a selector

Selector actions

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Introduction

As you may know, a procedure can include *actions,* some of which can alter a selector - these are detailed below.

Actions

Set selector value

Changes the value of the selector to some new value specified as a parameter to the action.

This value can be set using one of the named values for the selector (these values will be listed as protocol items on the *Variables* tab of the procedure editor. You can also use a numeric value, by storing the value in a variable. Examples of these methods of controlling a selector are shown below.

If the parameter falls outside the selector's range, the output will not be set.

Note that when a selector changes value, it reports its new value back to the procedure, so in the first example below, after setting the selector's value to 'Red', the Selector value changed event will fire.

Figure 1. This procedure sets the selector (which controls a range of coloured LEDs) to the 'Red' LED. 'Red' has been set up as one of the selector's named values.

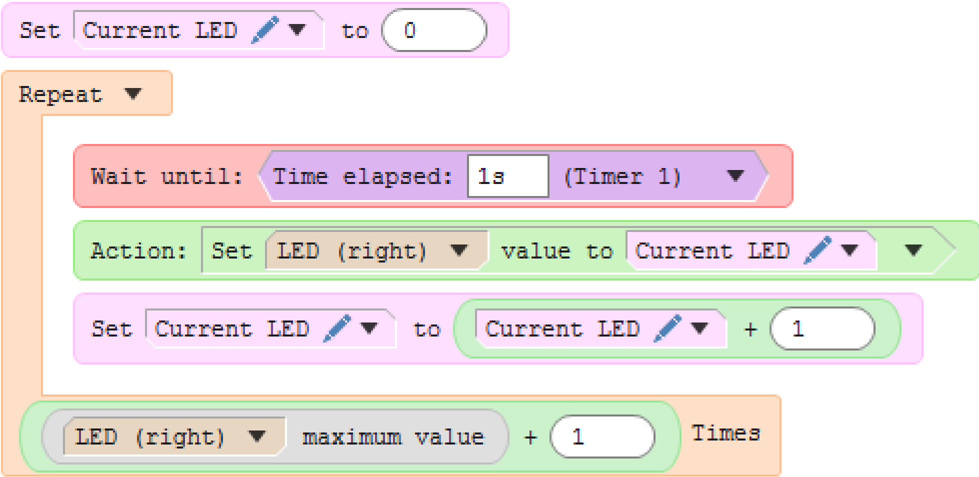


Figure 2. This procedure cycles through all possible values of the same LED in one-second intervals, using a numeric value.

See also:

- Setting up a selector
- Editing a selector
- Setting names for the different values of a selector

Test scheduling

Introduction

In the majority of experiments, the information you provide in the protocol is enough to define the order in which tests will be performed. This has the advantage that ANY-maze is then able to prepare a Test schedule, showing exactly which test on which animal you should perform next.

However, you may encounter situations in which the test order is simply too complex to be defined within the program, in which case you can opt to manually schedule the tests. A typical case when this is necessary is when you don't actually know beforehand when the animals will be tested - for example, they're tested soon after birth and you can't predict when they'll be born.

Although you may be tempted to simply choose manual scheduling in all cases (because it seems to make things more flexible), we don't recommend you do this as having ANY-maze organise the schedule for you is easier and helps to avoid mistakes.

Letting ANY-maze automatically schedule the tests

The first fundamental of test order is that the animals in an experiment are numbered, and animal 1 will be tested before animal 2. Nevertheless, there are still a number of options available to you.

In most experiments, you will probably want to perform the first stage for all the animals before proceeding on to the second stage - in a water-maze, for example, you would probably complete a 'Training' stage *for all the animals* before starting a 'Probe' stage. Nevertheless, there may be cases when you want to run through all the stages for one animal before you start the first stage for the second animal, and ANY-maze includes an option which allows you to do this.

Assuming you will complete one stage for all the animals before starting the next one, there's still the issue of the test order within stages which consist of more than one trial - for this, ANY-maze includes three possibilities:

1. You test all the animals through the first trial before starting the second trial.
2. You test the first animal through all its trials before starting the second animal.
3. This option is a little more complicated and allows you to group the animals and then test them within their groups. For example, you could put the animals into groups of say three. Thus animals 1,2 and 3 would be in the first group, animals 4,5 and 6 in the second, and so on. *Within* the first group, all the animals would then be tested in their first trial, then in their second trial, then their third etc. When the first group has finished then the second group will start. Thus this grouped system is effectively a mix of the previous two.

One last point about test order - when you actually run the tests, you can make alterations to the test order if you need to.

Scheduling tests manually

If the automatic scheduling won't suit your experiment, you can choose to *manually schedule tests* using the *Test scheduling* element in the protocol list. If you do this, you *will* still need to specify the stages your experiment will include and the number of trials there will be in each stage, however, ANY-maze won't actually assign any tests to the animals, you'll have to do this manually.

To manually schedule a test, you simply have to Add a test to the experiment, specifying which animal the test will be performed on. ANY-maze will expect you to perform the tests in the order you enter them. So, for example, if you add a test for animal 10 then one for animal 1 and then one for animal 3, you will test the animals in that order - 10, 1, 3. As you perform the tests for an animal, ANY-maze will simply assume that the first test the animal has is the first trial in the first stage, the second test is the second trial in the first stage (or the first trial in the second stage if the first stage only has 1 trial) and so on.

As mentioned above, it's usually easier to allow ANY-maze to schedule tests unless your schedule is simply too complex to be defined within the program.

What next?

After deciding how you want ANY-maze to schedule your tests, you should review the First stage that ANY-maze has automatically included in this protocol and also enter the duration of the tests in the stage. Then you should consider whether you want to add any further stages.

See also:

- The Test schedule report

Stages

Stages are used to define the sequence of tests you will perform in an experiment. For example, in a water-maze, you might want to perform four training tests followed by a probe test. In ANY-maze, such a sequence would consist of two *stages;* 'Training' and 'Probe', with the Training stage consisting of four trials and the Probe stage consisting of one.

Clearly, all protocols will have to include at least one stage, otherwise your experiment won't contain any tests - and in fact ANY-maze ensures this is the case by automatically including one stage in all new protocols.

Beyond just defining the number of tests, stages also define the order in which tests will be performed, the duration of tests, rules which can end stages (such as achieving some goal), and the positions of movable zones.

For more information about stages, refer to the topics below.

- An introduction to stages
- Setting up a stage
- Creating stage end rules
- Including an accustomisation period in a stage
- Setting the trials of a stage to start automatically at specific times
- Specifying the positions of movable zones during a stage
- Editing a stage
- Deleting a stage

An introduction to stages

Introduction

Stages are used in ANY-maze to define the tests which will be performed in an experiment, and as such they are a fundamental protocol element - in other words, all protocols *must* include at least one stage. In fact, ANY-maze automatically includes a *First stage* in all new protocols for you.

Each stage in an experiment consists of one or more *trials;* for example, in a water-maze you might have a 'Training' stage consisting of 4 trials, and a 'Probe' stage consisting of 1 trial. Of course, in a simple case such as a plusmaze, you would probably have just a single stage consisting of a single trial.

- Dividing an experiment into stages
- Test duration and ending tests
- The number of trials in a stage
- Specifying the testing order
- Specifying the position of movable zones
- Working with multiple apparatus

Dividing an experiment into stages

Although the division of an experiment into stages will usually be very obvious, the formal definition of a stage from ANY-maze's point of view is that the trials in a single stage all have the same characteristics. This means, amongst other things, that they all have the same duration and they share the same procedures. Although this defines situations in which you are *required* to divide an experiment into different stages, there's nothing to stop you creating two or more stages with the *same characteristics* if this is more logical in your experiment. For example, if in a water-maze the four 'Training' and one 'Probe' test are all 5 minutes long and if they will always end if the animal finds the water-maze island, then, *technically,* you could define a single stage consisting of five trials - nevertheless, it would probably be more logical *for you* to think of the experiment as having two stages.

Test duration and ending tests

Part of the definition of a stage is the duration of its tests. ANY-maze uses this information to time each test while it's running and to automatically end the test when the time is up. However, in some experiments, you may wish to end tests before this if the animal achieves some goal. For example, in a water-maze you would probably end the test if the animal finds the island.

As you may know, ANY-maze allows you to define procedures, which can be used to perform actions, some of which can cause the test to end. For example, in the water-maze you would probably want to include a procedure which would wait until the animal enters the island zone and then perform an action to end the test.

The number of trials in a stage

First a note about terminology: In ANY-maze the term *test* is a generic name which equates to what you would intuitively expect a test to be - i.e. an animal being placed in the apparatus, being tracked, and being removed from the apparatus. A *trial* on the other hand, is a single *test* within a stage. Thus, for example, ANY-maze will refer to 'Training stage, trial 1' and all the animals in the experiment will have this trial.

The *maximum* number of trials in a stage is part of the stage's definition, and can be any number from 1 to 50. However, you can also define stage end rules which can end the stage for an individual animal before it's had this number of trials.

For example, in a water-maze, you might want to train the animals to find the island - perhaps using the condition that they have to find the island within 30 seconds in two consecutive trials as a definition of 'trained'. This, then, would be your stage end rule. Therefore, although your stage might consist of a maximum of 6 trials, the actual number of trials an animal would receive would depend on how quickly it achieve the 'trained' condition - for example, for some animals this might be after 4 trials, while for others it might be after 5 or even 6 trials.

ANY-maze can even automatically retire animals which don't achieve the 'trained' condition within the maximum number of trials - a retired animal won't proceed on to the next stage in the experiment.

Specifying the testing order

Unless you choose to manually schedule your tests, then the stages will define the order in which the animals in an experiment will be tested.

The first fundamental of test order is that the animals in an experiment are numbered and animal 1 will be tested before animal 2. Nevertheless, there are still a number of options available to you.

In most experiments, you will probably want to perform the first stage for all the animals before proceeding on to the second stage. In the water-maze example, you would probably complete the Training stage *for all the animals* before starting the Probe stage. Nevertheless, there may be cases when you want to run through all the stages for one animal before you start the first stage for the second animal and ANY-maze includes an option which allows you to do this.

Assuming you will complete one stage for all the animals before starting the next one, then there's still the issue of the *test order* within those stages that consist of more than one trial - for this, ANY-maze includes three possibilities:

1. You test all the animals through the first trial before starting the second trial.
2. You test the first animal through all its trials before starting the second animal.

3. The last option is a little more complicated, and allows you to group the animals and then test them within their groups. For example, you could put the animals into groups of say three. Thus animals 1, 2 and 3 would be in the first group; animals 4, 5 and 6 in the second; and so on. *Within* the first group, all the animals would then be tested in their first trial, then in their second trial, then their third etc. When the first group has finished, then the second group will start. Thus this grouped system is effectively a mix of the previous two.

One last point about test order - when you actually run the tests, you can make alterations to the test order if you need to.

Specifying the position of movable zones

As you may recall from the information about zones, it's possible to define zones which aren't always in the same place, for example the reward arm of a T-maze might be on the left in some tests and on the right in others.

When you create such a *movable* zone, you can specify that it will always be in the same place for an individual animal but might change location from one animal to the next, or that it can change position between the trials of an individual animal.

In this latter case, you can optionally specify the positions that the zone should adopt during each of the stage's trials as part of the definition of the stage. For example, you might say that the reward arm of a T-maze is on the left in the 1st trial, on the right in the 2nd, etc.

Of course, the zone's positions might not follow this type of pattern, in which case you can choose to specify the position before the start of each test.

Working with multiple apparatus

If your protocol includes multiple apparatus, two water-mazes for example, then you can specify for each stage the way that ANY-maze assigns the animals to the apparatus.

In the simplest case, you can ask ANY-maze to *pre-assign* the animals to the apparatus, such that animal 1 would be assigned to the 1st piece of apparatus, animal 2 to the 2nd, animal 3 to the 1st, etc. (assuming just two pieces of apparatus) - see figure 1.

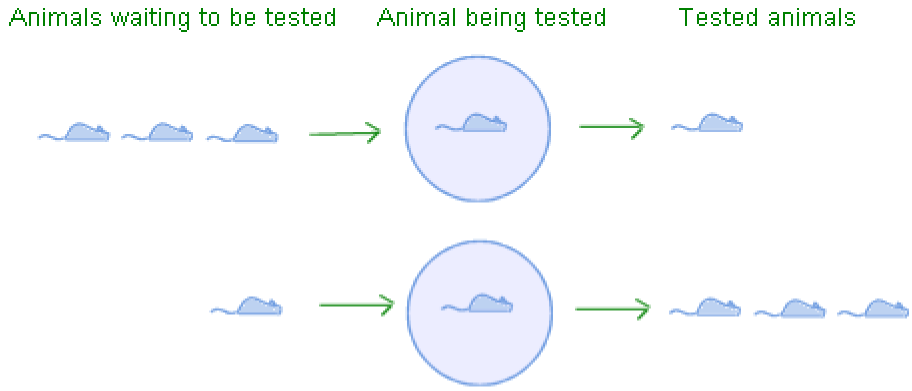


Figure 1. Animals 'pre-assigned' to two water-mazes.

While this simple assignment system seems very logical, it might be rather inefficient if tests can last for unequal durations and/or you are using some stage end rules. In these cases, it's possible for all the tests to end in one piece of apparatus while there's still a queue of animals waiting to be tested in another. To avoid this, you can choose to assign animals to the *first available piece of apparatus*. In this case, you won't know beforehand which piece of apparatus an animal will be tested on; it will just go to whichever one is free when its turn to be tested comes round - see figure 2.

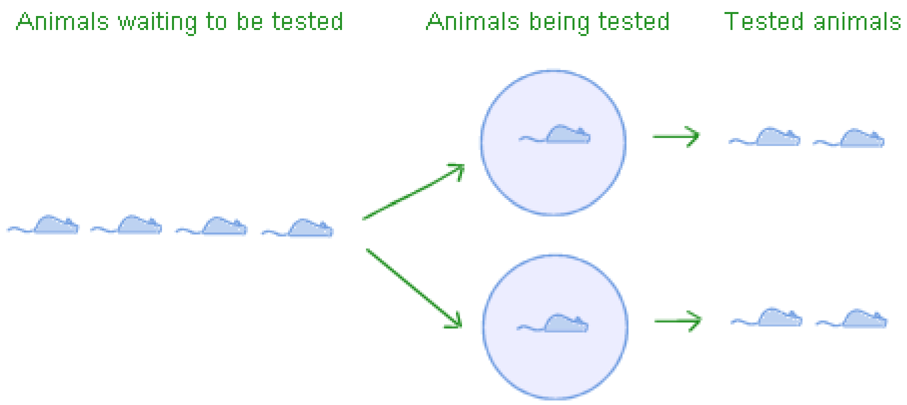


Figure 2. Animals being assigned to 'the first available maze'. This makes more efficient use of the mazes, although it can lead to an 'unbalanced' experiment with more animals being tested in one piece of apparatus than the other.

In most cases where a protocol consists of more than one stage, you will probably want to ensure that an animal which was tested on one particular piece of apparatus in the first stage is tested on the same apparatus in all its subsequent stages - ANY-maze includes this option too.

ANY-maze help topic H0367

Setting up a stage

Introduction

For a general introduction to stages, see An introduction to stages.

All protocols are automatically created with one stage called *First stage*. You should edit this stage to match the requirements of the first stage in your experiment, and then add any extra stages which you might need.

To add a stage, click the ✚ *Add item* button shown in the ribbon bar and select *New Stage* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the stage name.
2. Specify the maximum duration of tests in the stage.
3. Specify the maximum number of trials in the stage.
4. Specify the test order within the stage.
5. If your protocol includes multiple apparatus, specify how the animals should be assigned to the apparatus in this stage

After setting up the stage element itself, you can optionally use its sub-elements to:

1. Create stage end rules
2. Define an accustomisation period at the start of trials in the stage
3. Set trials in the stage to start automatically at specific times
4. Specify the position of movable zones in the stage

What next?

After completing these steps, you should consider whether you want ANY-maze to automatically start tests in your apparatus.

See also:

- Adding elements to a protocol
- Editing a stage
- Deleting a stage

Stage name

Don't use the word 'stage' at the end of the name you enter, because ANY-maze includes this automatically when required - so if you include it too, the name may sometimes appear as something like 'Training stage stage'.

In brief

Enter a name for the stage in the *Stage name* field on the stage's settings page. You must make an entry, but it can be anything you like.

Details

Stage names should uniquely and clearly identify a stage, because they'll be used to label groups of tests' results.

It's a good idea, although not required, to try to keep stage names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Test duration

In brief

In the *Test duration* field on the stage's settings page, enter the maximum duration of the tests in the stage. If you want to have different test durations in the same stage, then you should define two stages instead.

Details

The value you enter can use any of the units: d (days), h (hours), min or m (minutes), or s (seconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min', '8min 30s' or '510s'. If you don't specify any units, then seconds will be assumed.

Tests will always end after the duration you enter here, but you can also use procedures to end them before this. For example, in a water-maze, you might create a procedure which will wait until the animal enters the island and then perform an action which will end the test.

Limits

The maximum test duration is 7 days.

Things to consider when running long duration tests

If you plan to use long test durations (6 hours or more), then you may create quite large experiment files. For example, recording the animals' entire area at a frequency of 8 positions per second could use 3MB or more of disk space per hour. If tests lasted, say, 6 hours and you were testing 60 animals, this could create an experiment file of more than 1GB. ANY-maze can process large files like this, but results analysis will take longer.

In general, if you plan to run long tests, you should consider changing the recording frequency. For example, reducing it to 2 positions per second would reduce the size of the experiment file in the above example by 75%.

Another point to consider is whether you really need to record the entire area of the animal or whether recording just its centre point would suffice. This can reduce the size of experiment files by as much as 90% - together with a change to the recording frequency, this could reduce the experiment file in the example to less than 30MB.

Specifying the maximum number of trials in the stage

In brief

You should use the *Maximum number of trials in this stage* field on the stage's settings page to enter the number of trials that the stage should consist of. This is the *maximum* number, because you can also create stage end rules which can cause an animal to stop being tested before this if it achieves some type of goal - for example, finding the island in a water-maze within 30 seconds in 2 consecutive trials. Of course, if you don't include any stage end rules in this stage, then the number you enter here will be the number of trials which each animal will receive.

Limits

You must make an entry between 1 and 50.

Specifying the order in which animals will be tested

In brief

When a stage contains more than one trial, you can choose the order in which the animals will be tested *within* the stage. There are three options, which appear on the stage's settings page, and they're described in detail below. For a stage which contains just one trial, it doesn't make any difference which option you choose.

Normally, ANY-maze will complete one stage for all the animals before it moves on to the next one. However, you may want to create a protocol in which each animal will proceed through all the stages before the next animal is tested. If this is the case, then you should check the box on the Settings page which is displayed when the Stages... *element is selected in the protocol list.*

Details

In simple terms, animals will be tested within a stage in numerical order. That's to say that animal 1 will be tested first, then animal 2, etc. However, when a stage contains more than one trial, there's the question of how this basic order should be applied to the repeated trials. Essentially, there are two possibilities:

1. The animals are all tested in their first trial before any of them start their second trial. For example, with 6 animals and 2 trials, this could be represented as a testing schedule of:

Test 1	Animal 1, trial 1
Test 2	Animal 2, trial 1
Test 3	Animal 3, trial 1
Test 4	Animal 4, trial 1
Test 5	Animal 5, trial 1
Test 6	Animal 6, trial 1
Test 7	Animal 1, trial 2
Test 8	Animal 2, trial 2
Test 9	Animal 3, trial 2
Test 10	Animal 4, trial 2
Test 11	Animal 5, trial 2
Test 12	Animal 6, trial 2

2. Each animal is tested in all of its trials before the next animal is tested. For example, with 6 animals and 2 trials, this could be represented as a testing schedule of:

Test 1	Animal 1, trial 1
Test 2	Animal 1, trial 2
Test 3	Animal 2, trial 1
Test 4	Animal 2, trial 2
Test 5	Animal 3, trial 1
Test 6	Animal 3, trial 2
Test 7	Animal 4, trial 1
Test 8	Animal 4, trial 2
Test 9	Animal 5, trial 1
Test 10	Animal 5, trial 2
Test 11	Animal 6, trial 1
Test 12	Animal 6, trial 2

However, ANY-maze includes a third possibility, which effectively mixes these two by grouping the animals. Within each group, the animals are all tested in the first trial before any of them start the second trial; however, the second group will only start to be tested after the first group has completed all their trials. For example, with 6 animals in groups of 3 and 2 trials, this could be represented as a testing schedule of:

Test 1	Animal 1, trial 1
Test 2	Animal 2, trial 1
Test 3	Animal 3, trial 1
Test 4	Animal 1, trial 2
Test 5	Animal 2, trial 2
Test 6	Animal 3, trial 2
Test 7	Animal 4, trial 1
Test 8	Animal 5, trial 1
Test 9	Animal 6, trial 1
Test 10	Animal 4, trial 2
Test 11	Animal 5, trial 2
Test 12	Animal 6, trial 2

This might seem a rather esoteric way of organising your testing, but it can be very useful, particularly if you have large numbers of animals in an experiment.

Animal treatments and test order

A point which sometimes causes confusion in ANY-maze is the relationship between testing order and treatment. This usually arises when you're used to thinking about your test schedule in terms of treatments, for example:

Test 1	Drug A animal
Test 2	Drug B animal
Test 3	Drug C animal
Test 4	Drug A animal
Test 5	Drug B animal
Test 6	Drug C animal

This view of a test schedule is clearly important, because in most experiments you will want to ensure that the *treatments* are spread evenly across time. In ANY-maze, then, because animals are always tested in numerical order (subject to multi-trial issues described in the previous section), the issue is really one of which treatment each animal will receive. For example, if animal 1 receives treatment A, animal 2 treatment B, animal 3 treatment C, animal 4 treatment A etc. then the schedule will be the same as the one shown above.

This, of course, begs the question 'How do I control the treatments which the animals receive?' and this is something that's defined at the start of the protocol using the Treatment groups element.

Limits

If you choose to group animals, then the group size must be between 2 and 99 animals - a group of 1 not being a group at all.

Specifying how the animals should be assigned to the apparatus in this stage

In brief

If your protocol includes multiple apparatus, for example two water-mazes, then you can specify for each stage the way that ANY-maze assigns the animals to the apparatus for the tests in the stage. There are three options:

Pre-assign	The first animal is assigned to the first piece of apparatus, the second animal to the second, etc., until all the apparatus has one animal assigned to it. The next animal is then assigned to the first piece of apparatus again and so on.
First available	The animals are only assigned to the apparatus immediately before their tests. The next animal to test is simply assigned to the first piece of apparatus which is free (i.e. has no test running in it.)
Same as previous stage	This option is only available in the 2nd stage and above. The animals are tested in the same piece of apparatus in which they were tested in the previous stage.

Details

In the simple case of *pre-assigning* the animals to apparatus, animal 1 will be assigned to the 1st piece of apparatus, animal 2 to the 2nd, animal 3 to the 1st, etc. (assuming just 2 pieces of apparatus) - see figure 1.

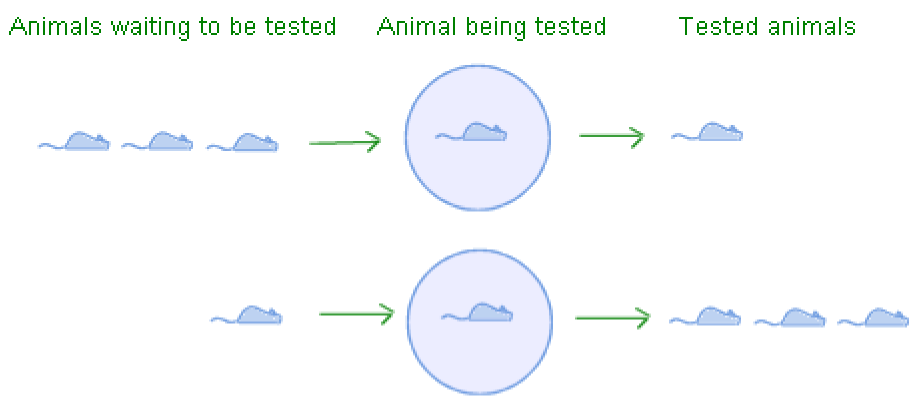


Figure 1. Animals 'pre-assigned' to two water-mazes.

While this simple assignment system seems very logical, it might be rather inefficient if tests can last for unequal durations and/or you are using some stage end rules. In these cases, it's possible for all the tests to end in one piece of apparatus while there's still a queue of animals waiting to be tested in another. To avoid this, you can choose to assign animals to the *first available piece of apparatus*. In this case, you won't know beforehand which piece of apparatus an animal will be tested on; it will just go to whichever one is free when its turn to be tested comes round - see figure 2.

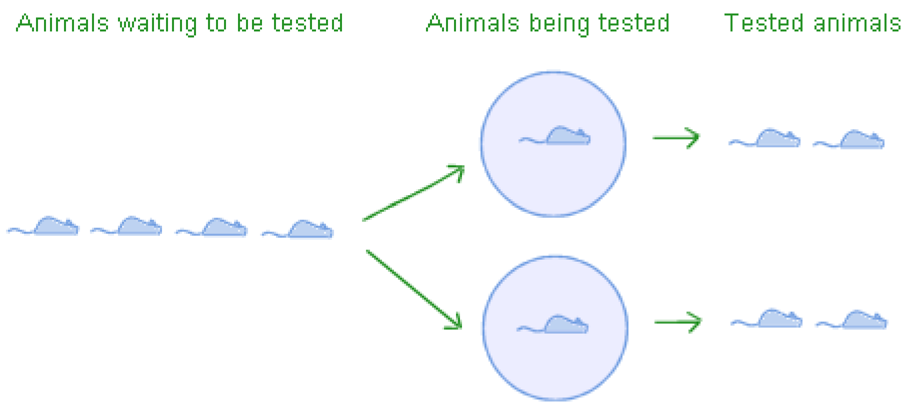


Figure 2. Animals being assigned to 'the first available maze'. This is makes more efficient use of the mazes, although it can lead to an 'unbalanced' experiment with more animals being tested one piece of apparatus than the other.

An important point to understand is that if a stage includes multiple trials, the animals will be assigned to a piece of apparatus in their *first trial* in the stage using the rules described above; they will then be tested on the *same* apparatus in all the other trials in the stage.

In most cases where a protocol consists of more than one stage, you will probably want to ensure that an animal which was tested on one particular piece of apparatus in the first stage is tested on the same apparatus in all its subsequent stages - in this case, you should choose the assignment option *Same as previous stage* for the second and subsequent stages.

Creating stage end rules

Introduction

In some experiments, particularly those related to memory, it's common to train animals to achieve a certain *goal*. For example, in a water-maze, the goal might be to find the island within a certain time in a number of consecutive tests.

Although it's possible to manually check when an animal achieves a goal such as this, and then instruct ANY-maze to stop its trials, it's much easier to get ANY-maze to check for you - which you can do using *Stage end rules*.

As the name implies, a stage end rule ends a stage for a particular animal; in other words, the animal stops having trials in the stage. The rules you can define in ANY-maze are very flexible, and should cover most situations; you can even define multiple rules for the same stage, so the trials end if the animal achieves either Goal A or Goal B.

The stage end rules sub-element

All stage elements in a protocol automatically include some sub-elements, one of which is used to set up the stage's *Stage end rules* - see figure 1.

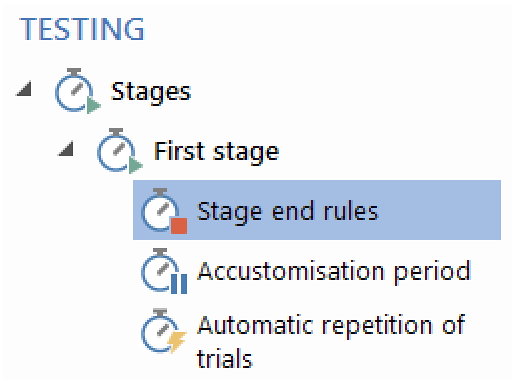


Figure 1. A stage's Stage end rules *sub-element is used to manage rules which can end the stage before an animal has had the maximum number of trials.*

Selecting this sub-element displays the stage end rules settings which you can use to create new rules, edit or delete existing rules, and set certain options.

Creating a stage end rule

In the settings pane of a stage end rule sub-element is a link titled *Help me create a new stage end rule* which, when clicked, opens the *Stage end rules wizard*. This wizard takes you a step at a time through the process of defining a new stage end rule. When you've finished defining a rule, it's added to the list of rules shown in the settings pane.

Editing or deleting a stage end rule

At any time, you can either edit or delete a stage end rule. To do this, you simply need to select the rule in the list of rules shown in the setting pane and click the *Help me edit the rule...* or *Remove the rule...* link respectively.

When you edit or remove a stage end rule that applies to an experiment's current stage (i.e. the stage which animals are currently being tested in), then ANY-maze will recheck all the animals to see whether any of them should, as a result of the change, either stop being tested or restart their tests - with any such effects simply being reflected in the test schedule.

However, unless you have selected the option to allow adjustments to completed stages, this will **only** affect the current, or future, stages - any stages which are complete won't be rechecked, even if such a check would cause an animal to require more tests.

In fact, this type of checking will also occur if other elements of a protocol - which could impact a stage end rule - are edited. For example, imagine a simple stage end rule that specifies that the stage should end if the animal enters the X zone within 30 seconds. Now you change the definition of the X zone, perhaps making it bigger. Some animals may now satisfy the rule, because in one of their existing tests they entered the new, enlarged area of the zone.

Manually ending a stage for an animal

Ordinarily, ANY-maze will automatically apply stage end rules without any intervention from you. However, there are two situations when you might want to end a stage manually. One is when an animal has *almost* satisfied a rule, and in your judgement it's enough to end the stage (perhaps the animal entered a zone in 31 seconds rather than the strictly-required 30 seconds); the second is when you want to use a rule which is just too complicated to be defined using the 'Stage end rule' wizard.

If you think that you may want to end stages manually in this way, you will need to check the box title *Allow the user to specify when an animal should stop having trials in this stage, or whether an animal should skip the stage entirely* in the stage end rule's settings pane. You will then be able to end trials manually, as described in Ending an animal's tests in a stage manually.

Skipping a stage for an animal

Ordinarily, ANY-maze will require that all animals pass through all stages. So, for example, even though a stage end rule might cause an animal to have only one trial in a stage, it would still have *a* trial. However, there might be situations in which you don't want an animal to have any trials in a stage at all - in other words you would like the animal to skip the stage.

If you think that you may want to skip a stage in this way, you will need to check the box title *Allow the user to specify when an animal should stop having trials in this stage, or whether an animal should skip the stage entirely* in the stage end rule's settings pane. You will then be able to skip the stage, as described in Skipping an entire stage for an animal.

Automatically retiring animals

In some experiments, you might want to remove animals which don't achieve some type of goal in any of the trials in the stage. This is commonly the case in a *training* stage, because failure to achieve the goal implies that the animal isn't trained.

Although it's easy to manually remove such animals from your experiment, you can also ask ANY-maze to do this for you by checking the box titled *Automatically retire any animal that reaches the end of this stage without satisfying any stage end rule*.

By the way, *retiring* an animal means that it will no longer be tested (i.e. it won't proceed to the following stage), but its results for the trials it has already had are retained.

Adjusting the trials in a completed stage

Usually, ANY-maze will not adjust the trials for an animal in any stages that it has completed, if the subsequent stage has been started. For example, if you will train animals in 6 trials of a *training* stage and then test them in two trials of a *probe* stage, then, once you've performed one of the probe trials, no adjustments to the training stage will occur. This rule will apply in all cases, including if you change the number of trials in the stage, alter any stage end rules, or even cancel a test in the training stage -

in all cases, ANY-maze *won't* update the animal's trials in the training stage. This behaviour is by design, as (in this example) it makes little sense to perform further training trials once you've performed some probe tests.

In general, you will probably want to maintain this 'no adjustments to completed stages' rule, as it helps to ensure that you don't inadvertently restart a completed stage. However, there may be situations when this is exactly what you do want to do, and this rule will prevent you. In this case, you can select the option to *Allow adjustments to completed stage* in a stage's *Stage end rules*. This will cause ANY-maze to adjust the trials to account for changes to the number of trials, the stage end rules, the cancellation of tests, etc., even if the stage has been completed.

You should be very careful about how you use the *Allow adjustments to completed stage* option, as it might cause new tests to be scheduled for animals in stages that they have already completed. For example, imagine you performed 6 training trials for animal 1 in the water-maze and the animal has also already had one probe trial. Now you change the number of training trials from 6 to 8. Normally ANY-maze will increase the number of training trials for all the animals which have not yet had a probe trial, but if the *Allow adjustments to completed stage* option is switched on, it will also add 2 training trials for animal 1, in other words it will expect you to go back to training the animal, even though it has had a probe trial – which doesn't really make much sense.

Also, if you use manual scheduling, this option can cause tests to move from one stage back into an earlier stage. For example, imagine you have 3 trials in the first stage and 3 in the second stage. You intend to schedule 3 tests for animal 1 and 3 for animal 2 – so these will be their first stage tests – but what you actually do is schedule 6 tests for animal 1. Nevertheless, you run the tests as you intended, so 3 for animal 1 and 3 for animal 2. Now you have a problem, you actually tested animal 1 three times and animal 2 three times, but ANY-maze thinks you tested animal 1 six times, with 3 tests in the first stage and 3 in the second. Realising your mistake, you change the animal number for tests 4, 5 and 6 to be animal 2. This *won't* fix the problem, because ANY-maze will still see these as second stage trials (because the animal has completed at least one second stage trial and therefore any prior stage will be considered to be completed); so it will now think that animal 2 had no first stage trials and 3 second stage trials – not exactly what you wanted. You *could* fix this by switching on the *Allow adjustments to completed stage* option; if you did this ANY-maze would then move the trials back into the first stage and you'd end up with 3 first stage trials on animal 1, and 3 first stage trials on animal 2 – perfect! Or perhaps not… In this case it would be very important to remember that while ANY-maze was performing the trials on animal 2 it thought they were trials in the second stage (on animal 1). So it will have used the test duration of the second stage trials, it will have applied any procedures that are specified for the second stage (and excluded any that are specified as being used just in the first stage) etc. For example, in the first stage you might play a sound to the animals and apply a shock 2 seconds later, while in the second stage you might just play the sound. So simply saying 'these were the really the first trials of animal 2, so change them to be part of the first stage' may actually be wrong – in the example, animal 2 would have heard the sound, but received no shock, so you can't really consider the test to be 'first stage' trials. In this case what you should probably do is simply delete animal 2. Of course, if the tests in the first stage are identical to the tests in the second stage, then in this example, moving the tests of animal 2 back into the first stage is exactly what you want and would fix the problem.

In summary, the *Allow adjustments to completed stage* option can be very useful, but it can also create subtle problems, so use it carefully. When you do use it we recommend that you switch it on, perform whatever adjustments you require, and then switch it off again. In fact, ANY-maze will always switch this option off in all stages, whenever you close a file.

ANY-maze help topic H0374

The Stage end rules wizard

The Stage end rules wizard helps you to set up rules which can cause an animal to stop having trials in a particular stage.

For example, in a training stage, you may be training the animals to perform some task within a certain time limit. You decide that if an animal performs the task within the limit in two consecutive tests, then it's considered to be trained. Thus in this stage, some animals (the quick learners) will perhaps only have 3 or 4 trials, while others may have 5, 6, 7 or more trials. Of course, you would still have a value for the maximum number of trials which an animal could receive; perhaps 8 in this example.

Of course, it would be perfectly possible for you to manually check your stage end rule (or rules) at the end of each trial to see whether the particular animal should continue to be tested or not. However, it's much easier to tell ANY-maze what your rules are, and let it do the checking for you - and you use this wizard to tell it about your rules.

- To move through the wizard, simply click the *Next* > button.
- If you want to go back to a previous step, perhaps to alter an answer you gave, then you can click the < *Back* button.
- You can cancel the wizard at any time, without creating a rule, by clicking the *Cancel* button.

Stage end rules wizard : Rule type

Each stage end rule that you create will be based around a particular *event* which has to occur. (Here, I'm using the term *event* rather loosely, because a rule might be based around something which *doesn't* occur, for example the animal *doesn't* enter a particular zone). Anyhow, the point is that you first need to choose what this *event* is going to be.

For example, if your rule is that 'the animal must enter the *Island zone* within 30 seconds of the start of the test in two consecutive trials', then the event would be that the animal enters the Island zone - you'll use the subsequent steps in the wizard to specify the additional details.

As there can be lots of possible rules, they're shown here in groups - for example, *Rules which relate to activity in the Island zone*.

- To see the rules inside a group, simply click the group.
- To choose the individual rule you want to use, simply click it - it'll then be shown highlighted.
- You won't be able to click the *Next >* button until you've selected a rule here.

If you want to use more than one rule, then you should use the wizard to define the first rule, and then run the wizard again to add the second rule, and so on. There's no limit to the number of rules you can define. When you have multiple rules, an animal's trials will end when *any* of the rules is satisfied.

Stage end rules wizard : Number of occurrences

In this step, you need to specify how many times the event you chose in the previous step has to occur.

If you select the option for *More than once*, then a field will appear where you can enter the actual number, which can be any value from 1 to 999.

Stage end rules wizard : Duration

In this step, you need to specify how long the situation you chose in the previous step has to last. For example, if you specified that the animal has to remain in a zone for certain amount of time, then you should enter the amount of time here.

The value you enter can use any of the units: d (days), h (hours), min or m (minutes), or s (seconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 1 second to 24 hours.

Stage end rules wizard : Distance

In this step, you need to specify how far the animal must travel (or not travel) in order to satisfy the rule. For example, if you specified that the animal must travel no more than a certain distance, then you should enter the distance here.

The value you enter should be in millimetres. You can enter any value from 1 to 999,999 millimetres.

Stage end rules wizard : Time limit

In this step, you can specify whether the event you've chosen has to occur within a certain amount of time of the start of the test. This is commonly the case in *training* stages, where you're training the animal to perform a certain task within a time limit - perhaps to find the island zone in a water-maze within 30 seconds.

If you specify that there is a time limit, then a field will appear which you can use to specify the actual value. Here you can use any of the units: d (days), h (hours), min or m (minutes), or s (seconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 1 second to 24 hours.

Stage end rules wizard : Number of trials

In this step, you can specify whether an animal's trials should end following the first trial in which it satisfies the rule, or whether it has to satisfy the rule in a number of repeated trials.

For example, in a water-maze training stage, you might have defined a rule that specifies that the animal has to find the island within 30 seconds of the test start. However, you probably wouldn't consider the animal to be 'trained' after the *first* trial in which it does this - after all, it might just be coincidence. So you would perhaps want to specify that the animal has to find the island within 30 seconds in 3 consecutive trials - in which case, you could be confident that it really does know where the island is.

- If you choose the option for a series of consecutive trials, then a field will appear where you can specify the number of consecutive trials - you can enter any value from 1 to the maximum number of trials in the stage.
- If you choose the option for X out of Y trials, then fields will appear where you can specify values for X and Y. For example, you might specify that the animal has to satisfy the rule in '3 out of 4 trials'. You can enter any values for X and Y, provided X ≤ Y and Y ≤ the maximum number of trials in the stage.

Stage end rules wizard : Finished

This is the final step of the Stage end rules wizard. Here, you should review the rule to check it's correct, and then give it a name.

- If you want to edit the rule, then simply use the *< Back* button to work backwards through the steps of the wizard to the appropriate page.
- The name you give for the rule will be used by ANY-maze when it reports the reason why a stage ended for a particular animal. It's a good idea to try to keep this name reasonably short; indeed, if you only have one rule then you might want to just use a name such as 'Trained' rather than repeating the rule's definition. This would mean, for example, that ANY-maze would report an animal as ending a training stage either because it 'Completed all trials' or because it was 'Trained'.

  You can enter any name you like up to a maximum of 80 characters.

Including an accustomisation period

Introduction

In some situations, you may wish to place an animal in your apparatus before you start a test so that it has a chance to get accustomed to it. For example, in a water-maze you might place the animal on the island before its first trial, so that it at least knows there's an island there.

Although you might think that you could simply place the animal in the apparatus, wait for your accustomisation period to end, and then start the test, this won't always work. A problem arises because ANY-maze assumes (in most cases) that the apparatus is empty, i.e. doesn't contain an animal, before a test starts. Therefore if the animal is already in the apparatus, the system will get confused. To resolve this, you need to tell ANY-maze if you plan to use an *accustomisation* period like this in your tests. This has the added benefit that the system will then time the accustomisation period for you too.

By the way, if you plan to include an accustomisation period, but you intend to *remove* the animals from the apparatus between the end of the period and the start of the test, then you don't *have* to tell ANY-maze about it at all.

By default, ANY-maze assumes that a stage won't include an accustomisation period; so if you don't want one, there's nothing for you to do.

Including an accustomisation period in a test is generally a good idea if the period is quite short, for example an hour or two at most, but if you intend to place the animal in the apparatus for a long period of time before you start the test (perhaps days), then using an accustomisation period is not very satisfactory because your computer will have to be on for the entire period and you won't be able to use it to perform any other experiments. In these circumstances, it's better to use the protocol's Tracking options *to specify that the animal will already be in the apparatus at the start of the test - see* this topic *for details of how this works.*

- The accustomisation sub-element
- Defining an accustomisation period
- Edit an accustomisation period

The accustomisation sub-element

Each *Stage* element in a protocol automatically includes some sub-elements, one of which can be used to set up the stage's accustomisation period - see figure 1.

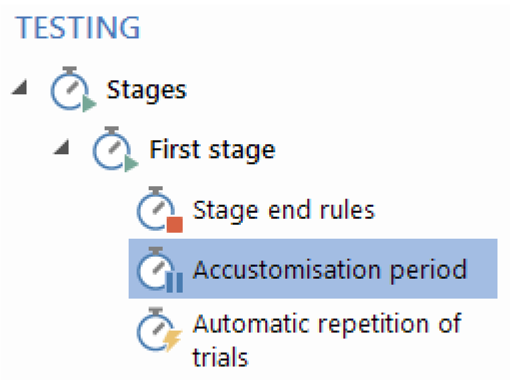


Figure 1. A stage's Accustomisation period *sub-element is used to define a period before the start of the test, during which the animal will be in the apparatus but won't be tracked.*

Selecting this sub-element displays the accustomisation settings, which you can use to specify whether an accustomisation period should be included, and how long it will be.

This option will only appear if the current protocol mode includes video tracking.

Defining an accustomisation period

If you want a stage to include an accustomisation period, then you should first check the box titled *Include an accustomisation period at the start of the trials in this stage,* and then enter the duration of the period.

The value you enter can use any of the units: d (days), h (hours), min or m (minutes), or s (seconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min', '8min 30s', or '510s'. If you don't specify any units, then seconds will be assumed. You can enter any value from 1 second to 24 hours.

Next, you should specify whether the period should be included at the start of all the trials in the stage. If you specify that the period won't be included at the start of all trials, then you should specify how many trials will use it. ANY-maze will then include the period before the *first N* trials, where 'N' is the number you enter. For example, entering 2 will include the period before the first and second trials only.

The final option when setting up an accustomisation period is to choose whether, when the accustomisation period ends, ANY-maze should automatically start or whether it should wait for you to start the test, either by 'auto-starting' it (i.e. by entering and leaving the camera's field of view) or by clicking the button.

For details about actually running tests which include an accustomisation period, see Running tests which include an accustomisation period

Editing an accustomisation

You can make any changes you like to an accustomisation period, including just turning it off, at any time.

Automatically starting the trials in a stage

Introduction

In some experiments, you may wish to test your animals in repeated trials at a certain set frequency. For example, you might want to track the animals for 10 minutes once every hour; alternatively, you might want to track the animals for 1 hour every day starting at 7am.

In either case, you can use the options to 'automatically start trials in a stage' to achieve this.

- Specifying when trials should start
- Side effects of automatically starting trials in a stage
- Controlling trials which are set to start automatically

Specifying when trials should start

Each *Stage* element in a protocol automatically includes some sub-elements, one of which is titled *Automatic repetition of trials*. Selecting this sub-element shows six options for how ANY-maze should automatically start the trials in a stage; they are:

Don't automatically repeat the trials in this stage

This is the default option for all stages; ANY-maze will simply end a trial and then wait for you to start the next one.

A set period after the previous trial starts

In this case the next trial will begin a certain period after the *start* time of the previous trial. This can mean that the trial may not start automatically if the period you specify is shorter than the duration of the previous trial. For example, if you specify that the trial should start 5 minutes after the *start* of the previous trial, and the previous trial lasts 10 minutes, then ANY-maze won't start the trial automatically, as it never starts trials whose start time is in the past.

The period you specify can use units of s (seconds), min or m (minutes), h (hours) and d (days). If you don't enter any units, then seconds will be assumed. You can enter any period from 0 to 1 day.

Note that this option will only automatically start trials 2, 3, etc. It won't start trial 1, as it has no *previous* trial.

A variable period after the previous trial starts

In this case, the next trial will begin a random time (between two limits) after the *start* time of the previous trial. This can mean that the trial may not start automatically if the period you specify is shorter than the duration of the previous trial. For example, if you specify that the trial should start between 5 minutes and 10 minutes after the *start* of the previous trial, and the previous trial lasts 20 minutes, then ANY-maze won't start the trial automatically, as it never starts trials whose start time is in the past.

Here, you should specify two times and ANY-maze will then choose a random time between them - for example, if you specify 5 minutes to 10 minutes, ANY-maze will choose a time between the two (inclusive), such as 6 minutes, 7 minutes 3 seconds, or 10 minutes.

The periods you specify can use units of s (seconds), min or m (minutes), h (hours) and d (days). If you don't enter any units, then seconds will be assumed. You can enter any period from 0 to 1 day.

Note that this option will only automatically start trials 2, 3, etc. It won't start trial 1, as it has no *previous* trial.

A set period after the previous trial ends

In this case, the next trial will begin a certain period after the *end* of the previous trial.

The period you specify can use units of s (seconds), min or m (minutes), h (hours) and d (days). If you don't enter any units, then seconds will be assumed. You can enter any period from 0 to 1 day.

Note that this option will only automatically start trials 2, 3, etc. It won't start trial 1, as it has no *previous* trial.

A variable period after the previous trial ends

In this case, the next trial will begin a random time (between two limits) after the *end* of the previous trial.

Here, you should specify two times and ANY-maze will then choose a random time between them - for example, if you specify 5 minutes to 10 minutes, ANY-maze will choose a time between the two (inclusive), such as 6 minutes, 7 minutes 3 seconds, or 10 minutes.

The periods you specify can use units of s (seconds), min or m (minutes), h (hours) and d (days). If you don't enter any units, then seconds will be assumed. You can enter any period from 0 to 1 day.

Note that this option will only automatically start trials 2, 3, etc. It won't start trial 1, as it has no *previous* trial.

Start each trial at the following time of day

In this case, you can specify a specific time of day (using the 24 hour clock) and ANY-maze will start the trials at this time. Clearly, this means that the system will run one trial per day.

Note that unlike the other options, this option *will* automatically start the first trial in the stage.

Side effects of automatically starting trials in a stage

Clearly, if the trials in a stage are to start automatically, then for each animal, you will need to run trial 1, then trial 2, then trial 3 and so on. (i.e. you can't run trial 1 for animal 1, then trial 1 for animal 2, and so on).

For this reason, if you use automatic repetition of trials, ANY-maze will automatically set the stage to use the option *Perform all the trials for each animal before starting the next animal*.

Specifying positions of movable zones

Introduction

As you probably know already, you can create zones in ANY-maze which can adopt different positions in different tests - so called movable zones. These zones come in two basic types, those which have a fixed position for a specific animal, but can alter position between animals, and those which can alter position between an individual animal's tests.

Specifying the positions of the first type of movable zones, those which only change position between animals, is simply done by specifying the position for each animal before you start the experiment - usually by using the Animals spreadsheet on the Experiment page. However, specifying the positions of the second type of movable zones is a little more complex and is described here.

- Ways in which a zone's position can alter within a stage
- Specifying a zone's position within a stage using the 'Zone position' sub-element

Ways in which a zone's position can alter within a stage

ANY-maze identifies a number of possible ways that a movable zone's position can be defined within a particular stage.

1. The zone simply isn't used in the stage - for example, in a water-maze, you might take the island out in some stages.
2. The position just moves about randomly between the trials and the animals.
3. The position just moves about randomly between the trials and the animals, but for a specific animal it doesn't repeat until it has adopted all the possible positions. For example, if there are 4 positions, numbered 1 through 4, then in the first test for an animal, the zone might be in position 3; this would mean that for the next test, it could only be in position 1, 2 or 4 because position 3 has already been used, so ANY-maze would randomly choose one of these positions.
4. The zone's position follows a fixed pattern between the various trials in the stage, irrespective of the animal being tested. For example, in the first trial the zone might be in position 1; in the second trial, in position 3; in the third trial in position 2; etc.
5. The position in a test is the same as the zone in which the animal ended its previous test. Note that for this option to work correctly, you must ensure that all the movable zones in your apparatus have positions with the same names. You should also be aware that ANY-maze matches the *names* of the positions - which doesn't necessarily mean they are physically located in the same place. For example, imagine you have a

circular piece of apparatus divided into 4 quadrants. You define two zones, A and B. Both zones have two possible positions, North and South. If you specified for both zones that North is the top-left quadrant and South is the bottom-left, then saying that the position of Zone A is the same as the position of Zone B in the previous test, would work as you would expect. But if you specified that the North position of Zone A is the top-right quadrant and the South position is the bottom-right quadrant, then if the animal ended one test in Zone B (which, let's say, was in the North, i.e. top-left, position) then the next test would have Zone A also in the 'North', although this would physically be in the top-right quadrant - probably not what you would want. Finally, note that when this option is selected, the user has to specify the position of the zone in the animal's first test.

6. The position follows some other, more complex sequence, perhaps following a fixed pattern that depends on *both* the trial number and the animal being tested.

Specifying a zone's position within a stage using the *Zone position* sub-element

ANY-maze automatically includes one *Zone position* sub-element in each stage of your protocol for each movable zone which can alter position between an individual animal's tests. For example, if a protocol includes two such zones, perhaps a 'Food' zone and a 'Water' zone, then each stage will include two sub-elements for defining the position of each zone - see figure 1.

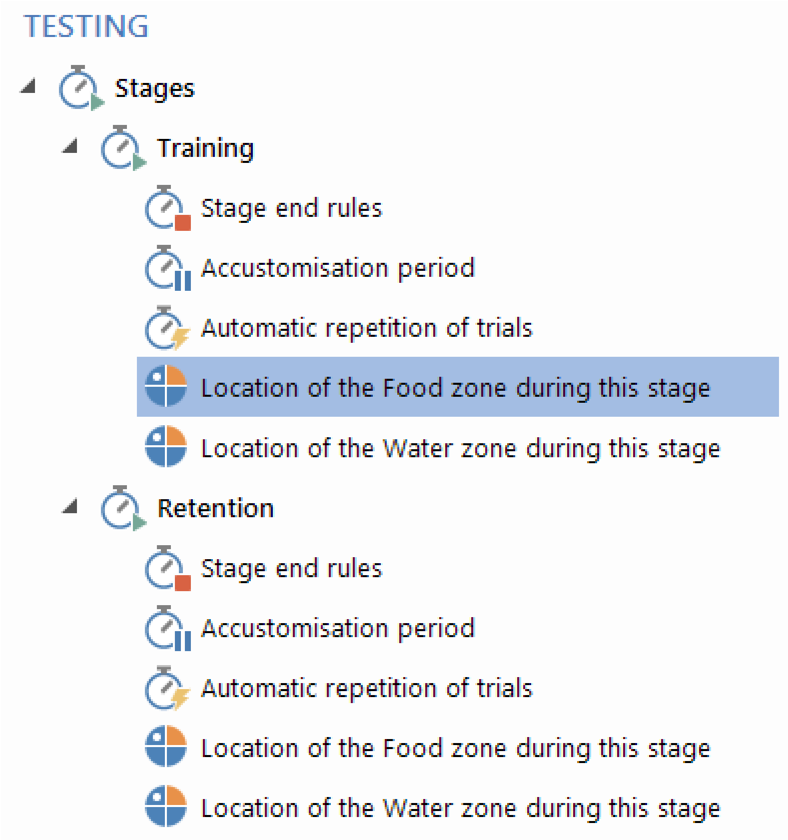


Figure 1. Each stage includes one zone position sub-element for each movable zone which can alter position between an individual animal's tests.

Selecting a zone position sub-element allows you to specify how the zone's position will be set during the stage, by selecting one of the possibilities described in the previous section.

1. If the zone is specified as not being used in the stage, then ANY-maze will simply ignore it completely during the stage's trials, and all measures for the zone in the stage will generate undefined results.
2. For a zone which follows a fixed sequence of positions, you can specify the sequence in the settings pane. This is very useful, because it means you won't need to set the position at the start of the tests because ANY-maze will already know it.
3. For a zone which adopts any random position, ANY-maze will simply choose a position for you at the start of each test. The position can be shown on the Test schedule report, so you know where to physically locate the zone.
4. For zones which follow a more complex sequence, *you* will need to tell ANY-maze the zone's position at the start of each test. You do this by clicking the position in the apparatus map before the test starts - see figure 2. This process is described more fully in Setting the positions of movable zones before a test.

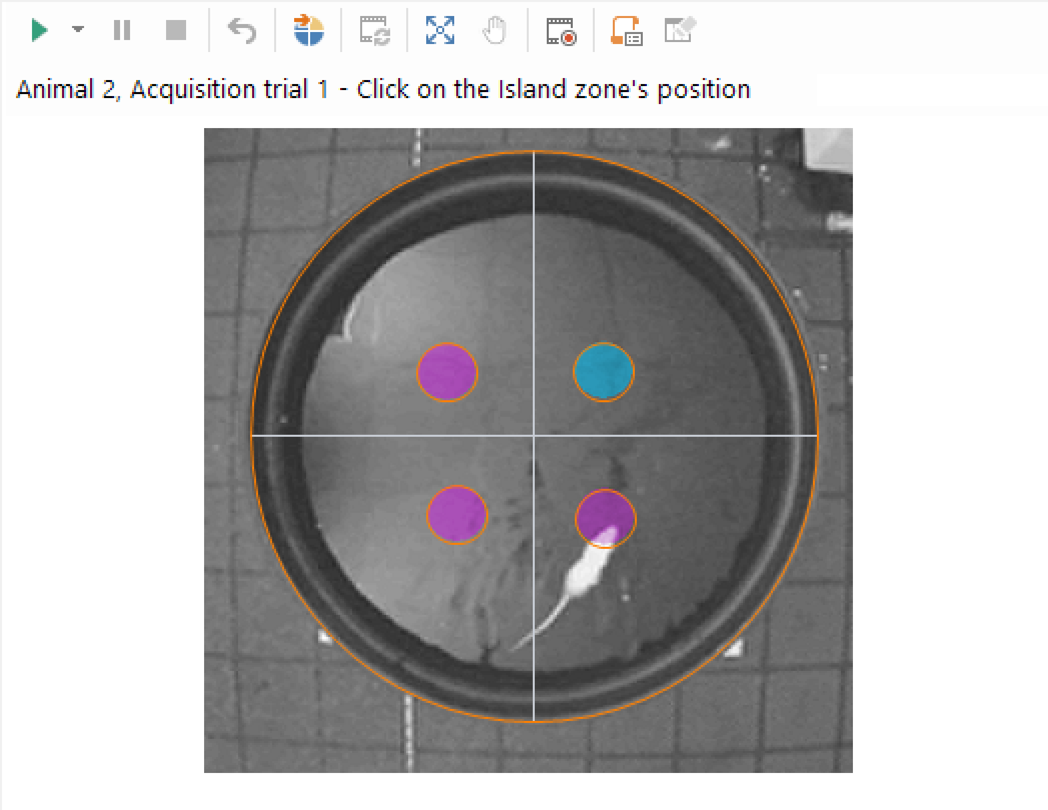


Figure 2. Specifying the position of an island in a water-maze. The purple areas indicate the positions the island can adopt; the blue area indicates the position the user has selected.

Editing a stage

Introduction

You can edit absolutely everything related to a stage at any time, whether before, during or after the experiment. However, there are a few things to be aware of.

Points to note

Changing the test duration will, of course, only affect future tests. Thus, changing this value can result in the tests in a stage having different durations; this can make the analysis and interpretation of results rather complicated.

You can always *increase* the number of trials in a stage; however, this won't necessarily affect all the animals:

- New trials WILL be added for any animals which *haven't started* the following stage (if there is one). In this respect, an animal is deemed to have *started a stage* if it has completed at least one trial in the stage.
- New trials WON'T be added for any animals which *have started* the following stage (unless you have selected the option to *Allow adjustments to completed stage* in the stage's Stage end rules sub-element).

Note that a repercussion of this rule is that in some circumstances, the experiment can go back a stage. For example, consider the situation where all the animals have completed stage 1 and animal 1 has completed its first trial in stage 2. In this case, the next test to perform will be a *stage 2* test, perhaps for animal 2. Now you add an extra trial to stage 1. Animal 1 WON'T receive this extra trial, because it's started stage 2, but the other animals WILL and the next test to perform will change to be a *stage 1* test - probably for animal 2. In other words, the experiment will have gone back to stage 1.

You can't decrease the number of trials in a stage to fewer than the maximum number of trials any animal has actually had in the stage. For example, if you specify that stage 1 has 6 trials and so far you've tested animal 1 in four of them, then you could reduce the number of trials to 5 or 4, but not fewer than this - because animal 1 has actually had 4 trials already. If you try to reduce the number of trials too much, ANY-maze will simply tell you that the change isn't possible.

Deleting a stage

Introduction

You can delete a stage, provided it's not the only stage in the protocol and it hasn't actually been started. In this respect, a stage is deemed to have been started if at least one animal has completed at least one trial in the stage.

Auto-starting tests

Introduction

There are, in fact, four different ways to start a test in ANY-maze (see Controlling testing), but all except auto-start require that the experimenter does something such as pressing a key on the computer's keyboard. The problem with this is that the apparatus may be in a different room to the computer, thus causing a delay between placing the animal in the apparatus and the start of the test, while the experimenter rushes to the computer to press the key. The ANY-maze auto-start feature addresses this problem.

- How auto start works
- Using fast auto-start
- Starting tests without using auto-start

How auto start works

Auto start works by tracking *you, the experimenter,* in the part of the video image which isn't the apparatus. When you enter this area to place the animal in the apparatus, ANY-maze registers your presence - and as you leave, it starts the test. Thus, the only delay is between the time you place the animal in the apparatus and the time you leave the camera's view. In some cases, this short delay can be a problem, in which case you can opt to use *fast auto-start,* which only waits for you to leave the immediate vicinity of the maze (there's more about this below).

You may wonder what happens if you just enter the camera's view for some other reason, to clean the apparatus for example - won't ANY-maze start a test when you leave? The answer is no, because to start a test you DO have to press a key; it's just that you press the key *before* you enter the camera's view. This *primes* the system, ready to start the test when you leave. Thus, a normal sequence of events is something like this:

1. You press a key on the computer's keyboard to tell ANY-maze to get ready to start a test.
2. You walk into the camera's view with the animal - ANY-maze starts to track *you*.
3. You place the animal in the apparatus.
4. You leave the camera's view as quickly as possible - when you leave, ANY-maze registers this and starts the test.
5. The test ends for some reason, and you enter the camera's view to retrieve the animal. As ANY-maze isn't waiting to start a test, it ignores you. You can enter and leave the camera's view any number of times, perhaps to clean the apparatus without

affecting the system.

6. When you're ready to start the next test, you go back to step 1, above.

Using fast auto-start

ANY-maze's fast auto-start feature is designed to work with water-mazes. It may provide satisfactory results in other types of apparatus too, but we make no claims for this.

Normally auto-start works by detecting you, the experimenter, in the area of the video picture which is not part of the apparatus and then waiting for you to leave - when you leave, the test starts. Although this works well in most apparatus, in the water-maze the animal will often start swimming immediately it's placed in the water, and unless you can exit the camera's view very quickly, the animal may have swum an appreciable distance before the test begins.

To overcome this, you can choose to use *Fast auto-start* which only requires that you exit the immediate area of the maze before the test starts. The way this works is that ANY-maze creates a thin 'halo' around the edges of the apparatus - see figure 1 - and looks for your hand entering and then exiting this area - the exit starts the test.

There is one caveat to fast auto-start;, it requires that there's good contrast between your hand/arm and the background of the area close to the apparatus - otherwise, ANY-maze simply won't see you.

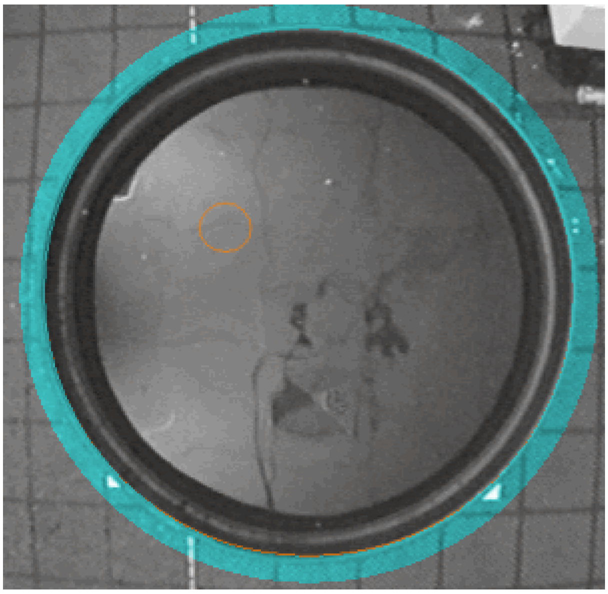


Figure 1. When using fast auto-start, ANY-maze waits for your hand/arm to enter and then exit the shaded area close to the apparatus - the exit triggers the test start. Note that in this example, you would need to be wearing white gloves or have a white sleeve in order for the system to 'see' you against the dark coloured background.

Starting tests without using auto-start

You don't have to use auto-start to start tests; instead, you can either:

- Press a key on the computer's keyboard.
- Click a button on the computer's screen.
- Press a switch connected to an input of an I/O device.

What next?

After completing these steps, you should consider what you want ANY-maze to record during your tests.

See also:

- Testing do's and don'ts

What to record while testing

Introduction

When tracking, ANY-maze attempts to identify the position of the animal in every image it receives from the camera, which may be as many as 30 images every second. However, it's generally unnecessary to record all these positions in order to analyse the animal's track and/or behaviour; indeed, doing so just serves to make the experiment file larger for no gain. For this reason, you can specify the maximum number of positions that the system should record each second.

As well as recording the animal's position during the test, ANY-maze can also record a digital video of the test. This can be very useful if you want to track the animal for long periods, and then visually review those parts of the test which contain interesting behaviours.

- Specifying the maximum number of positions the system should record each second
- Choosing whether to recording untracked frames
- Choosing not to record animal images
- Recording videos of tests

Specifying the maximum number of positions the system should record each second

As mentioned in the introduction, ANY-maze will attempt to identify the animal's position in every image it receives from the camera. However, it's not usually necessary to record all these positions in order to analyse the animal's track and/or behaviour. In fact, recording about 8 positions per second is usually fine for rats or mice that are walking at normal speeds. Of course, you might want to increase this, if you expect your animals will move very fast (panic response, for example), or perhaps decrease it if you intend to track for very long periods (24 hours, for example) as this will help to keep the size of the experiment file more manageable.

Choosing whether to recording untracked frames

This option is only available when you specify that the system should record every position of the animal.

Typically, ANY-maze receives 30 images (called 'frames') from the video source (usually a camera) every second. If it tracks the animal in all of these frames then, as you'd expect, you'll have 30 animal position results recorded every second, i.e. one every 33 milliseconds.

However, if in some of these frames, ANY-maze fails to determine where the animal is, then you'll have fewer than 30 frames recorded per second - i.e. there will be 'gaps' in the position results. For example, you might have positions for times 33ms, 66ms and 132ms - but the position for time 99ms is missing.

You can use the *Record frames in which the animal was not tracked* option to have ANY-maze record 'positions' for these gaps. Clearly, if ANY-maze doesn't know where the animal is (because it failed to track the animal) then it can't actually record the position, but it can record the time of the frame and a note that it couldn't determine the animal's location. For example, with this option switched on the Test data report might look like figure 1.

Time	Centre position X	Centre position Y
4.939s	337	290
4.973s	#N/A	#N/A
5.004s	337	287
5.043s	#N/A	#N/A
5.070s	373	240
5.109s	372	235

Figure 1. In this test ANY-maze failed to track the animal in the frames received from the camera at times 5.004s and 5.043s, this is why the position is shown as #N/A (not available). Nevertheless, every frame received from the camera is represented in the results.

Choosing not to record animal images

During a test, ANY-maze normally records not just the position of the animal's centre, but also the entire area of the video image which it considers to be the animal. This has a number of advantages, but it also makes experiment files quite large.

On modern computers, having large experiment files is really not a problem - and if you're working with short test durations (say less than 30 minutes), then there's no need to concern yourself about this. However, if you plan to track animals for long periods (for example, for days) then the size of the experiment file can become more of an issue. For example, if you intend to track sixteen animals for 72 hours each, then even if you chose to only record 4 positions each second, the experiment file could easily be larger than 1GB in size.

In cases where experiment files are likely to be very large, you may wish to turn off recording of the animal's area, which will reduce the file size dramatically - but you will lose some functionality.

Specifically, not recording the animal's area has the following effects:

- ANY-maze will no longer be able to analyse periods when the animal is immobile, whether you've turned on immobility detection or not. Note that the system may still indicate periods of immobility while actually performing a test, but the results will just show the animal as mobile the entire time.

- ANY-maze will no longer be able to use the percentage of the animal that's in a zone to analyse zone entries - it will just use the animal's centre point, irrespective of the setting you've made for the individual zones. In fact, *during the test* the system will continue to use the animal's area to detect zone entries, but it will use the centre point in the results. For this reason, if you intend to use this option, it's a good idea to alter all zones to use the animal's centre point to detect zone entries - this will make the behaviour during tests consistent with the analysis of test results.
- The distance from the animal to a zone's border will always be calculated based on the distance from the centre of the animal to the zone.
- The distance from the animal to a point will always be based on the distance from the animal's centre point.

In general, we don't recommend that you use this option. However, if you're running extremely long tests and immobility and/or precise zone entries are not required, then it can be a big help.

Recording videos of tests

⚠ *The videos ANY-maze records automatically are designed for use within the system, and are not suitable for playback in standard media players or for inclusion in PowerPoint presentations. If you want to record a video for these purposes, then refer to the Recording movies of tests for inclusion in a presentation topic.*

If you wish to automatically record a video of all the tests in your experiment, then you just need to check the box labelled *Automatically record videos of all tests.*

Automatically recording the empty apparatus

As you may know, ANY-maze tracks the animal by comparing the video pictures to a picture of the apparatus without the animal in it - the difference between the two is the animal. When performing tests 'live', ANY-maze is looking at the apparatus all the time, including when tests are not running, which means that when a test starts it has 'seen' the empty apparatus (just before the test began) which allows it to quickly determine the location of the animal.

However, when tracking from a video this is not the case, as the video will typically only show the apparatus *during* the test and therefore ANY-maze can have trouble determining the initial position of the animal, especially if it is motionless.

To address this, you can request that the system automatically appends a five second 'lead-in' showing the empty apparatus to videos of tests. This means that when you track from the video, ANY-maze will first 'see' the empty apparatus, which will allow it to quickly determine the location of the animal when the test begins.

If you switch on the option to *Automatically record videos of all tests,* then ANY-maze will automatically select the option to *Automatically record the empty apparatus at the start of the video,* and you will usually want to simply leave it switched on.

If for some reason you don't want to record the empty apparatus 'lead-in', then you can switch this option off - but be aware that will make it harder (although not impossible) to track the animal in the recorded video.

Recording quality

When you choose to record videos of tests, ANY-maze will compress the video files as they would otherwise be extremely large.

The system includes three standard video compression schemes - *Best quality, Medium quality* and *Low quality,* and you can define new ones yourself using the ANY-maze video recorder settings window, which can be accessed by clicking the *Settings* button.

In general, unless your tests are very long, recording at Medium quality is recommended. However, for long tests you may wish to use low quality so as to conserve disk space, or you might want to create your own custom compression scheme by clicking the *Settings* button and using the options in the ANY-maze video recorder settings window.

Playing the recorded videos

Of course, if you record a video, you'll presumably want to watch it - and this leads to the question of how to access the files which the system records.

In fact this is very simple - if ANY-maze records a video of a test, then it will include it in the *Related reports* list shown on the Tests page when the specific test is selected.

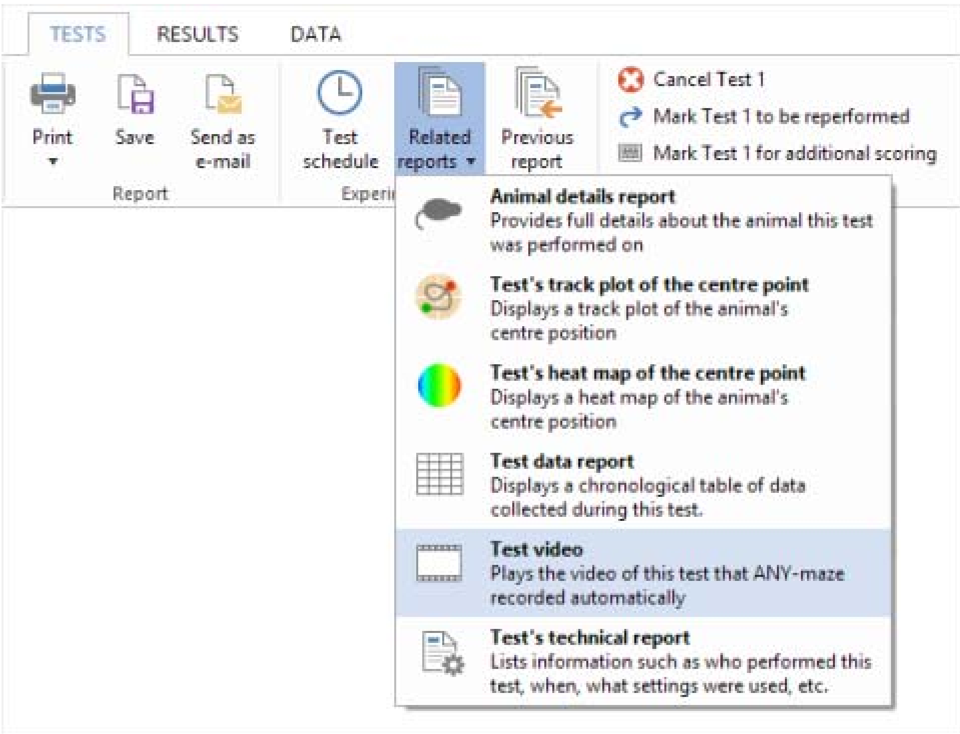


Figure 2. If ANY-maze records a video of an animal's test, then it automatically

includes a link to the video in the Related reports list.

Where the video files are saved

When you automatically record a video of a test, the video itself is stored in a folder with the same name as the experiment file, but with the word 'videos' appended to it. This folder is located in the same place as the experiment file. For example, if you have an experiment file *c:\my experiment\drug 123.szd,* then the videos will be located in the folder *c:\my experiment\drug 123 videos.*

If you wish to move the experiment then you should copy both the experiment file and this folder to the new destination (which could be a CD or DVD if you want to archive the experiment).

Controlling the video recorder during a test

Even if you're automatically recording a test, you can still control the video recorder manually. For example, you can pause the recorder using the button or stop it using the button.

In fact, you can use procedures to control the VCR as well. For example, you could set up a procedure which would pause the VCR when the animal becomes immobile and restart it when it becomes mobile again - this would offer a neat way of avoiding recording periods when the animal's asleep.

Indeed, you can even *start* the video recorder using a procedure. For example, you might want to record periods when the animal is in a certain zone of the apparatus - but if it never enters the zone, you're not interested in recording a video at all. In this case, rather than using the option on the *What to record while testing* page to automatically record every test, you could instead just create a procedure to switch on the video recorder when the animal enters the relevant zone. In this case, only those tests in which the animal actually entered the zone would have a video recorded.

By the way, if you record a video in this way, the video quality will be whatever is set on the *What to record while testing* page (that's why the quality settings are always enabled, even if the *Automatically record videos* button is not checked). Also, the video files will be recorded in the same place as they would have been recorded if the *Automatically record videos* button was used - see Where the video files are saved, above.

What next?

After completing these steps, you should consider what you want ANY-maze to display during your tests.

What to display while testing

Introduction

There's no necessity for ANY-maze to indicate anything other than the test time while tracking. However, you will probably want to see at least where it considers the animal to be.

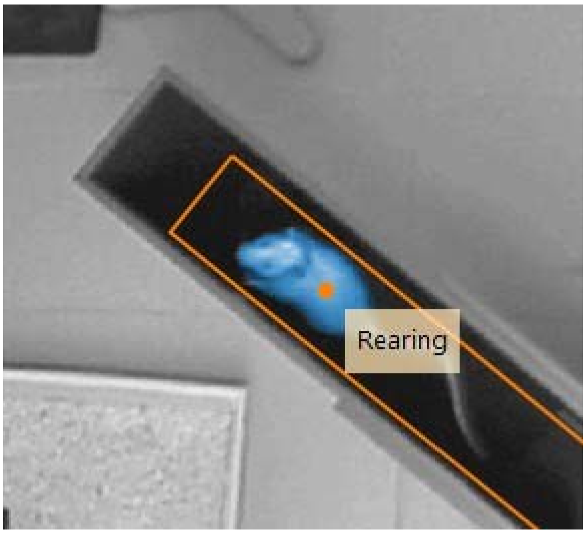


Figure 1. In this plusmaze test, ANY-maze is indicating the animal's entire area (in blue), the position of the animal's centre (the orange dot), and the fact that the experimenter is registering that the animal is rearing by pressing a key on the keyboard.

- Displaying time remaining or time elapsed
- Indicating the position of the animal
- Indicating which zone (or zones) the animal is in
- Indicating which 'behaviour' keys are pressed
- Indicating which switches are active
- Indicating whether the animal is immobile
- Indicating whether the animal is freezing
- Indicating whether the animal is rearing
- Choosing where the indicators are shown

Displaying time remaining or time elapsed

By default, ANY-maze will show the elapsed time in a test, i.e. it will count up - but if you prefer, it can show the time remaining in the test instead, i.e. count down.

Indicating the position of the animal

ANY-maze can always indicate the centre point of the animal during a test which it does using an orange dot (you can change the colour if you like). This is the most basic animal position indicator and usually you'll want to have it switched on.

You can also choose to have the entire area which the system considers to be the animal shaded in blue. However, you should take care interpreting this area as the system will sometimes shade less than the entire area of the animal (or sometimes more), particularly when there's poor contrast between the animal and the apparatus background. However, this doesn't necessarily mean that the tracking won't work correctly (although head tracking will be unreliable in these conditions); it just means that the system is unable to accurately detect the exact shape of the animal.

As well as showing the centre of the animal, ANY-maze can also indicate the position of the animal's head and/or tail, although you need to have switched on head tracking for these options to appear. In this case, you can also have the animal's head-centre-tail joined with lines and/or the animal's centre-head line extended beyond the front of the animal, to show its orientation.

3Indicating which zone (or zones) the animal is in

If your protocol includes any zones, you can choose to have the name of a zone shown when the animal is in it. Of course, zones are not mutually exclusive, so the animal may be in more than one zone at the same time. In this case, showing the zone names can be beneficial as zone highlighting may not make it clear that the animal is in more than one zone.

This option is also useful if you're scoring behaviours using keys, as the zone highlighting can make it a little harder to see exactly what the animal is doing.

Indicating which 'behaviour' keys are pressed

If you use any keys to score behaviours which ANY-maze can't detect automatically, such as grooming, then you can optionally have the behaviours' names shown when the animal is exhibiting the behaviour. This is particularly useful if you are using *toggle keys* (see key types) as it helps you keep track of which keys are on and which are off.

Indicating which input and outputs are active

If you have any inputs or outputs defined in the protocol, you can choose to have their names shown when they're active - this can be useful in situations when it would otherwise be unclear whether an input or output is active or not. For example, it might not be obvious to you, while watching a test, when a photobeam is broken by the animal; so including an indicator to show this would probably be a good idea.

Indicating whether the animal is immobile

If you've chosen to detect immobility, then you can choose to have the word 'Immobile' shown whenever ANY-maze considers that the animal is immobile. This can be particularly helpful if you're trying to determine the best immobility settings for your apparatus.

Indicating whether the animal is freezing

If you've chosen to detect freezing, then you can choose to have the word 'Freezing' shown whenever ANY-maze considers that the animal be freezing.

You can also choose to have the actual freezing score (see Freezing detection) shown, which can be particularly helpful if you're trying to determine the best freezing thresholds for your tests.

Indicating whether the animal is rearing

If your apparatus is viewed from the side, then ANY-maze will automatically detect periods when the animal is rearing. If you intend to use this feature in your results, then it's usually a good idea to switch on the rearing indicator to confirm that the detection is working OK.

Choosing where the indicators are shown

ANY-maze can either show the *Key, Switch, Immobile and Rearing* indicators next to the animal's position (as is the case in figure 1, above) or it can show them in the top right corner of the video image.

What next?

After completing these steps, you should consider whether you want to include any procedures in the protocol.

See also:

- Zone highlighting
- Changing ANY-maze colours

Procedures

Procedures are a way of instructing ANY-maze to respond to things that happen during a test, and take action to affect the way the test runs. They are incredibly flexible, and can be as simple as ending a test when an animal enters a zone, or can contain complex logic to interact with I/O devices.

Procedures were introduced in ANY-maze V5.10 as a replacement for Events and Actions. *They can be used to do anything that events and actions could do, and are simpler, more flexible, and easier to maintain.*

For more information about procedures, refer to the topics below.

- An introduction to procedures
- Setting up a procedure
- Editing a procedure
- Deleting a procedure
- Writing a procedure using the procedure editor
- Elements of a procedure
- Hints and tips for writing procedures
- Re-using parts of procedures
- Running procedures
- Debugging procedures
- Procedures and the end of a test
- Preventing a test from starting until an I/O device is ready
- Sharing and re-using procedures
- Procedure measures
- Time markers
- The changeover from Events and Actions to Procedures

Retaining the state of an I/O element from one test to the next

Introduction

The details outlined in this topic apply to all types of I/O output, although the explanation in the example is described for Output switches.

In some experiments, you may wish to maintain the state of outputs from one trial of an animal to the next. For example, in a radial arm maze with motorised doors, you might open and close the doors based on the arms the animal visits. At the end of one trial you might want to retain the current open/closed state of all the doors so that they are the same at the start of the animal's next trial.

Details

*IMPORTANT: In this topic, I'll describe how to retain the state of an output from one **trial of an animal** to the next trial of the same animal. This is not the same thing as retaining the state from one **test** to the next test, because sequential tests are not necessarily performed on the same animal.*

I'm going to continue with the example above, where we want to retain the opened/closed state of the doors of a radial arm maze from one trial to the next, but the technique I'll be describing can be applied to any type of output.

Clearly, there will be some rules that determine when a door should open and when it should close, but from the point of view of this discussion it doesn't matter what these are - all that we're interested in is how we can save the state of the doors at the end of a trial and restore it at the start of the next one (even if we exit from ANY-maze). So, for the rest of this topic we'll just assume that some existing procedure changes the state of the doors while the test runs, without worrying about its details.

As you may know, Procedures include lots of so-called *built-in variables,* whose values reflect something the system itself already knows about; for example, there are built-in variables for the day of the week (the system already knows what day it is), the test clock (the system knows how long the test has been running for) and, usefully for us, the state of outputs, such as the doors in our example (the system activates/deactivates the outputs, so it knows their state).

So if we want to store the state of the outputs at the end of a trial, all we need to do is note the state of these built in variables when the trial ends, and then use the note to set the state of the doors when the next trial begins. Of course, that raises an obvious question: where can we 'note' their state?

As well as built-in variables, procedures allow you to define your own 'user defined' variables - I won't go into all the details of these 'user defined' variables now (full details can be found here), but suffice to say that to add a new variable to a procedure you simply need to click the *Create variable* button (which is included in the ribbon bar when a procedure is being edited). This will open the *Create variable* window shown in figure 2, below.

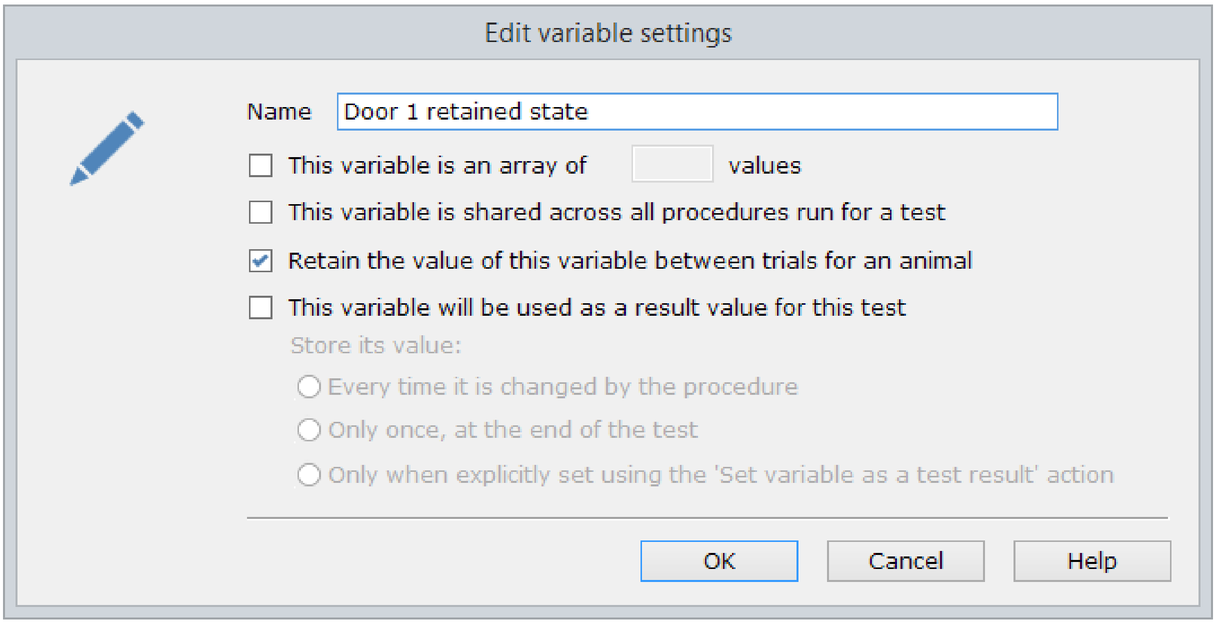


Figure 2. The Create variable window. Note that here, the option to 'Retain the value of this variable between trials for an animal' has been selected.

In this window, you can name the variable and also set various characteristics. What we want to do is to use a variable to store the state of a door between one trial and the next, so we would call our variable something like 'Door 1 retained state' and we would choose the option to *Retain the value of this variable between trials / stages* and then *For the current animal.* ANY-maze will automatically store this variable for us, and guarantee that whatever value it had at the end of one trial, it will have the same value at the start of the next trial of the same animal.

So, we now have somewhere to store the state of a door between trials, and we can also discover what the state of the door is at the end of a trial (using the built-in variable), so all that's left is to link this information together - which we can do using a procedure like the one in figure 3, below.

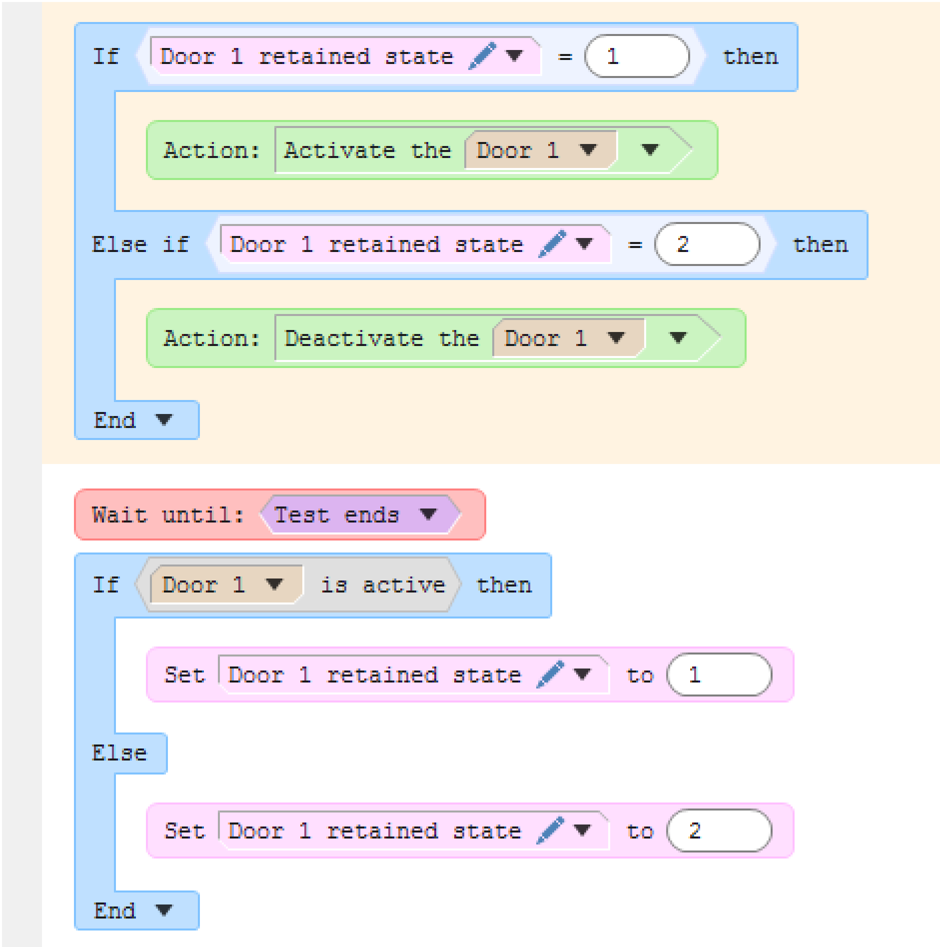


Figure 3. A procedure to retain the state of Door 1 between the trials of an animal. This procedure uses the variable 'Door 1 retained state' shown in figure 2.

It's easiest to understand this procedure by looking at the part with the white background first. Here, we start off by waiting until the test ends. When that happens, we look at the state of the 'Door 1' built-in variable, and if it is active (door open) we store a value of 1 in the 'Door 1 retained state' variable. If it is inactive (door closed) we store a value of 2. Remember, the value of the 'Door 1 retained state' variable will be stored by ANY-maze and the variable will have the same value in the animal's next trial.

Now let's look at the part of the procedure with the orange background. As you may know, the statements in the area with the orange background are run when a test is 'Ready' but before it begins. So what happens here is that we check the value of the 'Door 1 retained state' variable (which will be the value it had at the end of the previous trial) and if it is 1, we set Door 1 active (i.e. open it) and if it is 2, we set Door 1 inactive (i.e. close it). Voila - we have achieved our goal!

Well, almost... this does leave one question, what happens in the animal's first trial? Clearly, there's no previous trial and so the 'Door 1 retained state' variable will not have been set, so when the first trial is 'Ready' and the statements in the area with the orange background are run, what will they do? The answer is that they'll do nothing, and the reason is because the 'Door 1 retained state' variable will start off (as do all variables in ANY-maze) with a value of 0. And as you can see in the procedure listing in figure 2, when the variable has a value of 0 the procedure won't take any action at all - thus the doors will simply be left alone (presumably, some other procedure will set them to whatever their initial states should be).

Clearly, the procedure shown in figure 2 only processes door 1 and, of course, our radial arm maze will have more doors than this, so we'd need to either add similar procedures for doors 2, 3, etc., or edit this procedure to process the other doors as well; either approach is fine.

Working with other types of output

In the previous section, I described how to retain the state of some doors in a radial arm maze. Within ANY-maze, these doors were being controlled by *Output switches,* but what if you wanted to retain the state of some other type of output, for example, a shocker (so the intensity would be retained between trials) or a speaker (so what's playing is retained between trials)? The answer is that you could use exactly the same technique, as all of ANY-maze's output elements include built-in variables that reflect their current state.

Older experiment files (prior to ANY-maze version 6).

Older versions of ANY-maze (prior to version 6) used to have an option to *not* reset the output at the end of the test, for some outputs. If you have an experiment file where this option was selected, it will still appear for backward compatibility.

However, this option has been removed in newer versions of the software, as it would not quite achieve the desired effect. This is because:

- The state will be retained from one *test* to the next, but not from one *trial* to the next. For example, imagine that your test schedule is such that you will test animal 1 in its first trial, then animal 2 in its first trial and then animal 1 in its second trial, etc. What would happen is that after animal 1's first trial the state of the outputs (i.e. the open/closed state of the doors) would remain the same and so this would be the state that would apply to animal **2's** first trial... which is probably not what you want.
- Even if the above didn't apply (for example, in your schedule you might test animal 1 four times in a row), then if you exited ANY-maze after trial 2 and then restarted it again, the state of the outputs would be lost.

So, not resetting the outputs at the end of a test *doesn't* provide a way to achieve our goal, and for this reason, the option has been removed from the software. The solution is to use a Procedure to save the state of the outputs at the end of one trial, and restore it at the start of the animal's next trial.

See also:

- An introduction to procedures
- Built-in variables
- User-defined variables
- Output switches
- Speakers
- Analogue outputs
- Temperature controllers
- Lighting controllers
- Syringe pumps
- Shockers
- Laser controllers
- Selectors

An introduction to procedures

Introduction

Procedures essentially consist of a list of 'statements' that are run in sequence during a test. They can be used to wait for specific events to occur, to perform actions, or to execute complex logical operations to aid in the running of a test.

Procedures can be simple – for example, just executing a single action in response to an event occurring - or can be designed with complex logic that can respond to animal movement, I/O states, time, or any combination of these.

For example, a procedure in a water-maze could simply wait for the animal to enter the platform zone, then end the test. Or in a radial arm maze, a procedure could be used to open a series of doors based on the last zone entered in the animal's previous trial, and to respond to the animal's subsequent movement by stimulating other I/O outputs.

Some basic concepts

Here are a few general concepts that you need to know before creating procedures in ANY-maze:

- Any number of different procedures can run during a single test.
- A procedure consists of a number of 'statements' that are run successively.
- A procedure will run through its statements from start to finish (this may involve the procedure looping back to repeat a series of statements). Once the last statement has been executed, the procedure will stop running (the test itself will continue).
- If the procedure encounters a statement that causes it to wait (for example, waiting for a specific event to occur such as the animal entering a zone), the running of that procedure will pause until that condition is met. Once the condition is met, the procedure will continue executing.
- Procedures are set up using a graphical editor, which involves dragging statements and their parameters from a list of available items into the procedure.
- Procedures are split into two sections – 'before the test starts' and 'after the test starts'. Statements in the first section will run as soon as the test is ready, and then the procedure will wait for the test to start. Once the test starts (either because you have clicked the *Start* button, or the test is set to start automatically and ANY-maze has detected the animal in the apparatus) then the statements in the second section will run.

A simple example

The following procedure will wait for the animal to become immobile for 5s, and will administer a shock. This process will be repeated throughout the test.

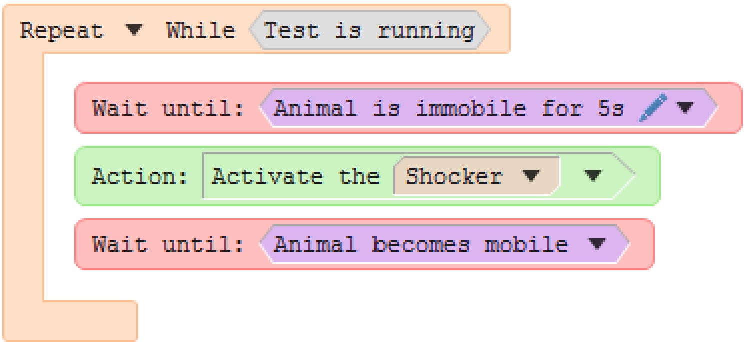


Figure 1. An example of a procedure that uses a number of statements to control a shocker, based on the movement of the animal.

See also:

- Setting up a procedure
- Editing a procedure
- Deleting a procedure
- Writing a procedure using the procedure editor
- Hints and tips for writing procedures
- Re-using parts of procedures
- Running procedures
- Debugging procedures
- Procedure measures
- The changeover from Events and Actions to Procedures

Setting up a procedure

Introduction

For a general introduction to procedures, see An introduction to procedures.

To add a procedure to a protocol, click the *Add item* button in the ribbon bar and select *New procedure* from the menu which appears.

The New procedure *menu option will be disabled until the protocol includes at least one apparatus element.*

What you need to do

Each of the steps below is a link; you can click them for more information.

1. Enter the procedure name.
2. Specify whether this is a sub-procedure
3. Specify whether the procedure is disabled.
4. Specify the stages that this procedure will run in.
5. If necessary, record some notes about the procedure.
6. Write the procedure using the procedure editor.

What next?

After completing these steps, you should decide whether you want to use Treatment groups in your experiment, and if you do, how you want ANY-maze to assign the animals to the groups.

See also:

- Adding elements to a protocol
- Editing a procedure
- Writing a procedure using the procedure editor
- Deleting a procedure

Procedure name

In brief

Enter a name for the procedure in the *Procedure name* field on the procedure's settings page. You must make an entry, but it can be anything you like.

Details

Procedure names should uniquely and clearly identify a procedure. If any problems are encountered in the running of a procedure (See Errors and warnings while running a procedure), then these will be listed in the Test details report against the name of the procedure.

If the procedure is used to end the test, then the name of the procedure will be used as the test end reason. This is useful because the test end reason can be used as an independent variable in the analysis of results.

Limits

You can enter anything you like up to a maximum of 32 characters.

Procedure is a sub-procedure

In brief

You can use the check box on the procedure's settings page to specify that a procedure should be a sub-procedure. A procedure flagged in this way will not run directly during a test, but will only run if it is specifically invoked by another procedure.

Details

A sub-procedure can be used to make a procedure easier to understand by moving chunks of code into their own self-contained units. This is particularly useful if you have similar code that is calle dmore than once in the same procedure, or even called from more than one different procedure. These self-contained units are called 'sub-procedures' and it is here that you specify that a procedure will be used in this way.

For more information, see Re-using parts of procedures.

Disable procedure

In brief

You can use the check box on the procedure's settings page to specify that a procedure should be disabled. This will prevent the procedure running during tests.

Details

Disabling a procedure is something that you may want to do while you're first writing the procedure(s) for a protocol. Setting up a procedure for a test may take some trial and error to get things running just right; this is particularly true if you have multiple procedures running for the same test (see Running procedures).

During this time, you're likely to want to try out perhaps one procedure at a time, to see if it's working correctly, without worrying about any errors that are encountered in the running of a procedure (See Errors and warnings while running a procedure).

This feature allows you to temporarily disable a procedure, so that it's not run during a test.

This feature shouldn't really be used once you actually start running an experiment. For example, if you want to run the procedure during a specific Stage of a test, but not for others, you should use the Stages option on the procedure's settings page to specify which stages that the procedure should run in. If you only want to run the procedure for certain trials, animals or even only on a specific day of the week, this can be achieved within the procedure itself by using built-in variables.

See also:

- Errors and warnings while running a procedure

Choosing the stages in which a procedure should run

In brief

You should use the list of stages on the procedure's settings page to select those stages in which you want the procedure to run. If you un-check a stage, then the procedure will not be run in *any* trials in that stage - it'll be like the procedure simply doesn't exist.

Details

In general, you will probably want ANY-maze to run a procedure in all the stages of an experiment - indeed, this is the default setting used by the system. However, there may be experiments in which the procedure is not required all the time.

For example, you might wish to shock the animal in some situation during a *Training* stage, but not during a *Retention* stage. In this case, you could un-check the *Retention* stage in the list of stages, for the procedure that administers the shock.

Recording procedure notes

In brief

If you want to, you can record notes about a procedure in the notes field on the procedure's settings page - there are no limits to what you can enter.

Details

You may wish to use the procedure notes to record information about the procedure; for example, a brief description of what the procedure does. This can be particularly useful if the protocol is to be shared between different users. It's also useful for your own benefit, if you come back to the procedure several months after writing it - you might not remember much about how the procedure works or what its purpose was!

Formatting

Since the procedure notes field is designed to help out other users of ANY-maze, you might want to format the text in such a way as to make it easier to read. For example, you might want to underline some titles, or make sections of the text bold, or a different colour (or both!).

When the procedure notes field is currently active (i.e. the cursor is in that field) then the *Procedure notes format* section of the ribbon bar will be enabled. This contains a number of options, which apply to any text that is currently selected in the notes field. (If no text is currently selected, then the formatting you select will apply to new text when you start typing).

Reset formatting	Resets the formatting of the selected text to the default formatting (i.e. removes any colour, bold, italic, etc.)
Text colour	Opens a dialog box allowing you to select a colour for the selected text.
Bold	Makes the selected text **bold.**
Italic	Makes the selected text *italic.*
Underline	Makes the selected text underlined.
Increase text size	Increases the size of the selected text by one point size.
Decrease text size	Decreases the size of the selected text by one point size.

Limits

There are no limits to the amount of text you can enter about a procedure.

Editing a procedure

Introduction

You can edit absolutely everything related to a procedure at any time, either before, during or after the experiment. However, although there are no restrictions on what you can edit, it's not usually a good idea to edit the statements in a procedure after an experiment has been started.

This is because unlike some other elements of a protocol (e.g. zones, points, etc.) any edits to a procedure will *only* take effect from the next test. Any tests that have already been run using a procedure will not be affected - they have been run with the previous procedure, and there's nothing that can change that.

See also:

- Editing the elements of a protocol
- Setting up a procedure
- Writing a procedure using the procedure editor
- Deleting a procedure

Deleting a procedure

Introduction

You can delete a procedure at any time, either before, during or after an experiment has been performed, as long as it has not been used to end the test (see below). However, please consider the notes below before doing so.

To delete a procedure, simply select it in the protocol list and click the *Remove item* button on the ribbon bar.

Things to consider

Any tests that have already been run will *not* be affected by the removal of a procedure - any actions that the procedure took will already have happened. Bear in mind that deleting a procedure before running further trials will mean that the trials will be run under different conditions to those already run - you may not want this to happen!

Restrictions

You cannot delete a procedure that has been used to end a test. When a procedure ends a test, the procedure is noted as the 'test end reason', and this can be used as an independent variable in the analysis of results. ANY-maze will prevent you from deleting such a procedure, so that this test end reason can still be used in analysis.

This is true if the procedure has ended the test using the End the test ***action****. A procedure can also end the test by specifying a specific test end reason, in which case the procedure itself is no longer needed and could be deleted. However, you should still think carefully before deleting the procedure!*

See also:

- Deleting protocol elements
- Editing a procedure

Writing a procedure using the procedure editor

Introduction

The ANY-maze procedure editor is used to write or edit a procedure. The editor appears to the right of the procedure settings pane when a new procedure is created, or an existing procedure is selected for editing.

The left-hand side of the procedure editor contains various lists of items that can be used in the procedure. These items are the building blocks of a procedure, and you just need to drag them into the right-hand side of the editor to form the procedure.

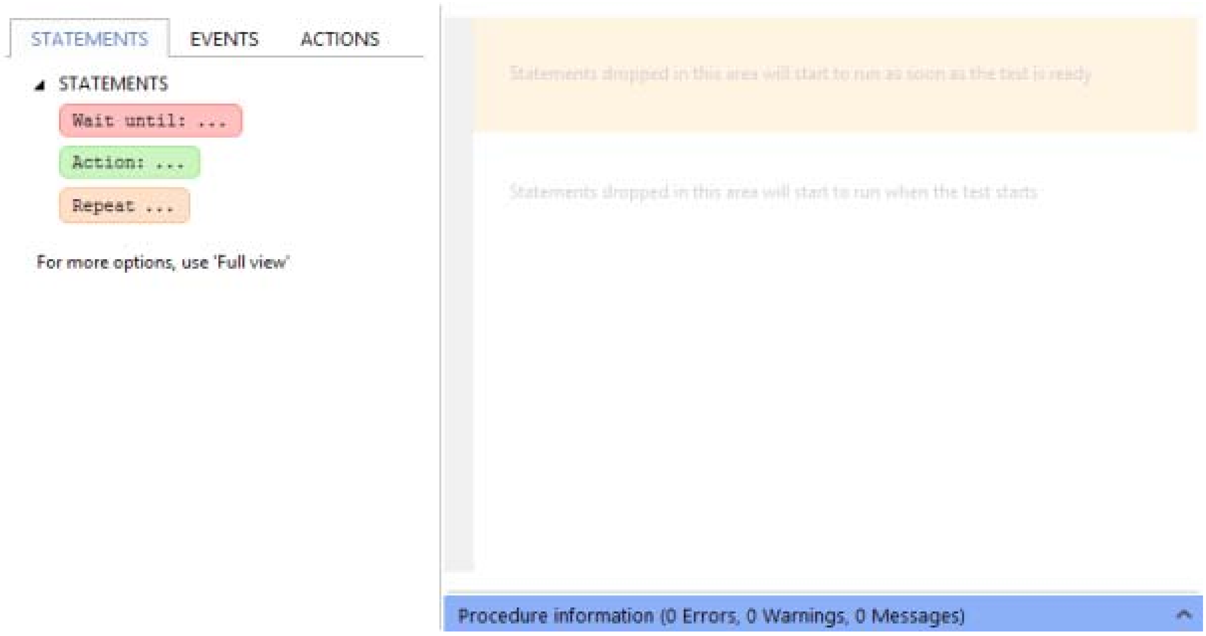


Figure 1. The ANY-maze procedure editor.

Writing a simple procedure

Rather than describe all the different aspects of a procedure, let's start by creating a very simple example. The concepts in this example will be useful for creating your own procedures, and hopefully it'll very quickly demonstrate both the power and simplicity of procedures.

For this example, let's say we have set up a water-maze experiment. There's a zone called 'Island', and we want to end the test when the animal enters the Island zone.

If you've used Events and Actions *in ANY-maze before, then you'll probably realise that this corresponds to a very simple Event (Animal enters the Island zone) and Action (End the test). In fact procedures use a very similar system - but can be much, much more powerful!*

If you want to work through this example yourself, you can do so using the example experiment 'Water-maze videoed experiment'. If you'd like to do this, open this example, switch to the Protocol page and select the *Add item* button from the ribbon bar. Select *New procedure* from the list, and on the procedure settings page, change the procedure name to 'Found Island'. (Don't worry about the fact that there's already a procedure there; this is just for the purposes of creating an example and won't be saved).

If the 'Water-maze videoed experiment' does not appear in the list of example experiments, you can download it from the ANY-maze web site at www.anymaze.com/ files/examples/water-maze example experiment.zip. *This contains two files (the example experiment, and the video that it uses), which you'll need to unzip into the 'Examples' sub-folder of the ANY-maze program folder - by default, this is* `C:\Program Files (x86)\ANY-maze` *on a 64-bit operating system, or* `C:\Program Files\ANY-maze` *on a 32-bit operating system.*

Note that the name of the procedure that you've just entered is important - we'll find out why later on in this worked example.

So what do we need our procedure to do? Well, the first thing to do is try and describe what you want to happen as a series of simple steps, i.e.

- Wait until the animal enters the 'Island' zone
- End the test

Let's create a procedure that does this.

The first thing we need the procedure to do is to wait for something to happen. So, let's drag a *Wait until* block into the procedure:

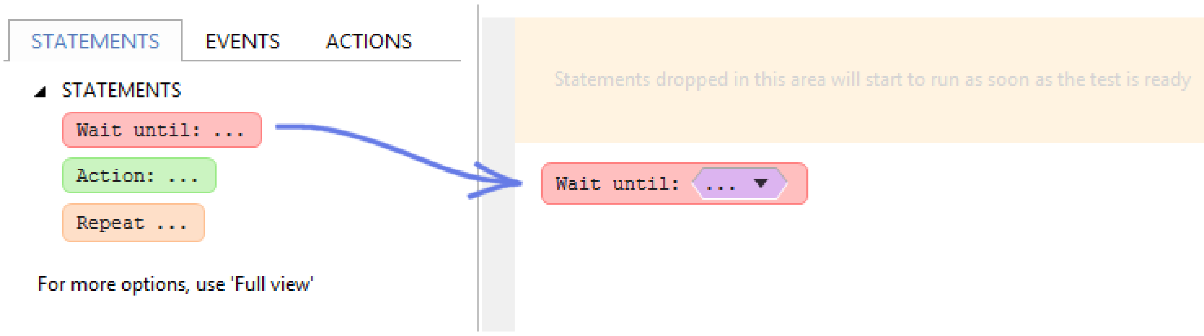


Figure 2. Dragging a Wait until *statement into the procedure*

There are two things to note here:

- Firstly, the *Wait until* block is something called a statement. This is something that controls the flow of the procedure, and a procedure is made up of one or more statements.
- Secondly, when we dragged the statement into the procedure, we dragged it to the *bottom* of the empty procedure, over the white area, and not the shaded area at the top. This is because these two sections actually represent different parts of the procedure - the shaded section will be run *before* the test actually starts, and the white bit will be run as soon as the test starts. Obviously the animal cannot enter the 'Island' zone before the test starts, which is why we've put it in the lower section of the procedure.

So now the procedure is waiting for something, but we need to tell it what it's waiting for. Things that happen in a procedure are called Events, so let's select the *Events* tab:

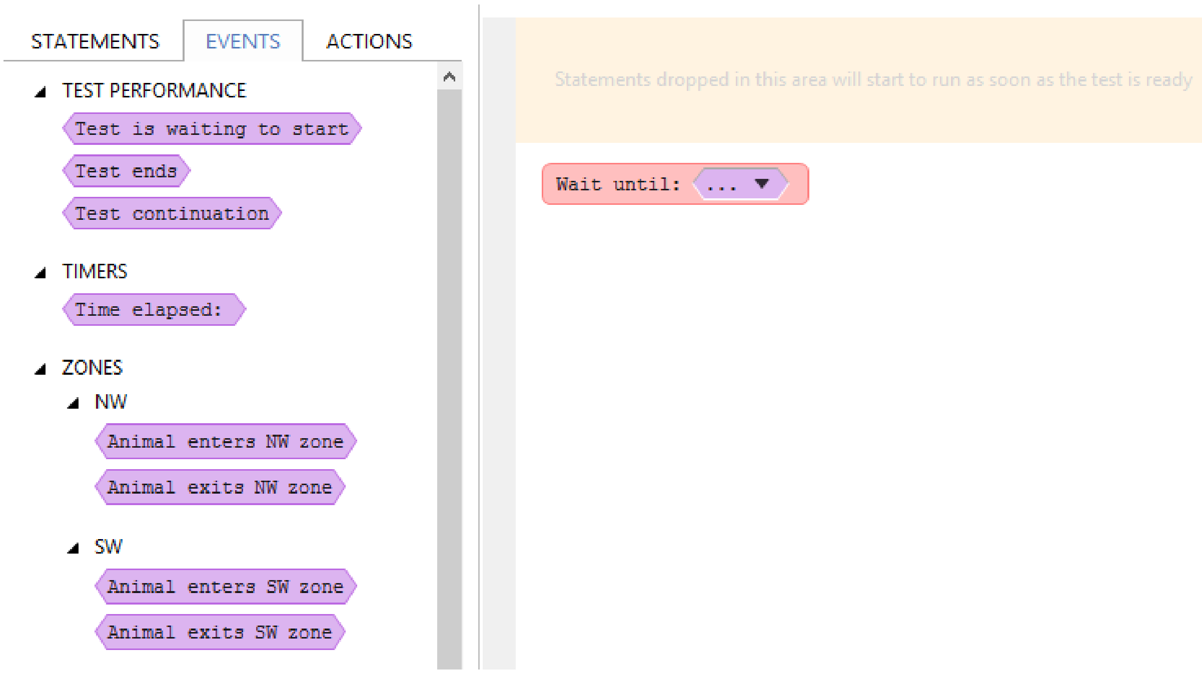


Figure 3. The Events tab.

Here we can see a list of all the possible events that are available to procedures. This contains animal position and activity events, zone entries and exits, and any events related to I/O devices being used by the protocol. In this case, we don't have any I/O, but this example experiment *does* have a number of zones, including the 'Island' zone that we want, and the events for these zones are at the bottom of the list. The headings in the list of events can be expanded or collapsed, just by clicking on them, so we can easily collapse the events that we're not interested in:

Figure 4. Collapsing the events we're not interested in.

Now we need to use the mouse to drag the *Animal enters Island zone* event, and drop it over the purple 'hole' in the *Wait until* statement:

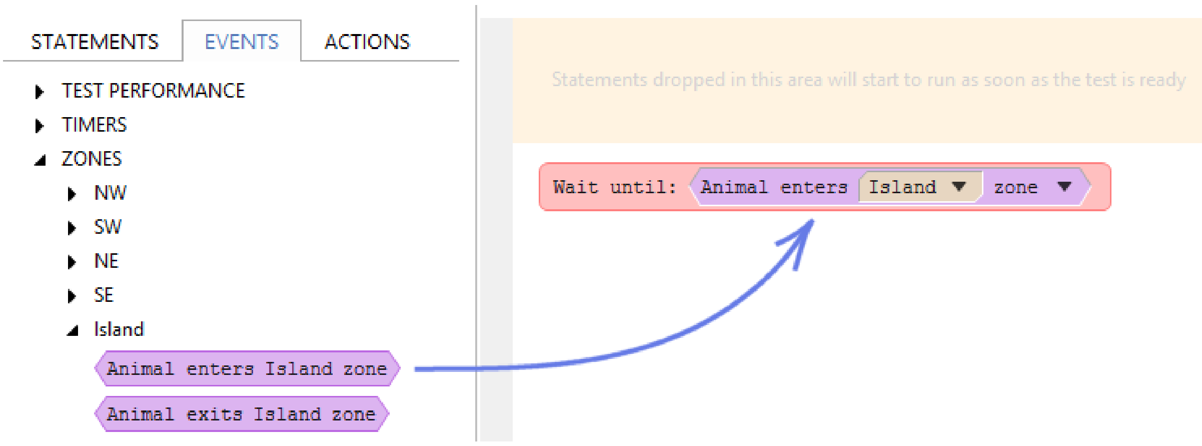


Figure 5. Dragging the event into the procedure.

Again, there are a couple of things to note here:

- The events are shaped with angled edges at each end; the 'hole' that we dragged it over had the same shape. In general, the ANY-maze procedure editor indicates what kind of object can be used in that space by the shape of its border. As you drag an item, a green outline will be shown over the places that it can be successfully 'dropped'.

- The items that we've dropped into the procedure editor have little drop-down arrows at the right-hand side. These can be clicked on to show all the possible options that can be used here. So in our example, we can drop the arrow next to 'Island' to quickly and easily change the zone whose entry we want to wait for, or we can drop the arrow at the right-hand side of the event itself to select an entirely different event. This allows for quick editing of a procedure, without having to find the building block on the left-hand side and drag it into the editor.

Now our procedure is waiting for the animal to enter the island zone, so next we need to tell it to end the test when that happens. Things that the procedure actually does that affect the test are called Actions, so we need to tell the procedure to take an action that will end the test. So we need to re-select the *Statements* tab, and this time, drag across an *Action* statement:

Figure 6. Dragging the Action *statement into the procedure.*

This is just the *statement* telling the procedure to execute an action, but we haven't yet told the procedure which action to execute. To do this, we need to switch to the *Actions* tab to see the available actions:

Figure 7. The Actions *tab.*

This tab will show you all the actions that a procedure can take to affect the test. This includes test control, actions to start/stop recording, and I/O outputs (if any have been set up in the protocol). Again, all the headings can be expanded to show you everything that's available. Full details of all these actions can be found under Actions available to procedures, but for now we can find the one we're interested in at the top of the list: *End the test*.

Again, note the shape of the actions - they're the same shape as the 'hole' in the statement we've just dragged into the procedure. So we can drag the *End the test* action into the procedure:

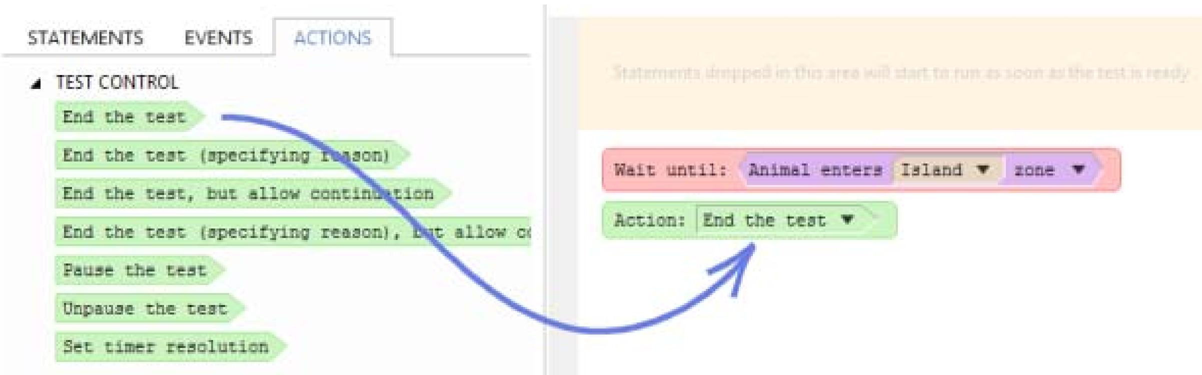


Figure 8. Dragging the action into the procedure.

An important point to note here is that every time a test ends, ANY-maze notes the reason for the test end. A test might end for one of a number of reasons - usually because the test duration elapsed, but it might also be because the user aborted the test, or (as in this case) because of a procedure action. These test end reasons can be used during analysis, for example, to include or exclude results from tests that ended for particular reasons. In this case, the test has ended because the animal found the island. When you use the *End the test* action, ANY-maze uses the name of the procedure as the test end reason. So you can see why, as I mentioned earlier, the name that you give the procedure is important!

We'll see in the **second worked example** *that you can generate your own test end reasons within a procedure, rather than relying on the name of the procedure. However that's slightly more complicated to set up, so if your procedure is relatively simple, it's best to just use the* End the test *action.*

Our procedure should now look like this:

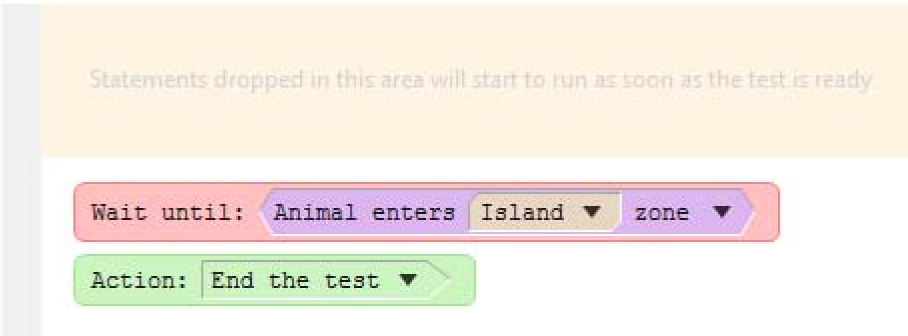


Figure 9. The completed procedure.

As with the event that we dragged earlier, you'll see that the action has a little drop-down arrow at its right-hand side. Again, this can be used to quickly and easily change the action that the *Action* statement will use.

In fact, you might have noticed earlier that this arrow appeared even *before* we added the action to the *Action* statement (and the *Wait until* statement had one as well, before we dragged the event into it). We *could* just have used these arrows to select the event / action that we wanted, rather than dragging them across from the left-hand side.

There's just one last thing to do before we've finished. Procedures *can* get quite complex, and if there are any errors in a procedure, it won't run (you'll be told about this before you try to run any tests, and you'll need to fix any errors before starting any tests). You can check that your procedure is valid at any stage, by clicking the *Check for errors* button in the ribbon bar. The bar at the bottom of the procedure editor should show that there are no errors or warnings:

Figure 10. The bar at the bottom of the procedure editor, showing that there are no errors or warnings.

If you forget to check the procedure like this, don't worry - the procedure will be automatically checked for errors when you move to a different page, or select another item in the protocol list. If there are any errors that will prevent the procedure from running, you'll be told about it and given the chance to correct them.

Summary

To create a procedure:

- Creating and editing a procedure involves dragging 'building blocks' from the left-hand side of the editor to the procedure on the right.
- The main 'building blocks' are called *statements*. These will run in order from top to bottom.
- Statements contain different-shaped 'holes', into which can be dragged parameters of the relevant type.
- Events and actions can be used in a procedure by selecting the relevant tab, and dragging the appropriate Event or Action into the relevant *Wait until* statement or *Action* statement, respectively.
- Within the procedure, you can sometimes use the drop-down list on the procedure item as a shortcut to bypass this dragging.
- Once you've finished writing a procedure, use the *Check for errors* button to make sure you haven't got any errors in the procedure you've just created.

What next?

An example that introduces some more procedure concepts can be found in the topic Writing a more complex procedure.

For even more advanced concepts, such as using multiple procedures and working with I/O, see A worked example using I/O. It's probably a good idea to work through this example even if you don't have any I/O connected, as it introduces some more useful features of procedures.

If you require any assistance with writing your own procedures, please contact ANY-maze technical support and we'll be happy to help.

See also:

- Elements of a procedure
- Hints and tips for writing procedures
- Re-using parts of procedures
- Errors and warnings while writing a procedure

A more complex example

Introduction

This topic describes how to create an example procedure to perform some slightly more complex functionality. If you haven't already done so, I highly recommend that you work through the simple example in the Writing a procedure using the procedure editor topic. This will give you some fundamental starting points.

This example will not guide you through the steps in quite as much detail as the previous one, but it will cover a few more aspects of what you can do with procedures.

What do we want the procedure to do?

Let's create a procedure that will end the test if the animal fails to enter a specific zone for a given length of time (say 20 seconds).

This is perhaps an unrealistic example, but it will serve to demonstrate some of the more advanced flexibility of procedures.

Set-up

Before we can write the procedure, we'll need to set up a protocol that has the right zones, so you'll need to do the following:

1. On the File page, select *New experiment* and then *New empty experiment*.
2. Give the protocol a name (e.g. 'Procedure example 2')
3. Select the *Treatment groups* item in the protocol list and specify that the animals should be randomly assigned amongst the groups.
4. Click the *Add item* button and select *New video source*.

   Give the video source a name and then select the 'Open field example video' from the drop-down list.

   If the 'Open field example video' does not appear in the list of example experiments, you can download it from the ANY-maze web site at www.anymaze.com/files/examples/open field example video.zip. *You'll need to unzip this file into the 'Examples' sub-folder of the ANY-maze program folder - by default, this is* `C:\Program Files (x86)\ANY-maze` *on a 64-bit operating system, or* `C:\Program Files\ANY-maze` *on a 32-bit operating system.*

5. Click the *Add item* button and select *New apparatus*.

Give the apparatus a name and then set up the apparatus as follows:

• Using the *Rectangle tool* button, draw the outline of the apparatus around the floor of the box.

• Move and resize the ruler to be along the edge of the box. In the *The length of the ruler line is (mm)* field, enter '400'.

• Use the *Grid settings* button to create a grid spaced at 10cm, crossing at the centre.

The resulting apparatus should look like this:

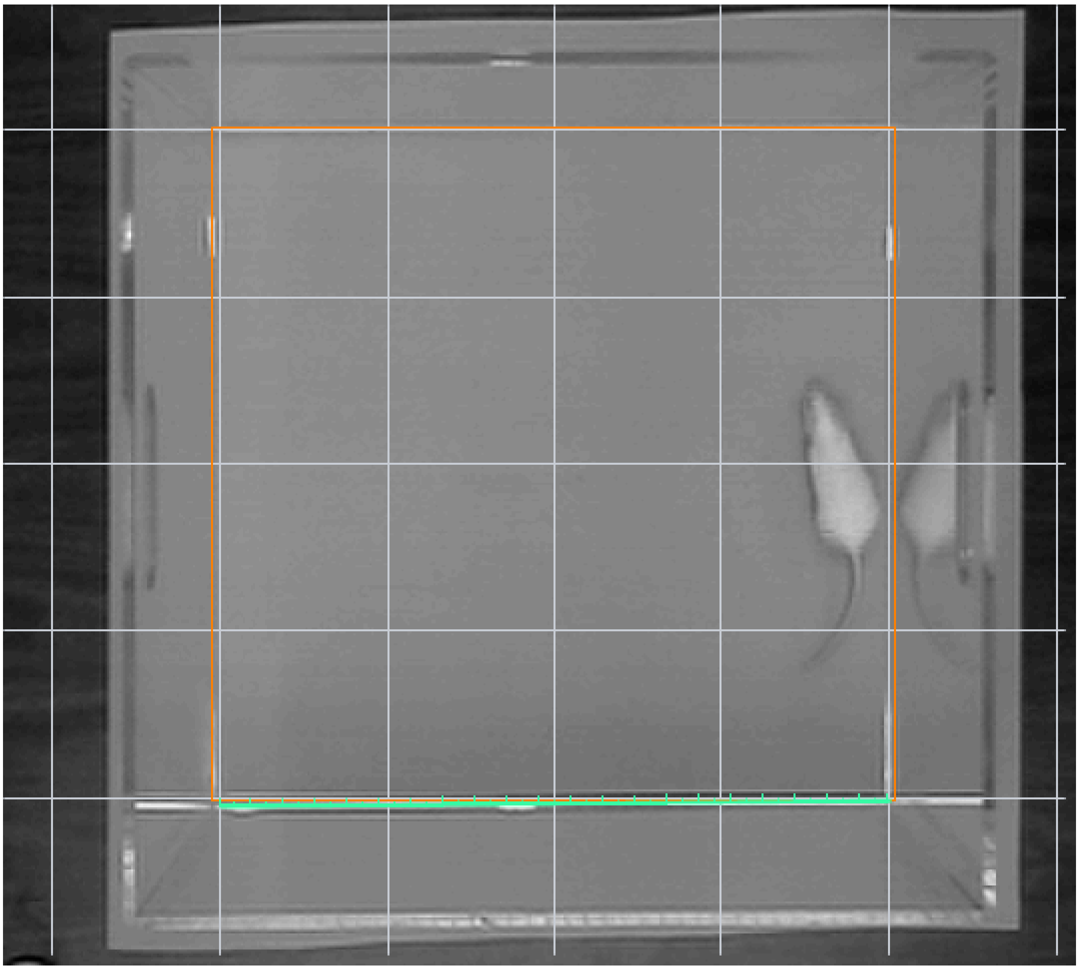


Figure 1. Set up the apparatus using the grid settings, so it looks like this.

6. Click the *Add item* button and select *New zone*.

   Specify a zone called 'Centre' and select the four squares at the centre, as follows:

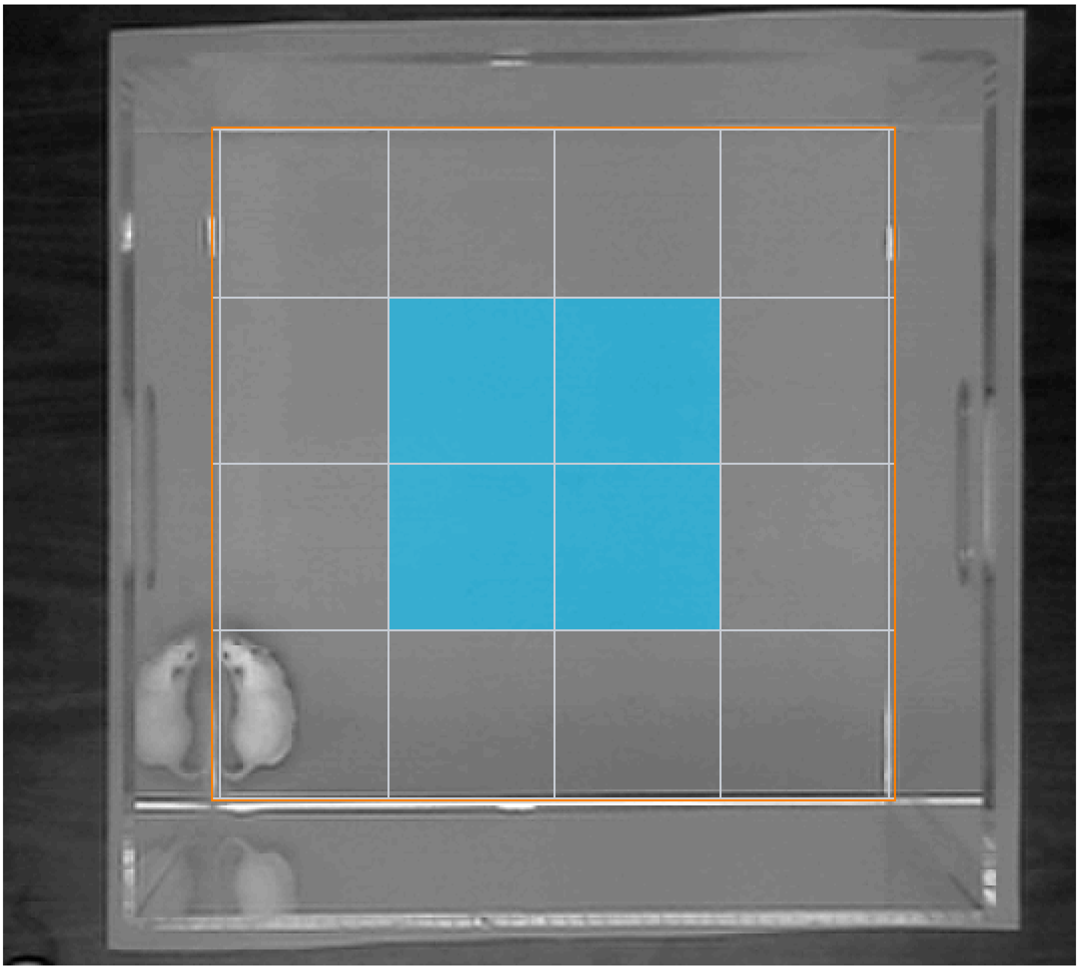


Figure 2. The centre zone.

7. Select the *Animal colour* item in the protocol list and specify that the animals are lighter than the background.
8. Under the *Stages* option in the protocol list, select *First stage* and enter a *Test duration* of 90s.

To finish off the setup, we need to switch to the Experiment page and enter at least one treatment and a number of animals, before saving the experiment file.

Creating the procedure

The first thing we need to do when creating a procedure is to try and summarise what it needs to do, in simple steps. In this case, we're waiting for the animal to enter the centre zone. If 20s elapses, without this happening, then we end the test.

The logic for this will be something along the following lines:

- Wait until the animal enters the zone, *or* 20s elapses.
- If 20s elapsed, then end the test.

Let's create the procedure - use the *Add item* button, and select 'New procedure'. Give it a name, say 'Procedure example 2'. (Note that you may remember from the first worked example that the naming of a procedure is important - but I've deliberately suggested a bad name here, which we'll come to later!)

Make sure the procedure is in 'Full' mode, rather than 'Simple' mode, using the *Full view* button on the ribbon bar. (If the procedure editor is already in 'Full' mode, you'll only see the *Simple view* button).

The first thing we'll do is drag in a Wait until statement:

Figure 3. The first Wait until *statement.*

But now we have a problem. We want to wait for one of *two* possible events to happen - either the animal entering the zone, *or* 20s elapsing. We can't add another *Wait until* statement, because the procedure would wait for the first event, and *then* wait for the second - and it would only get to the second *Wait until* statement if the first event had already occurred. Obviously, this is not what we want.

ANY-maze provides two Logical operators which can be used to check for multiple conditions at a time:

- AND This can be used to bundle together multiple True/False items, and will evaluate as True if ALL of them are True.

- OR This can be used to bundle together multiple True/False items, and will evaluate as True if ANY of them are True.

An important point, which is worth remembering, is that only one event can occur at a time in ANY-maze (and can therefore can be processed by procedures at a time) - so you can't use an *AND* operator to wait for multiple events. (In fact, if you try to do this in a procedure, it will result in an error when the procedure is checked for errors, or when you leave the procedure editor).

In our case, we need to wait for a zone entry OR time elapsed, so expand the *Logical operators* heading on the *Statements* tab, and drag the *OR* operator into the procedure:

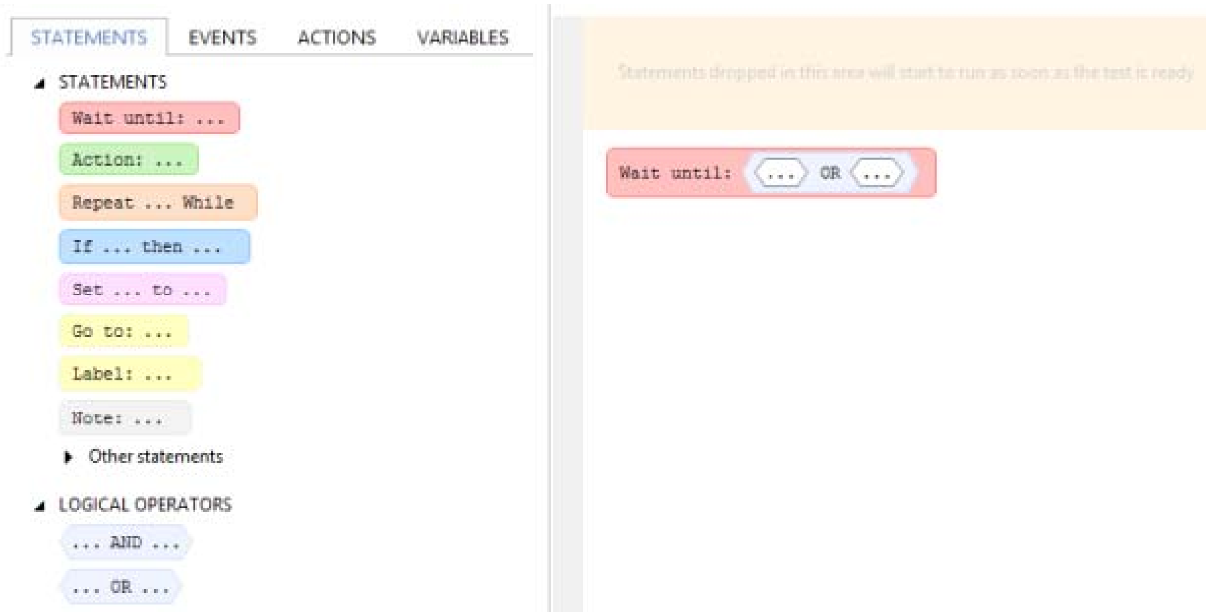


Figure 4. The OR *logical operator, dragged into the* Wait until *statement.*

A Wait until *statement* ***can*** *wait for multiple conditions as long as only one of them is an event - for more details, see the* ***Wait until statement*** *topic.*

Now we just need to fill in the two events that we're waiting for. Switch to the *Events* tab, and under the *Zones* heading, expand the *Centre* zone. Drag the *Animal enters Centre zone* event into the procedure as follows:

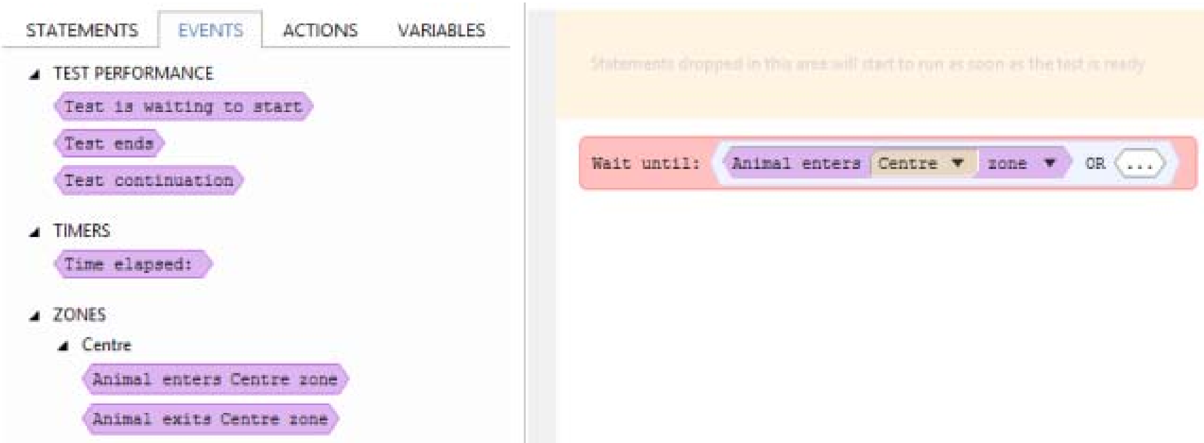


Figure 5. Dragging the zone entry event into the logical operator.

On the right-hand side of the *OR* operator, we need to specify a 'time' event, i.e. a specific time has elapsed. Just above the zone events in the list of events is a section entitled *Timers,* which contains a single event called *Time elapsed.* Drag this into the procedure, on the right-hand side of the *OR* operator.

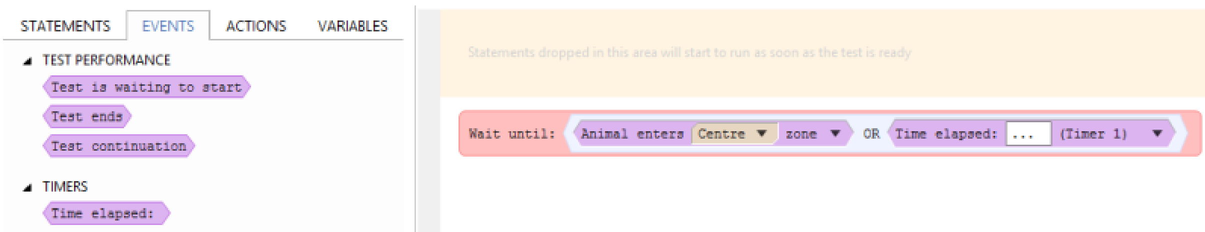


Figure 6. Dragging the timer event into the logical operator.

To finish setting up this timer, all we need to do is enter in the 'hole' in the timer event the time that we're waiting for - so click on the empty parameter and type in '20s':

Figure 7. Entering the time for the timer event.

Along with other places in ANY-maze that accept a time, you can type a time here in any valid format - for example '1min 25s', '0.5s', etc.

Some things to note here, before we continue:

- The *Time elapsed* event will start timing as soon as it is encountered by the procedure. So in this example, it starts as soon as the test has started; however if there were other procedure statements before it, the timer would not start timing until all those statements had run and the *Wait until* statement was first encountered.
- You may notice that the timer shows the text '(Timer 1)' after it. It's possible to have a procedure wait for multiple *Time elapsed* events, at different stages of a procedure, so each one is given a unique ID - this allows statements further on in the procedure to refer to this timer without any ambiguity. We'll see more on this in just a moment.
- What happens if there are *more* than two possible events that you want to wait for? For example, you might have four corner zones, and you want to wait for the animal to enter *any one* of the four. In this case, you simply 'nest' these operators - i.e. you can put an *OR* operator inside another *OR* operator (and then another *OR* operator inside that).

  In fact, this applies to all Logical operators - any number of *AND* and *OR* operators can be nested inside each other, to any level (although I wouldn't advise doing too much of this, as it can make the procedure very difficult to read!)

The procedure is now set up to wait for the relevant events, but what do we want to do when one of them happens? Well, if it's the zone entry event that occurred, we have nothing further to do - but if it's the 20s elapsed, we need to end the test. So we need a way of checking which event it was that happened.

The way we do this in a procedure is to use an If statement. This statement checks for one of a number of conditions, and can execute a different set of statements depending on which condition is true. In our procedure, we only need to check for one thing - whether it was the time that elapsed (rather than the zone entry) which caused the procedure to continue running after the *Wait until* statement. So we'll just use a simple form of the *If* statement.

Switch to the *Statements* tab, and you'll see the fourth statement in the list is the *If* statement. Drag it into the procedure, below the *Wait until* statement, as shown here:

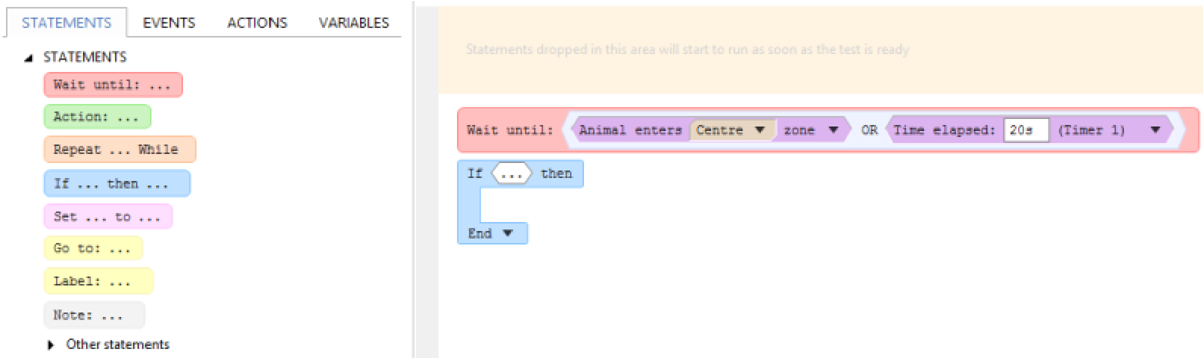


Figure 8. Dragging the If *statement into the procedure.*

You'll notice that the *If* statement has been expanded to become a 'container'. We can put statements *inside* this container, and then if the condition is true, the procedure will run the statements inside the container.

We now need to enter the condition that we're checking for; so we want to add the 'timer' event as a condition. There are a couple of ways we could do this:

1. Switch to the *Events* tab, then drag the *Time elapsed* event from here into the relevant 'hole' in the *If* statement.
2. Copy the *Time elapsed* event from its existing position on the right-hand side of the *Wait until* statement.

   To do this, hold down the 'Ctrl' key on the keyboard, and drag the event using the mouse from the right-hand side of the statement into the 'hole' in the *If* statement. (As you drag the event, you should see a little '+' next to the image of the event being dragged, to show that it's being copied, rather than moved. If you don't hold down the 'Ctrl' key, you'll *move* the event from one place to another).

Whichever of these methods you choose, you should end up with the procedure looking like this:

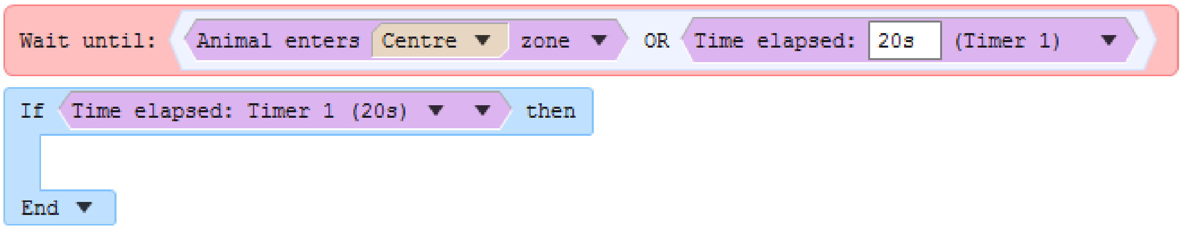


Figure 9. The Time elapsed event, dragged or copied into the If *statement.*

You'll notice that the timer event in the *If* statement refers to 'Timer 1'. This clarifies that we're looking at the same timer that the *Wait until* statement was waiting for. It's obvious in this case, as there's only one timer in the procedure, but in more complex procedures with multiple timers, you might need to refer to the timer ID to know which one is which. If necessary, you can use the drop-down arrow to the right of the timer to select between different timers that the procedure has previously waited for.

Note that for more advanced procedures, *If* statements can have more conditions - see the worked example using I/O for a simple example of this.

The last thing to add to the procedure is to end the test when the timer has elapsed. If you've gone through the first worked example, then you'll already have seen how a procedure can end a test, but we'll introduce something slightly more advanced here.

Ending a test is an Action, so switch to the *Actions* tab. The first action you'll see in the list is the *End the test* action, which we used in the first example. Just below that, however, is the action *End the test (specifying reason)*. Drag this action into the procedure, inside the *If* statement:

Figure 10. Dragging the action into the procedure.

You might remember from our first example that when you use the *End the test* action, the name of the procedure is used as the test end reason (and that this reason can be used in analysis). However, at the start of this example, we named this procedure 'Procedure example 2', and we don't really want that to be listed in our test results as the reason that the test ended! This new action that we've just dragged into the procedure gives us the opportunity to create our own test end reason (incidentally, one that you could reuse from different procedures in the same protocol if you wanted to).

This test end reason is a 'parameter' to the *End the test* action, and you'll see that it's currently set to '...' (which is what ANY-maze displays when it has nothing set for a specific parameter).

Just to the right of the empty parameter is an edit button ( ), which allows you to edit the parameter, and a drop-down arrow that allow you to select from any pre-existing test end reasons in the protocol (for example, any that you might have set up for other procedures).

Click the drop-down arrow, and a menu will drop which contains the option *Create new test end reason*. If you select this option, the Test end reason settings window will open; the default name will include our not-very-useful procedure name, so edit it to say something like 'Failed to enter Centre within 20s' and click *OK*. Remember that this reason will be used in test results, so it needs to be succinct but explanatory.

Your procedure is now complete, and should look like this:

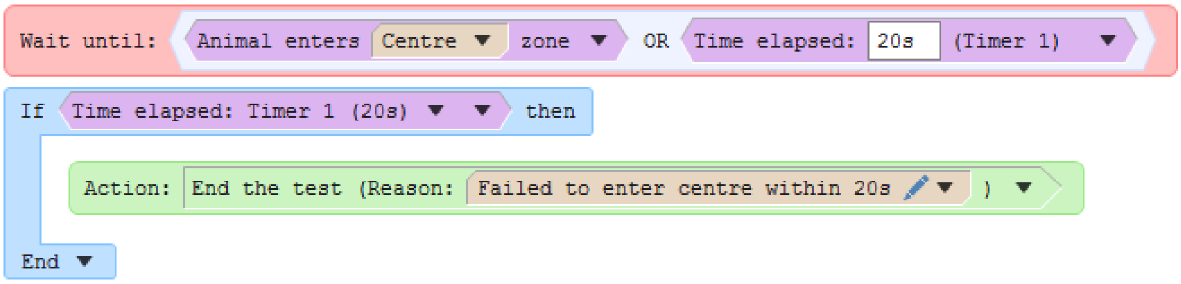


Figure 11. The completed procedure.

Summary

We've seen a few new concepts in this example:

- The ability to wait on multiple events
- The introduction of an *If* statement
- Specifying your own test end reasons.

Although this example only uses these in a very basic way, hopefully you can see that the options available could be quite flexible and can allow your procedures to perform some quite complicated test control.

What next?

If you use I/O with ANY-maze, I strongly suggest that you also look at the worked example using I/O. Even if you don't have any I/O connected, it's probably a good idea to work through this example anyway, as it introduces some more useful features of procedures.

Otherwise, I suggest that you have a play with the procedure editor, and try some things out for yourself to see what it can do! Take a look at the *Events* tab on the procedure editor, and expand all the headings to see what your procedure can respond to; on the *Actions* tab, expand the headings to see the things that your procedure can do to affect the test (this will be particularly useful if you have any I/O set up in your protocol).

If you require any assistance with writing procedures for your specific experiments, please contact ANY-maze technical support and we'll be happy to help.

A worked example using I/O

Introduction

This topic describes an example walkthrough to manipulate input/output devices (usually abbreviated to I/O) using procedures. If you haven't already done so, I highly recommend that you work through the simple example in the Writing a procedure using the procedure editor topic. This will give you some fundamental starting points. It would also be a good idea to work through the more complex example too.

The steps in this example will not be explained in quite as much detail, but we'll cover more aspects of what you can do with procedures.

It's highly unlikely that you will have the right I/O devices to actually make this example work, but this walkthrough is intended to give you some ideas of what's possible with procedures and I/O in ANY-maze. Once you've completed this example, I suggest you use the concepts learnt to try writing your own procedures, using the I/O that you do *have available.*

What do we want the procedure to do?

Let's assume that we have an experiment in which the animal will receive some sort of reward (a pellet). However, in order to receive the pellet, it must press a lever first. If the animal tries to get the reward *without* first pressing the lever, it will get a shock.

Set-up

Before we can write a procedure, we'll need to set up a protocol that has the right zones and I/O, so you'll need to do the following:

1. On the File page, select *New experiment* and then *New empty experiment*.
2. Give the protocol a name (e.g. 'Procedure example 3')

   Specify that the protocol mode should be 'Video tracking mode with input/output'.
3. Select the *Treatment groups* item in the protocol list and specify that the animals should be randomly assigned amongst the groups.
4. Click the *Add item* button and select *New video source*.

   Give the video source a name and then select the 'Open field example video' from the drop-down list.

If the 'Open field example video' does not appear in the list of example experiments, you can download it from the ANY-maze web site at www.anymaze.com/files/examples/open field example video.zip. *You'll need to unzip this file into the 'Examples' sub-folder of the ANY-maze program folder - by default, this is* `C:\Program Files (x86)\ANY-maze` *on a 64-bit operating system, or* `C:\Program Files\ANY-maze` *on a 32-bit operating system.*

5. Click the *Add item* button and select *New apparatus*.

   Give the apparatus a name and then set up the apparatus with a simple rectangular border, and a rectangular area in the top right corner:

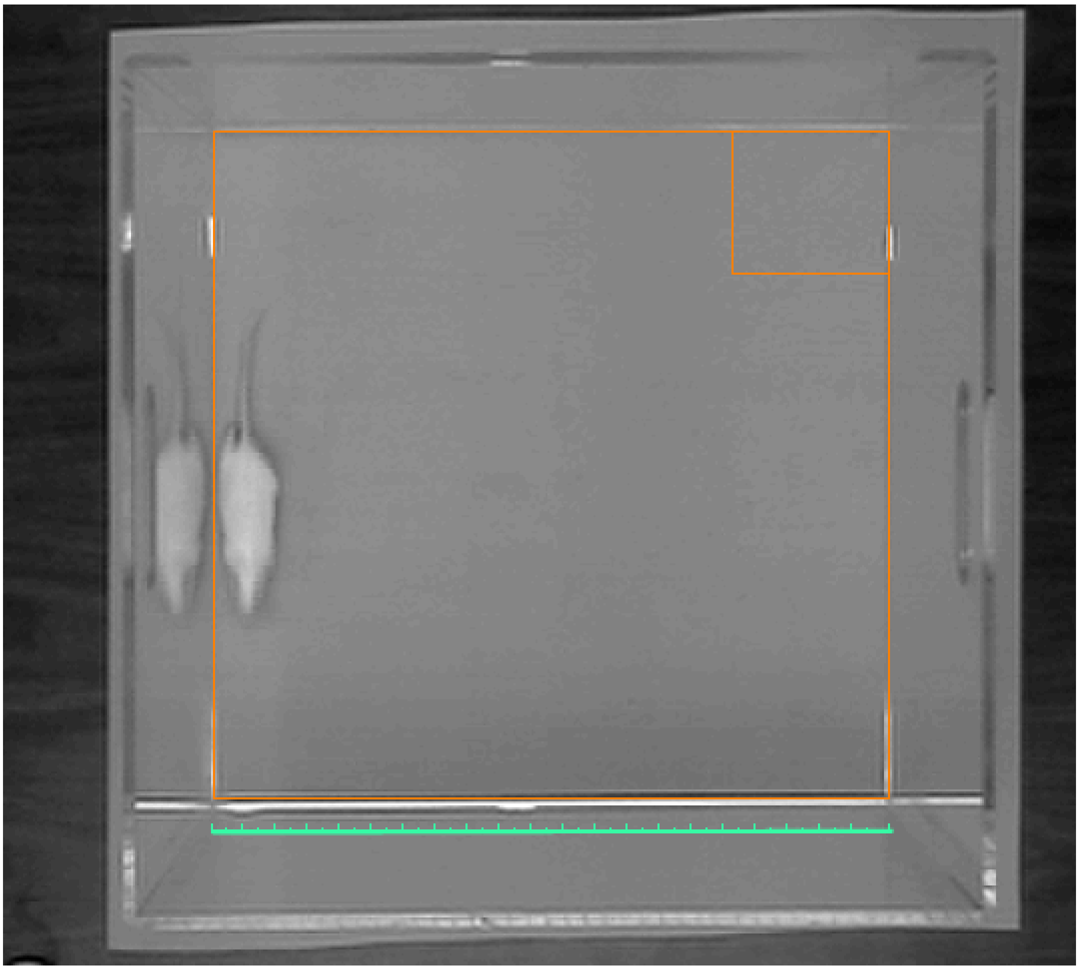


Figure 1. Set up the apparatus with an area in the top right corner.

6. Click the *Add item* button and select *New zone*.

   Specify a zone called 'Pellet dispenser' and select the top right-corner.

7. Click the *Add item* button and select *New I/O device*.

8. Click the *Add item* button and select *New output item* then *New output switch*. Give it a

name of 'Pellet dispenser'. (When you leave this item, select *Ignore* on the warning message about not having selected a port).

9. Click the ➕ *Add item* button and select *New output item* then *New shocker*. Give it a name of 'Shock floor' and an arbitrary duration and intensity.

   Again, when you leave this item, select *Ignore* on the warning message about not having selected a port.

10. Click the ➕ *Add item* button and select *New input item* then *New On/off input*. Give it a name of 'Lever'.

    Once again, when you leave this item, select *Ignore* on the warning message about not having selected a port.

11. Select the *Animal colour* item in the protocol list and specify that the animals are lighter than the background.

12. Under the *Stages* option in the protocol list, select *First stage* and enter a *Test duration* of 90s.

Now we're ready to create the procedure!

What does the procedure need to do?

So how do we go about creating this procedure? Well, the first thing we need to do is try and break the logic up into simpler parts.

To do this, we'll split the procedure into:

- Logic to do with the lever
- Logic to do with the pellet dispenser

Rather than try to incorporate all this in a single procedure, we'll actually create *two* procedures to implement this logic.

We'll create a 'variable' that allows us to keep note of whether the animal has pushed the lever, and share this variable between both the procedures. A variable is a kind of temporary storage for the procedure, which allows you to keep track of a specific value during the procedure. We can set our variable in one procedure, when the lever is pressed, then use this variable in the other procedure to determine whether to activate the pellet dispenser.

We *could* use a simple True/False value (i.e. 'Has the lever been pressed?'), but in order to demonstrate some other capabilities of a procedure, we'll actually count the number of times the lever is pressed and call the variable 'Lever press count'. This value will be incremented every time the animal presses the lever.

This gives us the following logic:

- **Procedure 1:**

  Wait until animal presses the lever

  Increment 'Lever press count' by 1

  We'll use a loop to ensure that this procedure runs repeatedly throughout the test.

- **Procedure 2:**

  Wait until animal enters the 'Pellet dispenser' zone

  If the lever press count is 1 or more

      Activate the pellet dispenser

  otherwise

      Activate the shock floor

  Reset the lever press count to zero

  Again, we'll use a loop to ensure that this procedure runs repeatedly throughout the test.

As you can see, breaking apart the logic for the test makes each of these procedures quite simple in what it needs to achieve.

Creating the first procedure

We'll start with the 'Lever press' procedure. Create a new procedure (using the *Add item* button, and selecting *New procedure*). Give it a name of 'Lever press'.

Make sure the procedure is in 'Full' mode, rather than 'Simple' mode, using the *Full view* button on the ribbon bar. (If the procedure editor is already in 'Full' mode, you'll only see the *Simple view* button).

The first thing we'll do is create the variable that will be used to keep track of lever presses. Use the *Create variable* button to create a variable, and edit its name to 'Lever press count'.

This variable will be shared between procedures, so make sure you check the box labelled *This variable is shared across all procedures run for a test.*

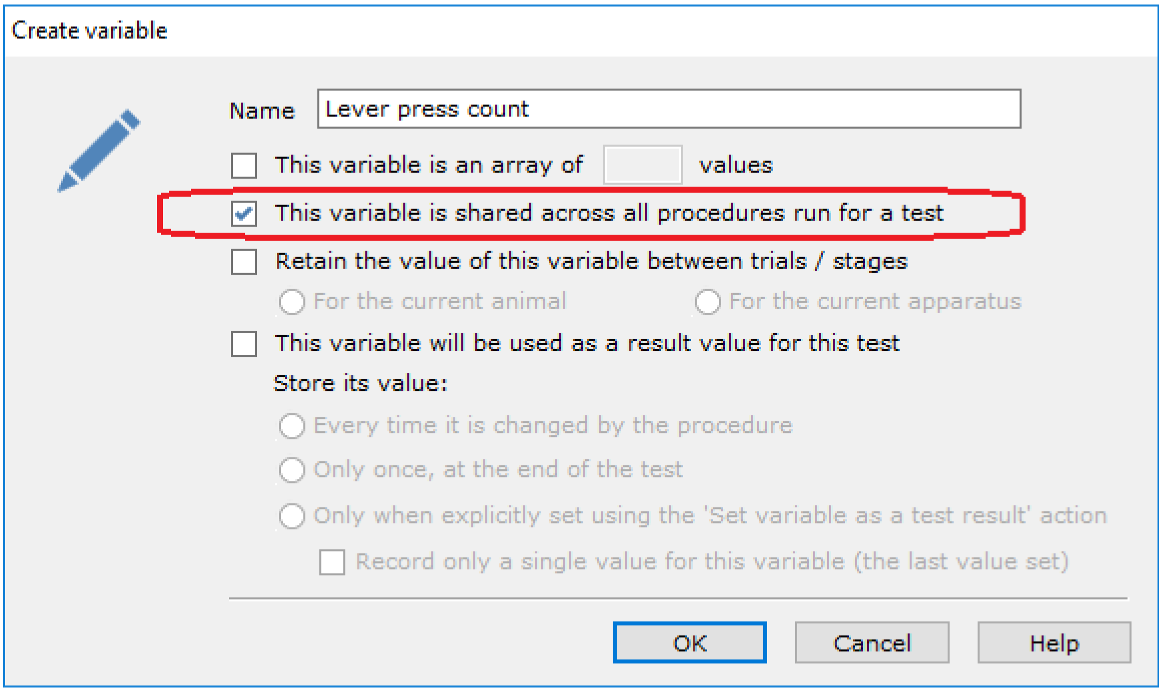


Figure 3. Setting up the 'Lever press count' variable. Note that this variable is shared across all procedures.

Once you've created the variable, you'll see it's been added to the list of variables on the *Variables* tab:

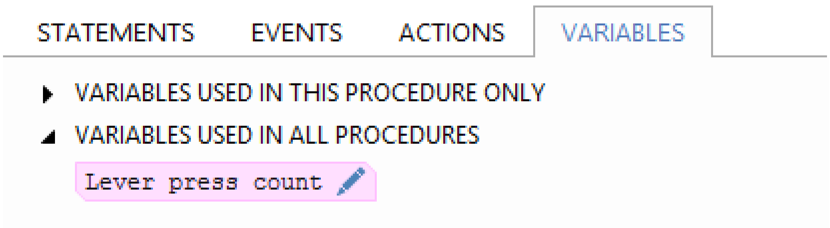


Figure 4. The Variables *tab, showing the new variable.*

Now we can start writing the procedure. Just to remind ourselves, this will wait until the animal presses the lever and then increment the 'Lever press count' variable. This will be done continually throughout the test, so we'll put it in a loop.

We'll start off with the loop - to do this, select the *Statements* tab, and drag the *Repeat* statement into the procedure:

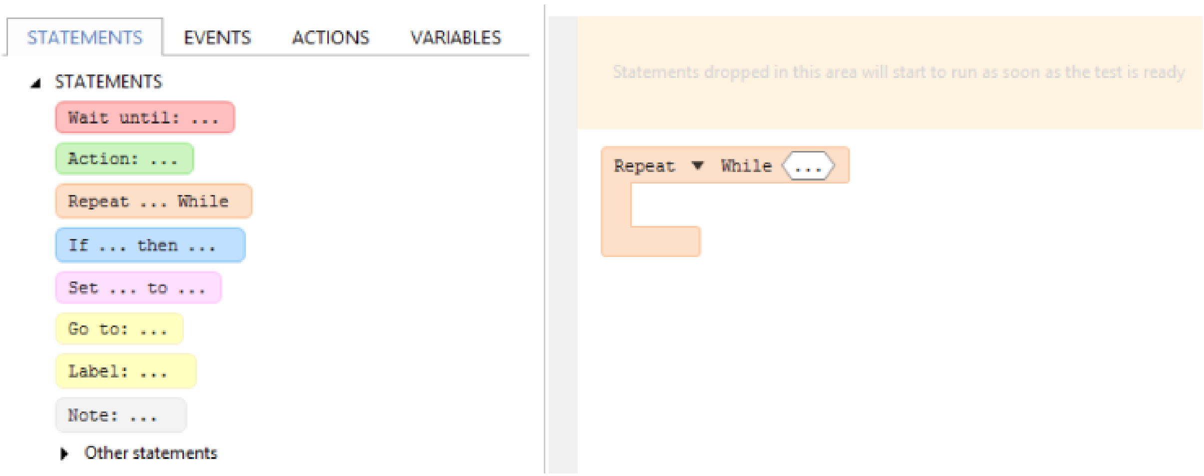


Figure 5. Dragging the Repeat *statement into the procedure.*

We *could* now drop the little arrow next to the 'While' on this statement, and select 'Forever'. This would run this loop for the entire duration of the test. However, we'll do something slightly different (but which will have exactly the same effect), using a built-in variable.

Select the *Variables* tab. You'll see a number of headings under this tab, one of which is expanded to show the variable we created earlier. Just to see the wealth of information on offer, right click anywhere in the pane under the tab, and select the *Expand all* option - this will open all the headings to show all the available built-in variables. All of these are available to allow your procedures to use information about the currently-running test. Take a moment to look through these variables, just to see the kind of things that ANY-maze can tell the procedure. Moving the cursor over any of the items will show a tool tip that gives some more explanation.

The variable we're interested in is under the *Test information* heading - it's called *Test is running*. Note the shape of this built-in variable - it has two angled ends, which means it's a True/False value, and can be used in the procedure anywhere where a True/False value is permitted - in this case, as a 'condition' of the *Repeat* statement.

This particular variable will be True once the test starts, and will continue to be True while the test is running. Once the test ends, this variable will be False.

To use this variable in the procedure, just drag it from the left-hand side into the 'hole' in the *Repeat* statement:

Figure 6. Dragging the Test is running *variable into the procedure.*

Now that we've got the loop in place, we can set its contents. Whatever sequence of statements we include within this loop will run continuously throughout the test.

First of all, we need to wait for an event to occur, so select the *Events* tab. If you have already read the Hints and tips for writing procedures topic, you may recall that we have a shortcut for adding a 'wait' for an event - dragging an *event* into a procedure will automatically create a *Wait until* statement for it.

The event we will wait for is the lever press, so make sure the *On/off inputs* heading is expanded, and select the *Lever* sub-heading to expand it. You should see events here for activation and deactivation of the lever; select the *Lever activated* event, and drag it into the middle of the *Repeat* statement:

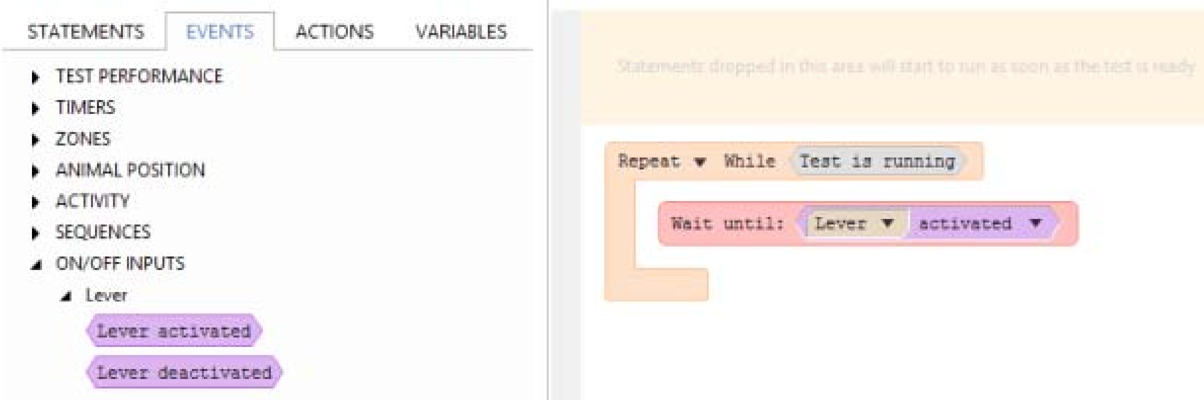


Figure 7. Dragging the 'Lever activated' event into the procedure. This automatically adds the Wait until *statement.*

Once the animal presses the lever, we need to increase the value of the 'Lever press count' variable. Select the *Variables* tab, and you should see the 'Lever press count' variable under the *Variables used in all procedures* heading. Drag the variable into the procedure, inside the *Repeat* statement and after the *Wait until* statement (here we're using another shortcut - dragging a user-defined variable into a procedure will automatically create a *Set value* statement for the variable):

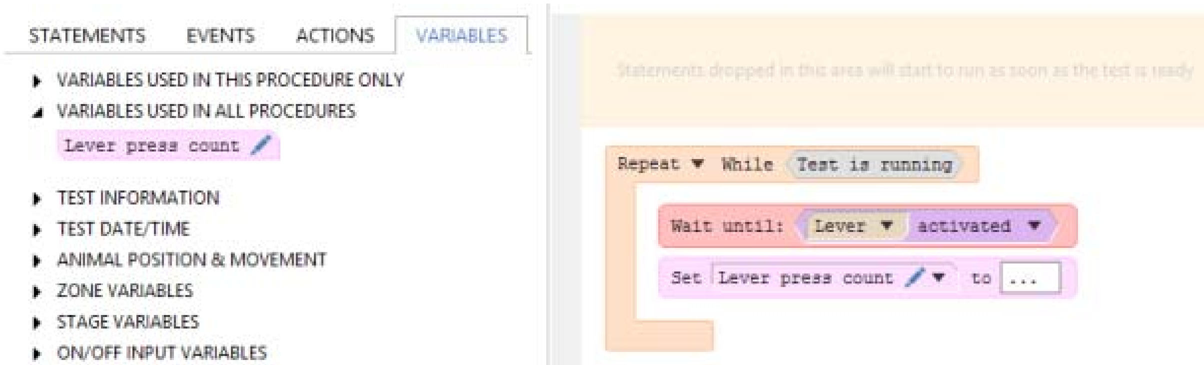


Figure 8. Dragging the 'Lever press count' variable into the procedure. This automatically adds the Set value *statement.*

Now we need to enter the second parameter (the right-hand side) of the *Set value* statement. We're incrementing the value of the 'Lever press count' variable, so what we're doing is the following:

Set 'Lever press count' to 'Lever press count' + 1

For this, we need the maths operator '+', which is on the *Statements* tab. Switch to this tab and expand the *Maths functions & operators* heading. You should see all the available maths operators, including '+' which is at the top of the list. Drag this into the procedure, over the 'hole' on the right-hand side of the *Set value* statement:

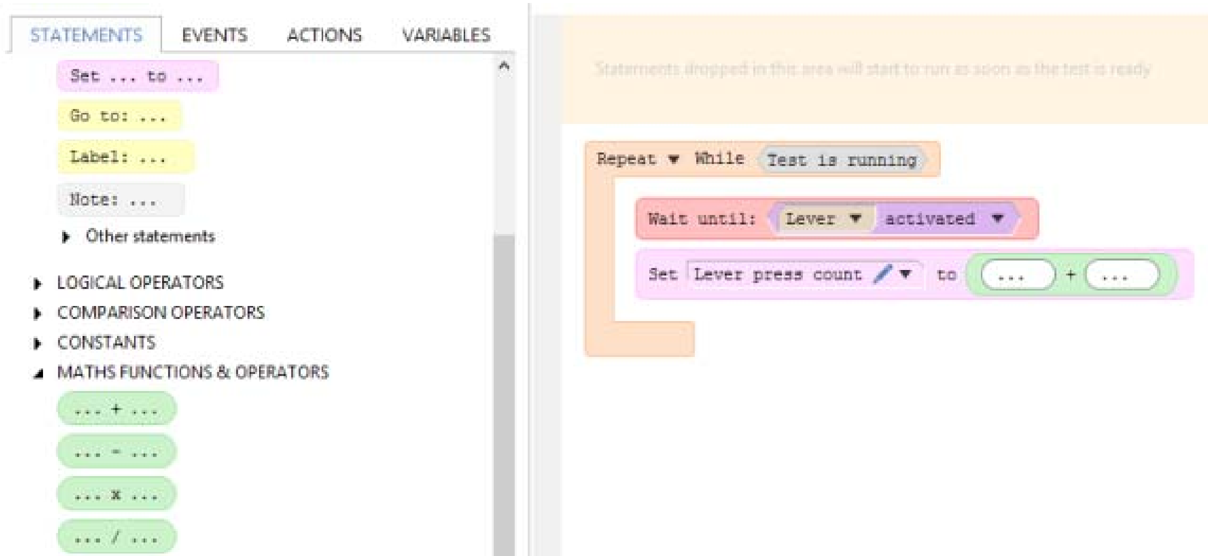


Figure 9. Dragging the '+' operator into the procedure.

Note here the shape of the operators, and also the shape of the 'holes' in the operator that we've just dragged. They're 'lozenge' shaped, i.e. with curved ends - in the ANY-maze procedure editor, this shape represents a numerical value. So you can see that the '+' operator accepts two *numerical* parameters, and will give a *numerical* value as its result.

Now we need to fill in the two sides of this operator. The right-hand side is just the value 1, so click on the right-hand 'hole' and type in the value 1:

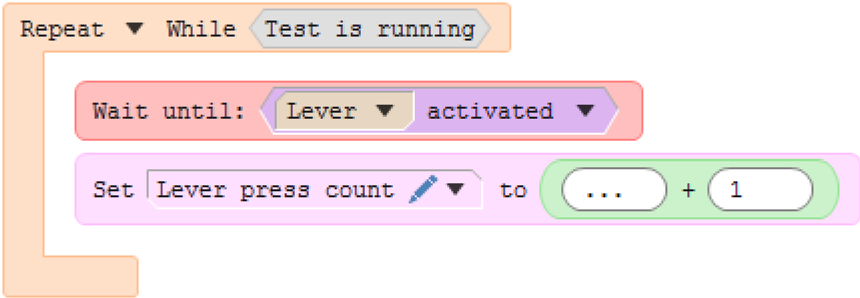


Figure 10. Entering the value 1 into the right-hand side of the addition operator.

For the left-hand side, we need to insert the 'Lever press count' variable. There are a couple of ways we could do this:

1. Switch to the *Statements* tab, then drag the variable from here into the relevant 'hole' in the '+' operator;
2. Copy the 'Lever press count' variable from its existing position on the left-hand side of the *Set value* statement.

   To do this, hold down the 'Ctrl' key on the keyboard, and drag the variable using the mouse from the left-hand side of the statement into the 'hole' on the right. (As you drag the variable, you should see a little '+' next to the image of the variable being dragged, to show that it's being copied, rather than moved. If you don't hold down the 'Ctrl' key, you'll *move* the variable from one place to another).

Whichever of these methods you choose, you should end up with the following:

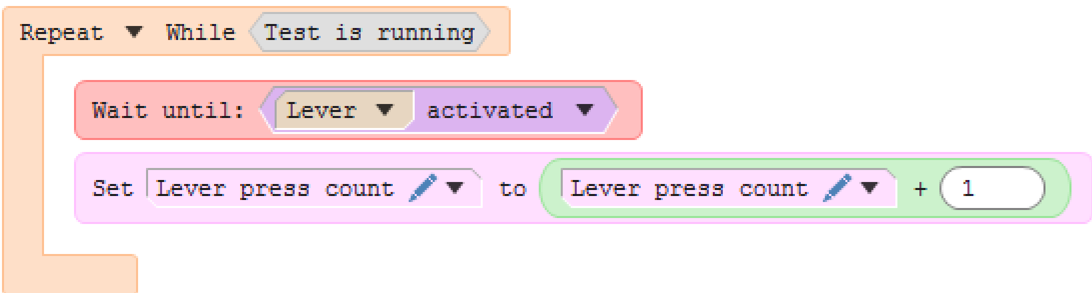


Figure 11. The complete 'Lever press' procedure.

One point to note here is that we haven't done anything about initialising the value of the 'Lever press count' variable at the start of the procedure. That's OK, because all variables are initialised to zero at the start of a procedure (unless you've changed their settings to specify that they should retain their value between trials for an animal).

Before we continue, just click the *Check for errors* button to make sure that the procedure doesn't have any errors.

So now the first procedure is set up, let's have a go at creating the second. As a reminder, this procedure will wait for the animal to enter the 'Pellet dispenser' zone; if it's already pressed the lever, then it will get a pellet; if not, then it will get a shock via the shock floor.

Creating the second procedure

Create a new procedure (using the *Add item* button, and selecting *New procedure*). Give it a name of 'Pellet dispenser'.

Once again, our procedure will continue to run throughout the test, so we'll again use a loop - switch to the *Statements* tab and drag the *Repeat* statement into the procedure. This time, just drop the list next to 'While' and select 'Forever':

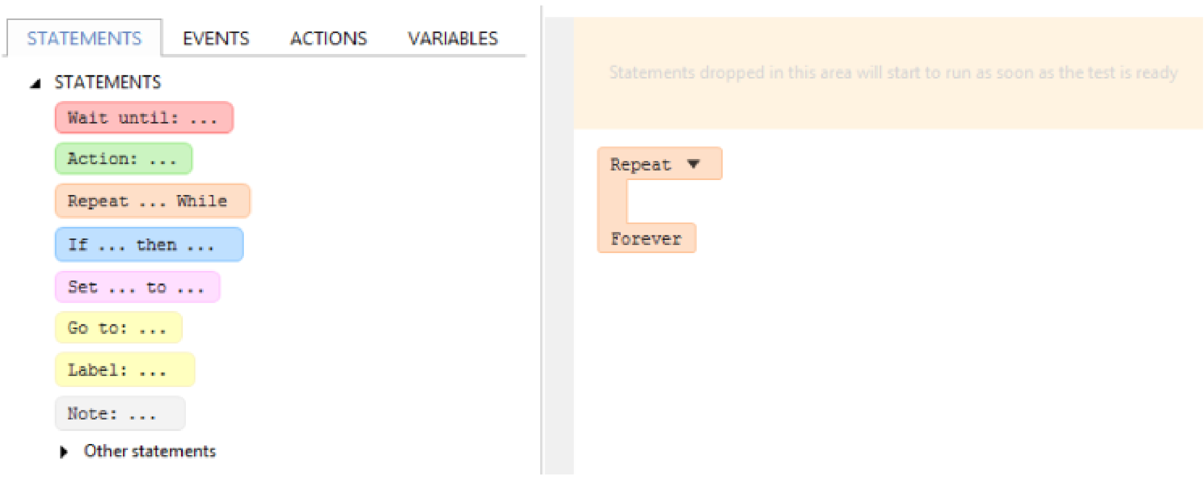


Figure 12. The Repeat *loop for the second procedure.*

Now we'll add the event that we're waiting for (i.e. the animal's entry to the pellet dispenser zone). Select the *Events* tab, find the 'Pellet dispenser' zone in the list of zones, and expand it to see the events for zone entry and exit. Drag the *Animal enters pellet dispenser zone* event into the procedure, inside the *Repeat statement:*

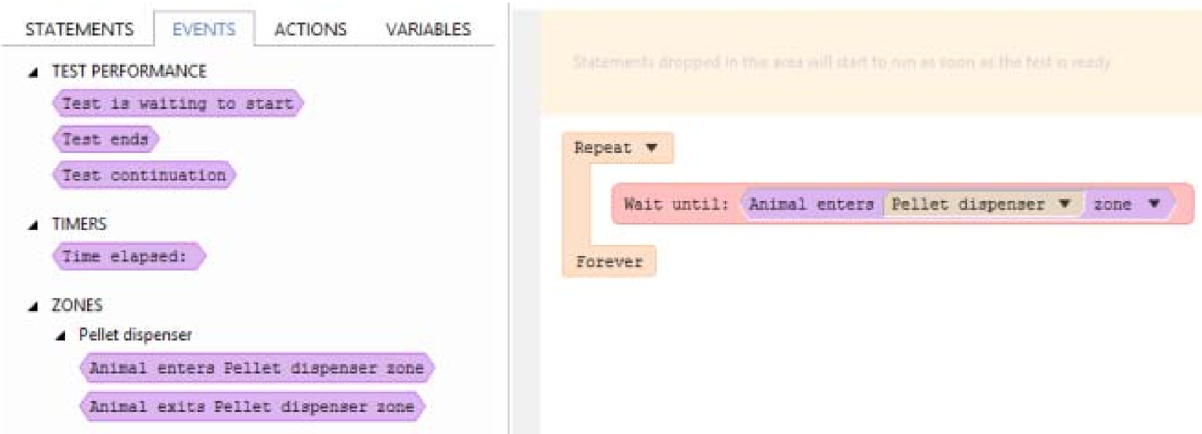


Figure 13. Dragging the Wait until *statement into the procedure.*

Once the animal enters the zone, we need to check whether it has previously activated the lever. This will use the value of the 'Lever press count' variable, which is being incremented by our first procedure ; if the lever press count is greater than zero, then we'll activate the pellet dispenser; if not we'll activate the shock floor.

How do we determine whether the value of the 'Lever press count' variable is greater than zero? We use a Comparison operator to compare the variable's value to 0. Switch to the *Statements* tab, and expand the *Comparison operators* heading. You'll see a number of comparison operators, allowing you to compare whether values are equal, not equal, greater than, greater than or equal, less than, less than or equal. In fact, we could use quite a few of these:

- Not equals (*not* =), to check whether the value is not equal to zero
- Greater (>), to check whether the value is greater than zero
- Greater than or equals(>=), to check whether the value is greater than or equal to 1

We'll just use the 'greater than' operator (the last-but-one in the list), so select that and drag it in to the procedure, just under the *Wait until* statement (again, we're using one of the shortcuts described in the Hints and tips for writing procedures topic to automatically create the *If* statement for the operator).

You should end up with the following:

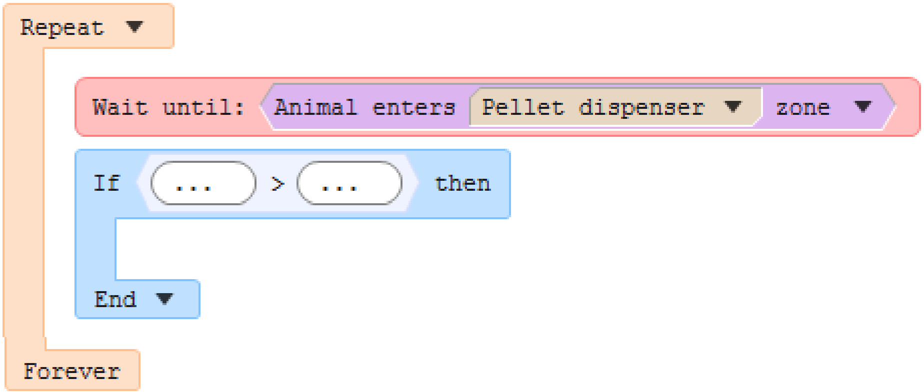


Figure 14. Dragging the 'greater than' operator into the procedure, which automatically creates the If *statement.*

We're comparing the value of the 'Lever press count' variable, so select the *Variables* tab, and drag the 'Lever press count' into the left-hand side of the 'greater than' operator. Click on the right-hand side and type in '0' to give the following:

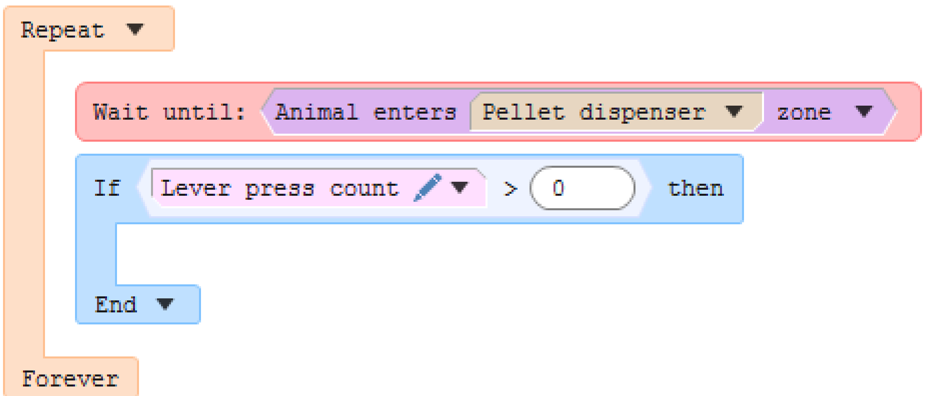


Figure 15. Entering the parameters to the 'greater than' operator.

What happens if the lever press count is greater than zero? We activate the pellet dispenser. This is done using an Action, so switch to the *Actions* tab. By default, only the test control actions are expanded, but you can right click and select the *Expand all* option to see all the actions available. The pellet dispenser is a switch, so look for it under the *Output switches* heading; there are a number of actions available for this switch. Drag the *Activate the switch* action into the *If* statement in the procedure:

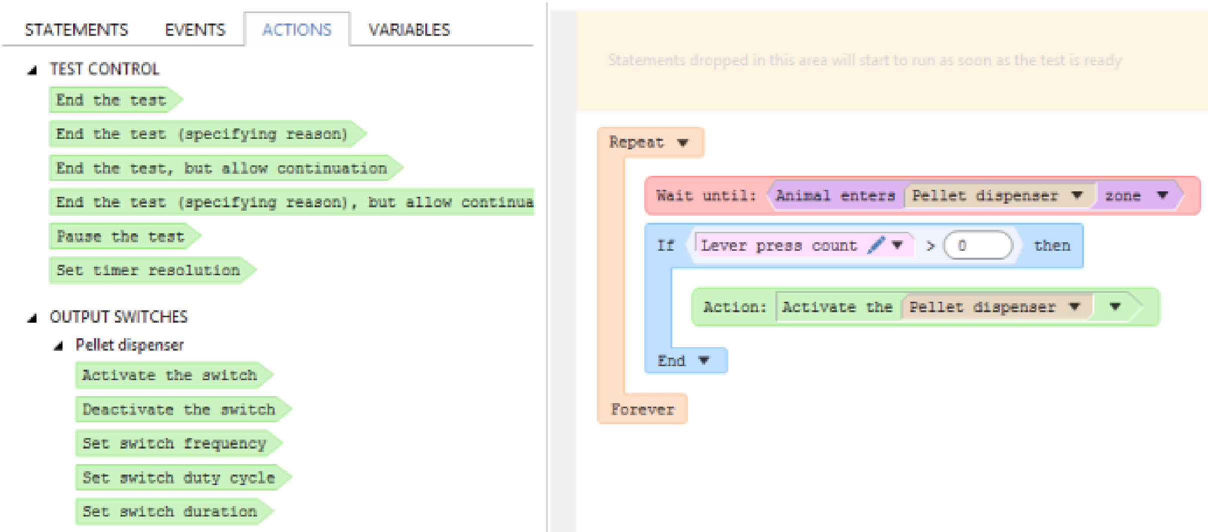


Figure 16. Dragging the Activate the switch action into the If *statement.*

If the animal *hasn't* pressed the lever, then the value of the 'Lever press count' variable will be 0. So what will happen then? As the procedure stands, the *If* statement will just drop through without taking any action, and will go back into the *Repeat* statement - i.e. it will wait for the animal to enter the pellet dispenser zone again.

Since that's not what we want (we want to activate the shock floor, if the animal hasn't pressed the lever), we need to add something to the *If* statement. You can do this using the little drop-arrow next to the 'End' part of the 'If' statement. If you click on this, you'll find that as well as the 'End' option, you also have the following choices:

- *Else If* - You can add as many 'Else If' conditions to the *If* statement as you like - each one will be evaluated in turn until one evaluates as true. If a condition is true, then the statements within this section of the *If* statement will run. If none are true, then the *If* statement will drop out of the bottom and do nothing (unless there's an 'Else' defined - see below).
- *Else* - This allows you to define a sequence of statements to be run if none of the 'If' or 'Else if' conditions are true. If none of them are true, then the statements within this 'Else' section will run.

Clearly, what we need to do is define an 'Else' section to this statement - if the 'Lever press count > 0' condition is not true, we need to activate the shock floor.

Select 'Else' from the drop-down list, which will create a second section to the *If* statement. On the *Actions* tab, find the *Shockers* heading and within it, the 'Shock floor' shocker. The first action in this list is to activate this shocker, so drag this action into the second half of the *If* statement as follows:

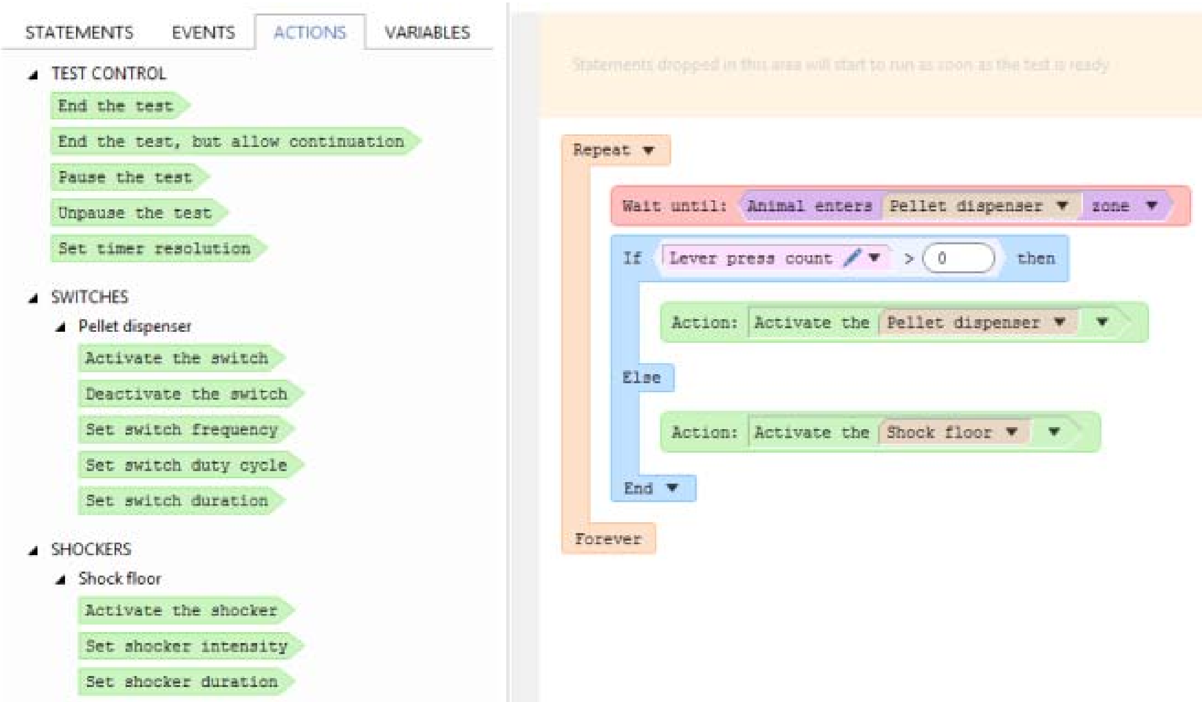


Figure 17. Dragging the 'Activate the shocker' action into the If *statement.*

We're nearly there, but there's just one thing we still need to do. As it stands, this procedure only needs the animal to activate the lever *once*, and then every time they enter the 'Pellet dispenser' zone they will get another pellet (even if they haven't activated the lever again). This is because the 'Lever press count' variable is only ever incremented, and never reset. To ensure that the animal must press the lever *every* time they want a pellet, we must reset the 'Lever press count' variable after the successful activation of a pellet.

The simplest way to achieve this is to select the *Variables* tab, then drag the 'Lever press count' variable below the 'If' statement, as follows:

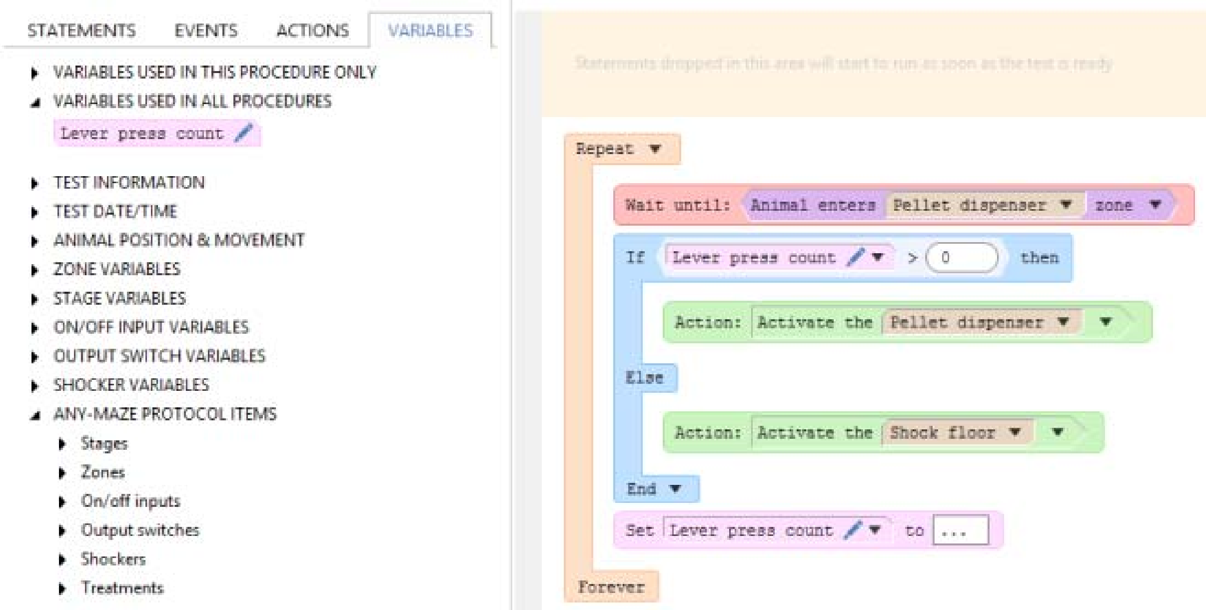


Figure 18. Dragging the 'Lever press count' into the procedure, to reset it.

Of course, we need to provide the value to reset to, so click on the right-hand parameter of this *Set value* statement, and enter '0'. This leaves us with the following completed procedure:

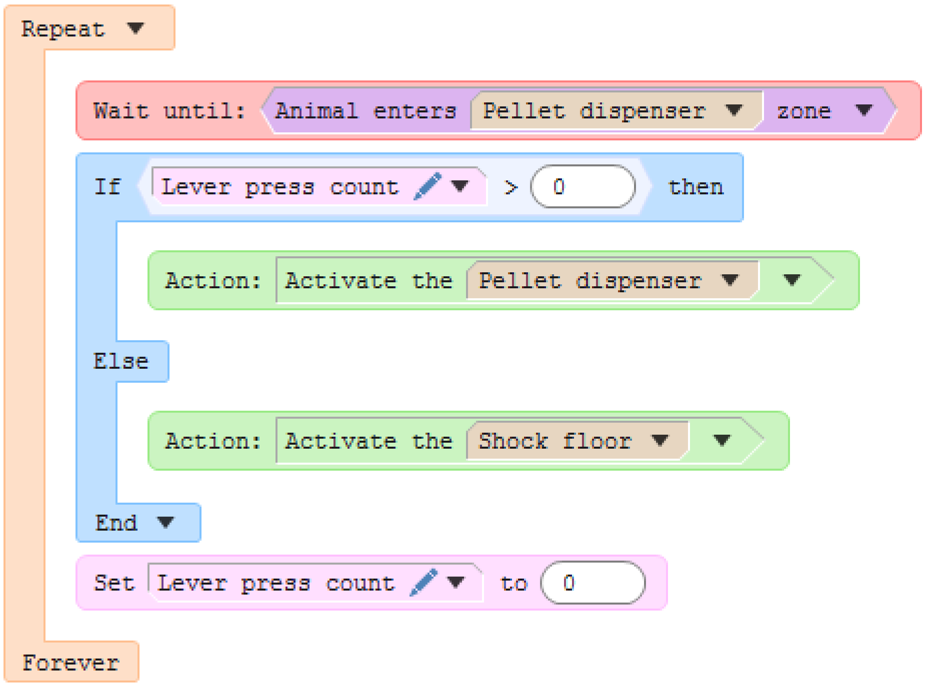


Figure 19. The complete 'Pellet dispenser' procedure.

The resetting of the variable could also have been inserted immediately after the activation of the pellet dispenser; this would have exactly the same effect.

To finish off, click the *Check for errors* button to make sure that the procedure doesn't contain any errors.

Summary

This example has taken you through some of the more advanced features of procedures:

- Using multiple procedures
- User-defined variables
- *If* statements
- Conditional operators

There's much more available, but this should give you an idea of the flexibility and power that procedures can offer.

What next?

Looking at all the items under the tabs on the procedure editor window will show you all the options that are available, and you can find out more on all the building blocks available to a procedure in the Elements of a procedure topic.

Now that you've seen some of the ways that a procedure can use I/O, I suggest that you have a play around and see if you can get procedures to manipulate any I/O devices that you have. Use the details above, along with the tool tips for all the procedure items, to help you.

If you require any assistance with writing procedures for your specific experiments, please contact ANY-maze technical support and we'll be happy to help.

ANY-maze help topic H0403

Errors and warnings while writing a procedure

Introduction

Before you can run a test, any procedures that have been defined must be valid. If there are any errors in a procedure, then the test will not run (you'll be warned of this on the Tests page and given the opportunity to edit the procedures at this point).

If you want to run a test without a procedure - say you're in the middle of writing a procedure, and you know that it's not yet working properly - you can disable the procedure *to prevent it being used.*

The ANY-maze procedure editor allows you to run a check on a procedure at any time, to see if it is valid - just select the *Check for errors* button.

Procedures will still run if they contain any warnings; it is only errors that will stop a test from starting.

Checking for errors

You can check a procedure you're currently editing for errors using the *Check for errors* button. This will check through the procedure to see if it's able to run during a test.

Errors and warnings are shown in an information window at the bottom of the procedure editor. If there are no errors or warnings, then the window will remain closed and its title will be blue, showing that there are no problems:

Figure 1. Procedure information, if there are no errors or warnings.

If there are any errors or warnings, then the window will open, and its title will change colour to red (if there are any errors) or yellow (if there are only warnings). All errors and warnings will be listed within the window, and a small error or warning image will be shown in the left-hand margin of the procedure itself, at the position where the problem has occurred. These images will be light in colour, but clicking on one of the lines in the information window at the bottom will highlight that specific error or warning in the procedure, and darken the image to make it clear where in the procedure the problem is.

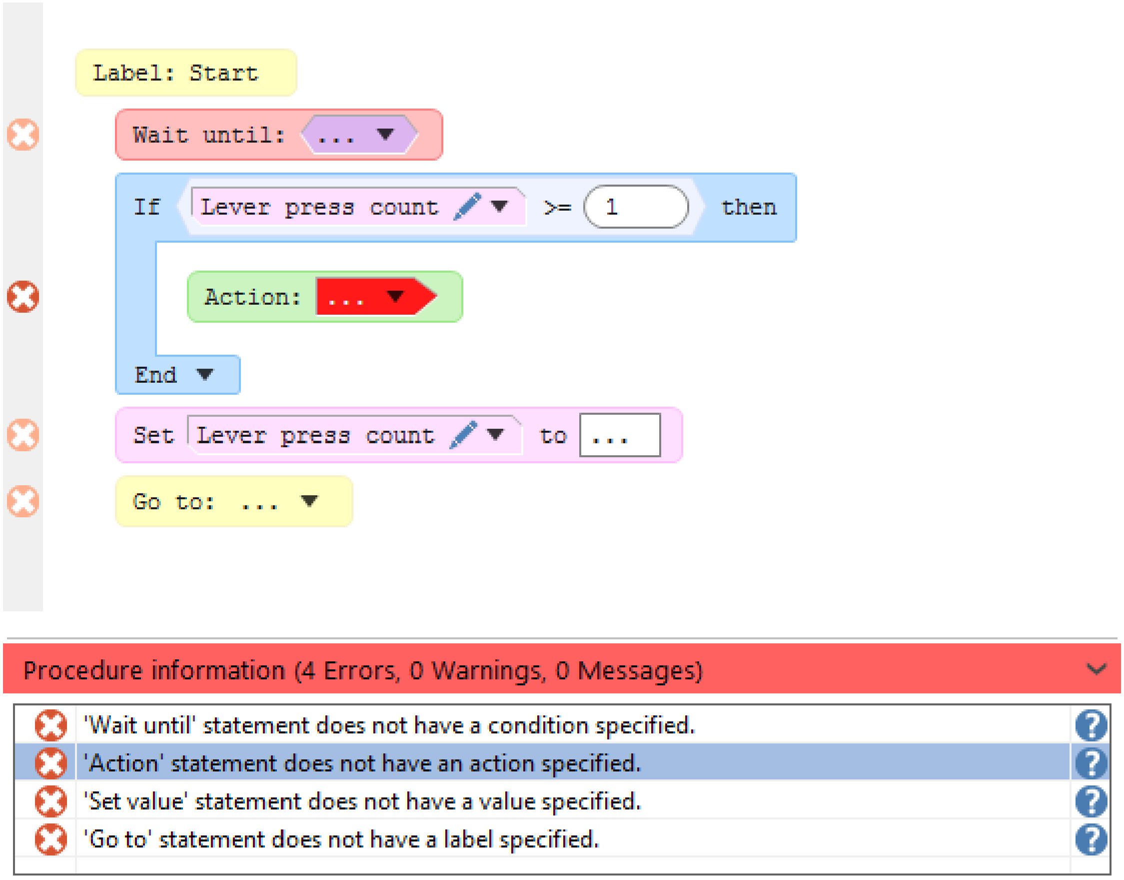


Figure 2. A procedure with errors. One of the errors has been selected in the information window, and is highlighted in the procedure.

The information window can be opened or closed using the little arrow at the right-hand side of its title.

More information

The most common errors when writing a procedure are those that result in a parameter to a statement being missing - for example, waiting for an event but forgetting to specify which event to wait for (like the first error in the example in figure 2, above). However ,there are a number of possible errors that may occur when writing a procedure, and they are listed below.

Errors and warnings

There are a number of errors and warnings that can be generated when you check the procedure for errors. If you need further information, click the ? link for the relevant line in the list of errors and warnings at the bottom of the procedure editor to go to an explanatory help topic.

The following errors and warnings may appear in this list:

Errors

- Error: Both sides of an 'AND' operator cannot contain events.
- Error: Cannot use an event in an 'if' statement without first waiting for that event to occur.
- Error: The left-hand side / right-hand side of the operator must be specified.
- Error: Both sides of the operator cannot be 'time elapsed' events.
- Error: The event must have a parameter specified.
- Error: The function must have both parameters specified.
- Error: Statement does not have a parameter specified.
- Error: The expression must have a parameter specified.
- Error: The action/event must be specified.
- Error: Labels must be unique.
- Error: The variable passed to the 'Set variable as a test result' action must be a single value variable, set up to be a result for tests.
- Error: Cannot use a 'Wait' statement before the test has started.
- Error: Cannot wait for the 'Test is waiting to start' event after the test has started.
- Error: Cannot use a variable that relies on tracking information before the test starts.
- Error: The procedure tries to set the location of a movable zone after the test has started.
- Error: The procedure tries to prevent/allow test start after the test has started.
- Error: To set the position of a movable zone from a procedure, you must set up the zone's location for this stage to 'The location will be set by a procedure'.
- Error: Cannot end the test before it has started.
- Error: Cannot use treatments within this procedure: the protocol has been set up not to use treatments.
- Error: Cannot call a sub-procedure recursively.

Warnings

- Warning: This procedure has no executable statements.
- Warning: This procedure does not include a 'Wait until' statement.
- Warning: The Repeat/If statement does not have a clause specified.
- Warning: The procedure uses the 'Test is running' variable before the start of the test.

- Warning: The variable passed to the 'Randomise array' function is not an array variable.
- Warning: The variable passed to the 'Initialise array' function must be an array variable.
- Warning: The variable passed to the 'Copy array' function must be an array variable.
- Warning: This event will never occur.
- Warning: This variable is not in use for the current protocol settings.

See also:

- Errors and warnings while running a procedure

Error: Both sides of an 'AND' operator cannot contain events

Description

As a test is performed, ANY-maze tracks the animal and also monitors the I/O ports set up in the protocol. Whenever anything changes (for example the animal's position, or the state of an I/O port), this is stored in the test's results and, internally, ANY-maze generates a 'test event'. These 'test events' are loosely (but not exactly) what you'll see on the Test data report.

It's important to realise that these events occur *one at a time*; each one is passed in turn to a procedure to deal with. If a procedure is currently waiting for that event to happen, using a Wait until statement, then it will use this opportunity to move forward in the procedure and run subsequent statements (until the procedure ends or it reaches another *Wait until* statement). Otherwise, the procedure will go no further and will let other procedures process that same 'test event'.

Because these events are processed one after the other, it's only possible for a procedure to receive one event at a time. It therefore follows that it can only *wait* for one event at a time. The procedure can wait for more than one event, as long as it is waiting for *one of* those events to happen, and not *all of* them to happen. For this reason, you can't have events on both sides of an *AND* logical operator.

If a procedure attempts to use two events within the *AND* logical operator, this error will be displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure is checked for errors or you exit the procedure editor.

Examples

The following procedure will generate this error, because it is waiting for two events to happen at the same time.

Figure 1. This procedure will cause this error, because it uses events on both sides of the AND *operator.*

If you want your procedure to wait until the point that the animal has entered two different zones, you can often use built-in variables to get the information you require. The following example shows how the procedure in figure 1 can be re-written to achieve this:

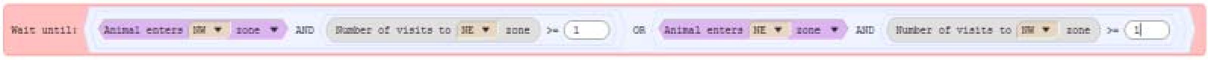


Figure 2. This procedure uses a built-in variable to check whether the animal has previously *entered the NE zone, and the event for entry to the NW zone, to wait for the animal to have entered both zones.*

Note that it's perfectly OK to use events on either side of the *OR* logical operator, as in figure 3, because you're waiting for *one of* the events to happen, and not both.

Figure 3. It's OK to use events on either side of the OR *logical operator.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement
- Logical operators

Error: Cannot use an event in an 'If' statement without first waiting for that event to occur

Description

If you want a procedure to wait for an event to happen, this *must* be done using a Wait until statement, and not an *If* statement.

More information

It's a common mistake to try and write a procedure like the following:

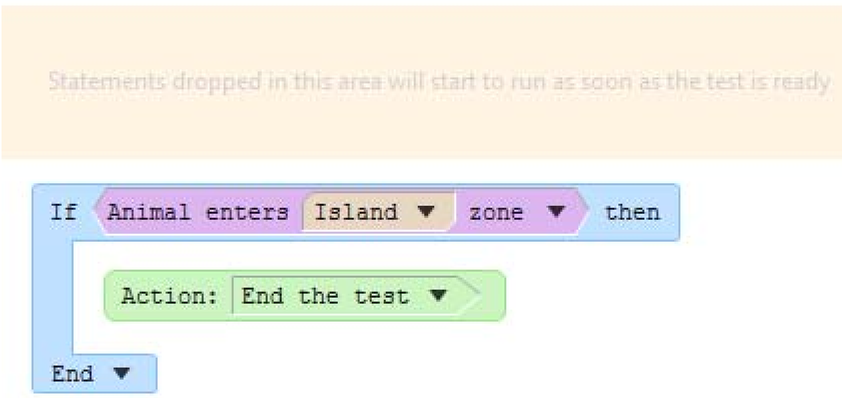


*Figure 1. This procedure is **not** valid, because it uses the event in an* If *statement instead of a* Wait until *statement.*

To see why the above procedure wouldn't work, let's consider how a procedure actually runs. Procedures run through their statements in order, starting when the test starts, and only stopping at the end of the procedure or when it reaches a Wait until statement which means it's got to hang around and wait for something specific to happen. This means that each statement is only processed at a single point in time, then the procedure moves on.

Looking at the procedure in figure 1, you can see that as soon as the test starts, the procedure gets to the If statement. It runs this *If* statement by evaluating its condition to see if it is true. The *Animal enters Island zone* event isn't occurring at this exact time (even if the animal starts the test *already in* the island zone, the procedure is currently processing the 'test start', and there hasn't yet been an event to say that the animal has entered the zone), so the condition is not true. The procedure therefore doesn't execute the statements within that condition, and drops out of the *If* statement. There are no more statements in the procedure, so the procedure finishes. Obviously, now that the procedure has finished, it can no longer detect a zone entry to the island zone when it happens.

The solution to this is to rewrite the procedure using a Wait until statement:

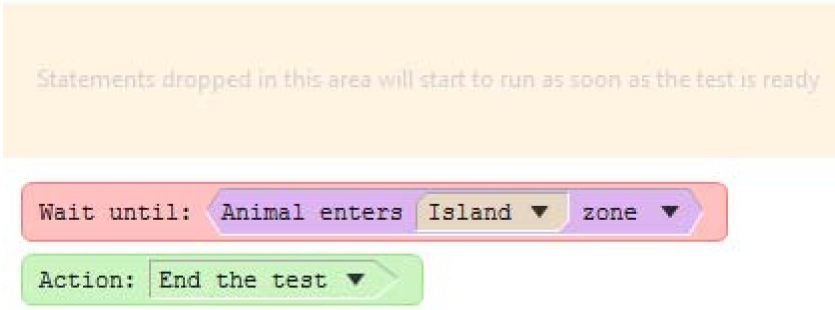


Figure 2. This procedure will wait until the zone entry event happens, before ending the test.

Again, if we consider how this procedure actually runs, we can see how this will function correctly where the procedure in figure 1 would not. At the start of the test, the procedure gets to the *Wait until* statement. The condition for the *Wait until* statement is not true (as we ascertained above), so the procedure 'pauses' and lets other procedures have a chance to run.

As the test progresses, the tracking continually reports animal movements, and whenever this happens, a 'test event' is generated for the new position. Each time this happens, the procedure gets the chance to run again. Each time it starts at its current statement - the *Wait until* statement - and evaluates it to see if the event is what the procedure is waiting for. A simple animal movement is *not* what the procedure is waiting for, so again it pauses at the same point, and lets other procedures have a turn. This happens repeatedly throughout the test, whenever something new happens - the animal moves, freezes, the state of an I/O input changes, etc.

At some point, the animal's movement may mean that it's now in the Island zone. When this happens, ANY-maze generates a 'zone entry' event which is passed to all procedures. Once again, this procedure is given its chance to run; it looks at the condition of its current statement, the *Wait until* statement, to see if the event is what it's waiting for. It **is** the right event now, so the condition is true; the procedure moves on and executes the next statement, which is to end the test. The ending of the test also ends the procedure.

If a procedure is written which uses an event in an *If* statement without first using it in a *Wait until* statement, this error will be generated in the list of errors and warnings at the bottom of the procedure editor when you check the procedure for errors or exit the procedure editor.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement

Error: The left-hand side / right-hand side of the operator must be specified

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when using any mathematical, logical or comparison operator that accepts two parameters (i.e. values that the operator will act upon). This applies to:

- Maths operators such as +, -, *x*, /, *Mod* etc.
- Comparison operators such as >, <, = and *not* =
- Logical operators *AND* and *OR*

More information

Everything in an ANY-maze procedure that accepts a parameter (i.e. the thing or things that it operates on) must have all parameters specified. You can't leave any empty 'holes' in any statements of a procedure.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Maths functions and operators
- Comparison operators
- Logical operators

Error: Both sides of the operator cannot be 'Time elapsed' events

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when you are using any Logical operator ('AND' or 'OR'), and you've used a 'Time elapsed' event on both sides of the operator.

More information

When a procedure waits for a 'Time elapsed' event, it starts waiting at the time that it first encounters the Wait until statement (*not* the time elapsed from the start of the procedure). So if the same wait statement is waiting for two *different* times, it will start waiting for them both at the same time. Obviously, this means that the shorter of the two times will always elapse first. ANY-maze prevents you getting into this situation by flagging this situation as an error.

The following procedure snippet will cause this error:

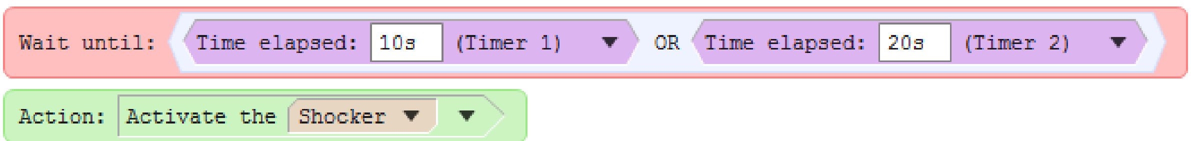


Figure 1. When waiting for two 'Time elapsed' events, one of them will always occur first. For this reason, the statement above does not make sense, and so ANY-maze will flag this as an error.

If you need to perform some action based on a time having elapsed *since the start of the test*, you have a number of options:

- Wait for the test clock to reach a specific time (note that you'll need to wait for the test clock to be **greater than or equal to** this time).
- Create your own user-defined event to occur at a certain time after the test starts, and wait for this event.

- Use Wait until statements with 'Time elapsed' events, but make the times *cumulative* so that each one adds to the total time since the start of the test.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Waiting for times to elapse
- Logical operators

Error: The event must have a parameter specified

Description

There are several events in ANY-maze that occur for a specific protocol item. For example, the *Animal enters zone* event must specify the zone, the *Sequence completed* event must specify the sequence, and the *Rotary encoder clockwise rotation* event must specify the rotary encoder.

If you've used an event such as these in your procedure, but have not specified the protocol item, then this error will be displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure is checked for errors, or when the procedure error is edited.

The missing parameter will show '...' within the procedure editor; to fix the error, simply click on the drop-down arrow to the right of the missing item and you'll be given a list of possible values to select from.

More information

Everything in an ANY-maze procedure that accepts a parameter (i.e. the thing or things that it operates on) must have all parameters specified. You can't leave any empty 'holes' in any statements of a procedure.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Events available to procedures

Error: The function must have both parameters specified

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure uses a function that needs two parameters - i.e. values that the function acts on - but one or both of these parameters has not been specified.

This applies to the following functions:

- Join function, to join two strings
- Pow function, to return a number to the power of another number
- Random function, to generate a random number between two values
- Round function, to round a number to a given number of decimal places

The missing parameter will show '...' in the procedure editor; you need to type in a relevant value or drag an expression into the place of the missing item.

More information

Everything in an ANY-maze procedure that accepts a parameter (i.e. the thing or things that it operates on) must have all parameters specified. You can't leave any empty 'holes' in any statements of a procedure.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Maths functions and operators
- Other functions

Error: Statement does not have a parameter specified

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a statement doesn't have some required information specified. This can apply to:

- A Wait until statement that does not have an event specified
- An Action statement that does not have an action specified
- A Repeat statement that does not have a condition specified
- An If statement that does not have a condition specified
- A Set value statement that does not have a variable or value specified
- A Go to statement that does not have a label specified

If not specified, the required information for the statement will be displayed as '...'. You can usually click the drop-down arrow to the right of the missing information, and select from the resulting list.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Statements available to procedures

Error: The expression must have a parameter specified

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a built-in variable or function is used in a procedure that does not have its parameter specified - i.e. the item that it is to be applied to.

For example, the *Number of visits to a zone* built-in variable needs to specify which zone it is - this zone is a 'parameter' to the built-in variable. Any built-in variable or function that requires a parameter must have that parameter specified.

Example

The following procedure uses two built-in variables; one of them has a missing parameter (i.e. the zone has not been specified), which will result in this error.

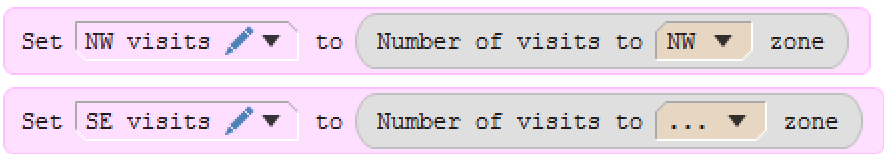


Figure 1. The first use of this built-in variable is set correctly; the second has a missing parameter (which zone it is) and will result in this error.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure

ANY-maze help topic H0449

Error: The action/event must be specified

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when an action or event does not contain all the required information.

More information

Some actions in ANY-maze require extra information to be specified; for example, an action to activate an I/O output switch must specify which switch it is. Likewise, when waiting for an event, there are events that require some secondary information; for example the *Animal enters zone* event must specify which zone it is.

If an action of event does not specify this extra information, this error will be generated.

Example

In the following section of a procedure, the first *Wait until* statement is missing a zone, and the *Action* statement is missing the switch that it is to activate.

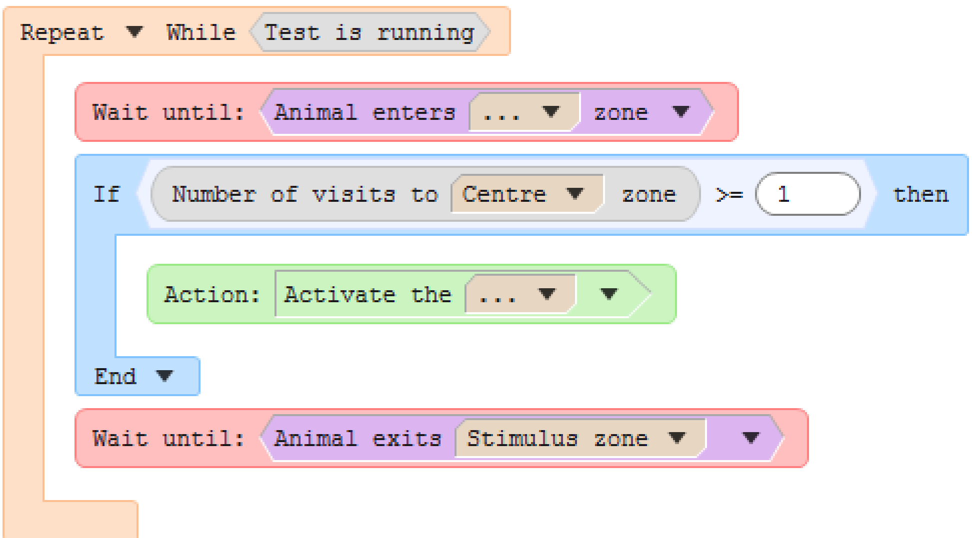


Figure 1. A section of a procedure with missing parameters to an event and an action.

The list of errors and warnings at the bottom of the procedure editor will list more specific details of what is missing:

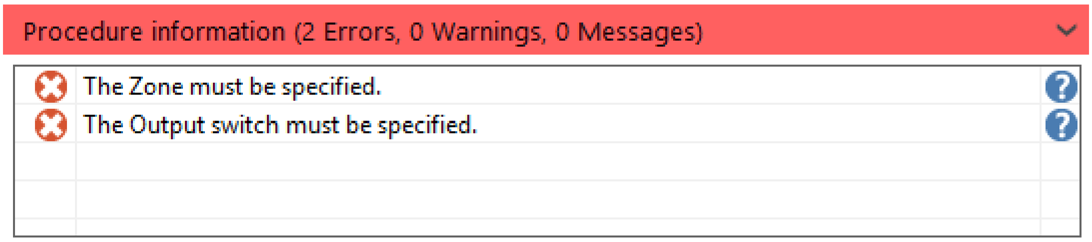


Figure 2. The list of errors for the procedure in figure 1.

Clicking on the line in the list on figure 2 will highlight the error

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Events available to procedures
- Actions available to procedures

Error: Labels must be unique

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor if the procedure contains more than one label with the same name.

More information

Labels are used in a procedure in conjunction with Go to statements to give you control over the flow of a procedure. A procedure uses the name of the label to determine which label to jump to, so if there is more than one label in the procedure with the same name, the procedure can't differentiate between them.

For this reason, you need to ensure that the names of all your labels are unique within a procedure. Note that labels are case-sensitive, which means that the same names with different capitalisation are treated as different labels - i.e. 'Start' is a different label to 'start'. However, I'd advise that you *don't* use the same word with different capitalisation, as it could make the procedure quite confusing!

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Label statement
- Go to statement

Error: The variable passed to the 'Set variable as a test result' action must be a single value variable, set up to be a result for tests

Description

A procedure can use user-defined variables to keep track of a procedure's internal status, calculate values, and generally contain any value that you want. Under some circumstances, you might decide that you want the value of a variable to be available as a result of that test.

Procedures allow you to achieve this by defining a variable as a 'result variable', which means that the procedure can note its value at any time, and that value will then be available in the test's results.

This option is only available for variables that store a numeric *value, and that are single-value variables, rather than array variables.*

If you try to use the *Set variable as a test result* action, and within it you use a variable that has not been set up to be a result variable, then this error will be generated when you check the procedure for errors or exit the procedure editor.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- User-defined variables
- Result variables

Error: Cannot use a 'Wait until' statement before the test has started

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure uses an invalid Wait until statement before the test starts, i.e. in the shaded section at the top of the procedure editor.

This does *not* apply to a wait statement that is waiting for an I/O event, a *Time elapsed* event, or a *Test is waiting to start* event. These are the *only* events that can occur before the test starts.

I/O events (for example, lever presses or new temperature sensor values) can occur before the test starts, and you may want to wait for these events in order to determine whether the apparatus is in the desired state to start the test - for example, wait until a temperature controller has reached the desired temperature. The *Prevent test start* and *Allow test start* actions can be used to prevent or allow the starting of the test if an I/O device is not yet ready. For more information on this, see Preventing a test from starting until an I/O device is ready.

The *Time elapsed* event can be used before a test starts if you are waiting for something to change in order to start the test. For example, your test might rely on the value of a specific Field being set before the test can start. You can find out more details about this in the topic Preventing a test from starting until an I/O device is ready. The example here waits for an I/O event, but you can apply the same principle to check that the value of a field has been entered, for example. In this case, waiting for a Time elapsed event within the loop will simply ensure that the procedure runs more efficiently, and the time elapsed should be relatively short (usually less than a second or so).

Note that the *Test is waiting to start* event will only happen if your protocol is set to start tests automatically. If it's not, then the procedure will just start running the code in the un-shaded area as soon as you click the (Start test) button on the Tests page.

A procedure will start to run the statements in the shaded section at the top of the procedure when the test is 'ready'. A test can actually become ready under different situations:

- *If you change something in the protocol list, then return to the* Tests page*;*
- *When a test ends, the test becomes 'ready' for the next test.*

So it's possible that any test set-up performed by a procedure could run more than once for a given test; you'll need to bear this in mind if the procedure initialises any I/O, for example, as this may be done multiple times under some circumstances.

If you've set up your protocol to **auto-start tests***, then you can wait for the* Test is waiting to start *event in the shaded section of the procedure, and then do your test's initialisation - this will mean that it will only be initialised when the user actually starts a test, and not at any other time (such as* *a change to the protocol settings, or the end of a previous test).*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Running procedures

Error: Cannot wait for the 'Test is waiting to start' event after the test has started

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure uses a Wait until statement to wait for the *Test is waiting to start* event, in the un-shaded area at the bottom of the procedure.

The *Test is waiting to start* event can *only* occur *before* the test starts, so cannot be used in the un-shaded part of a procedure, which runs *after* the test has started. It must be used in the shaded area at the top of the procedure editor.

Bear in mind that the Test is waiting to start *event will not be generated unless the protocol is set to use* Automatic starting of tests.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Automatic starting of tests

Error: Cannot use a variable that relies on tracking information before the test starts

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure uses a built-in variable related to tracking of the animal in the shaded section of the procedure, which is run before the test starts.

More information

Several of the built-in variables provided by ANY-maze to ascertain the current state of a test are only updated when the system starts tracking the animal, i.e. after the test has started. All details of the animal's position or speed, activity, immobility and freezing state are only available once the test has started and tracking has begun. Likewise, information about the animal's position relative to zones (distance from zones, which zone the animal is in, etc.) is not available until the test has started and ANY-maze can determine the position of the animal, and therefore which zone it is in.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Built-in variables

Error: The procedure tries to set the location of a movable zone after the test has started

Description

If you've set up a zone to be a movable zone, you can select for each stage of the test whether the location of the zone should be set by a procedure (rather than setting it to be the same for each animal, or setting it manually when each test starts). This is set up in the stage's settings page, under *Location of the zone during this stage*.

If you've set up your stage in this way, you can use the *Set location of movable zone* action to specify the location of the movable zone. However, this **must** be done in the shaded area at the top of the procedure editor, and not in the main body of the procedure - the location of a movable zone can't be altered after the test has started.

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure is checked for errors, or when the procedure editor is exited.

Example

When using the *Set location of movable zone* action, it must be dragged into the shaded area at the top of the test, as shown below:

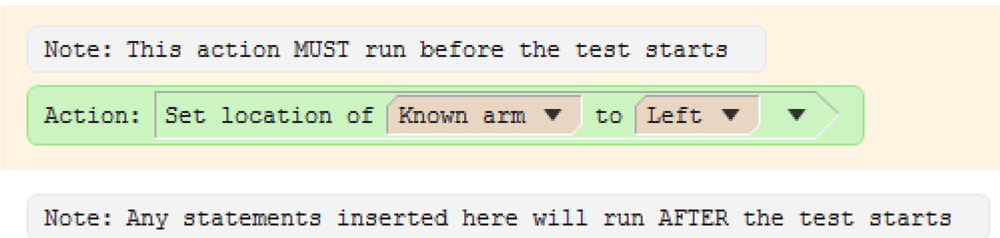


Figure 1. The Set location of movable zone *action can **only** be used in the shaded section of the procedure, before the test starts -* not *the un-shaded section at the bottom of the procedure.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Setting up a stage
- Specifying the positions of movable zones during a stage

Error: The procedure tries to prevent/allow test start after the test has started

Description

There are two actions available to a procedure which are designed to allow some control over when a test can start. These are the *Prevent test start* and *Allow test start* actions, and they are designed to be used with external I/O devices.

If you are using some I/O in your tests, you may need to ensure that the I/O is initialised in a certain way before a test can start. For example, you may have a device with hot and/or cold plates, and you'd like to ensure that these are at a certain temperature before the test is allowed to start. These actions are designed to allow you to wait for the appropriate state of the I/O, before allowing the test to start.

These actions can have no effect *after* the test has actually started, and so including them in the non-shaded area at the bottom of the procedure (the part that runs after the test has started) is not allowed. If they are included here, then this error is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure is checked for errors, or when the procedure editor is exited.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Actions available to procedures

Error: To set the position of a movable zone from a procedure, you must set up the zone's location for this stage to 'The location will be set by a procedure'

Description

If you've set up a zone to be a movable zone, you can change its location for each test using a procedure. This is specified for each stage of the experiment, in the stage's settings page, under Location of the zone during this stage.

If you try and set the movable zone's position in a procedure, using the *Set location of movable zone* action, but you *haven't* specified in the stage that this zone's location should be set by a procedure, then this error will be shown in the list of errors and warnings at the bottom of the procedure editor when the procedure is checked for errors, or when you exit the procedure editor.

More information

To specify that the location of a movable zone can be set by a procedure, select the relevant stage in the protocol list and then the entry for *Location of the ... zone during this stage*. This will allow you to specify that the zone's location will be set by a procedure, by selecting the *The location will be set by a procedure* option.

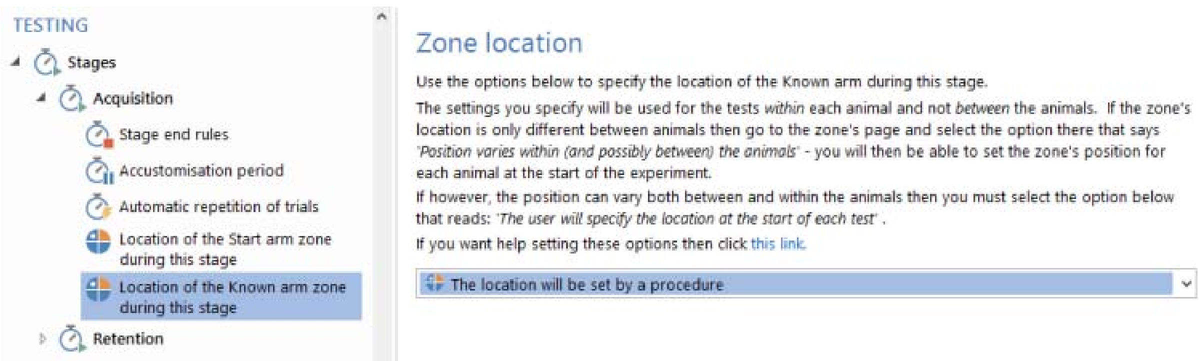


Figure 1. Selecting the option for a stage to allow a procedure to set the location

of a movable zone.

Note also that the *Set location of movable zone* action can **only** be used in a procedure *before* the test starts - i.e. in the shaded area at the top of the procedure. You can't change the location of a movable zone *after* the test has started.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Specifying the positions of movable zones during a stage
- Setting up a stage

Error: Cannot end the test before it has started

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure tries to use an *End the test* action before the test has started.

This applies to the following actions:

- *End the test*
- *End the test (specifying reason)*
- *End the test, but allow continuation*
- *End the test (specifying reason), but allow continuation*

Example

The following procedure will generate an error, as it may try to end the test before it has started. The shaded area represents statements that will run before the test starts; the non-shaded area is run *after* the test has started.

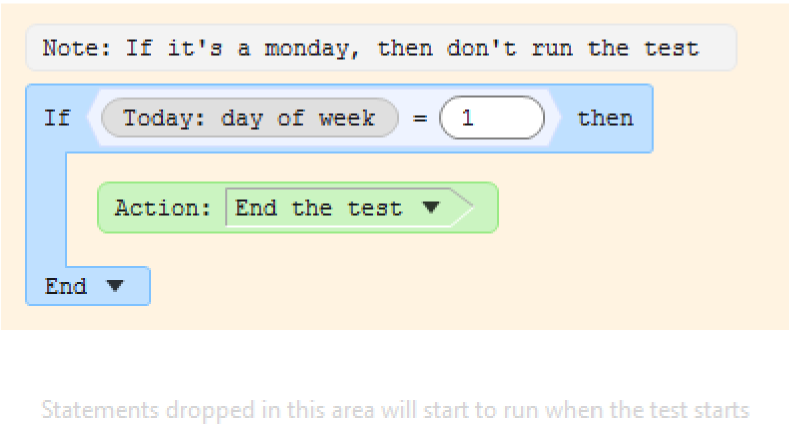


Figure 1. This procedure will cause this error, because it may try to end the test before it has started.

To prevent this error, simply move the statements that use the *End the test* action to be *after* the test starts (i.e. at the start of the un-shaded section of the procedure).

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Actions available to procedures

Error: Cannot use treatments within this procedure - the protocol has been set up not to use treatments

Description

This error is displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure attempts to use the *Treatment* built-in variable, but the *Treatment groups* element of the protocol has been set up not to use treatment groups.

More information

If the protocol has been set up to *not* use treatment groups, the *Treatment* built-in variable will not be available to the procedure. However, it's possible to import a procedure created in a different protocol, which might contain this built-in variable.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Treatment groups

Cannot call a sub-procedure recursively

Description

Sub-procedures allow you to separate out chunks of procedure code, both simplifying procedures and allowing repeated tasks to be called from multiple places (or even multiple procedures). One sub-procedure can also invoke another sub-procedure.

However, a sub-procedure cannot invoke itself - this will lead to the procedure being called recursively, and potentially getting into an infinite loop with no exit.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Re-using parts of procedures

Warning: This procedure has no executable statements

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure does not contain any statements that will affect the test at all (i.e. it doesn't contain any Action statements and it doesn't change the value of any variables).

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure

Warning: This procedure does not include a 'Wait until' statement

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure does not contain any Wait until statements.

More information

A procedure runs through its statements in turn, until it reaches the end of the procedure. If it encounters a loop along the way (either through the use of a Repeat statement or a Go to statement with a Label statement), it repeatedly runs through that loop until the right condition occurs for it to exit the loop (or the procedure finishes). It is only when a procedure reaches a *Wait until* statement that it stops to take a breather, and allows other procedures the chance to run.

If a procedure doesn't contain a *Wait until* statement, there is a danger that it will get 'stuck' in a continuous loop, which will prevent other procedures from getting a chance to run themselves. (In fact, ANY-maze tries to detect this situation and give other procedures a chance to run, but it does make the procedure quite inefficient when this happens).

The best way to prevent any problems is to ensure that the procedure has at least one *Wait until* statement. It is only when the procedure reaches such a statement that it 'takes a breather' and allows other procedures or processing to occur.

Example

The following example shows a procedure with a loop, which doesn't contain a *Wait until* statement. It continually monitors the animal's position, and outputs this position in (x,y) co-ordinates on the display. This procedure will run, although inefficiently, and may slow down processing of other procedures and tracking.

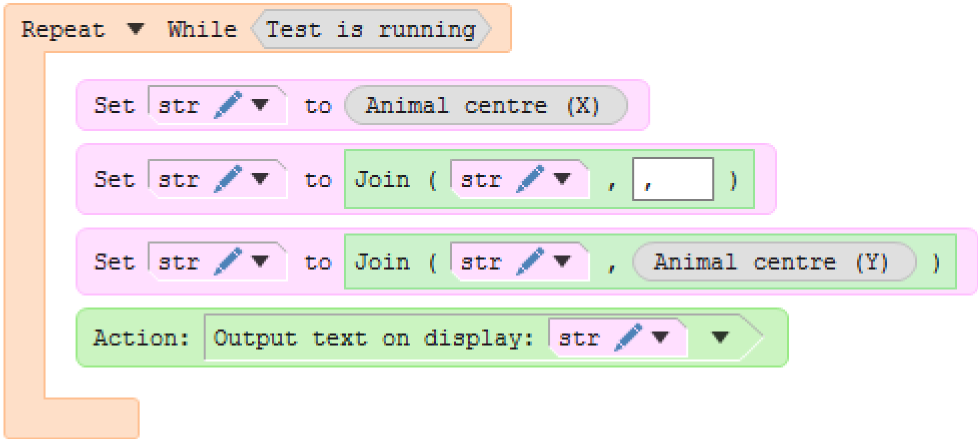


Figure 1. A procedure that is stuck in a loop without a Wait until *statement. Although this procedure will work, it will be inefficient and may stop other procedures from running correctly.*

This warning can be easily fixed by adding a *Wait until* statement:

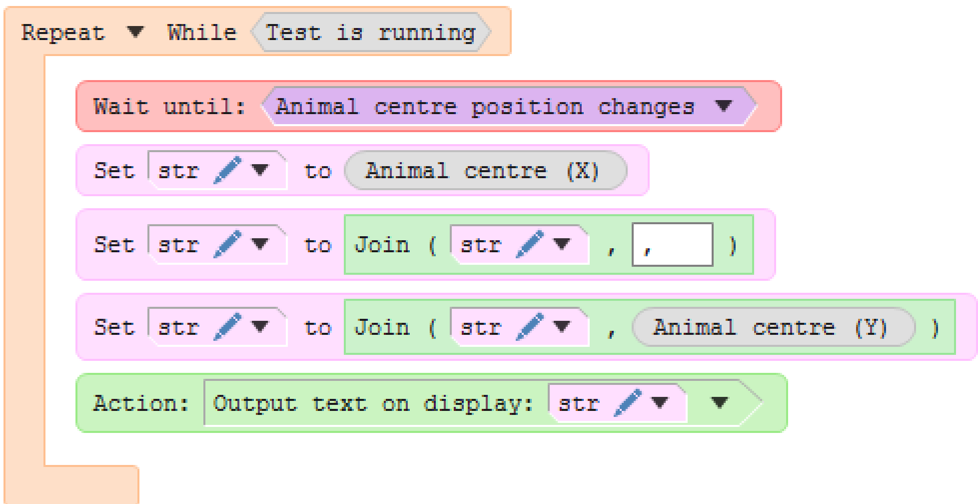


Figure 2. The solution to this is to insert a Wait until *statement. The procedure will do exactly the same thing, but will run more efficiently and also allow any other procedures to run properly.*

If the warning given for the procedure in figure 1 is ignored, and it is used in a test anyway, a run-time warning *will also be generated.*

Note that there are circumstances where it *is* OK to have a procedure without a *Wait until* statement - for example, if you are just running a procedure to set up some I/O at the beginning of a test. If this is the case, you can just ignore this warning.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement

Warning: The Repeat/If statement does not have a clause specified

Description

Within a procedure, Repeat statements and If statements are 'containers' for other statements; that is, they have groups of statements called 'clauses' that run when certain conditions are true. A *Repeat* statement has a clause that runs while (or until) the statement's condition is true; this clause is the group of statements contained within the statement. An *If* statement has a clause for each condition that it evaluates; it must have at least one condition, and therefore at least one clause.

If a *Repeat* statement or an *If* statement do not contain any statements as a clause, then this warning will be generated in the list of errors and warnings at the bottom of the procedure editor when the procedure is checked for errors.

Note that if this warning is ignored, the procedure will still run, but the statement will not do anything if the condition is true. This will be ignored for *If* statements, as it's possible to write a valid procedure with an empty clause for an *If* statement, but for *Repeat* statements, another warning will be generated when the procedure encounters this statement while running.

Example

The following procedure doesn't have any statements contained within the *Repeat* statement, so will generate this error.

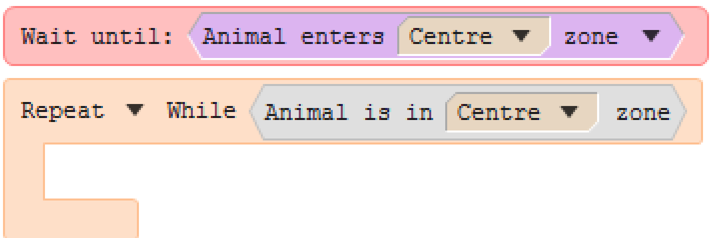


Figure 1. The repeat statement in this procedure doesn't have a clause.

If you receive this warning for a *Repeat* statement, you should ensure that the statement has a clause. For an *If* statement, you should review the procedure to see if it's really what you intend. If it is, then simply inserting a Note statement as the clause, with some explanatory text like 'Do nothing', will prevent this warning from being displayed.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Repeat statement
- The If statement
- The Note statement

Warning: The procedure uses the 'Test is running' variable before the start of the test

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when it tries to use the *Test is running* built-in variable before the test has started.

The procedure editor splits the procedure into two sections - the first, with a shaded background, is the section that runs *before* the test starts. Using the *Test is running* variable is irrelevant in this section of the procedure, because its value is *always* False. The flag can be used in the lower, un-shaded section of the procedure to check whether the test is still running, i.e. the test has not yet ended.

Note that when a test stops, the procedure will also stop running, although if it is in the middle of a sequence of statements at the point that the test stops, it will continue to run until it reaches the end of these statements. If ANY-maze detects that the procedure is stuck in a loop at this point, the procedure will be forcefully exited.

Example

The following procedure will generate an error for the first use of the *Test is running* variable, because it's before the test starts. The second use, after the test starts, is OK.

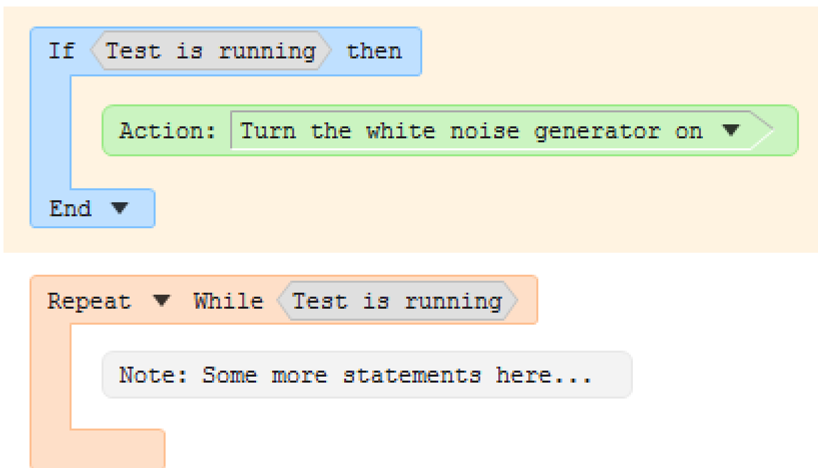


Figure 1. This procedure will generate an error for the first use of the Test is

running *variable, but not the second.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Built-in variables

Warning: The variable passed to the 'Randomise array' function is not an array variable

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure uses the Randomise array statement, but the variable used is not an array variable.

More information

The *Randomise array statement will take an array variable with several values, and shuffle them into a random order. If the variable used by the statement is not an array variable, then its contents cannot be shuffled.*

If this warning is ignored, then when the procedure encounters this statement while it is running, it simply won't do anything and will move on to the next statement in the procedure.

To set a variable up as an array variable, click the button on the variable within the procedure itself, or on the *Variables* tab. When the Variable settings window opens, select the *This variable is an array of ... values* option, and enter the number of values that there should be in the array.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Other statements
- User-defined variables
- Variable settings window

Warning: The variable passed to the 'Initialise array' function must be an array variable

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure uses the Initialise array statement, but the variable used is not an array variable.

More information

If the variable used by the statement is not an array variable, then its contents cannot be initialised using a list of values. If this warning is ignored, then when the procedure encounters this statement while it is running, it simply won't do anything and will move on to the next statement in the procedure.

To set a variable up as an array variable, click the button on the variable within the procedure itself, or on the *Variables* tab. When the Variable settings window opens, select the *This variable is an array of ... values* option, and enter the number of values that there should be in the array.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Other statements
- User-defined variables
- Variable settings window

Warning: The variable passed to the 'Copy array' function must be an array variable

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when the procedure uses the Copy array statement, but one of the variables used (either the variable to copy, or the variable to copy it to) is not an array variable.

More information

If one of the variables used by the statement is not an array variable, then its contents cannot be copied to or from another array. If this warning is ignored, then when the procedure encounters this statement while it is running, it simply won't do anything and will move on to the next statement in the procedure.

To set a variable up as an array variable, click the button on the variable within the procedure itself, or on the *Variables* tab. When the Variable settings window opens, select the *This variable is an array of ... values* option, and enter the number of values that there should be in the array.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Other statements
- User-defined variables
- Variable settings window

Warning: This event will never occur

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure contains one or more Events which, due to changes made to the experiment's protocol, will not fire during a test.

Details

Settings in an ANY-maze protocol may be changed at any time, either before, during or after tests have been run. If a procedure uses an event that depends on one of these settings, then the procedure will no longer work as expected. For example, if a procedure is waiting for the animal to become immobile, but the protocol is edited to turn off Immobility detection, the *Animal becomes immobile* event will never happen.

As part of its checking of a procedure (either when the procedure is checked for errors, or when you leave the procedure editor), ANY-maze will warn you about these events.

Reasons why events will not fire

The checking of a procedure will detect the following reasons why an event may not fire:

- **Auto-starting of tests is turned off in the protocol.** This applies to the *Test is waiting to start* event. When tests are started automatically, clicking the (Start test) button will put the test into a 'waiting to start' state, while ANY-maze waits for the experimenter to put the animal into the apparatus. When tests are started manually, however, clicking the button will actually start the test immediately, and so the *Test is waiting to start* event will never occur.
- **Immobility detection is turned off in the protocol.** The *Animal becomes mobile* and *Animal becomes immobile* events will only be triggered if the protocol is set up to detect immobility.
- **Freezing detection is turned off in the protocol.** The *Animal freezes* and *Animal unfreezes* events will only be triggered if freezing detection is turned on in the protocol.
- **Immobility is turned off in the protocol, and no keys are defined to count as activity.** ANY-maze only registers the animal as being 'active' if immobility detection is turned on, or a key has been set up to count as an activity. Only then can ANY-maze trigger the *Animal becomes active* and *Animal becomes inactive* events.

- **This item is not available in the current protocol mode.** Several items in the protocol are only available for selected protocol modes. For example, the *Video tracking mode* does not include I/O, and so none of the I/O inputs or outputs will trigger any events (e.g. *On/off input activated, Speaker starts playing,* etc.). There may be other events that are not available, if the selected protocol mode is different to that used when the procedure was written.
- **Head tracking is not enabled in the protocol.** Some events, for example *Animal head position changes* or *Is animal oriented towards a zone,* rely on ANY-maze tracking the animal's head and/or tail. If tracking the animal's head is turned off, these events will no longer occur.
- **The current protocol mode does not use video tracking.** Events such as *Animal centre position changes* will not be triggered if the current protocol mode does not include video tracking.
- **The virtual switch is only available after the test has finished.** Virtual switches can use measures that are only available at the *end* of a test, and not during the test itself - for example, the *Relative animal length* measure is only calculated when a test finishes. If a virtual switch relies on this measure, it cannot be used *during* the test itself, and so the event for this virtual switch being active will never occur while a procedure is running.

For all of the above, ANY-maze will register a warning for the procedure, but the procedure will still run. It is up to you to determine whether the failure of the event to occur will cause you problems in your tests.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Events available to procedures
- The Wait until statement

Warning: This variable is not in use for the current protocol settings

Description

This warning is displayed in the list of errors and warnings at the bottom of the procedure editor when a procedure contains one or more built-in variables which, due to changes made to the experiment's protocol, are not used during a test.

Details

Settings in an ANY-maze protocol may be changed at any time, either before, during or after tests have been run. If a procedure uses a variable that depends on one of these settings, then the procedure will no longer work as expected. For example, if a procedure checks the value of the *Animal is frozen* variable, but the protocol is edited to turn off freezing detection, the value of the *Animal is frozen* variable will *always* be False.

As part of its checking of a procedure (either when the procedure is checked for errors, or when you leave the procedure editor), ANY-maze will warn you about any variables which are not valid with the current protocol settings.

The following built-in variables are affected by this:

- **Animal centre position** variables are only available if the current protocol mode includes video tracking. If not, then all position values will be zero.
- **Animal is mobile** - This is only available if the protocol is set up to detect immobility. If not, then this variable will always be True.
- **Animal is active** - This is only available if immobility detection is enabled in the protocol, or if a key has been set up to count as an activity. If not, then this variable will always be True.
- **Animal is frozen** and **Freezing score** - These are only available if the protocol is set up to detect freezing. If not, then the *Animal is frozen* variable will always be False, and the *Freezing score* will always be 0.
- **Head Distance from Zone**, **Head distance from point**, **Is Oriented Towards Zone** etc. rely on tracking of the animal's head. If head tracking is turned off in the protocol, then these variables will always return False (or #N/A for numerical values).

ANY-maze help topic H0470

Elements of a procedure

There are a number of procedure 'building blocks' that can be used to create a procedure:

- Statements, which are the top-level items that control the flow of the procedure. They will be executed one after another as the procedure runs.
- Events, which represent the things happening in the apparatus as the test is run. These could be events due to animal movement (the animal enters a zone, the animal freezes, etc.) or I/O events (a lever was pressed, etc.), or other events such as the test ending. Note that these events are similar to the events in the old Events and Actions system; a basic set of events is listed automatically on the left-hand side of the procedure editor, and new events can be set up by clicking the *Create event* button, which will allow you to create more complex events.
- Actions, which are used when the procedure needs to tell the ANY-maze software to actually do something – for example, end the test, or turn on an I/O device or adjust its settings. These are similar to the actions that were available in the old Events and Actions system, but allow more flexibility in the parameters that can be specified for these actions.
- Variables, which can be one of two types:

  1. There are a number of Built-in variables that a procedure can retrieve from the ANY-maze software to determine the current state of the test - for example, the animal's current speed or its distance from a specific zone.

  2. You can also define your own variables for use in the procedure – these are values that the procedure can use for whatever purpose it requires, for example, to maintain a count of the number of times the animal enters a zone. To create a variable for use within a procedure, click the *Create variable* button.

Each of these procedure elements has its own tabbed section in the procedure editor window, containing the items of that type that are available to procedures.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

Statements available to procedures

Introduction

Statements are the top-level items that will be executed one after another as the procedure runs.

The following statements are available to a procedure:

- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements

Click on the link in the list above for more information on the statement.

The *Statements* tab also contains the following items:

- Logical operators
- Comparison operators
- Constants
- Maths functions and operators
- Other functions

Click on the link in the list above for more information.

See also:

- Elements of a procedure

The *Wait until* statement

Introduction

The *Wait until* statement allows a procedure to respond to a specific event occurring in a test. For example, it could wait for the animal to enter a specific zone, a switch to be activated, or a specific time period to elapse.

Execution of the procedure will pause until this specific event occurs, at which point, the next statement after the *Wait until* statement will be executed.

For full details of the events available to a procedure, see Events available to procedures.

Example

In the following example, the procedure will wait until the animal enters the 'Island' zone. If it does, the action will be executed (to end the test). If the animal never enters the island zone, then the procedure will never run the action (and the test will end when its test duration has elapsed).

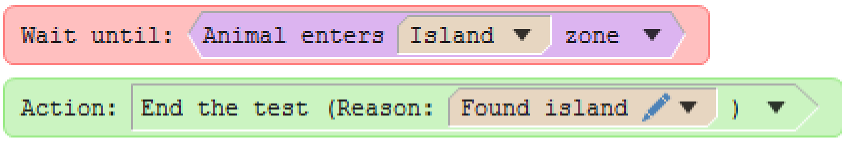


Figure 1. An example of a procedure with a Wait until *statement.*

More information

A procedure should almost always contain a Wait until statement at some point, particularly if the procedure contains a loop - either within a *Repeat* statement or within a loop created by a *Label* statement and a *Go to* statement. An exception to this is a procedure that simply does some set-up for the test (for example, setting up some I/O devices before the test starts). In this case, it's perfectly valid for it not to contain a *Wait until* statement.

If your procedure doesn't have any Wait until statements, you'll get a warning *in the procedure editor when you check the procedure for errors, or leave the procedure.*

A *Wait until* statement allows the procedure to pause and allow other processing to be done (including allowing other procedures to run). If a procedure contains a loop, but no *Wait until* statements, then the statements will just run continually without letting other things have a chance to take their turn. (In fact ANY-maze tries to detect this situation, and compensates for it by allowing other procedures to run anyway, but this is an inefficient way to run a procedure and so it's much better to have a *Wait until* statement in a procedure with a loop).

As an example, consider the following procedure, which uses a loop but does not include a *Wait until* statement.

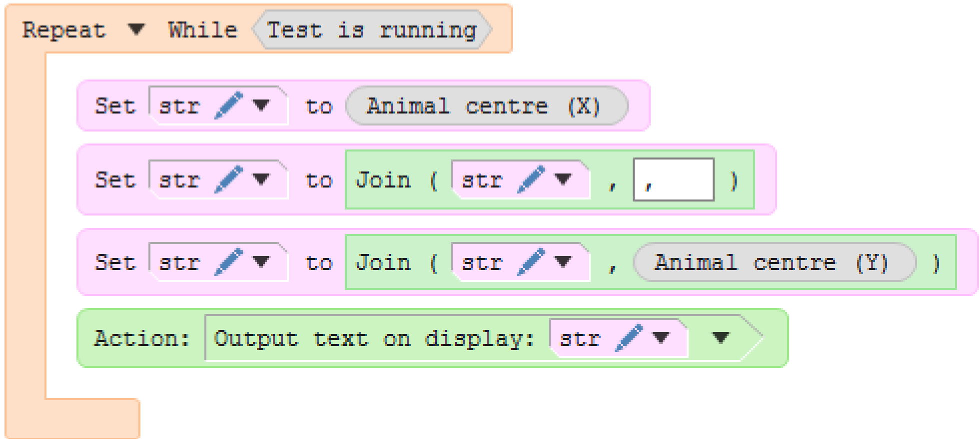


Figure 2. A procedure that contains a loop without a Wait until *statement.*

This procedure will repeat this loop continuously throughout the test, but the absence of a *Wait until* statement will mean that it won't take a break and give other procedures the chance to run. To fix this, you need to insert a *Wait until* statement.

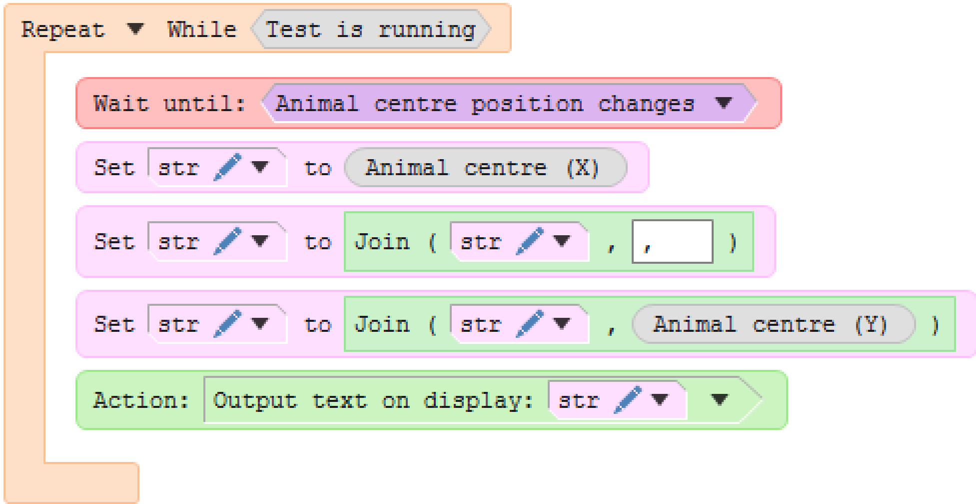


Figure 3. The solution to this is to insert a Wait until *statement. The procedure will do exactly the same thing, but will run more efficiently and also allow any other procedures to run properly.*

If the warning given for the procedure in figure 2 is ignored, and it is used in a test anyway, a run-time warning *will also be generated.*

A *Wait until* statement will usually be used to wait for an event. However, it can actually be used to wait for *anything* that evaluates to a True or False value. This could be the value of a built-in variable, a user-defined variable, or any expression that evaluates to True or False. It can also use Logical operators to look for any or all of multiple True/False values at the same time.

There is one limitation to this, however - if a *Wait until* statement uses the logical *AND* operator, you can't use events on both sides of the *AND*. This is because the events that occur within an ANY-maze test occur one at a time, and so the procedure can't actually receive more than one event simultaneously.

So what would happen if you actually wanted to wait for two things? Let's say that you had four zones for the corners of an apparatus (NW, NE, SW and SE) and wanted to wait for the animal to have entered BOTH of the top corners, in any order (i.e. the animal to enter NW and then NE, or NE and then NW). You might be tempted to write something like the following, but this is NOT a valid procedure:

Figure 4. This procedure is NOT valid; it contains an error and cannot be run.

These two events *cannot* occur at the same time, so the procedure (if allowed to run) would wait forever. So how *would* you achieve what you want?

ANY-maze provides you with a number of built-in variables that the procedure can use to ascertain the state of the test. In most cases, you should be able to find one that will help you out, although if you can't, then you can use a user-defined variable to keep track of a specific state and then use that in the *Wait until* statement.

In this case, what you really want to know is when the animal enters the NW zone (having *already* entered the NE) or when the animal enters the NE zone (having *already* entered the NW). ANY-maze provides you with a built-in variable that tells you how many times the animal has visited a specific zone, so you can use this in conjunction with the event to achieve this objective:

Figure 5. Using events AND built-in variables to wait for an animal to have entered both zones.

You could also achieve this using multiple procedures - one procedure that waits for a NW zone entry and sets a 'NW entered' shared variable; another procedure that waits for a NE zone entry and sets a 'NE entered' shared variable; and a third that waits for 'NW entered' AND 'NE entered'.

Waiting for times to elapse

ANY-maze waits for times to elapse using the *Time elapsed* event. Note that this is *not* the time elapsed since the beginning of the test, but is the time elapsed from the point that the procedure gets to the *Wait until* statement that contains the *Time elapsed* event.

If you want to wait for the time elapsed since the beginning of the test, use the Create event *button to open the* ***Event trigger wizard*** *and select 'A specific duration since the start of the test' from the list of 'Triggers that relate to time'.*

Within an ANY-maze procedure, there is a time resolution of approximately 50ms. This should be perfectly fine for most procedures, and you'll only need to change this if you need a finer resolution than this. You might need a finer resolution, for example, if you have a switch that you need to toggle on and off with a very short interval.

If you need to do this, you'll need to use the *Set timer resolution* action to alter the timer resolution for this procedure. So if you need your switch to be toggled for 10ms, you'd need to change the resolution to 10ms or less, as follows:

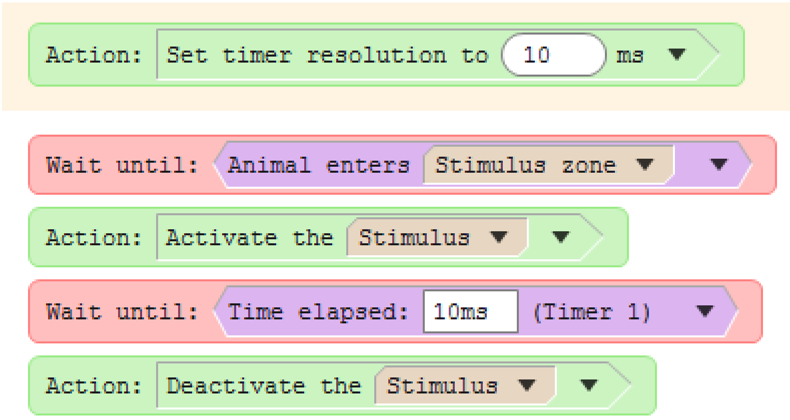


Figure 6. This procedure alters the time resolution at the start of the test, and then uses it to ensure a small time interval for the toggling of an output switch.

Changing the timer resolution in this way only affects the procedure that it is used in, and not any other procedures. However it does mean that the procedure needs to process its statements much more often, using more of the computer's processing power, so it's not a good idea to alter the resolution unless you really need to (or are advised to do so by ANY-maze technical support).

The timer resolution can be set to anything between 1ms and 60 seconds; however, note that there's a small margin of error, particularly for I/O devices, and a resolution of 1ms actually means a time interval of approximately 1-2ms.

Limitations

For most possible events, a *Wait until* statement cannot be used before a test starts (i.e. in the shaded section at the top of the procedure editor). This is because most events cannot occur before a test start (the animal cannot enter any zones, for example). The only events that can be used in a *Wait until* statement before the test starts are I/O events, a *Time elapsed* event, or a *Test is waiting to start* event.

I/O events (for example, lever presses or new temperature sensor values) can occur before the test starts, and you may want to wait for these events in order to determine whether the apparatus is in the desired state to start the test - for example, wait until a temperature controller has reached the desired temperature. The *Prevent test start* and *Allow test start* actions can be used to prevent or allow the starting of the test if an I/O device is not yet ready - See Preventing a test from starting until an I/O device is ready for more details.

The *Time elapsed* event can be used before a test starts, if you are waiting for something to change in order to start the test. For example, your test might rely on the value of a specific Field being set before the test can start. You can find out more details about this in the topic Preventing a test from starting until an I/O device is ready. The example here waits for an I/O event, but you can apply the same principle to check that the value of a field has been entered, for example. In this case, waiting for a Time elapsed event within the loop will simply ensure that the procedure runs more efficiently, and the time elapsed should be relatively short (usually less than a second or so).

Note that the *Test is waiting to start* event will only happen if your protocol is set to start tests automatically. If it's not, then the procedure will just start running the code in the un-shaded area as soon as you click the ▶ (Start test) button on the Tests page.

If a procedure is waiting for an event before the test starts, then this wait will be 'aborted' when the test *actually* starts, and execution of the procedure will jump directly to the first statement in the un-shaded area of the procedure (i.e. the area that runs after the test has started).

See also:

- Statements available to procedures
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements

The *Action* statement

Introduction

The *Action* statement allows a procedure to do something that will affect the test. This might be to activate or alter some kind of I/O item, for example, or to pause or end the test itself. Actions can also be used to output some explanatory text to the display, to generate a result that can be used in analysis, or to aid in debugging a procedure.

For full details of the actions available to a procedure, see Actions.

Example

In the following example of a couple of lines taken from a procedure, there are two action statements used, which will execute when the animal enters the 'Centre' zone.

The first action will activate the 'Shock floor' output switch, and the second will output some text to the display indicating that this has happened.

The text that is output to the display in this case will also appear in any video that is recorded of this test.

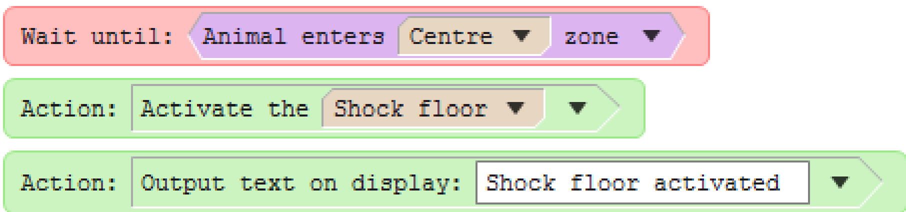


Figure 1. An example of a procedure with two Action *statements*

The example above uses an action to output text to the display, although you could also use the What to display while testing *settings in the protocol list to do the same thing, by showing the active output ports on screen during a test.*

See also:

- Statements available to procedures
- The Wait until statement

- The Repeat statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements

The *Repeat* statement

Introduction

The *Repeat* statement allows simple loops to be inserted into a procedure, to enable the procedure to execute a set of statements a specific number of times, or to run continually until the test ends. Within the procedure editor, the *Repeat* statement visually 'contains' a set of statements, which are the statements that will get repeated.

The repeat statement has four options that will determine exactly how it will repeat. These options determine what causes the procedure to 'drop out' of the loop, and are selected using a drop-down list at the bottom of the statement:

- It can repeat *forever* (actually, for as long as the test is running)
- It can repeat a specific number of times
- It can repeat *until* a specific condition is true, which means that the statements contained within the loop will execute at least once, and then the condition will be evaluated. If the condition is true then the *Repeat* statement will drop out, otherwise the *Repeat* statement's contents will execute again.
- It can repeat *while* a specific condition is true. In this case, the condition will be evaluated *first;* the repeat statement's contents will only be executed if the condition is true.

Examples

The following code will wait for the animal to enter a reward zone, and will then dispense a pellet. However, it will only do this three times:

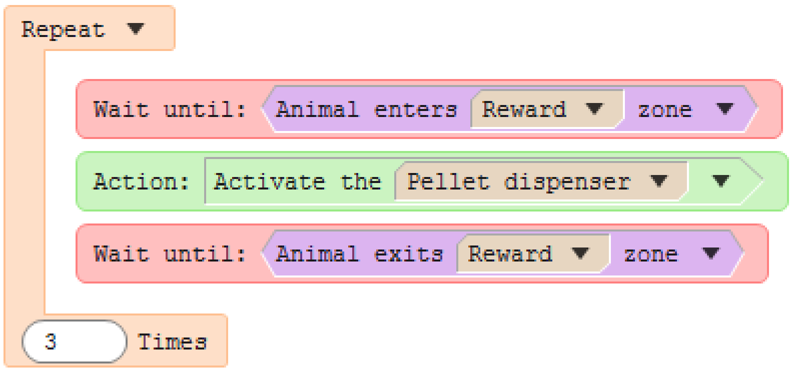


Figure 1. Repeat 'A number of times' will repeat the statements contained within it a specified number of times.

The following two examples illustrate the subtle (but important!) difference between repeat *while* and repeat *until.*

The first example will evaluate the condition repeat *while* the condition is true. This means that it will evaluate the condition *before* running any of the contents.

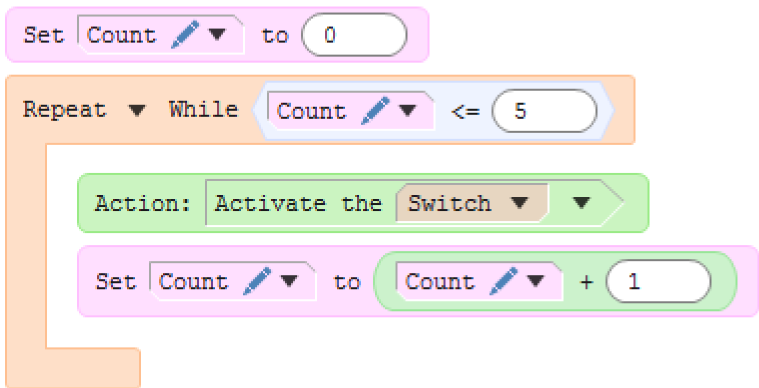


Figure 1. Repeat 'While' will evaluate the condition first, *and will only execute the contents if the condition is True.*

In this case, the 'Count' variable starts as 0; the condition is less than or equal to 5 so the contents will run and the count is incremented. Again, the condition is true, so the contents will run again. The third time, the count is 2; the fourth time 3; the fifth time 4; and the sixth time 5; and each time the contents of the statement will run. The seventh time it is run, the count variable's value is checked - it is now 6, so the contents of the repeat loop will not be run. So altogether, the switch will have been activated six times.

The second example will repeat the contained statements *until* the condition is true.

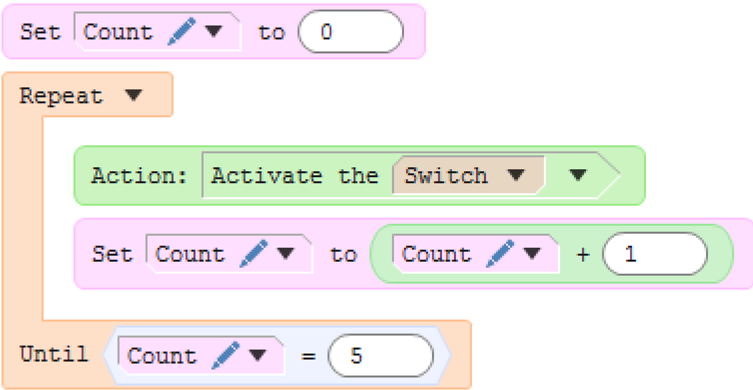


Figure 2. Repeat 'Until' will run the contents before *evaluating the condition.*

Regardless of the value of the 'Count' variable, the contents of the statement will run at least once. The count will then be incremented to 1 and the condition checked - it will be true, so the repeat statement will run again. After this second run, the count will be 2. After the third run it will be 3, after the fourth run it will be 4, and after the fifth run it will be 5. At this point the condition is true so the repeat statement will drop out. Altogether, in this example, the switch will have been activated 5 times.

So you can see that even though the procedure looks very similar in each case, the way it runs is actually subtly different.

More information

Repeat statements can 'contain' any statements within the loop. This can include further *Repeat* statements or *If* statements, and these in turn can contain *Repeat* statements or *If* statements, 'nested' to any level.

Within the ANY-maze procedure editor, dragging a *Repeat* statement will also drag its contents. Similarly, deleting the statement will also delete *all* its contents. If you want to keep the contents within the statement, but delete the *Repeat* statement itself, you'll need to drag the contents *out* of the *Repeat* statement first, to a different point in the procedure, before deleting it.

See also:

- Statements available to procedures

- The Wait until statement
- The Action statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements

The *If* statement

Introduction

The *If* statement, or 'Conditional' statement, will take one of a number of actions depending on whether a certain condition is true.

If statements with a single condition

The simplest form of the conditional statement will check one single condition, and take appropriate action if that condition is true. For example, the following example procedure will end the test if the animal does not enter a specific zone for 30 seconds.

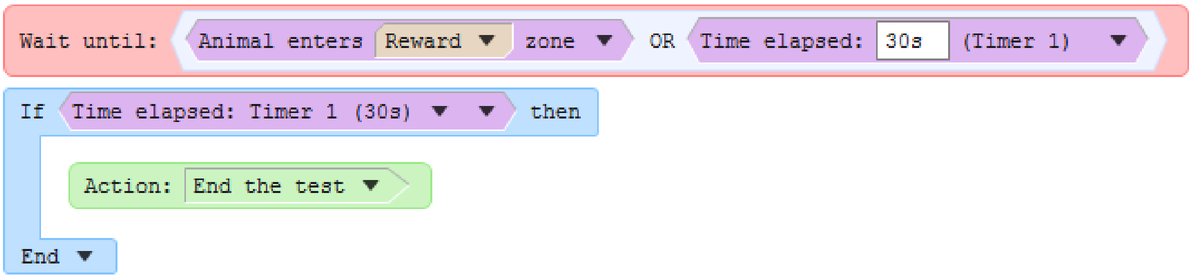


Figure 1. This simple If *statement will wait for either a zone entry, or a specific time to elapse. It uses an* If *statement to see which event it was that occurred, and if it was the* Time elapsed *event, the test is ended.*

To do this, it must wait for one of two things to happen - either the animal enters the zone, OR 30 seconds elapse without the animal entering the zone. We can use a logical 'OR' operator to wait for one of two events to happen, but then we are left with the situation that we need to take a *different* action depending on which event actually occurred. This is where the *If* statement comes in.

The *If* statement allows the procedure to check which event *did* actually occur, and then run a different set of statements accordingly.

Conditional statements with multiple conditions

Conditional statements can have any number of conditions, which they evaluate one after the other, in order. When the procedure reaches the first condition that evaluates to True, the statements within that section of the *If* statement will be run.

If statements can also have a 'catch all' condition called 'Else', which contains code that runs if *none* of the other conditions are true.

The following example shows this:

Figure 2. This If *statement evaluates a number of conditions; if none of these conditions are true, the statements in the* Else *clause will run.*

More information

A common mistake when writing a procedure is to use an *If* statement instead of a *Wait until* statement when waiting for an event to occur.

When you're deciding what you need a procedure to do, you might think to yourself 'Well, if the animal enters the island, I want to end the test'. So you drag in the *If* statement, and drag in the *Animal enters Island zone* event as its condition.

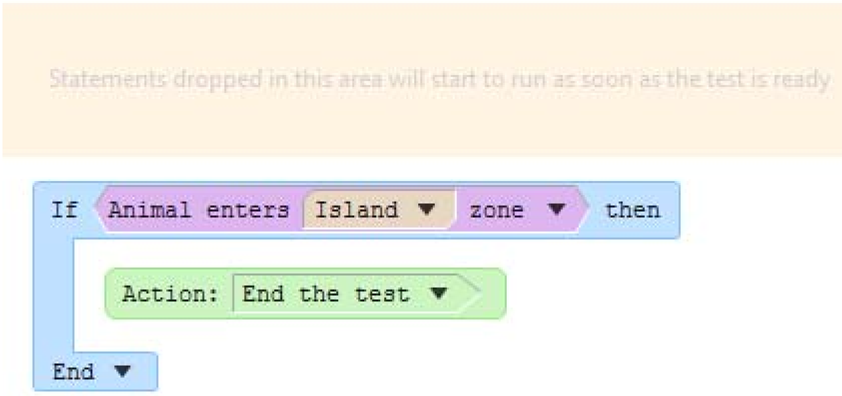


Figure 3. This procedure is incorrect (and an error will be generated when the procedure is checked for errors, or when you exit the procedure editor).

However, this isn't what an *If* statement is designed for. An *If* statement is used to evaluate a specific condition, at a single point in time in the procedure - so the chance of the procedure running the *If* statement at the exact point that the *Animal enters Island zone* event occurs is pretty slim!

When thinking about writing a procedure, you need to consider what the *procedure* is going to do, and remember that the procedure will just run through each of its statements in turn until it has finished. In this example, it's going to do nothing *until* the animal enters the island - i.e. it's going to *Wait until* the animal enters the island - before continuing.

Adding further conditions

When you drag an *If* statement into the ANY-maze procedure editor, it will contain a single condition and space for the statements to execute if that condition is true. To add further conditions, select the drop-arrow next to the 'End' at the bottom of the statement, and select whether you require another condition ('Else if'), or a catch-all condition ('Else') if no other conditions are valid.

You can also select 'End' from the list, but this won't have any effect and will leave the existing 'End' in place.

Within the procedure editor, if you need to remove a condition from an *If* statement, just right-click on the 'arm' of the condition (where the 'Else if' or 'Else' text is displayed) and you'll see a menu appear. Select the last item, 'Delete this condition' to delete the condition.

Note that this will also delete *all* the statements contained within this condition! If you don't want this to happen, select multiple statements using the left-hand margin of the editor and drag those statements out of the *If* statement, to somewhere else in the procedure, before deleting the condition.

If statements can 'contain' any number of statements within each condition. This can include further *If* statements or *Repeat* statements, and these in turn can contain *If* statements or *Repeat* statements, 'nested' to any level.

Within the ANY-maze procedure editor, dragging an *If* statement will also drag all its contents. Similarly, deleting the statement will also delete *all* its contents, in all its clauses.

See also:

- Statements available to procedures
- The Wait until statement
- The Action statement
- The Repeat statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements

ANY-maze help topic H0421

The *Set value* statement

Introduction

The *Set value* statement is used to set the value of a user-defined variable. User-defined variables allow a procedure to keep an internal note of things that are important to a procedure - for example, a 'counter' to keep track of the number of times specific things happen.

The *Set value* statement has two parameters - the first of these is the variable whose value is being set, and the second is the new value to set the variable to. This new value can be a simple numeric, text, or True/False value, or could be the result of any operation that results in one of these values (for example, the result of a Maths function or operator, or the value of another variable).

Example

The following example procedure uses the *Set value* statement twice - firstly to initialise the value of a count to zero (which is done before the test starts), and then (within a loop) to increment a counter. Note that the *value* of the variable is set to the result of a mathematical function which adds 1 to the previous value of the variable.

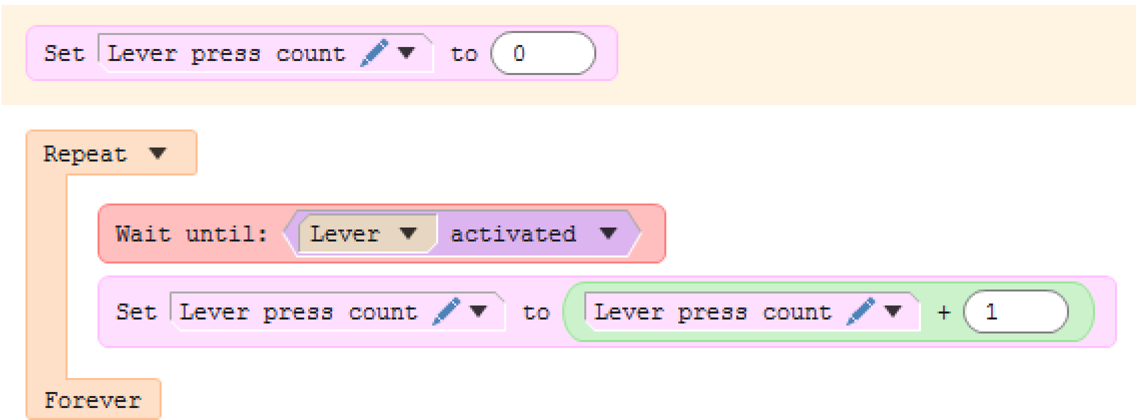


Figure 1. A procedure showing the use of the Set value *statement to initialise and then increment a variable used as a count.*

Note that any variable can be set to a numerical, text, or True/False value (for the tech savvy, this means that variables do not have a 'type').

More information

A *Set value* statement can only be used to alter the value of a user-defined variable, and *not* a built-in variable. This is because built-in variables can *only* be read by a procedure, and not changed.

Shared variables

If a user-defined variable is defined to be shared between procedures, what happens if more than one procedure tries to change the value of the variable at the same time?

In fact, this can't happen, because of the way that procedures run - ANY-maze will look at each procedure in turn, and allow each one to run as much of the procedure as it can, until it either finishes running, or encounters a *Wait until* statement. It will then move on to the next procedure in the protocol.

Because of the way this works, only one procedure is running a statement at a time, so only one procedure can change the value of a variable at a time. However, it is quite possible that a procedure could change the value of a variable, only for another procedure to change it again almost immediately afterwards. For this reason, if you're writing multiple procedures for a protocol, you need to carefully consider how they will interact with each other.

See also:

- Statements available to procedures
- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements

The *Go to* statement

Introduction

The *Go to* statement is used in conjunction with a Label statement, to perform simple loops or jumps in a procedure.

A *Go to* statement can jump to a label anywhere in the procedure. If the label is *before* it in the procedure, then this will usually result in a simple loop in the procedure (i.e. repeating a sequence of statements multiple times). If the label is *after* it in the procedure, then more complex logic can be performed - for example, missing out a section of the procedure depending on certain conditions in the test.

Example

The following example shows a *Go to* statement being used to perform a simple loop, which executes continually throughout the test.

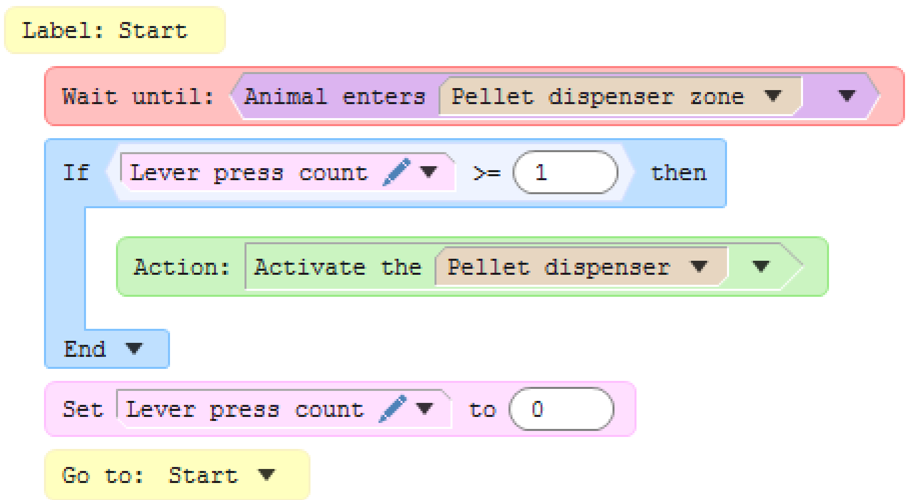


Figure 1. A procedure showing the use of the Go to *statement to perform a simple loop.*

This same logic could also have been implemented using a Repeat *statement.*

More information

A *Go to* statement can be used inside a container *If* statement or *Repeat* statement, to 'jump' out of that statement and continue the procedure running from a different point. Any statements within the container that occurred after the *Go to* statement will not be run.

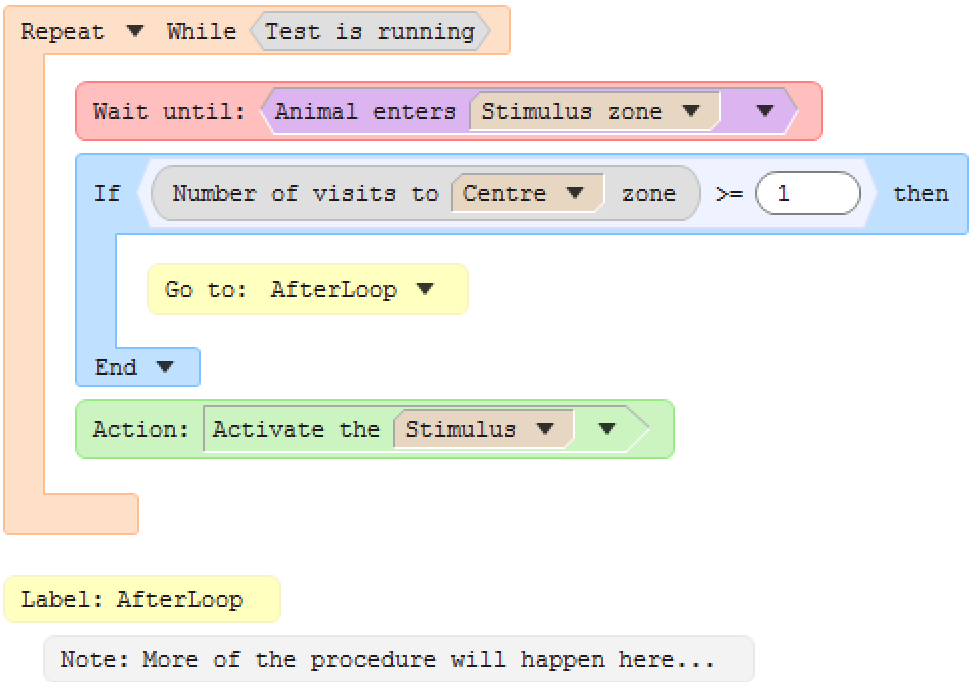


Figure 2. A procedure showing the use of the Go to *statement to jump out of a* Repeat *statement. If the animal has already entered the 'Centre' zone, then the stimulus will not be activated.*

See also:

- Statements available to procedures
- The Label statement
- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement

- The Run sub-procedure statement
- The Note statement
- Other statements

The *Label* statement

Introduction

The *Label* statement is usually used in conjunction with a Go to statement, to perform simple loops in a procedure. It can also be used without a *Go to* statement, to simply show the start of a new section of a procedure.

Within the procedure, the statements after a label are indented to show that it is the start of a new section of the procedure.

Example

The following example shows a *Label* statement used in conjunction with a *Go to* statement, to perform a simple loop which executes continually throughout the test.

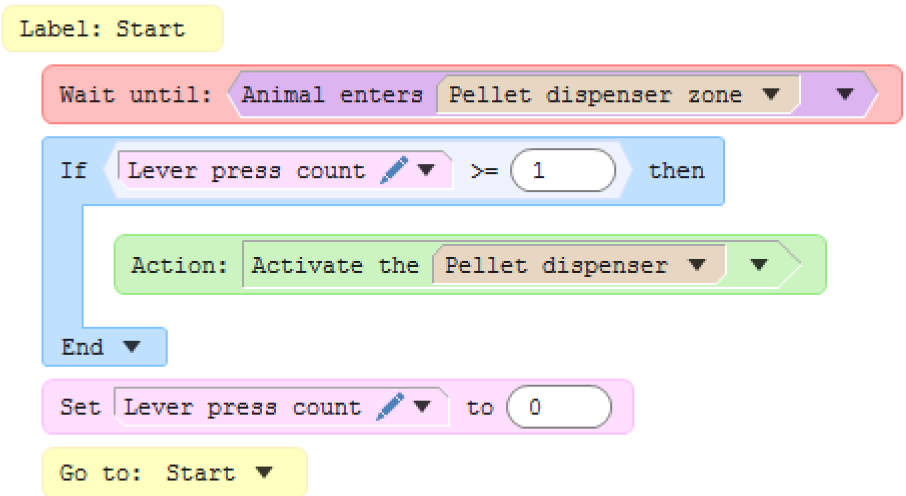


Figure 1. A procedure showing the use of the Label *statement along with a* Go to *statement, to perform a simple loop.*

This same logic could also have been implemented using a Repeat *statement.*

More information

Labels within a procedure *must* have unique names, to ensure that a *Go to* statement can differentiate between them. If you try and define a procedure with multiple labels with the same name, an error will be generated when you check the procedure for errors or exit the procedure editor.

A *Label* statement can be inside a container *If* statement or *Repeat* statement, so that a *Go to* statement can jump into the container and execute from there. However, this will result in quite complex logic within the procedure, so I don't recommend that you do this!

See also:

- Statements available to procedures
- The Go to statement
- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement
- The Run sub-procedure statement
- The Note statement
- Other statements

ANY-maze help topic H0424

The *Run sub-procedure* statement

Introduction

The *Run sub-procedure* statement is used to invoke a sub-procedure, which allows chunks of procedure code to be moved into their own

A *Run sub-procedure* statement can run any procedure that has been set up as a sub-procedure. The same sub-procedure can be invoked any number of times within the same procedure, and can also be run from more than one procedure. A sub-procedure can even be run from within another sub-procedure.

More information

More information on sub-procedures, and an example of how to use them, can be found under Re-using parts of procedures.

See also:

- Statements available to procedures
- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Note statement
- Other statements

The *Note* statement

Introduction

The *Note* statement has no effect as far as running a procedure is concerned, and exists solely to allow you to include helpful comments about the procedure, either for yourself or for anyone else looking at the procedure.

Imagine coming back to a protocol with a complicated procedure in a year's time, and trying to understand what it does - the more notes you've left yourself, the easier this will be! I highly encourage you to make as many notes about the procedure as possible, either using this statement or using the notes field on the procedure's settings page.

Example

The following example shows the use of *Note* statements to indicate to other ANY-maze users how the procedure is using a variable.

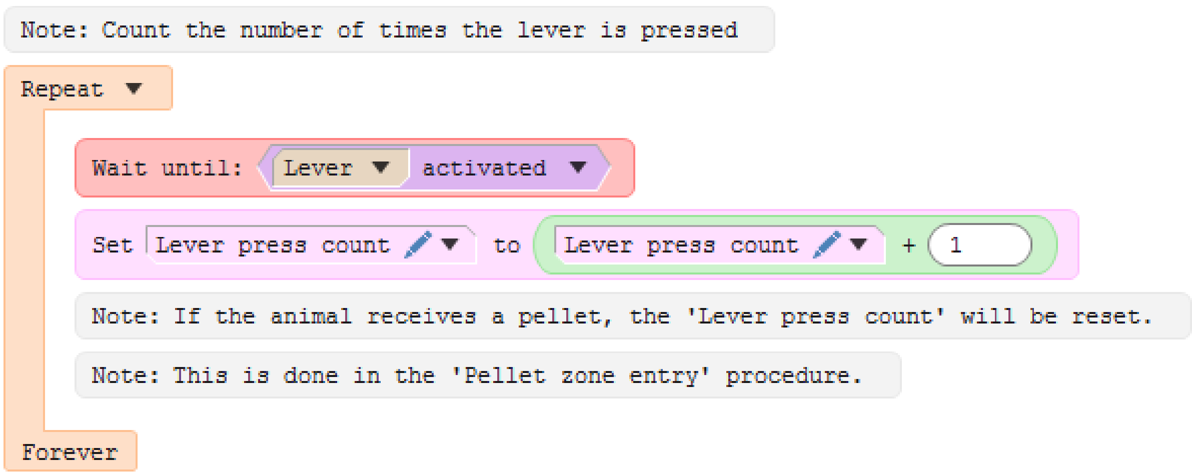


Figure 1. A procedure showing the use of the Note *statement to give helpful information to anyone else looking at the procedure.*

See also:

- Statements available to procedures
- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- Other statements

Other statements

Introduction

Along with the basic statements that are the building blocks of all procedures, there are some more esoteric statements that are used in more advanced procedures.

Most of these statements are to add useful functionality when working with user-defined variables, particularly variables that are an array of values.

The functions covered in this topic are as follows:

- The Initialise array statement
- The Randomise array statement
- The Copy array statement
- The Set to #N/A statement

The *Initialise array* statement

The *Initialise array* statement is a simple way to set up the different elements of an array variable without needing to set up a loop. The values for the array are entered separated by commas.

The following example initialises an array variable with 5 values to contain the values 1000, 2000, 3000, 4000 and 5000 in numerical order. These will be used later as tone frequencies to play through an AMi's speaker.

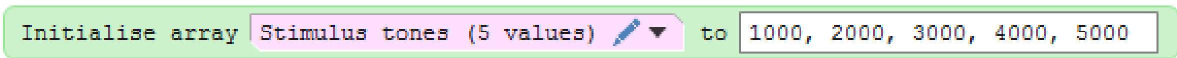


Figure 1. Using the Initialise array *statement to set up the values of an array variable.*

If there are more comma-separated values than the array will hold, any extra values will be ignored. If there are fewer values than the array will hold, the other values will be set to zero.

If the number of values used to initialise the array is not the same as the number of values in the array, you will get a warning when the procedure is checked for errors. If this is not fixed when writing the procedure, then when the procedure is run, another warning will be listed in the Test *details report.*

The Initialise array *statement can be used at any point in a procedure, and doesn't necessarily have to be used at the start.*

An array variable can be initialised to anything that can be represented as text - a numerical value as in figure 1 above, text itself, or True/False values. The following example demonstrates setting up an array with some True/False values, and another array with some names.

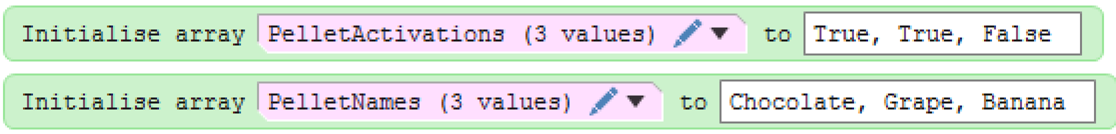


Figure 2. Using the Initialise array *statement to set up the values of an array variable.*

If you need to set up an array to contain something other than numbers, text, or True/False values (for example, positions of a movable zone) then simply enter the names of these as text. ANY-maze can convert names to the relevant zone position when needed.

If you ever change the name of your zone positions in the protocol list, you will need to remember to change the array initialisation in the procedure too!

The *Randomise array* statement

On occasions, the values in an array will need to be initialised to a random, rather than a pre-defined, order. Let's assume that you want the tone frequencies in the above example to be played in a random order; you would use the *Randomise array* statement to put them in a random order:

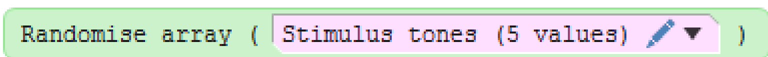


Figure 3. Using the Randomise array *statement to change the values in the array from figure 1 to a random order.*

The resulting array will now contain the values in a random order (e.g. 2000, 5000, 1000, 4000, 3000).

Since the order is changed entirely at random, it is possible (although very unlikely!) that the randomisation process results in the array contents being left in the same order as they started. The larger the array, the less likely this is to happen.

If you want to simply generate a single random number, see the *Random* function under Maths functions and operators.

The *Copy array* statement

This statement copies the contents of one array to another.

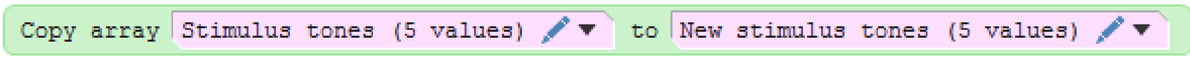


Figure 4. Copying the contents of an array variable into a different array variable.

If the array being copied is larger than the array it's being copied to, then the extra values will be ignored and not copied. If the array being copied is smaller than the one it's being copied to, then the extra array indexes will still contain their previous values.

If the arrays do not both contain the same number of values, then you will get a warning when the procedure is checked for errors. If this is not fixed when writing the procedure, then when the procedure is run, another warning will be listed in the Test details report.

The *Set to #N/A* statement

If you're using a variable within a procedure to measure a specific value, what happens if the thing you're measuring never happens?

For example, you might be measuring the number of times that the animal was grooming while in a specific zone. But what happens if the animal never *enters* the zone? You probably wouldn't want to record the value as zero, because that doesn't tell you whether the animal entered the zone and never groomed itself, or whether it never entered the zone at all.

The value #N/A is used for just such a purpose. N/A stands for 'Not Applicable' and can be used to differentiate a result from zero.

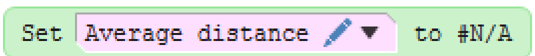


Figure 5. Initialising the value of a variable to #N/A.

You can use the *Is undefined* function to test whether the value of a variable is #N/A.

Don't try and use the value #N/A within a mathematical function - it's not a valid numerical value. If *you do this then the result of the function will be #N/A.*

See also:

- Statements available to procedures
- The Wait until statement
- The Action statement
- The Repeat statement
- The If statement
- The Set value statement
- The Go to statement
- The Label statement
- The Run sub-procedure statement
- The Note statement
- Other statements
- Maths functions and operators

Events available to procedures

Introduction

Events represent things that may occur as the test is run. These could be events due to animal movement (the animal enters a zone, the animal freezes, etc.) or I/O events (a lever was pressed, etc.), or other events such as the test ending.

If you've used previous versions of ANY-maze, these events are similar to the events in the old 'Events and Actions' system.

A basic set of events is listed automatically on the left-hand side of the procedure editor under the *Events* tab; you can set up more complex events by clicking the *Create event* button in the ribbon bar. These user-defined events are usually similar to the basic events, but with an additional time constraint or a count of a number of events occurring.

For example, the basic list of events will include an event for *Animal becomes immobile*, but using the *Create event* button you can set up an event that will occur when the animal has remained immobile for a specific period of time.

There are a great many events in the procedure system, and they are described below. You can also see what is available by expanding all the event groups in the *Events* tab of the procedure editor. Events will be available for all relevant items in the protocol list - zones, points, animal activity, key presses, I/O inputs, and more. Hovering over the relevant event will show a tool tip giving you more information on that event.

An event in a procedure ***must*** *be used within a* ***Wait until statement****. Once the procedure has waited for an event, you can then use it in an* ***If statement****. This would be useful if, for example, you had waited for one of a number of events to happen. However, an event* ***cannot*** *be used in an* If *statement without first being waited for.*

If you try and use an event in an If *statement without first waiting for it, this will result in an error in the procedure editor, and the test will not run.*

Test performance events

Test is waiting to start	This event will fire when the test is waiting to start. This will be when you click the (Start test) button for the apparatus.
Test starts	This event should be used with care. It is not

	normally used in a procedure (since the procedure will *automatically* wait for the test to start, after running the shaded section at the top of the procedure, and before running the un-shaded section at the bottom). However, there *may* be circumstances when you need to use this event in the shaded area at the top of the procedure - for exmaple, waiting for an I/O reading such as a temperature sensor.
Test ends	This event will fire when the test is ended, either manually or by the procedure. The procedure can use an action to end the test.
Test continuation	This event will fire when the user manually forces the test to continue, after it has been ended automatically. This event will only fire if the test was stopped using the action *End the test, but allow continuation* (or *End the test (specifying reason), but allow continuation*).

Timer events

Timer events, or *Time elapsed* events, will fire when a specified time period has elapsed.

Note that the *Time elapsed* event will start timing from the first point that it is encountered by the procedure, and *not* from the start of the test.

If you want to wait for the time elapsed since the beginning of the test, use the Create event *button to open the* **Event trigger wizard** *and select* A specific duration since the start of the test *from the list of* Triggers that relate to time.

Timer events are slightly different to normal events. Since you may have a procedure that waits for multiple *Time elapsed* events to occur (with each one potentially waiting for a different time), each *Time elapsed* event is given a unique numerical ID. So if you wait for a time to elapse, and then use that event in an *If* statement, you can match up the *Time elapsed* event using its timer ID.

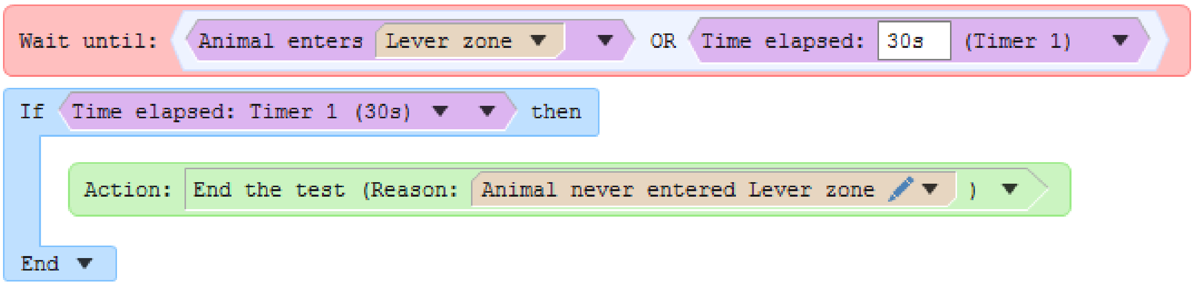


Figure 1. This procedure waits for a specific time to elapse; it can then use that same timer in the following If *statement.*

When entering the time to wait for, you can either type in a valid time (using the units d, h, m, s, ms for days, hours, minutes, seconds and milliseconds) or use an expression that returns a numerical value. If you use an expression (or just type in a number, without any units), then the value will be assumed to be in seconds.

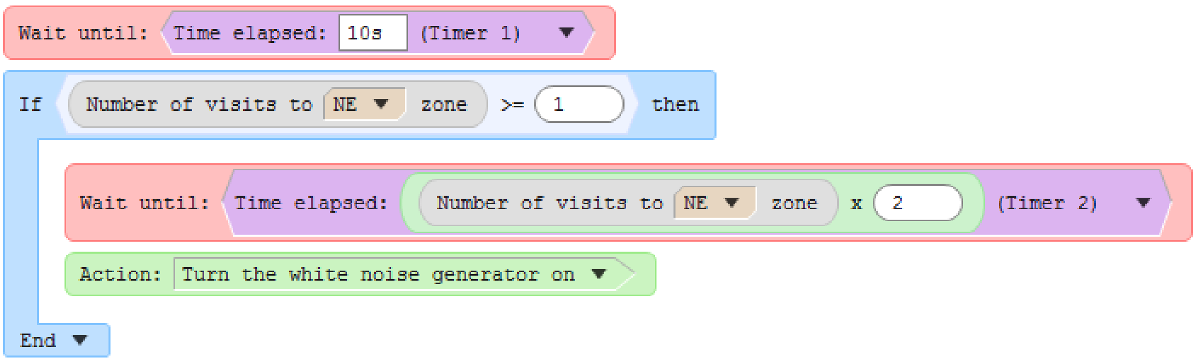


Figure 2. Times can be entered as text, or as a numeric expression, in which case the number evaluated will be assumed to be in seconds.

Zone events

Animal enters zone	This event will fire when the animal enters the zone.

Animal exits zone	This event will fire when the animal exits the zone.
Animal becomes oriented towards centre of zone	This event will fire when the animal becomes oriented towards the centre of the zone. To determine this, the procedure uses the settings made in the protocol's Analysis options > Movement towards and away from zones and points sub-element.
Animal becomes oriented away from centre of zone	This event will fire when the animal is no longer oriented towards the centre of the zone. To determine this, the procedure uses the settings made in the protocol's Analysis options > Movement towards and away from zones and points sub-element.

Point events

Animal becomes oriented towards point	This event will fire when the animal becomes oriented towards the point. To determine this, the procedure uses the settings made in the protocol's Analysis options > Movement towards and away from zones and points sub-element.
Animal becomes oriented away from point	This event will fire when the animal is no longer oriented towards the point. To determine this, the procedure uses the settings made in the protocol's Analysis options > Movement towards and away from zones and points sub-element.

Animal position events

Animal centre position changes	This event will fire when the animal's centre position changes.
Animal head position changes	This event will fire when the animal's head position changes. This event is only available when head and tail tracking is turned on in the protocol.
Animal tail position changes	This event will fire when the animal's tail position changes. This event is only available when head and tail tracking is turned on in the

	protocol.

These events can happen quite frequently during the test; perhaps around 30 times per second.

Mobility events

Animal becomes mobile	This event will fire when the animal becomes mobile.
Animal becomes immobile	This event will fire when the animal becomes immobile.

The protocol's immobility settings are used to determine when the animal becomes mobile or immobile.

Activity events

Animal becomes active	This event will fire when the animal becomes active (either through movement, or a key marked as an activity key is pressed).
Animal becomes inactive	This event will fire when the animal becomes inactive (either through lack of movement, or a key marked as an activity key is released).

Freezing events

Animal freezes	This event will fire when the animal becomes frozen.
Animal unfreezes	This event will fire when the animal becomes unfrozen.

The protocol's freezing detection settings are used to determine when the animal freezes and unfreezes.

Sequence events

Sequence completed	This event will fire when the sequence is completed.

Key press events

Key starts	This event will fire when the key is pressed.
Key stops	This event will fire when the key is released.

On/off input events

On/off input activated	This event will fire when the on/off input is activated.
On/off input deactivated	This event will fire when the on/off input is deactivated.

Signal input events

Signal changed	This event will fire whenever a new value is received from the signal.

This event can occur very frequently, depending on the update rate of the signal - up to 1kHz (1,000 new values every second).

Sensor events

Sensor changed	This event will fire whenever a new value is received from the sensor.

Rotary encoder events

Rotary encoder rotation	This event will fire when the rotary encoder performs a full rotation, either clockwise or anti-clockwise.
Rotary encoder clockwise rotation	This event will fire when the rotary encoder performs a full rotation in the clockwise direction.
Rotary encoder anti-clockwise rotation	This event will fire when the rotary encoder performs a full rotation in the anti-clockwise direction.
Rotary encoder direction reversed	This event will fire when the direction of the rotary encoder is reversed.
Rotary encoder pulse	This event will fire when a pulse of the rotary encoder is detected (in either direction).
Rotary encoder clockwise pulse	This event will fire when a clockwise pulse of the rotary encoder is detected.
Rotary encoder anti-clockwise pulse	This event will fire when an anti-clockwise pulse of the rotary encoder is detected.

Output switch events

Output switch activated	This event will fire when the output switch is activated.
Output switch deactivated	This event will fire when the output switch is deactivated.

Speaker events

Speaker starts playing	This event will fire when the speaker starts playing a sound.
Speaker stops playing	This event will fire when the speaker stops playing a sound.
Speaker reaches the end of the sound file	This event will fire when the speaker reaches the end of the file it is playing.

Analogue output events

Analogue output changed	This event will fire when the value of the analogue output changes.

This event can occur fairly frequently - perhaps 30 times a second.

Temperature controller events

Temperature controller ramp started	This event will fire when the temperature controller starts ramping towards its target temperature.
Temperature controller ramp ended	This event will fire when the temperature controller finishes ramping to its target temperature, i.e. the target temperature is reached.

Lighting controller events

Lighting controller switched on	This event will fire when the lighting controller is switched on.
Lighting controller switched off	This event will fire when the lighting controller is switched off.
Lighting controller light level changes	This event will fire when the lighting

	controller's target light level is changed.
Lighting controller ramp started	This event will fire when the lighting controller starts ramping towards its target light level.
Lighting controller ramp ended	This event will fire when the lighting controller finishes ramping to its target light level, i.e. the target light level is reached.

Syringe pump events

Syringe pump started	This event will fire when the syringe pump is started.
Syringe pump stopped	This event will fire when the syringe pump is stopped.
Syringe pump stalled	This event will fire when the syringe pump stalls.
Syringe pump changed direction	This event will fire when the syringe pump changes direction, from infuse to withdraw (or vice versa).

Shocker events

Shocker activated	This event will fire when the shocker is activated.
Shocker deactivated	This event will fire when the shocker is deactivated.

Laser controller events

Laser controller activated	This event will fire when the laser controller is activated.
Laser controller deactivated	This event will fire when the laser controller is deactivated.

Selector events

Selector value changed	This event will fire when the value of the selector changes.

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Virtual switch events

Virtual switch started	This event will fire when the virtual switch is started.
Virtual switch stopped	This event will fire when the virtual switch is stopped.

User-defined events

If none of the pre-defined events in the list are quite right for you, you can create your own. To do this, use the *Create event* button in the ribbon bar. This will open the Event trigger wizard which will lead you a step at a time through the process of defining the trigger for the event.

User-defined events are mostly events which apply counts or durations to the standard events listed under the *Procedures* tab of the procedure editor, for example:

- A zone entry event is listed in the procedure editor for each zone; however, the Event trigger wizard allows you to set up an event that will occur after a specific *number* of entries to that zone.
- You can set up an event to trigger when the animal *fails* to enter the zone for a specific amount of time.
- Triggering when a sequence is completed a specific number of times.
- There are some handy events for times; for example, triggering repeatedly at certain time intervals, or at a specific time of day.

Once you've set up a user-defined event, it will be available to all procedures in the protocol.

Actions available to procedures

Introduction

There are a great many actions available in the ANY-maze procedure system, and the best way to see what is available is to expand all the action groups in the *Actions* tab of the procedure editor.

Hovering over the relevant action will show a tool tip giving you more information on that action.

Contents

Actions are grouped into the following categories. Click the links to see a description of the actions in each category.

- Test control
- Zones
- Output switches
- Speakers
- Analogue Outputs
- Temperature controllers
- Lighting controllers
- Syringe pumps
- Shockers
- Laser controllers
- Selectors
- White noise
- Video recorder control
- Messages & Text
- Analysis
- Plug-ins
- Run a sub-procedure
- Run a program

Test control

This group contains actions that affect the running of a test.

End the test

This will simply end the test, using the procedure's name as the test end reason.

The test end reason can be used as an independent variable in the analysis of results.

End the test (specifying reason)

This will simply end the test, allowing you to specify the reason (if it is different to the name of the procedure).

The test end reason can be used as an independent variable in the analysis of results.

End the test, but allow continuation

This will end the test, using the name of the procedure as the test end reason, but allows the test to continue if the user restarts it using the *Start test* button.

The test end reason can be used as an independent variable in the analysis of results.

End the test (specifying reason), but allow continuation

This will end the test, allowing you to specify the reason (if it is different to the name of the procedure), and will allow the test to continue if the user restarts it using the *Start test* button.

The test end reason can be used as an independent variable in the analysis of results.

Pause the test

This will pause the test. Note that **no** events will be detected by the procedure until the test is resumed!

There is no way to re-start a paused test from within a procedure - this *must be done manually. This is because once paused, a procedure never gets any events - so you can't use a test event to trigger the un-pausing of a test! However, it* is *sometimes useful to pause the test using a procedure - for example, if the animal falls off an open arm of a Plusmaze, you could use a procedure to notify you via SMS message and then pause the test - then once you've put the animal back in the maze, you would re-start the test manually.*

Prevent test start	Prevents the test from starting. The procedure can use this action to prevent the user from starting the test, if some condition has not been met. For example, a procedure could be used to monitor the temperature of the water in a water-maze, and prevent the test from being started until the temperature is between a range of 20 - 24°C. The parameter to this action is a piece of text that will be displayed above the apparatus image on the Tests page, and should explain to the experimenter why the test cannot be started. This text must be specified. For more details, see Preventing a test from starting until an I/O device is ready.
Allow test start	Allows the test to start again, after starting has been prevented by the *Prevent test start* action. For more details, see Preventing a test from starting until an I/O device is ready.
Set timer resolution	This will set the resolution of the internal timer for the procedure to a value between 1ms and 60s. This affects the frequency at which ANY-maze will attempt to process the events that happen during a test (animal movement, I/O inputs, etc.).

The default timer resolution is 50ms, but it may be that the procedure requires a finer resolution (for example, if you're toggling an output switch on and off for a duration less than 50ms). Setting this timer resolution to a lower value means that the procedure will be checking for new events much more often, thus making the computer work harder and giving less time for it to process other things (for example, analysing video frames).

Since it has the potential to slow down tracking, we advise that you don't change this resolution unless you've been asked to by ANY-maze technical support.

Note that there are some subtleties with ending the test, if you want a procedure to respond to the *Test ends* event. For more details, see Procedures and the end of a test.

Zones

The following actions are available for zones:

Set zone label	Sets a piece of text on the screen over this zone. If the zone consists of more than one non-contiguous area, the label will appear over the largest of these areas.
Remove zone label	Removes the label previously set using the 'Set zone label' action.
Set location of movable zone	Sets the position of the zone. This action can **only** be used *before* the test starts, i.e. it is only valid to drag it into the shaded area at the top of the procedure editor, and not into the white area at the bottom. Using this action after the test starts will result in an error. *This action is only listed if the zone has been set up as a movable zone, and the zone can have its position set by a procedure.*

White noise

The following actions control the white noise generator, which outputs sound via the computer's speakers.

Turn the white noise generator on	Turns on the generation of white noise from the computer's speakers.
Turn the white noise generator off	Turns off the generation of white noise from the computer's speakers.

It's possible for a procedure to turn on the white noise, but then finish without turning it off. If this happens, you can turn the white noise generator off from the I/O page using the *Play noise* button, or adjust its volume using the *Speaker volume* button.

Video recorder control

These actions control the recording of the test to an ANY-maze video (.szd) file. For full details of what is recorded and where these video files will be stored, see What to record while testing.

The available actions are:

Start the video recorder	Starts recording to the video file.
Stop the video recorder	Stops recording to the video file and closes the file.
Pause the video recorder	Pauses recording.
Unpause the video recorder	Un-pauses a previously-paused recording.
Label the video with text	Inserts a 'label' in the video. This is just a marker which you can use when replaying the video to get to a specific point. On ANY-maze's *Video page,* for example, when the video is opened, clicking the *Jump to label* button will show a list of all the markers that have been added to the video. Labelling the video with a specific marker can be useful in a presentation - if you label certain key points in the experiment, you can then quickly and easily jump to that point in the video using these labels.

Messages & Text

The actions in this group are all to do with displaying text, sending messages, or noting debugging information to aid in fixing problems in a procedure.

Issue an alert message	This can be used to issue a message to yourself or another user. There are three ways that the procedure can do this:

- Display a pop-up message on the computer's screen
- Send an e-mail to a specific address
- Send a text message to a given phone number

When you drag this action into a procedure, the Sending an alert message window will open, allowing you to define the text of the alert message, and what kind of alert you would like.

Output text on display

This will show the text on the apparatus display as the test is running. The text will continue to be displayed until either it is changed again by this action, or removed using the *Remove text from display* action.

The text displayed in this way will also be recorded to any video of the test that you are recording.

Remove text from display

This will remove any text previously added to the display using the *Output text on display* action.

Output text to debug window

This will show a line of text on the procedure output debug window for this test.

For full details of the procedure output debug window, see the Debugging procedures topic.

Analysis

These actions allow your procedures to link up to the ANY-maze analysis system.

Set a time marker

This notes a specific marker point in the results of the test.

As a test is performed, ANY-maze tracks the animal and also monitors the I/O ports set up in the protocol. Whenever anything changes (for example the animal's position, or the state of an I/O port), then this is stored in the test's results. Internally to ANY-maze, a 'test event' will be generated. These 'test events' are loosely (but not exactly) what you'll see on the Test data report.

Setting a time marker allows the procedure to insert its own specific labelled marker at any point in this stream of test results, and this marker can then be used in one of the following ways:

- As a reference point for a time period (See Analysis across time).
- As a point for measurement of the baseline for a signal input (See Setting up a signal's baseline).

For full details, see the Time markers topic.

Set variable as a test result

This action is used in conjunction with a Result variable, to set a value which can be used in analysis.

To set a variable as a test result, you must specify in the variable's settings window that it is to be used as a result variable, and that its value will be noted 'only when explicitly set'. Once you've done this, the value will be noted by the procedure whenever this action is used, and this will allow a number of procedure measures to be calculated by ANY-maze results analysis (e.g. minimum value, maximum value, sum of values or count of the number of times it was set).

Plug-ins

This group contains a single action to trigger an Action plug-in.

Trigger Action plug-in

This action is used to trigger an Action plug-in. An ANY-maze plug-in is a third-party piece of code that can be written to perform any task that you like - for example, to log some information, or to trigger some external hardware that can't be controlled using the I/O devices available to ANY-maze.

Run program

This group contains a single action to allow an external program to be run.

Run a program

This action is used to run a separate Microsoft Windows executable or batch file.

When you drag this action into a procedure, the *Run program* window will open, allowing you to define the command-line for the program.

You might use an external program, for example, to run a program that can control some external I/O that is not directly controllable from ANY-maze.

Run sub-procedure

This group contains a single action to allow this procedure to run another procedure (called a 'sub-procedure').

Run sub-procedure	This action is used to run a different procedure. To run another procedure in this way, you must use the Procedure's Settings page to specify that this procedure is a sub-procedure.

See also:

- The Action statement

Variables available to procedures

Introduction

Variables can be used in a procedure to ascertain or keep track of the state of a test. There are various Built-in variables available to you, which you can use to retrieve the state of the test, and you can create your own User-defined variables to keep your own note of the state of a procedure.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

See also:

- Built-in variables
- User-defined variables

Built-in variables

Introduction

Built-in variables are a way for a procedure to be able to ask the ANY-maze software about anything to do with the state of the current test. They can be anything from a simple flag specifying whether the test is currently running, to the animal's current position, to a note of how many times the animal has entered a specific zone.

There are many of these built-in variables available in the ANY-maze procedure system, and they are listed below. You can also see what is available by expanding all the groups in the *Variables* tab of the procedure editor. Variables are available for the apparatus itself, as well as for all relevant items in the protocol list - zones, animal activity, key presses, I/O, etc. Hovering over the relevant variable will show a tool tip to give you more information on that variable.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

Built-in variables are available to a procedure for a number of aspects of the test:

- Test information
- Test date/time
- Animal position and movement
- Zone variables
- Point variables
- Stage and trial variables
- Sequence variables
- Key variables
- On/off input variables
- Signal input variables
- Sensor variables
- Rotary encoder variables
- Output switch variables
- Speaker variables
- Pellet dispenser variables
- Analogue output variables

- Temperature controller variables
- Lighting controller variables
- Syringe pump variables
- Shocker variables
- Laser controller variables
- Selector variables
- Field variables
- Virtual switch variables
- ANY-maze protocol items

More information

Built-in variables can only be read by a procedure, and not changed, so you can't use a *Set value* statement to alter the value of a *built-in* variable.

They can be one of a number of types - this is visible from the shape of the variable in the list in the procedure editor. Shapes with angled edges are True/False values, shapes with rounded edges are numeric values, and squares with cut-off corners are ANY-maze protocol items as set up in the protocol list (e.g. zones, sequences, etc.).

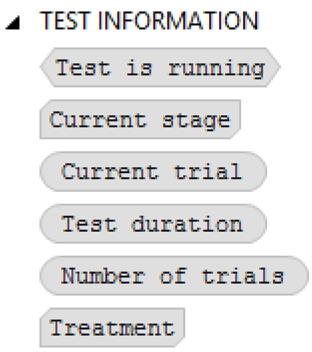


Figure 1. The test information built-in variables, showing their different shapes. Test is running *is a True/False value;* Current trial, Test duration *and* Number of trials *are numeric values, and* Current stage *and* Treatment *are ANY-maze protocol items as set up in the protocol list.*

You can use a built-in variable *anywhere* in a procedure where an expression of the variable's type can be used - for example, in a Maths function or operator, a logical operator, or to set as the new value for a User-defined variable.

The built-in variables available are listed below:

Test information

Test is running	(*True/False*) True if the test is currently running; False if it is not yet running or has been stopped.
Current stage	(*Protocol item*) The stage that the current test is part of. The stages in the protocol are listed as protocol items on the *Variables* tab of the procedure editor.
Current trial	(*Numeric*) The current trial within this stage. The first trial is 1, the second is 2, etc.
Test duration	(*Numeric*) The programmed duration of the test (as specified for the current Stage), in seconds.
Number of trials	(*Numeric*) The number of trials programmed for the current Stage. This may not be the same as the number of trials actually performed, if (for example) the animal completes a 'Training' stage before all programmed trials have been performed.
Apparatus	(*Protocol item*) The apparatus in which the current test is running. The apparatus for the protocol are listed as protocol items on the *Variables* tab of the procedure editor.
Treatment	(*Protocol item*) The treatment applied to the animal for this test. The treatment must have been specified for the animal on the Experiment page, otherwise a run-time error will occur when the procedure attempts to use this variable. The available treatments are listed as protocol items on the *Variables* tab of the procedure editor.
Animal number	(*Protocol item*) The animal number (*not* the Animal ID) of the animal for this test.

Test date/time

Test clock	(*Numeric*) The number of seconds that have elapsed since the start of the test.
Today: day	(*Numeric*) The day of the month on which this test is being performed, from 1 to 31.
Today: month	(*Numeric*) The month of the year on which this test is being performed, from 1 to 12.
Today: year	(*Numeric*) The calendar year in which this test is being performed.

Today: day of week	(*Numeric*) A numeric representation of the day of the week on which this test is being performed. 1 = Monday, 2 = Tuesday, up to 7 = Sunday.
Current time: hour	(*Numeric*) The 'hour' part of the current time of day at which the variable is read, from 0 to 23.
Current time: minute	(*Numeric*) The 'minute' part of the current time of day at which the variable is read, from 0 to 59.
Current time: second	(*Numeric*) The 'second' part of the current time of day at which the variable is read, from 0 to 59.

Animal position and movement

Animal centre (X) as % of width	(*Numeric*) The X position of the animal's centre point, as a percentage of the width of the apparatus. The apparatus dimensions that are used for this value are the smallest square that completely encompasses the apparatus map defined in the protocol.
Animal centre (Y) as % of height	(*Numeric*) The Y position of the animal's centre point, as a percentage of the height of the apparatus. The apparatus dimensions that are used for this value are the smallest square that completely encompasses the apparatus map defined in the protocol.
Animal centre (X)	(*Numeric*) The X position of the animal's centre point, in pixels relative to the left-hand side of the apparatus image. This value is the same as that shown for the animal's X position on the Test details report.
Animal centre (Y)	(*Numeric*) The Y position of the animal's centre point, in pixels relative to the top of the apparatus image. This value is the same as that shown for the animal's Y position on the Test details report.
Animal centre time	(*Numeric*) The time at which the position of the animal's centre point was last measured. This value is in ms since the start of the test.
Animal head (X) as % of width	(*Numeric*) The X position of the animal's head, as a percentage of the width of the apparatus. The apparatus dimensions that are used for this value are the smallest square that completely encompasses the apparatus map defined in the protocol. This value is only available when head and tail tracking is turned on in the protocol.
Animal head (Y) as % of height	(*Numeric*) The Y position of the animal's head, as a percentage

	of the height of the apparatus. The apparatus dimensions that are used for this value are the smallest square that completely encompasses the apparatus map defined in the protocol. This value is only available when head and tail tracking is turned on in the protocol.
Animal head (X)	(*Numeric*) The X position of the animal's head, in pixels relative to the left-hand side of the apparatus image. This value is the same as that shown for the head's X position on the Test details report. This value is only available when head and tail tracking is turned on in the protocol.
Animal head (Y)	(*Numeric*) The Y position of the animal's centre point, in pixels relative to the top of the apparatus image. This value is the same as that shown for the head's X position on the Test details report. This value is only available when head and tail tracking is turned on in the protocol.
Animal head time	(*Numeric*) The time at which the position of the animal's head was last measured. This value is in ms since the start of the test. This value is only available when head and tail tracking is turned on in the protocol.
Animal tail (X) as % of width	(*Numeric*) The X position of the animal's tail, as a percentage of the width of the apparatus. The apparatus dimensions that are used for this value are the smallest square that completely encompasses the apparatus map defined in the protocol. This value is only available when head and tail tracking is turned on in the protocol.
Animal tail (Y) as % of height	(*Numeric*) The Y position of the animal's tail, as a percentage of the width of the apparatus. The apparatus dimensions that are used for this value are the smallest square that completely encompasses the apparatus map defined in the protocol. This value is only available when head and tail tracking is turned on in the protocol.
Animal tail (X)	(*Numeric*) The X position of the animal's tail, in pixels relative to the left-hand side of the apparatus image. This value is the same as that shown for the tail's X position on the Test details report. This value is only available when head and tail tracking is turned on in the protocol.
Animal tail (Y)	(*Numeric*) The Y position of the animal's tail, in pixels relative to the top of the apparatus image. This value is the same as that shown for the tail's X position on the Test details report. This value is only available when head and tail tracking is turned on in the protocol.

Animal tail time	(*Numeric*) The time at which the position of the animal's tail was last measured. This value is in ms since the start of the test. This value is only available when head and tail tracking is turned on in the protocol.
Animal's speed	(*Numeric*) The animal's current speed, in m/s.
Animal is mobile	(*True/False*) True if the animal is currently mobile; False otherwise. This variable is only available if the protocol is set up to detect periods when the animal is immobile.
Animal is active	(*True/False*) True if the animal is currently active; False otherwise. An animal is considered to be active if it is mobile (assuming mobility is being detected) *or* if it's performing some other behaviour which has been defined to be an activity (for example, a key press).
Animal is frozen	(*True/False*) True if the animal is currently frozen; False otherwise. This variable is only available if the protocol is set up to detect freezing.
Freezing score	(*Numeric*) This value represents the animal's 'freezing score', which is used to determine whether the animal is freezing. This variable is only available if the protocol is set up to detect freezing.
Most recent on/off input	(*Numeric*) On/off inputs in ANY-maze can have index values specified, which allows the sequence of activations to be analysed (this can be useful, for example, to detect a sequence of activations of doors within the RAPC apparatus). This variable will return the index of the most recently activated on/off input - of course, for this to work, there must be at least one on/off input in the protocol set up with an index.

Zone variables

Animal is in zone	(*True/False*) This variable is True when the animal is in the zone, and False if it is not.
Distance from Zone	(*Numeric*) The distance of the animal's centre point from the zone's border (in metres). If the animal is currently *in* the zone, then this variable will be 0.
Head distance from Zone	(*Numeric*) The distance of the animal's head from the zone's border (in metres). If the animal's head is currently *in* the zone, then this variable will be 0. This value is only available when head and tail tracking is turned on in the protocol.
Distance from border	(*Numeric*) The distance of the animal's centre point from the

	zone's border (in metres), if the centre is *within* the zone. If the animal is *not* within the zone, then this function returns 0.
Head distance from border	(*Numeric*) The distance of the animal's head from the zone's border (in metres), if the head is *within* the zone. If the animal's head is *not* within the zone, then this function returns 0. This value is only available when head and tail tracking is turned on in the protocol.
Number of visits to zone	(*Numeric*) The number of separate occasions that the animal has entered this zone so far in the test. If the animal has never entered the zone, this value will be 0.
Is animal oriented towards zone	(*True/False*) True if the animal is currently oriented towards the zone. To determine this, the procedure uses the settings made in the protocol's Analysis options > Movement towards and away from zones and points sub-element.
Location of Zone	(*Protocol item*) This variable is only available for movable zones, and gives the location of the zone for this test. The possible locations of the movable zone are listed as protocol items on the *Variables* tab of the procedure editor.

Point variables

Distance from point	(*Numeric*) The distance of the animal's centre from the point (in metres).
Head distance from point	(*Numeric*) The distance of the animal's head from the point (in metres). This value is only available when head and tail tracking is turned on in the protocol.
Is animal oriented towards point	(*True/False*) True if the animal is currently oriented towards the point. To determine this, the procedure uses the protocol's Analysis options > Movement towards and away from zones and points sub-element.

Stage and trial variables

Last trial for stage	(*Numeric*) The last trial for this animal in the given stage. If the animal has not yet completed the stage, this variable's value will be undefined.

Sequence variables

Duration of sequence	(*Numeric*) The amount of time, in seconds, that the animal took to perform this sequence the last time it was completed.

If the sequence has never been completed, this value will be 0.

Key variables

Key is pressed	(*True/False*) True while the key is pressed, False otherwise.

On/off input variables

On/off input is active	(*True/False*) True if the on/off input is currently active, False otherwise.
Most recent on/off input	(*Numeric*) The index of the most recently activated on/off input. This will only be available if you've specified an index value for any on/off inputs in the protocol; if so, then this value will be the index specified for the on/off input that was last activated. If no inputs have yet been activated, this value will be -1.

Signal input variables

Signal value	(*Numeric*) The current value of the signal. The units for this value will depend on the units that have been defined in the set-up of the signal.

Sensor variables

Sensor value	(*Numeric*) The current value of the sensor. The units for this value will depend on the sensor.

Rotary encoder variables

Rotary encoder RPM	(*Numeric*) The rotational velocity of the rotary encoder, in RPM.
Pulses per revolution	(*Numeric*) The number of pulses per revolution of this rotary encoder. This value is set for a specific rotary encoder when an I/O device is configured.

Output switch variables

Output switch is active	(*True/False*) True if the switch is currently active; False otherwise.

Speaker variables

Speaker is playing	(*True/False*) True while the speaker is playing a sound; False otherwise.
Speaker source	(*Protocol item*) The currently selected audio source for the speaker (File, Tone or White Noise). The available speaker sources are listed as protocol items on the *Variables* tab of the procedure editor.
Speaker volume	(*Numeric*) The current volume of the speaker, as a percentage of full volume.
Speaker frequency	(*Numeric*) The current frequency of the tone to play from the speaker, in Hz. Note that if the speaker source is not set to be 'Tone', then this might not be the sound that is currently playing.
Speaker duration	(*Numeric*) The duration for which the speaker will play a sound, in seconds.
Speaker files repeat	(*True/False*) True if audio files played through the speaker can repeat; False if they are played only once.

Pellet dispenser variables

Pellet dispenser is busy	(*True/False*) True if the pellet dispenser is currently dispensing a pellet, and hs not finished. This is only relevant for Ugo Basile pellet dispensers.

Analogue output variables

Output value	(*Numeric*) The current level of the analogue output, in Volts.

Temperature controller variables

Controller is ramping	(*True/False*) This will be True if the temperature controller is currently ramping to its target temperature; False otherwise.
Controller temperature	(*Numeric*) The current temperature of the temperature controller, in degrees (currently only supported by OPAD cages). This may not be the same as its 'target' (programmed) level.
Programmed temperature	(*Numeric*) The programmed temperature of the temperature controller, in degrees (currently only supported by OPAD

cages).

Programmed ramp duration	(*Numeric*) The programmed duration of the ramping of the temperature of the temperature controller, in seconds (currently only supported by OPAD cages).

Lighting controller variables

Controller is on	(*True/False*) True if the lighting controller is currently switched on; False otherwise.
Controller is ramping	(*True/False*) True if the lighting controller is currently ramping to its target light level; False otherwise.
Current light level	(*Numeric*) The current light level of the lighting controller, in lux. This may not be the same as its 'target' (programmed) level.
Programmed light level	(*Numeric*) The programmed light level of the temperature controller, in lux.
Ramp duration	(*Numeric*) The programmed duration of the ramping of the lighting controller, in seconds.

Syringe pump variables

Syringe Pump is running	(*True/False*) True if the syringe pump is currently running; False otherwise.
Syringe Pump direction	(*Protocol item*) The direction of the syringe pump (Infuse or Withdraw). The available directions are listed as protocol items on the *Variables* tab of the procedure editor.
Processed volume	(*Numeric*) The volume processed so far by the syringe pump, in µl.
Target volume	(*Numeric*) The target volume of the syringe pump, in µl.
Flow rate	(*Numeric*) The flow rate of the syringe pump, in µl/min.

Shocker variables

Shocker is active	(*True/False*) True if the shocker is currently active; False otherwise.
Shocker intensity	(*Numeric*) The current intensity of the shock produced by the shocker, in mA. If the shocker is currently inactive, this value will be 0.
Shocker duration	(*Numeric*) The current duration of the shock produced by the

	shocker, in seconds. If the shocker is currently inactive, this value will be 0.
Shocker intensity limit	(*Numeric*) The maximum limit for the intensity of the shock, as specified in the protocol, in mA.
Shocker duration limit	(*Numeric*) The maximum limit for the duration of the shock, as specified in the protocol, in seconds.

Laser controller variables

Controller is active	(*True/False*) True if the laser controller is currently active; False otherwise.
Controller is a pulse train	(*True/False*) True if the laser controller is currently active, and outputting a 'pulse train' signal; False if it is off, or outputting a continuous (DC) signal or a pulse sequence defined in a pulse train file.
Controller is continuous	(*True/False*) True if the laser controller is currently active, and outputting a continuous (DC) signal; False if it is off, or outputting a pulse train or a pulse sequence defined in a pulse train file.
Controller is playing a pulse train file	(*True/False*) True if the laser controller is currently active, and outputting a pulse sequence defined in a pulse train file; False if it is off, or outputting a pulse train or a continuous (DC) signal.
Controller frequency	(*Numeric*) The current frequency of the laser controller, in Hz. This frequency is only used when the laser controller is outputting a pulse train.
Controller duty cycle	(*Numeric*) The current duty cycle of the laser controller, in percent. This duty cycle is only used when the laser controller is outputting a pulse train.
Controller duration	(*Numeric*) The current duration that the laser controller will be on for, in seconds. A duration of 0 means that the controller will remain active until explicitly turned off.
Controller intensity	(*Numeric*) The current intensity of the laser controller, in Volts.

Selector variables

Selectors are only available on certain types of device (such as the Ugo Basile Operon).

Selector Value	(*Protocol item*) The current value of the selector.
Selector Maximum value	(*Numeric*) The maximum value that the selector can be set to, as a numeric value. This is to allow the procedure to be able

to determine the maximum value in order to cycle through all possible selector values. For more information, see Selector actions.

Although the current value of a selector is returned as a protocol item, *it can also be evaluated as a numeric value. If the current value of the selector is copied into a user-defined variable, then it will be stored as a numeric value. This doesn't cause any problems with other parts of the procedure, because selector values can be set as either a* protocol item *or a numeric value.*

Selector is busy	(*True/False*) True if the selector is currently busy making the change; False otherwise. This is only relevant for certain types of selector (the Revolver in an Ugo Basile Operon, for example).
Selector Value Text	(*Text*) The text specified in the protocol for the given numeric value of the selector.

Field variables

Field value	The current value of this field - this will depend on how the field has been set up. If it's a number field, then a *numeric* value will be returned; if it's a text field, the value will be returned as whatever text was entered; if it's a choice field, a *protocol item* will be returned - the possible values are listed as protocol items on the *Variables* tab of the procedure editor.

Virtual switch variables

Virtual switch is active	(*True/False*) True if the virtual switch is currently active; False if it is inactive.

ANY-maze protocol items

Some of the variables listed above return one of a list of 'known values' from the protocol as a return value. For example, the direction of a syringe pump is always 'Infuse' or 'Withdraw', and the location of a movable zone can only be one of a set of predefined locations.

All of the possible values of these, and anything else defined in the protocol, are listed under the *ANY-maze protocol items* sub-heading. These values can then be used anywhere in a procedure - for example, to compare with the return value from a function (using a comparison operator), or to store in a user-defined variable.

ANY-maze help topic H0431

User-defined variables

Introduction

User-defined variables can be used as a kind of temporary storage within a procedure. They can be used to keep track of anything that you want to keep note of, and can be changed in any way throughout the lifetime of the procedure. They can be numerical values, True/False values or text, and can be passed to any function in a procedure that requires a value of that type.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

Example

It's quite difficult to explain exactly what a variable is, so perhaps an example will help. The following procedure shows the use of variables to calculate the time interval between the animal entering a zone and the time it starts grooming.

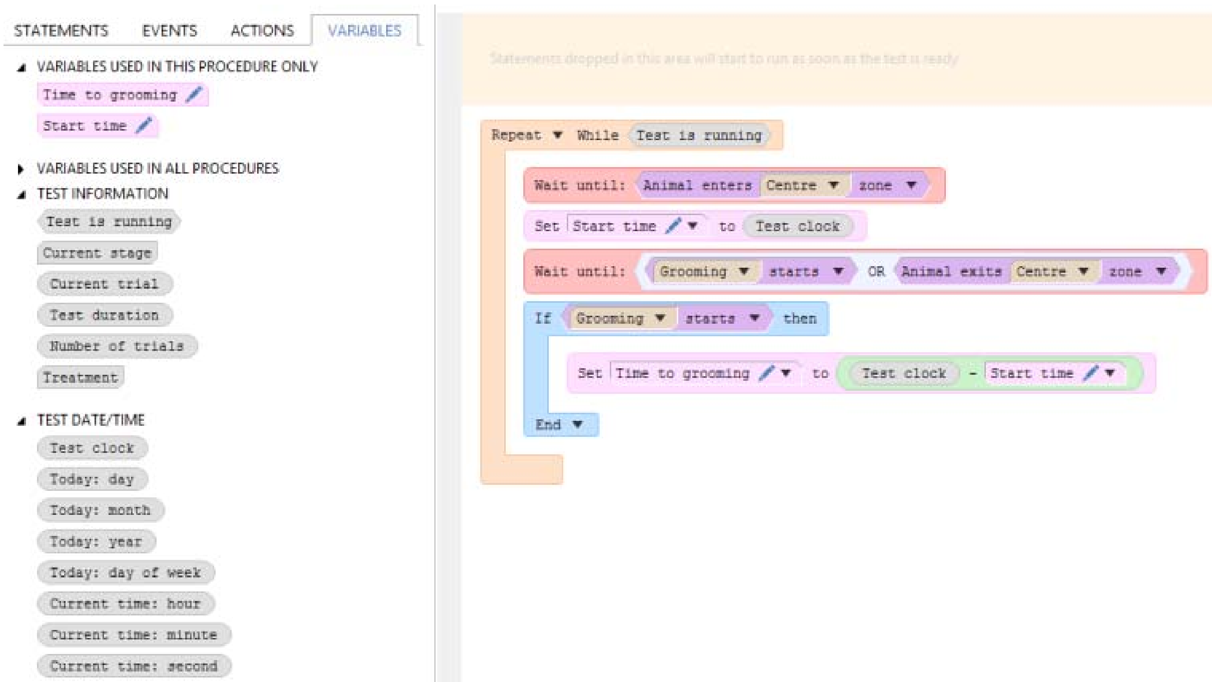


Figure 1. This procedure uses the 'Start time' variable to note the time that the animal enters the zone, then uses this in a calculation to determine the duration.

This example uses two variables - one called 'Start time', which is used to note the time at which the animal enters the zone, and the other called 'Time to grooming' which uses the previously-noted start time, together with the *Test clock* built-in variable to calculate the time it took to start grooming after entering the zone.

Note that if this value was something that could be useful in your test results, you could set the 'Time to grooming' variable up to be a Result variable - see below for further details.

To see more about user-defined variables, click on one of the links below:

- Creating a variable
- Editing a variable
- Deleting a variable
- Using a variable in a procedure
- Sharing variables between procedures

- Sharing variables between trials for an animal
- Sharing variables between tests in an apparatus
- Array variables
- Result variables

Creating a variable

Before you create a variable of your own, it's a good idea to take a look at the list of built-in variables, to see if ANY-maze already reports the information that you want the procedure to keep track of (for example, you might set up a procedure that counts the number of zone entries, but that's unnecessary as there's already a built-in variable that will give you that information).

To create a user-defined variable, click the *Create variable* button on the ribbon bar. A window will open, allowing you to set up the properties of the variable - see the Variable Settings window topic for full details.

When the procedure editor is in 'Simple' mode, the Create variable *button is not shown.*

Once a variable has been created, it will be listed in the Variables tab, and can then be dragged into the procedure to the right location.

Editing a variable

Once you've created a variable, you can edit it using the button on the variable, either under the *Variables* tab on the left-hand side of the procedure editor, or from any of the places you've used the variable within the procedure itself. For full details of the settings that you can edit, see the Variable settings window topic.

Deleting a variable

To delete a variable, you can right click on the variable in the list under the *Variables* tab on the left-hand side of the procedure editor. This will give you the option to *Delete this variable*. Confirming the 'Are you sure?' message will remove the variable from the list under the *Variables* tab, and will also remove the variable from all the places that it is used in the procedure.

Note that this may result in one or more 'holes' being left in the procedure - i.e. the expressions that the variable was used in will now have missing parameters - which will need to be resolved before you can use the procedure.

If you right click *on the variable in the procedure itself, on the right-hand side of the procedure editor, you'll see a drop-down menu with the option to* Delete this parameter. *This will NOT delete the variable itself; it will simply remove it as a parameter to the expression it is being used in.*

Using a variable in a procedure

To use the current value of a variable, simply use the variable as a parameter to the expression that you want. For example, in figure 1, above, the 'Start time' variable is subtracted from the *Test clock* using the - operator.

To set a new value for a variable within a procedure, use the Set value statement.

Note that if the value of a variable is used before it is set, it will have the value of 0 (if used as a (*Numeric*) value), False (if used as a (*True/False*) value), or empty text (if used as a (*Text*) value).

Note that all variables start with a value of 0, so if they are used before being set, the variable will be evaluated as 0. If the variable is used somewhere that expects a (*True/False*) value, the variable will be evaluated as *False,* and a (*Text*) value will be evaluated as '0'.

3Sharing variables between procedures

Variables can be used to store any internal state within a single procedure, but they can also be set up to be shared between procedures. This can be incredibly useful, as it allows you to share state between multiple procedures, and therefore split the procedure logic into smaller, easier-to-understand procedures that all have access to this information.

For example, you might have one procedure that sets a variable's value, and then another that checks the value and takes some appropriate action based on it. The worked example using I/O demonstrates an example of this, so if you haven't done so already, I'd advise following this example through to see how to do this.

The tech-savvy among you might wonder what happens if two procedures try to access the same variable at exactly the same time. In fact, this can't happen - since each procedure runs through a series of statements before the next procedure gets the chance to run, each statement runs in complete isolation and there's no danger of the variable being accessed by two procedures at the same time. However, it is certainly worth considering the order in which each procedure might do things, and how this might affect the value of a shared variable.

To specify that a variable should be shared between procedures, use the *This variable is shared across all procedures run for a test* option in the Variable settings window.

3Sharing variables between trials for an animal

A variable will usually hold its internal state for an individual test - i.e. a specific trial for an animal. However, there are some circumstances under which you may need to know some information about the animal's previous trial - for example, you may decide on the location of a movable zone based on the zone that the animal visited last in its previous trial.

By default, a variable's value is reset at the beginning of every test - what this means is that if the variable is used as a numeric value, it will be reset to zero; if it's a True/False value, it will be set to False; and if it's a text value, it will be set to '0'.

You can, however, choose to keep the variable's value between subsequent tests of the same animal, which means that the variable will *not* be reset at the start of each test and will therefore keep the value that it had at the end of the previous trial for that animal.

To share a variable between subsequent trials, turn on the *Retain the value of this variable between trials/stages* option on the Variable Settings window, and select the *For the current animal* option.

At the beginning of the animal's first trial, the variable will have the same value as any other variable that has not yet been set - if it's used as a numeric value, it will be zero; if used as a True/False value, it will be False; and if it's a text value, it will be '0'.

If you run some tests on an animal, then close ANY-maze to resume the testing in another session, the current value of all variables that are shared between trials will be noted in the experiment file, so that they can be recalled before running further tests.

3Sharing variables between tests in an apparatus

In a similar way to storing a variables's value between trials for an animal, you can also choose to store a variable between tests that run in an apparatus (regardless of which animal the test is run on). This might be useful, for example, if you need to start an animal in a different part of the apparatus to the one that the previous animal finished in.

Choosing to keep the variable's value between subsequent tests in the same apparatus will mean that the variable will *not* be reset at the start of each test and will therefore keep the value that it had at the end of the previous test in that apparatus.

To turn this on for a variable, you should select the *Retain the value of this variable between trials/stages* option on the Variable Settings window, and select the *For the current apparatus* option.

At the beginning of the first trial in the apparatus, the variable will have the same value as any other variable that has not yet been set - if it's used as a numeric value, it will be zero; if used as a True/False value, it will be False; and if it's a text value, it will be '0'.

If you run some tests in an apparatus, then close ANY-maze to resume the testing in another session, the current value of all variables that are shared between tests for that apparatus will be noted in the experiment file, so that they can be recalled before running further tests.

Array variables

Array variables are simply variables that hold a fixed number of values of a given type. This allows you to process multiple values using the same variable, rather than having to create a number of similarly-named variables.

For example, let's say that you have an apparatus with four zones. You want to play a tone when the animal enters each zone, but you want the time delay to be different for each zone.

Instead of setting up four variables to represent the time delay for each zone, you could set up a single *array* variable with four values. These four values would represent the time delay for each of the four zones:

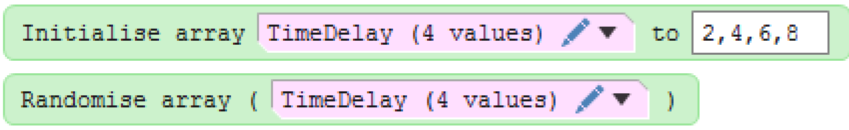


Figure 2. Setting up the array to contain four values. The values are initialised and then randomised, so they're in a different order.

You then use each of the values in the array variable by referring to it by *index*. The four values in this array are accessed by indexes 1 to 4.

For example, you could use the array variable set up in figure 2 above as follows.

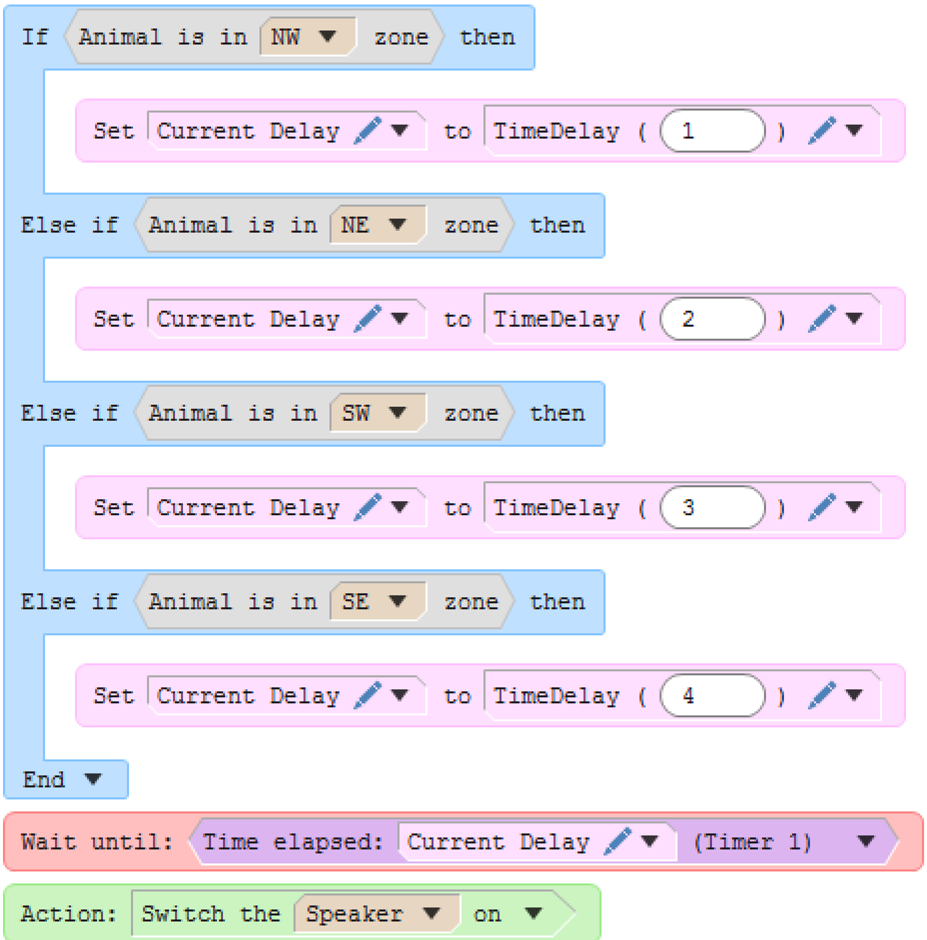


Figure 3. A section of the procedure that uses the array variable from figure 1. Note that the part of the procedure that waits for the animal to enter one of the

four zones has been omitted.

This part of the procedure shows that different values in the array can be accessed using the indexes 1, 2, 3 and 4. The relevant value is then copied into a variable with a single value ('Current Delay') and this variable is then used in the *Time elapsed* event, to wait for the relevant time for the current zone.

Result variables

You may have a variable in a procedure that contains a value that you would find useful in the analysis of a given experiment - either as a single result value for a test, or to make note of a minimum, maximum or average value. In this case, you could set up a variable to be used as a test result, and use this in analysis.

In the example in figure 1 above, you could edit the 'Time to grooming' variable to be a result variable, and specify that it's to be noted 'every time it is changed by the procedure'. This would then make the average, minimum, maximum, count and sum of these values to be available as procedure measures, i.e. available in the test's results.

For more details about result variables and how to use them, see Creating and using result variables.

See also:

- Built-in variables
- Result variables
- The Variable Settings window

Creating and using result variables

Introduction

Variables are useful within a procedure to maintain internal state and help in the running of the procedure, but what if you're using a variable to calculate some value that you'd like to use in your analysis?

In this case, you would set up the variable to be used as a 'result variable'. This would cause the value of the variable to be treated as a test result, meaning it would appear in all the places that results are normally available – for example, on the Results page and Data page.

Example

To make this clearer, here's an example. Imagine you want to know the duration of the animal's second visit to a specific zone (let's not worry for the moment about *why* you might want to know this!). There's no standard result in ANY-maze that could tell you this, but you could easily calculate it in a procedure. You'd wait until the animal enters the zone for the second time, note the time this occurs in a variable (call it Time1), then wait until the animal leaves the zone and calculate the difference between the time now and Time1, which would give you the 'Second visit duration'. You could store this value in a variable of that name.

Here's what the procedure could look like:

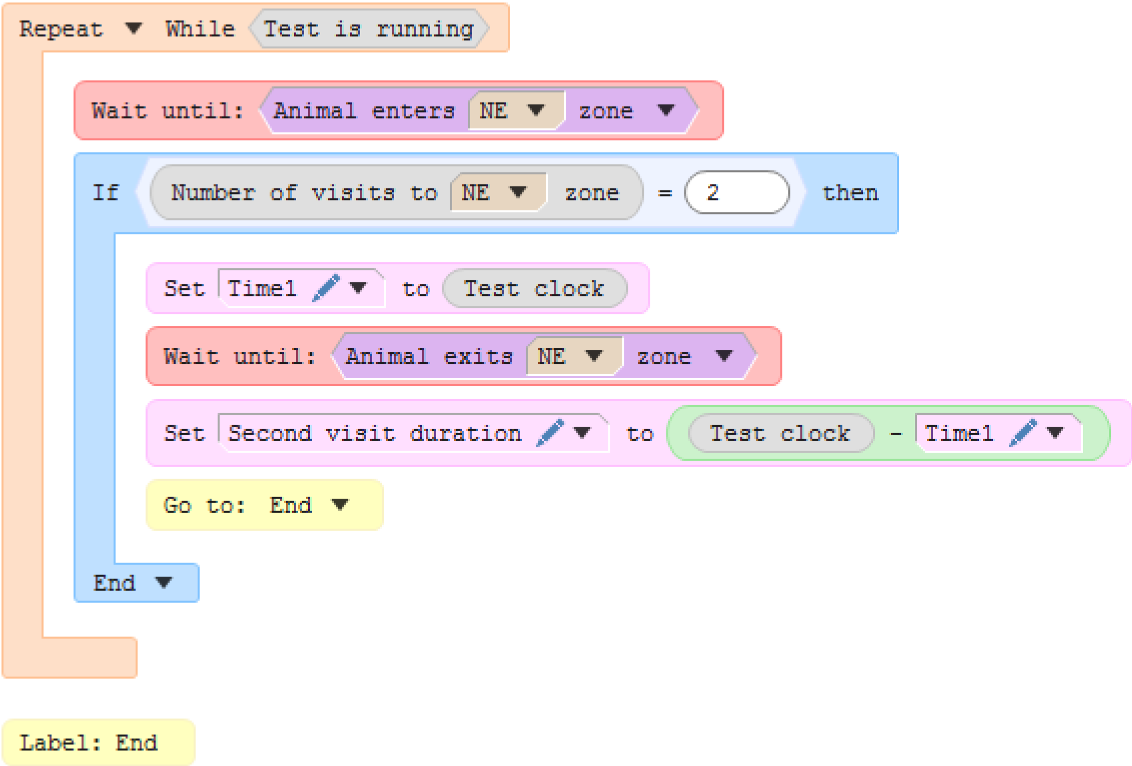


Figure 1. The procedure to measure the time the animal spends in a zone on its second visit. This value is saved in the 'Second visit duration' variable.

But, if that's all you did then the procedure system would know the 'Second visit duration', but the rest of ANY-maze wouldn't. However, if you set the 'Second visit duration' to be a **result variable,** then it would appear in the result measure lists throughout the system – see figure 1. In this way, the value could be analysed, reported, exported, etc. - just like any other result.

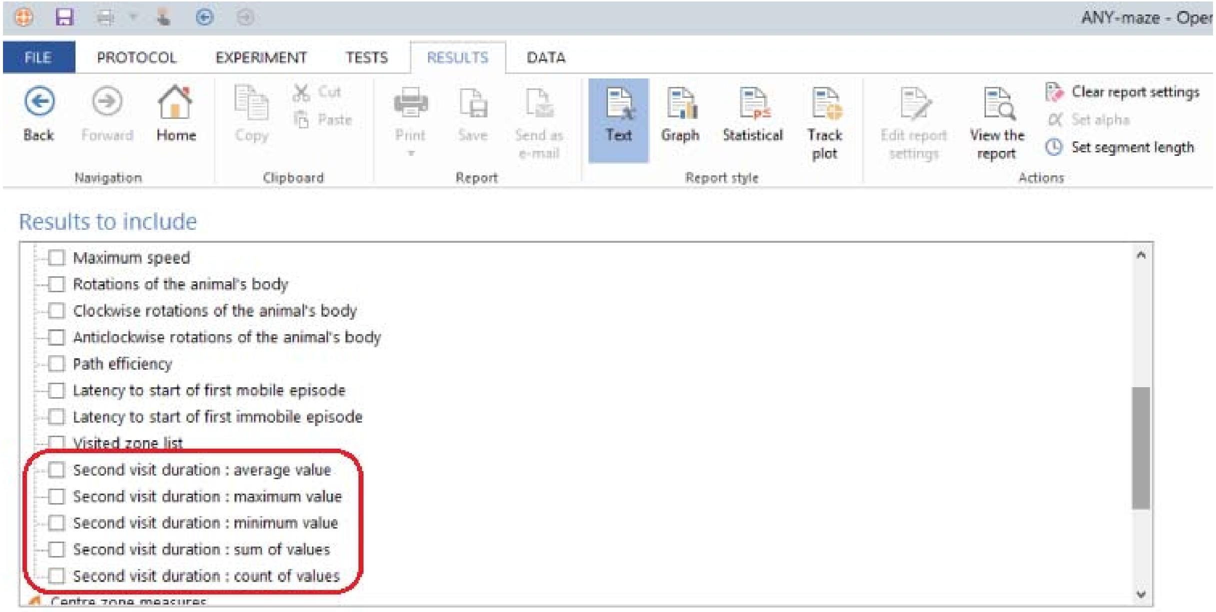


Figure 2. The 'Second visit duration' variable described above, visible in the list of measures on ANY-maze's Results page.

Setting up a variable as a result variable

To set up a variable as a result variable, you need to edit the variable's settings (using the button on the variable in the procedure editor) to open the Variable settings window.

Select *This variable will be used as a result value for this test*, and then choose when you want the value of the variable to be recorded. You have three options here:

- *Every time it is changed by the procedure*

  Every time the variable is changed by the procedure, using the Set value statement, the value will be noted.

- *Only once, at the end of the test*

  Whatever value the variable has when the test ends, will be used as a test result.

- *Only when explicitly set using the 'Set variable as a test result' action*

  In order to note the value of the variable, you will need to explicitly use the *Set variable as a* *test result action*

The second of these options ('only once...') will simply make a note of the value of the variable once, when the test ends. Since there's only one value, then the only measure available for analysis will be 'value'.

The first and last of these options will potentially result in the variable's value being noted several times during a test. This will make more measures available - maximum, minimum, average, etc.

For more details of the results available, see Procedure measures.

More information

The current value of a result variable can be set as a column in the Test data report. This is done using the Test data report option under the *Results, reports and data* section of the protocol list.

Limitations

A variable can only be used as a test result if it has a numeric value. Variables that have been set to True/False, or that contain text, cannot be used as test results.

See also:

- The variable settings window
- Procedure measures
- User-defined variables

Logical operators

Introduction

Logical operators allow the procedure to check for multiple events or conditions occurring. A logical operator takes two 'parameters' (i.e. things whose True/False values are checked), and returns True or False depending on whether any or all of the parameters themselves evaluate to True or False.

Logical operators can be one of the following:

Operator	Description
AND	True if BOTH its parameters are true.
OR	True if EITHER of its parameters are true.

Example

The following example will wait for one of two different events to happen. If *either* of these events occurs, the procedure will continue and use the *If* statement to determine which event it was.

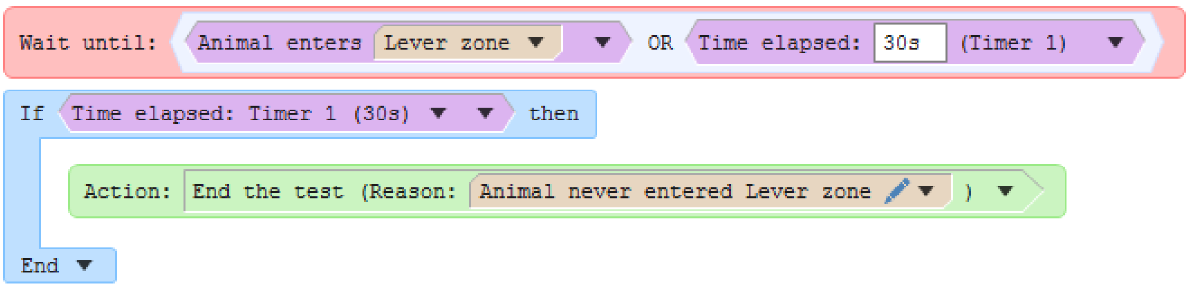


Figure 1. This procedure will wait for either the animal to enter the zone, OR the specified time to elapse. If EITHER of these occur, the procedure will move on.

More information

Logical operators can be 'nested', to allow more than two conditions to be tested at a time. This means that you can use a logical operator as one of the parameters of another logical operator. Any number of logical operators (or other True/False expressions) can be nested in this way.

Figure 2. This example shows an OR *operator nested inside an* AND *operator.*

If you realise that you've used *AND* instead of *OR*, or vice versa, you can quickly and easily change the operator in the procedure editor by right clicking on the operator and selecting *Change operator* from the resulting drop-down menu. This will change the operator without affecting either of its parameters.

See also:

- Elements of a procedure

Comparison operators

Introduction

Comparison operators allow a procedure to compare two different values. The two values of the operator are the 'parameters' to the operator.

Comparison operators can be one of the following:

Operator	Description
=	True if the left-hand parameter equals the right-hand parameter
not =	True if the left-hand parameter is NOT equal to the right-hand parameter
<	True if the left-hand parameter is less than the right-hand parameter
<=	True if the left-hand parameter is less than OR equal to the right-hand parameter
>	True if the left-hand parameter is greater than the right-hand parameter
>=	True if the left-hand parameter is greater than OR equal to the right-hand parameter

Example

The following example compares the value of a variable to 1. If the variable's value is greater than or equal to 1, the procedure will perform an action.

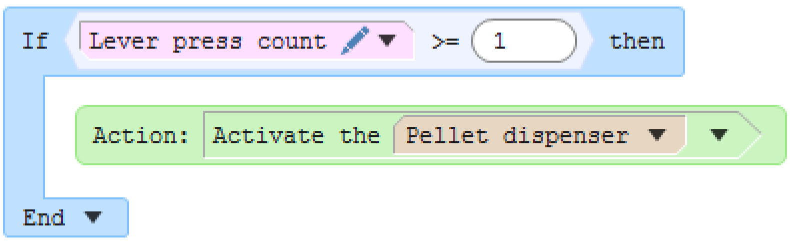


Figure 1. In this example, the value of the 'Lever press count' variable is compared to 1; if greater than or equal to 1 (i.e. if the lever has ever *been pressed) then the procedure will activate the 'pellet dispenser' switch.*

More information

These operators can accept any expression (including those that are made up of other expressions themselves). When both parameters of a mathematical operator are numeric, the result of these operations should be obvious. In fact most of these operators (>, >=, <, <=) can *only* be used with numerical expressions.

However, an expression in a procedure can be either numeric, True/False, or a text value. If the expression contains a user-defined variable, then it's possible that the resulting type of the expression is not known until the procedure actually runs. In this case, ANY-maze will try to evaluate the expression as a number when it encounters it in the procedure. In order to do this, it will use 1 and 0 for True and False, and will attempt to convert any text expressions to a number (this is unlikely to succeed, unless the text expression obviously represents a numeric value, for example '25' or '3.14159265').

The 'equals' (=) and 'not equals' (*not* =) operators can be used with *any* type of expression. To be equal, the expressions must both be of the same type (i.e. both numeric, both True/False, or both text) as well as evaluate to the same value.

In this way, the 'equals' (=) and 'not equals' (*not* =) operators can be used to compare built-in variables, constants, user-defined variables, and the *ANY-maze protocol items* found at the bottom of the *Variables* tab.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

The following example shows the 'equals' comparison operator being used to determine the current position of a 'known' arm in a Y-maze.

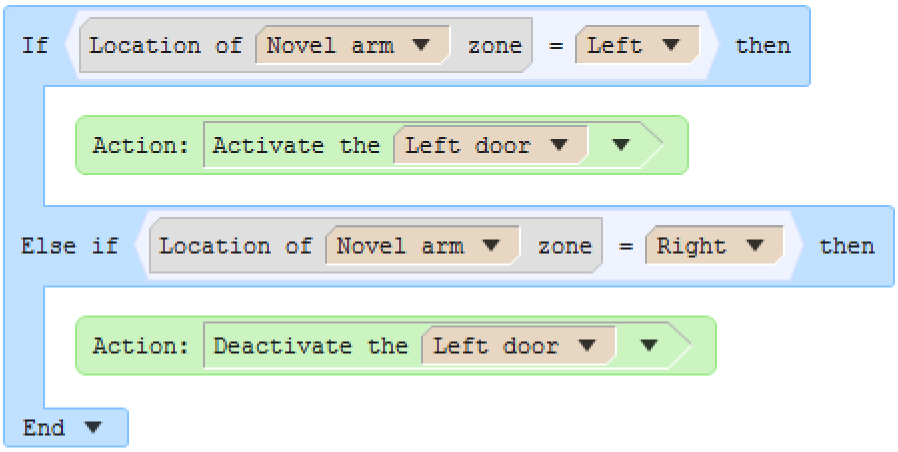


Figure 2. Using the 'equals' comparison operator to determine the current location of a movable zone.

If you change your mind about which operator you're using (for example, you want to change 'greater' (>) to a 'greater than or equals' (>=), just right-click on the operator in the procedure editor. You'll get a drop-down menu which includes the *Change operator* option; selecting the new operator will change the operator while leaving both of its parameters intact.

Constants

Introduction

The following constants can be used in comparison operators and other expressions within a procedure:

Constant	Description
True	Can be used to see if the result of an expression is true
False	Can be used to see if the result of an expression is false
Pi	The numerical value representing the ratio of a circle's circumference to its diameter (approximately 3.14159265)

Example

The following example uses the constant *True* in a comparison.

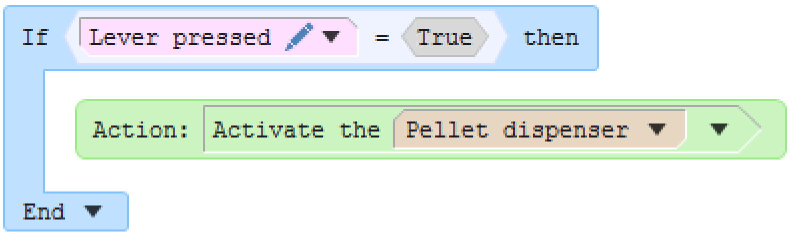


Figure 1. Using the constant True *to check against the current value of a variable.*

More information

These constants can be used almost anywhere in a procedure where an expression of the relevant type is required, for example:

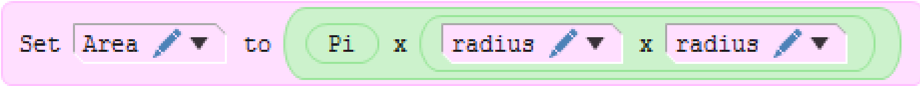


Figure 2. Using the constant Pi *to calculate the area of a circular arena.*

Maths functions and operators

Introduction

ANY-maze includes a number of mathematical functions and operators that allow a procedure to perform calculations.

These functions and operators all take one or two *parameters*, i.e. values that are used by the function or operator. For example, the + operator takes two parameters (the two values that are being added together) and the *Sqrt* function takes one parameter (the value whose square root is being calculated).

Comparison operators can be one of the following:

Function/Operator	No. parameters	Description
+	2	Gives the result of adding the left-hand parameter to the right-hand parameter.
-	2	Gives the result of subtracting the right-hand parameter from the left-hand parameter.
x	2	Gives the result of multiplying the left-hand parameter by the right-hand parameter.
/	2	Gives the result of dividing the left-hand parameter by the right-hand parameter.
Mod	2	Gives the remainder, when the left-hand parameter is divided by the right-hand parameter.
Sqrt	1	Gives the square root of the parameter.
Pow	2	Gives the first parameter to the Nth power, where N is the second parameter.
Abs	1	Gives the absolute value of the parameter, i.e. if the parameter is negative, it returns the value with the negative sign removed.
Log	1	Gives the base 10 logarithm of the parameter.
Ln	1	Gives the natural logarithm (base e) of the parameter.
Sin	1	Gives the sine of the parameter, which is assumed to be an angle in degrees.
Cos	1	Gives the cosine of the parameter, which is assumed to

		be an angle in degrees.
Tan	1	Gives the tangent of the parameter, which is assumed to be an angle in degrees.
Sin-1	1	Gives the inverse sine of the parameter, i.e. the angle for which the parameter is the sine.
Cos-1	1	Gives the inverse cosine of the parameter, i.e. the angle for which the parameter is the cosine.
Tan-1	1	Gives the inverse tangent of the parameter, i.e. the angle for which the parameter is the tangent.
Round	2	Rounds the first parameter to the number of decimal places specified by the second parameter.
Random	2	Generates a random number between the values specified by the two parameters. The second parameter must be greater than the first parameter.

To randomise the contents of an array, see the Randomise array *statement in the Other statements topic.*

Example

The following section taken from a procedure uses some of the operators and functions listed above to perform a simple calculation of distance, using built-in variables for the animal's current centre point X and Y, and user-defined variables to aid in the calculation. These user-defined variables have been noted from the animal's centre point earlier in the procedure.

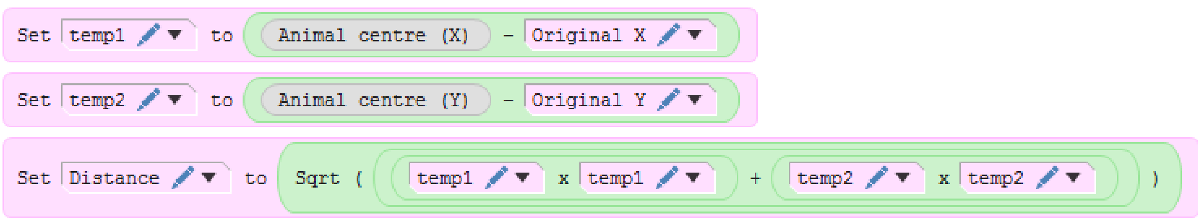


Figure 1. This example uses the +, - *and* x *operators, together with the* Sqrt *function, to calculate a distance.*

More information

The parameters to mathematical functions and operators can either be simple numeric values, or they can be any expression that evaluates to a numerical value (for example, another mathematical function from the list above).

If any of the parameters of one of these mathematical functions evaluates to #N/A, the result of the function will also be #N/A.

You will often want to use the value of a User-defined variable as a parameter to one of these functions, but since the procedure editor doesn't know what the value of a variable will be until it runs, it is quite possible that when the procedure tries to evaluate the mathematical operator or function, the variable does not contain a numerical value.

If a non-numerical value is passed to one of these mathematical functions or operators, it will evaluate as #N/A. This is shown in figure 2.

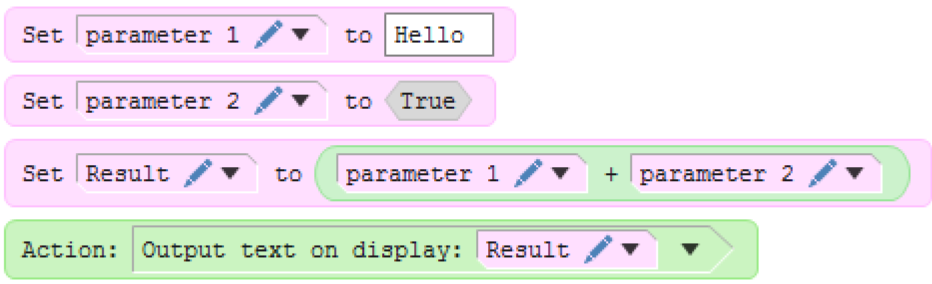


Figure 2. This section of a procedure is valid to write, but the values that are used in the + operator are the text "Hello" and the constant True. These can't be added together, so the + operator evaluates to #N/A, and the text displayed on the screen is "#N/A". A warning will also be generated.

Where possible, ANY-maze will try and evaluate a variable as a numerical value for all the functions and operators listed above. If the variable contains True or False, it will evaluate to 1 or 0 respectively. If it contains text, then the procedure can only successfully extract a numerical value if the text is obviously a number, for example '5' or '-1.234'

In the example above, if the 'parameter 1' variable were set to *False*, then the + operator would evaluate to True + False, i.e. 1 + 0, so the value assigned to the 'Result' variable would be 1.

A common mistake when entering parameters to functions like this is to type in the name of the variable, rather than dragging the variable into the procedure. If you do this, you'll get an error like the following:

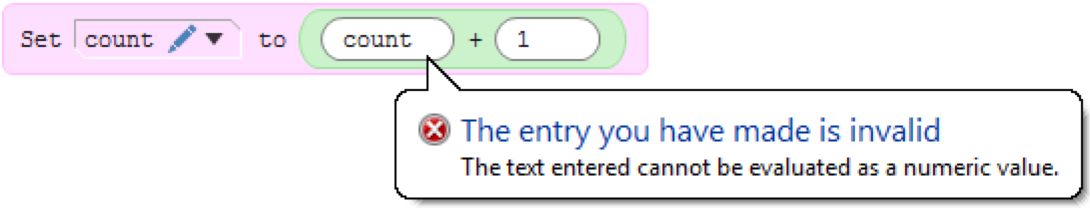


Figure 3. A procedure statement where the name of the variable has been typed into a + operator, rather than dragging in the variable from elsewhere. A warning message is shown.

If you've used one of the mathematical operators (+, -, *x* or */*) and you realise you need to change the one you're using, (for example, you want to change *x* to +), just right-click on the operator in the procedure editor. You'll get a drop-down menu which includes the *Change operator* option; selecting the new operator will change the operator while leaving both of its parameters intact.

Other functions

Introduction

There are a couple of esoteric functions that may be useful when writing procedures:

- 'Is undefined' function
- Join function
- Size of array function
- Sum array values function

The *Is undefined* function

- There are a number of functions and operators within the ANY-maze procedure system which return a numeric value. If any parameters to these are not valid, then the function or operator will return #N/A.

The *Is undefined* function can be used to check whether a variable's value is undefined (i.e. set to this '#N/A' value).

The *Join* function

The *Join* function can be used to join two pieces of text together.

The *Size of array* function

The *Size of array* function can be used to return the number of slots, or indexes, in an Array variable.

The *Sum array values* function

The *Sum array values* function can be used to return the sum of all numerical values in an Array variable. If there are any non-numeric values in the array, this function will return #N/A.

Note that True/False values are evaluated as 1 and 0, respectively, so you can easily see if there are any True values in an array of True/False values simply by checking for a non-zero return from this function.

Example

The following example uses the *Is undefined* function to check a variable's value, and then the *Join* function to create text to output on the display.

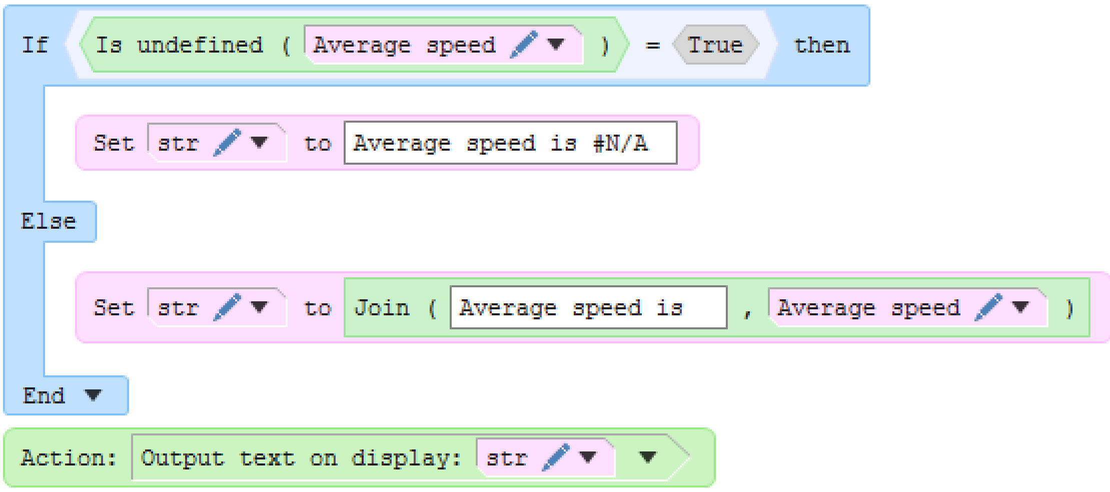


Figure 1. Example using the Is undefined *and* Join *functions.*

Hints and tips for writing procedures

Introduction

Procedures can be very flexible, but this also means that they can be quite complex. This page outlines some things to remember, and gives some tips on how to get the best out of your procedures.

Hints and tips

- Procedures run statements one after the other, then stop.
- A procedure's lifetime is not the same as that of a test.
- Procedures can be made to 'loop' or 'repeat'.
- Procedures usually require a 'Wait until' statement to run properly.
- You can delete a statement from the procedure by dragging it out of the procedure.
- You can delete a parameter from a statement by dragging it out of the statement it is contained in.
- You can select multiple statements at a time, to move, copy, or delete them.
- Split different logical areas into different procedures.
- Some shortcuts to make writing procedures faster.
- Copying and moving items within a procedure.

Procedures run statements one after the other, then stop.

When a procedure runs, it will execute one statement after another, until it reaches a point where it can't execute a statement. This might be because the procedure ends (i.e. there *are* no more statements), or it might be because the procedure has been asked to *Wait until* something happens - and that thing hasn't happened yet.

If the procedure's current statement is a *Wait until* statement, everything that subsequently happens in the test (e.g. animal movement, I/O input, etc.) will be processed against that statement to see if it is relevant. If it's the thing that the procedure is waiting for, then the procedure will be able to continue running its subsequent statements. If not, then the procedure continues to wait for the next thing to occur.

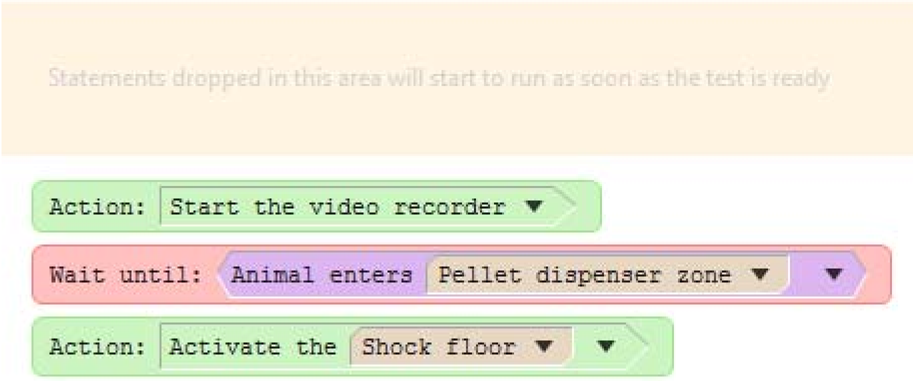


Figure 1. In the procedure above, as soon as the test starts, the procedure statements will start running. The first statement will be run (turning the video recorder on), then the second statement will wait until the animal enters the 'Pellet dispenser' zone. All subsequent events that occur within the test (animal movement, I/O inputs, etc.) will be checked against this wait statement, but ignored until the one occurs that it is waiting for (entry to the relevant zone). Only at that point will the procedure continue running, and will turn on the 'Shock floor' shocker. The procedure will then end.

In the example above, after the shocker is turned on, the procedure will stop running (but the test will continue).

A procedure's lifetime is not the same as that of a test.

Once a procedure has finished running all its statements, it will stop. This doesn't mean that the *test* will stop, though - the test will carry on running until its test duration has elapsed, or until it is ended manually, or by another procedure using the *End the test* event.

If a test ends while a procedure is still running, then the procedure itself will end. The exception to this is if the procedure is actually waiting for the *Test ends* event - in which case, it will receive the event and continue running statements until it reaches the next *Wait until* statement.

Note that there are some subtleties with procedures that wait for the *Test ends* event, particularly regarding test continuation - for full details, see the Procedures and the end of a test topic.

Procedures can be made to 'loop' or 'repeat'.

If you don't want a procedure to finish, but you want it to repeat the same sequence of actions, there are two ways you can do this:

You can use a *Repeat* statement, and enclose the statements that you want to repeat within the loop:

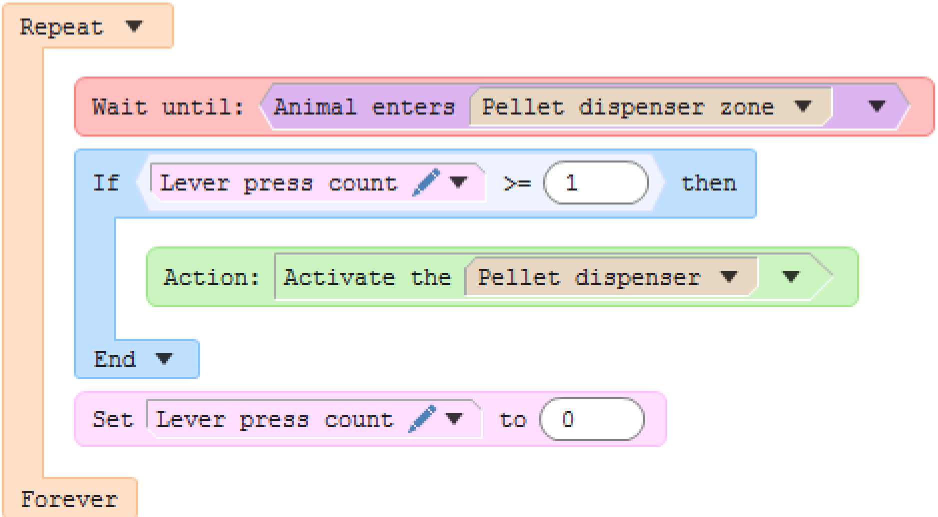


Figure 2. This procedure will continue to loop, forever (or until the end of the test).

Or you can use a *Label*, and then a *Go to* statement to send control back to a specific point in the procedure:

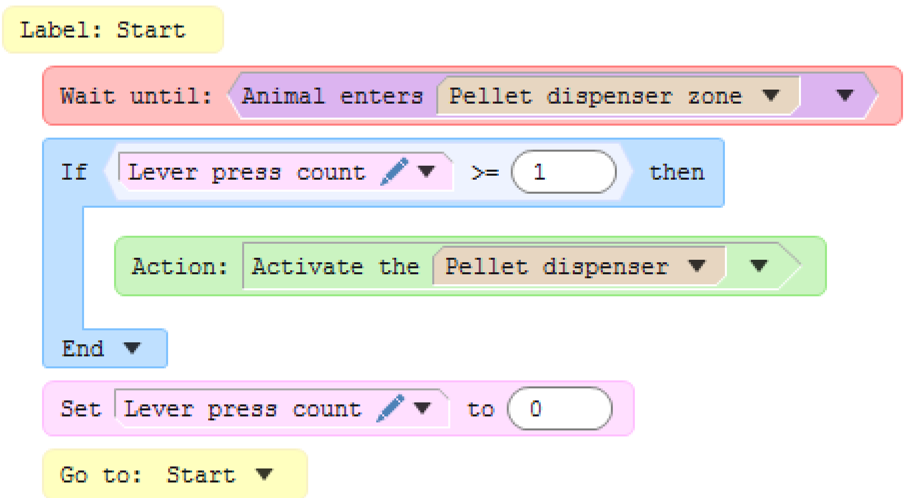


Figure 3. This procedure will behave in exactly the same way as the one in figure 2, above, but uses a label and a Go to *statement, instead of a* Repeat *statement.*

Note that *Label* statements and *Go to* statements can be used to jump forwards as well as backwards in the procedure.

Procedures usually require a Wait until *statement to run properly.*

Since a procedure will just run the statements in the order that it is written until it has to stop, then if a procedure contains a loop of any kind, there is a danger that the procedure will get 'stuck' in a continuous loop. This will prevent other procedures from getting a chance to run themselves. (In fact, ANY-maze tries to detect such a situation, and *does* let other procedures have a chance to run, but this is very inefficient and you should avoid this happening).

The best way to prevent any problems is to ensure that the procedure has at least one *Wait until* statement. It is only when the procedure reaches such a statement that it 'takes a breather' and allows other procedures to run when they need to, or other processing to occur.

For more details of this, see the *Wait until statement topic*.

Note that there are circumstances where it *is* OK to have a procedure without a *Wait until* statement - for example, if you are just running a procedure to set up some I/O at the beginning of a test.

If you write a procedure without a Wait until *statement, you'll get a warning when you check the procedure for errors or exit the procedure editor.*

You can delete a statement from the procedure by dragging it out of the procedure.

The easiest way to delete a statement from a procedure is to drag it off the right-hand pane of the procedure editor, and drop it anywhere else (for example, you can drop it on the left-hand pane of the editor).

You can also right click on the statement itself, and you'll get a menu giving you the chance to delete it.

Lastly, you can press the *delete* key on the keyboard to delete any currently-selected statements (i.e. those highlighted in blue).

You can delete a parameter from a statement by dragging it out of the statement it is contained in.

The easiest way to delete a parameter from a statement is to drag it off the statement, and drop it anywhere in a blank space within the right-hand pane, or anywhere off the pane.

You can also right click on the parameter itself, and you'll get a menu giving you the chance to delete it.

You can select multiple statements at a time, to move, copy, or delete them.

If you want to move, copy or delete multiple statements at a time, you need to select all the statements first.

To do this, you need to move the cursor over the grey 'margin' at the left-hand side of the procedure editor pane - when you do this, the cursor arrow will flip horizontally to point to the right instead of to the left. You can then select multiple statements in one of the following ways:

- Holding down the mouse button and dragging the cursor up or down within the grey margin will select all contiguous statements to the right of the cursor;
- You can use the 'Ctrl' key while you click on individual statements, to select and un-select the statement to the right of the cursor;
- To select a contiguous range of statements, click the mouse button against the first statement, then move the mouse and hold down the 'Shift' key while you select the last statement.

The key presses described in these last two points are the same as those that you would use to select multiple files within the Windows file explorer.

Split different logical areas into different procedures.

Multiple procedures can be run at the same time, which makes it easier to perform complex logic as part of the test. It's advisable to split any complex logic into multiple procedures, as this makes it much easier to understand.

For example, let's say you have a pellet dispenser, but you only want to activate this if the animal has previously pressed a lever. If the animal goes near the pellet dispenser *without* first activating the lever, you want to give them a shock.

The logic for such an experiment can quickly get complicated, such that trying to write this all in a single procedure could get messy - so you'd be better off writing it as two much simpler procedures, where the logic for each one is self-contained, i.e.

- Procedure 1: If the animal presses the lever, note that he's done so in a variable (the variable must be 'shared' between procedures)
- Procedure 2: If the animal goes near the pellet dispenser, check whether he's pressed the lever (by looking at the value of the variable) - if so, then dispense the pellet, otherwise activate the shocker.

This logic is the subject of the third walkthrough example using I/O.

Some shortcuts to make writing procedures faster.

There are a number of shortcuts that you can use to make it quicker and easier to write a procedure:

- Dragging an Action from the *Actions* tab into an empty space in a procedure will automatically create an Action statement, and will insert the action into the statement.
- Dragging an Event from the *Events* tab into an empty space in a procedure will automatically create a Wait until statement, and will insert the event into the statement.
- Dragging a Variable from the *Variables* tab into an empty space in a procedure will automatically create a Set value statement, and will insert the variable as the first parameter of this statement.
- Dragging a Comparison operator from the *Statements* tab into an empty space in a procedure will automatically create an If statement, and will insert the operator as the condition for this statement.

Copying and moving items within a procedure.

Rather than dragging everything from the left-hand side of the procedure editor to the right, you can copy or move existing items within the procedure itself, and then edit them.

Dragging an item with the mouse will move it, and holding down the 'Ctrl' key while you drag will copy rather than move it.

Once you've copied an item, you can change its contents using a drop-down arrow on the item - for example, you can change which zone an event refers to, which event is being waited for by a Wait until statement, or the action of an Action statement.

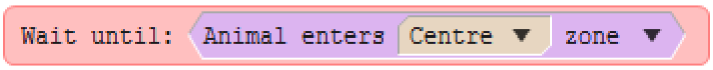


Figure 4. You can change which zone the event refers to by dropping the arrow to the right of 'Centre'. You can change which event is being waited for by dropping the arrow to the right of the event itself.

Re-using parts of procedures

Introduction

Procedures can be flexible on what they can achieve in a test, but this means that they can become very complex. Often, a procedure will repeat certain code, or the same thing needs to be done by more than one procedure running as part of a test.

Details

To reduce procedure complexity, ANY-maze allows you to define *sub-procedures* which can be invoked from within any procedure. This allows you to keep Procedures shorter, and if you need to change any code that needs to be repeated, you only need to change it once (in the sub-procedure) and not multiple times.

You can share information between a Procedure and the sub-procedure(s) that it calls using variables; simply set up the variables to be shared between Procedures for a given test.

Example

As an example, let's consider a piece of apparatus that contains two arms; at the end of each arm is a pellet dispenser. When the animal approaches the end of the arm (a 'target zone'), a pellet will be dispensed. This will also cause either a light or a noise stimulus to be switched on (at random); this stimulus will only be switched off when the animal retreats back down the arm to an area at the start of the arm.

A procedure to do all this will be long and fairly complicated:

Figure 1. An example of a procedure that could benefit from the use of a sub-procedure.

So how can we reduce the complexity of this Procedure? It should be fairly obvious that the bulk of the code is repeated, i.e. the code that turns on the pellet dispenser, turns on the stimulus, and waits for the animal to retreat again before turning it off. The only difference between that repeated code is *which* pellet dispenser and stimulus need to be turned on.

The first thing we'll do is create three variables to note which pellet dispenser, light stimulus and noise stimulus should be used. These variables will be shared between procedures, which means that the main procedure can set them and the sub-procedure can then use them. Then, we simply need the main procedure to set these three variables before calling the sub-procedure. This leaves us with the following main procedure, which I'm sure you'll agree is much simpler!

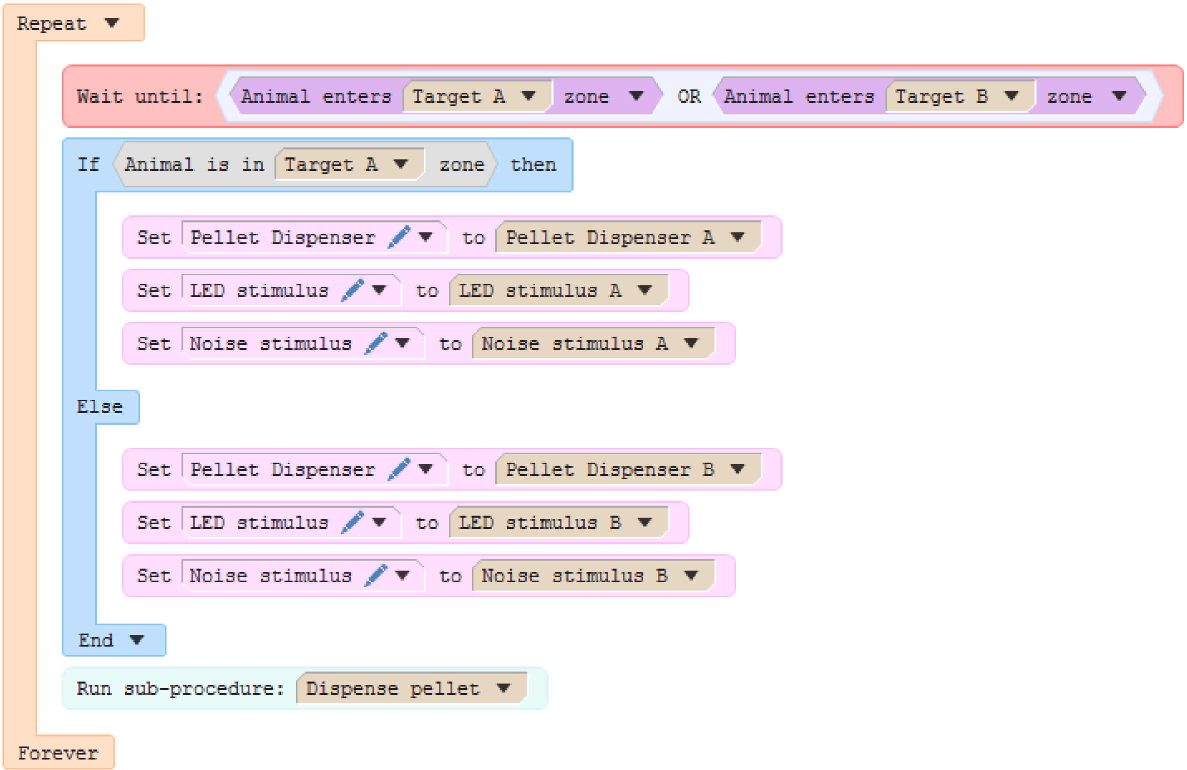


Figure 2. The same procedure, reworked to call a sub-procedure.

That just leaves the matter of defining the sub-procedure. The first thing you'll need to do when creating the sub-procedure is check the box on the Procedure's settings page titled *This procedure is a sub-procedure*. This will mean that ANY-maze won't actually run this procedure, unless you call it directly from within another procedure. Then, the sub-procedure itself contains the repeated code from the main procedure, but instead of directly accessing the specific I/O (pellet dispenser, light stimulus and noise stimulus), we simply refer to the variables that the main procedure set for us:

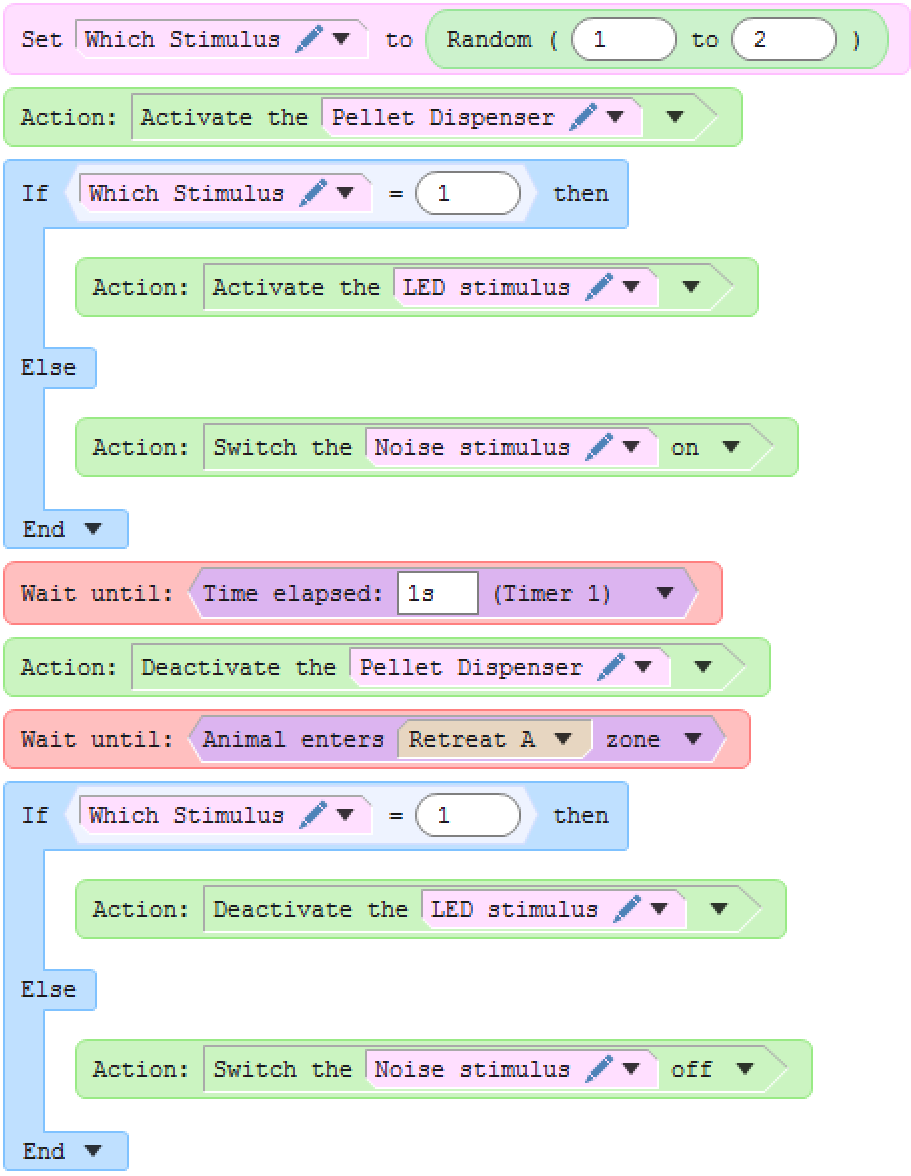


Figure 3. The sub-procedure that is called from the main procedure.

As you can see, using sub-procedures can greatly reduce the code in a given procedure; in fact the same sub-procedure can be invoked from within more than one different procedure. Not only that, a sub-procedure allows you to separate key bits of functionality into easier-to-understand chunks, since there is less code in each procedure.

See also:

- Setting up a procedure

* The Run sub-procedure statement

Running procedures

Introduction

Procedures will start to run when the test is 'ready'.

A test will not start if there are any procedures that contain errors. If you need to run a test anyway, and for some reason don't need a procedure to be working, you can disable it on the procedure's settings page.

Within a procedure, there are two sections: statements that are run *before* the test starts (where the background of the procedure editor is shaded) and statements that are run *after* the test starts (where the background is not shaded).

Statements that run before the test starts will be executed when the test becomes 'ready'. This is usually immediately after the previous test ended. All statements in the first section of the procedure will run, and then the procedure will wait for the test to start.

Note that this could cause unexpected effects, because tests can become 'ready' in multiple situations - for example, when the user makes an edit to the protocol and then switches back to the Tests page. If the statements that run before the test starts are set to choose a random arm of a radial maze and then dispense a pellet into that arm, for example, then if the test become ready multiple times, multiple pellets will get dispensed. One way round this would be to wait for the *Test is waiting to start* event before dispensing the pellet (although this event only happens if Automatic starting of tests is turned on).

As soon as the test starts (either using the ▶ button, or automatically, when the animal is put into the apparatus), the statements in the second section of the procedure will start to run.

The following procedure initialises some I/O before the test starts:

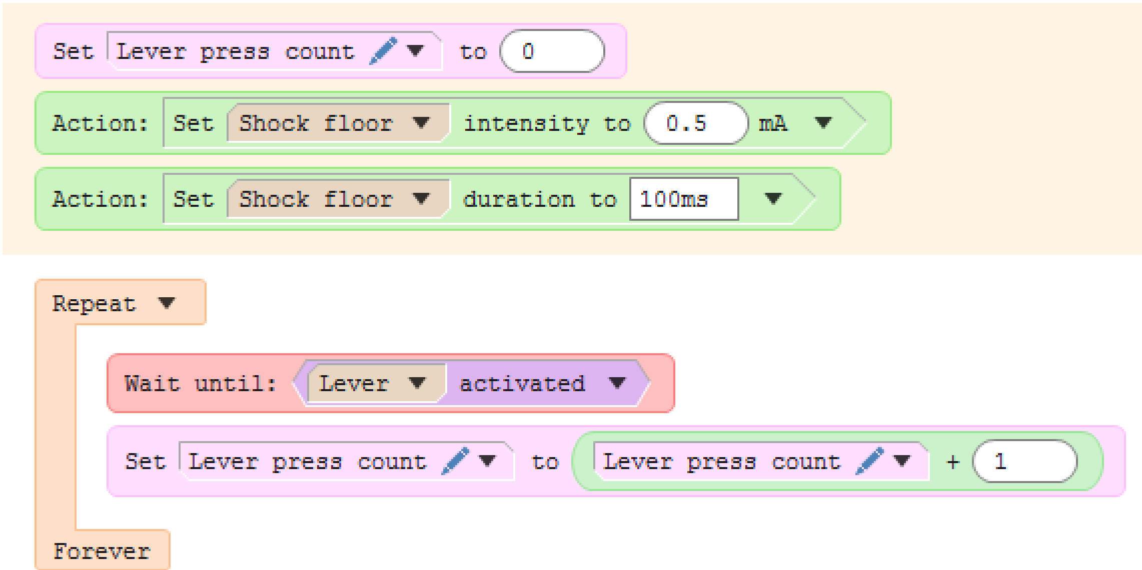


Figure 1. An example of a procedure that runs some statements before the test starts (to initialise a variable, and set up some I/O) and some statements after the test starts (waiting for the animal to press a lever)

Running multiple procedures

If you've defined multiple procedures in a protocol, they will all run simultaneously.

What this means in practice is that each procedure will be given a chance to run - up to the point where it ends, or needs to wait for an event - and then the next procedure will get the chance to do the same.

This means that the *order* of the procedures may be important, since the first procedures defined will get the chance to run before the others. This is especially true if the procedures contain any variables that are shared between procedures.

In future versions of ANY-maze, procedures will be able to be re-ordered by dragging and dropping them in the protocol list.

Technical stuff

You can ignore this section unless you want a deeper understanding of how procedures work.

As ANY-maze performs a test, it tracks the animal in real-time and also monitors the I/O ports set up in the protocol. Whenever anything changes (for example the animal's position, or the state of an I/O port), then internally to the software, a 'test event' will be generated. These 'test events' are loosely (but not exactly) what you'll see on the Test data report.

A test event might be due to a change of animal position, zone entry/exit, lever press, etc. and several of these occur every second.

As a procedure runs, it will take the current test event (the very first one will be a *Test is waiting to start* event) and pass it to the first statement of the first procedure. The procedure statement will be executed based on that event.

Usually, this will cause a series of statements to be run until the current statement is a Wait until statement. If the event that the statement is waiting for is *not* the test event that is being processed, then the procedure will temporarily stop running and the 'test event' will be passed to each other procedure, in turn, until they've all had a go. If any procedure's current statement is a *Wait until* statement, and it's waiting for this specific event to occur, then it will continue running statements until it reaches another *Wait until* statement (or the end of the procedure).

It therefore follows that a procedure that doesn't contain any *Wait until* statements, but contains a loop where it continually runs the same sequence of statements, won't give any other procedures the chance to run. (In fact, ANY-maze tries to detect such a situation, and tries to ensure that all procedures are given a chance, but this is an inefficient way to run a procedure). This is why you should usually have at least one *Wait until* statement in a procedure).

See also:

- Procedure statements
- The Wait until statement

Errors and warnings while running a procedure

Introduction

Even if a procedure has been checked for errors in the procedure editor, there is still a chance that it may not run correctly. This can occur if, for example, the result of an expression is not valid (e.g. trying to divide a value by zero). Errors of this type are called 'run-time errors' (because they're only detected when the procedure is run) and if ANY-maze detects any while a procedure is running, it will list them in the Test details report.

Examples of errors

Most 'run-time' errors will occur because the result of an expression is not valid. This might be because the expression uses a variable which does not contain the right type of expression (numeric, True/False, or text), or a numerical calculation results in an invalid number (for example, trying to take the square root of a negative number).

The procedure editor has no idea what the result of these expressions will be until the procedure is actually run, so it can't flag them up as an error in the procedure editor; they will only be caught when the procedure is actually run.

In general, a procedure that contains errors will note the error and then continue to run (if possible).

Note that if a mathematical operation results in an invalid value, the result will be set to #N/A. The procedure will carry on running, but if the value is used in any other mathematical functions, then the result of those will also be #N/A. Note that you can test whether a number is undefined in a procedure using the 'Is undefined' function.

Some examples of the kind of things that will cause run-time errors are:

- Trying to use a variable or value of the wrong type. The most obvious of these is trying to use a text variable in a mathematical function, but there will be other times when ANY-maze can't evaluate a value as the correct type. In general, ANY-maze tries to convert where possible, so if a True/False value is required, then zero will be false and non-zero will be true. Converting a numerical or True/False value to text will always succeed.
- Using an invalid index for an array variable. If a variable has been set up to be an array with 5 values, then you can only access the indexes of this variable using the values 1 to 5. If you're using the result of an expression to determine the index, and the expression's result is not within this range, an error will be reported (and the value #N/A will be returned for the value of the variable at that index).
- Using values in a mathematical operation that will cause it to overflow. For example, calculating the square root (*Sqrt*) of a negative number, dividing by zero, or calculating the inverse sine (

Sin-1) of a number that is not in the range -1.0 to 1.0.

How you get notified of any errors

If any errors or warnings occur when a test is run, a yellow information bar will appear at the top of the ANY-maze screen, informing you of this fact. You'll also find the (test debug output) button is highlighted; if you click on this button then the Test debug output window will open so you can view the error(s) and/or warning(s) that occurred.

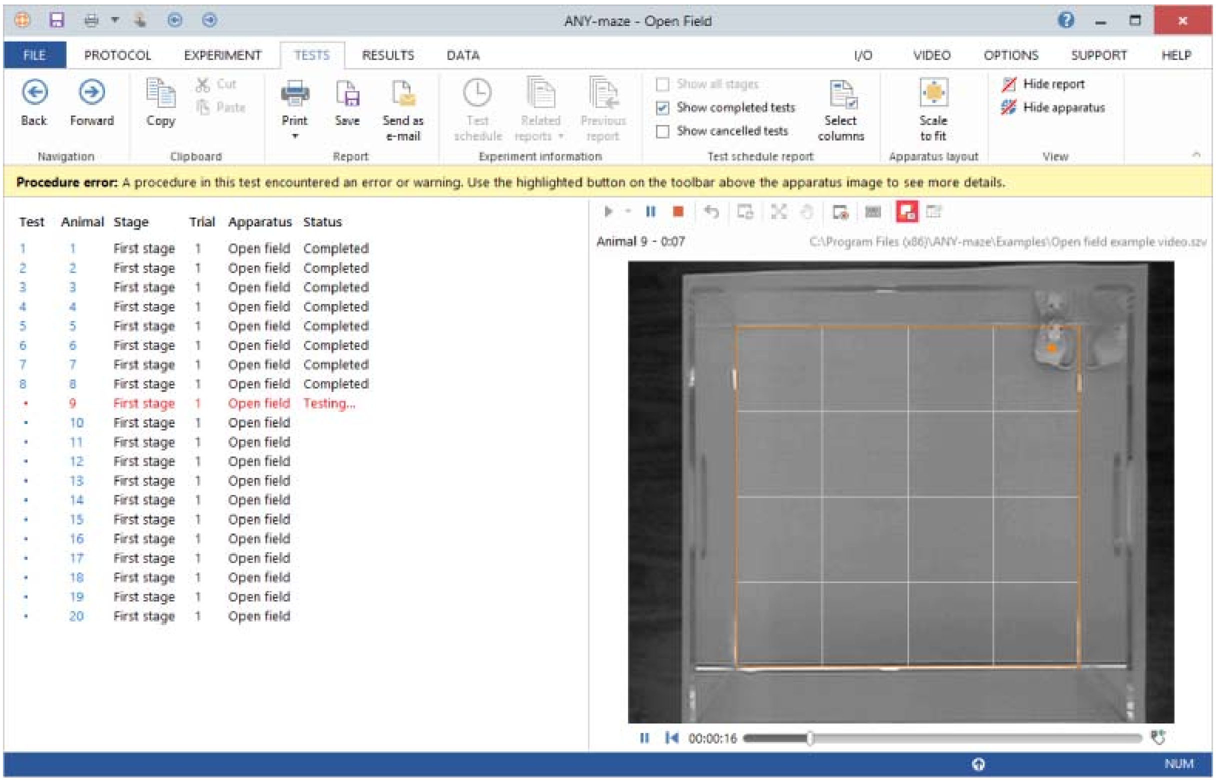


Figure 1. The information bar that appears if a procedure encounters any errors or warnings during the test.

You'll also get a message box displayed at the end of the test. Any errors and warnings will be listed, alongside the name of the procedure and the test clock at the time the error occurred, in the Test details report for the test.

Errors and warnings

The following errors and warnings can happen when the test is run:

Errors

- Error: The procedure was aborted while it was still executing statements.
- Error: The procedure took too long waiting for the variable to update.
- Error: The procedure could not run a sub-procedure.

Warnings

- Warning: This procedure was disabled and was therefore not used in this test.
- Warning: The procedure was unable to keep up with the events generated by the test.
- Warning: The procedure appears to be stuck executing a statement.
- Warning: Cannot evaluate the expression as a [numeric] value.
- Warning: Cannot evaluate the expression as a time period.
- Warning: Cannot evaluate the expression as an [ANY-maze protocol item].
- Warning: The parameter to the variable was invalid.
- Warning: Division by zero.
- Warning: The repeat count is invalid.
- Warning: The index to the array variable was not a valid value.
- Warning: The selector value is invalid.
- Warning: A negative time value was used in a 'Time elapsed' event.
- Warning: The parameter to the function was not a positive numeric value.
- Warning: The parameter to the function was a negative value.
- Warning: The expression resulted in a numerical overflow.
- Warning: The number of items in the list passed to 'Initialise array' did not match the number of indexes in the array.
- Warning: The sizes of the source and destination arrays passed to 'Copy array' did not match.
- Warning: Could not set the location of the movable zone, because this zone is not set up for the current stage to 'The location will be set by a procedure'.
- Warning: The procedure tried to set the location of the zone, but this does not represent a valid position for this zone.

- Warning: The procedure tried to determine the location of '...', but this is not a valid movable zone.
- Warning: The procedure tried to access the treatment for this test, but the treatment has not been set.
- Warning: The timer resolution set for the procedure was invalid.

See also:

- Errors and warnings while writing a procedure

Error: The procedure was aborted while it was still executing statements

Description

A test may finish for one of a number of reasons. The most common of these will be because the Test duration set up for the stage has been reached, but it might also end because a procedure has used the *End the test* Action. You might also abort the test by using the button.

Under all of these circumstances, any procedure that is still running will continue running, until the point that it reaches the end of its list of statements, or encounters a Wait until statement. Since the test has finished, it's not possible for any further events to occur, so the procedure is stopped at this point. (Note that it's still possible under some circumstances for the procedure to receive a *Test continuation* event - for more details, see Procedures and the end of a test).

If the procedure hasn't encountered a *Wait until* statement after a short period of time and is still running, then it will be aborted in mid-flow and this warning will be generated. This will usually only happen if the procedure is stuck in a loop, without a *Wait until* statement.

This warning is displayed in the Test details report after the test has stopped.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement

Error: The procedure took too long waiting for the variable to update

Description

Within ANY-maze, a procedure has access to a large number of built-in variables which allow it to keep track of the current state of a test, including the state of any I/O devices used by the test. All of these variables start off in an 'unknown' state, and they only get updated to a known and valid value once ANY-maze has started tracking and has determined where the animal is, which zone it is in, etc.

Some of these variables return the state of an I/O port, for example, the *Switch is active* variable returns the state of an output switch, or the *Temperature controller temperature* variable returns the current temperature of a temperature controller. Obviously, in order to return the state of these ports, ANY-maze must ask the I/O device for its state. This is done at the start of a test, and also when the procedure uses an action to change an I/O output on the device. When an action is used in this way, the built-in variable is temporarily marked as having an 'unknown' value, and only when the action has succeeded, and ANY-maze has read the resulting state of the device, is this variable updated to a valid value.

So we can see that at various points during the test, some of ANY-maze's variables will temporarily be in this 'unknown' state. This should, however, be for a *very* short period of time - only a fraction of a second. If the procedure asks for a value which is currently in this 'unknown' state, it will sit and wait until this value is updated. ANY-maze will give all variables up to a second to update (most of them will be updated much, much quicker than that) and only if the variable hasn't correctly been updated within this time will this error be generated.

More information

If this error occurs, it is most likely that one of your I/O devices is not responding correctly when the procedure asks it to perform an action.

If this error occurs, the value of the built-in variable will evaluate to #N/A (for numeric variables), False (for True/False variables) or empty text (for text variables).

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Built-in variables

- Actions available to procedures

Warning: This procedure was disabled and was therefore not used in this test

Description

This warning is displayed in the Test details report after a test is run, if the named procedure was disabled and therefore not used in this test.

More information

You should double-check that the disabled procedure was not necessary for the test in question, although you should already have been warned about the disabled procedure on the *Tests page* before starting the test.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The 'Disable procedure' setting

Warning: The procedure was unable to keep up with the events generated by the test.

Description

As a test runs, it continually generates 'test events' which are the things that happen due to animal movement, I/O inputs, etc. All of these 'test events' are passed to each procedure to deal with.

A procedure takes each of these test events and runs as many statements as it can, based on that event. If the procedure is currently in a Wait until statement, then it won't do anything unless the 'test event' that it receives is the thing it's waiting for. If it is, then it moves to the next statement; otherwise, it will go no further and will wait for the next 'test event' to see if that is what the procedure is waiting for.

If a procedure can't immediately handle a test event (if, for example, it's in the middle of processing the previous event), then this event will be put in a queue for the procedure to deal with when it can. However, if the procedure is stuck in a loop for some reason, or the procedure receives a great many events in quick succession, then the queue will soon build up and the procedure won't be able to deal with it fast enough.

Often, the procedure will finish what it's currently doing and will start to process the queue of events again. If this *doesn't* happen (i.e. the queue of events waiting to be processed does not reduce in size), then the procedure will generate this warning. The procedure will carry on running, but it does usually indicate a problem with the way that the procedure has been written, since it can't keep up with the 'test events' that are being generated. The best way to fix such a problem is to ensure that the procedure regularly waits for things to happen using a Wait until statement, since 'waiting' is what empties the queue quickest and allows other procedures to have a chance at processing test events too.

If you're still not sure why you're getting this warning, then please contact ANY-maze technical support and we'll be happy to help.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement

Warning: The procedure appears to be stuck executing a statement.

Description

Often, you'll want a procedure to repeat some sequence of events throughout all or part of a test - for example, you want to repeatedly take some action when the animal enters a zone. In this case, your procedure would contain some way of looping back to a particular point in the test, either using a Repeat statement or a Go to statement in conjunction with a Label statement.

Since a procedure works by running through its statements in order, one after the other, there's a danger that it could get stuck in a loop and keep running the same statements over and over again, without ever taking a break and allowing other procedures to run. Procedures only stop and take a break when they encounter a Wait until statement, so if the loop does not contain any *Wait until* statements then it won't allow other procedures to run (in fact, ANY-maze tries to detect this situation and allow other procedures to run, but it's an inefficient way to run and not recommended).

If the procedure finds itself repeatedly running the same set of statements, without encountering a *Wait until* statement, then you'll see this error in the Test details report after the procedure has finished running. The warning tries to indicate which statement it thinks it is repeating, to give you some idea of where in the procedure it is happening.

Example

Consider the following procedure, which uses a loop but does not include a *Wait until* statement.

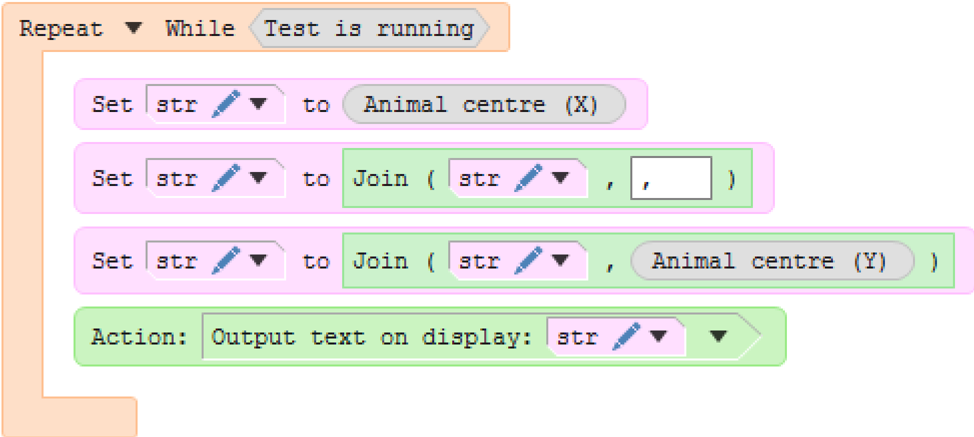


Figure 1. A procedure that contains a loop without a Wait until *statement.*

This procedure will repeat this loop continuously throughout the test, but the absence of a *Wait until* statement will mean that it won't take a break and give other procedures the chance to run. To fix this, you need to insert a *Wait until* statement.

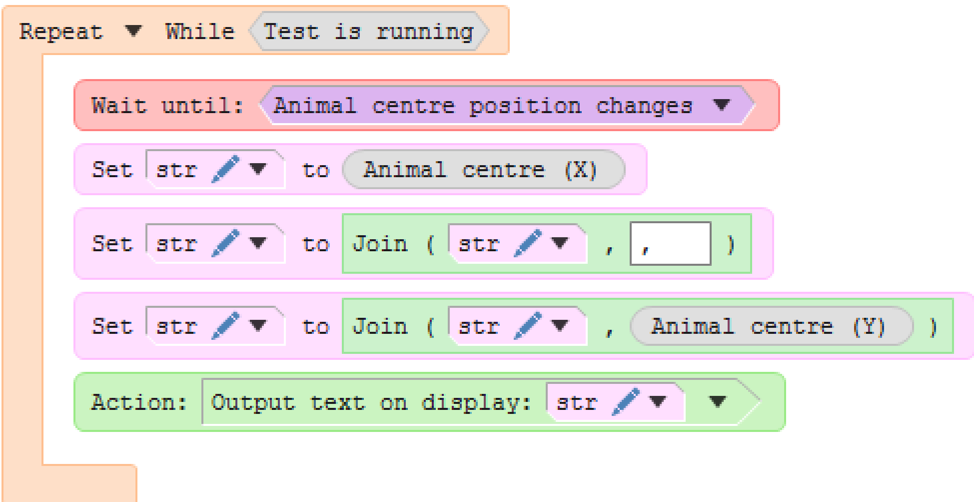


Figure 2. The solution to this is to insert a Wait until *statement. The procedure will do exactly the same thing, but will run more efficiently and also allow any other procedures to run properly.*

If you're still not sure why you're getting this warning, then please contact ANY-maze technical support and we'll be happy to help.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement
- The Repeat statement
- The Go to statement
- The Label statement

Warning: Cannot evaluate the expression as a [numeric] value

Description

Expressions used in a procedure can be one of three types - numeric, True/False, or text. All of the functions and operators in a procedure will act on one of these specific types of expression (for example, a Maths function or operator or a Comparison operator will act on numeric values, and a Logical operator will act on True/False values).

If an expression in a procedure is not of the correct type, then this warning will be generated and listed in the Test details report.

More information

This warning is most likely to occur for an expression which is expected to be numeric, but is actually a text value, for example:

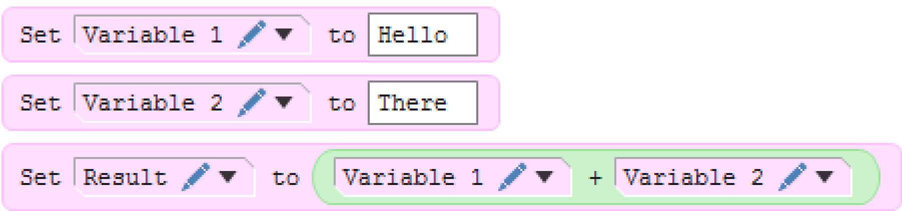


Figure 1. This rather contrived example will result in this error, since 'Hello' and 'There' are text, and the procedure can't evaluate them as numbers for the + operator.

If you wanted to join two pieces of text in this way, you should use the Join *function, rather than the + operator. See* Other functions *for more details on this function.*

ANY-maze will try and convert the values used in the expression to the correct type. For example, a True/False value will be converted to 1 or 0, and if a text value contains a number (e.g. '1' or '3.1415'), then these can be successfully converted to numeric values.

If ANY-maze can't convert the values, and this warning is generated, then the actual result of the expression will be indicated when the warning is listed in the Test details report. This result will be #N/A for numeric expressions, False for True/False expressions, and empty text for text expressions.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure

Warning: Cannot evaluate the expression as a time period

Description

A procedure can use the Wait until statement to wait for a number of possible Events, one of which is the *Time elapsed* event. When this event is used, you need to enter the time to wait as part of the event. This time can either be entered as text (for example, '30min' or '100ms') or can be the result of another numerical expression; if it's a numerical expression that is used, ANY-maze will assume that the value is in seconds.

If the procedure can't evaluate the expression (or whatever text you type in) as a time, this warning will be generated in the Test details report.

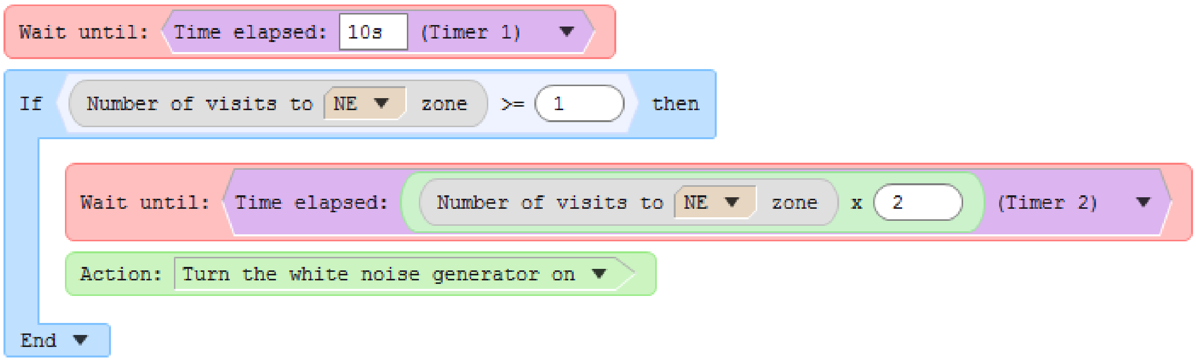


Figure 1. This procedure waits for two Time elapsed *events; the first is for a specific time, the second uses the result of a numeric expression which is assumed to be in seconds.*

More information

If this warning is generated, then the value used for the time elapsed will be 0. This means that the procedure will wait for 0 seconds to elapse, so will effectively drop straight through this *Wait until* statement without waiting at all.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement
- Events available to procedures

Warning: Cannot evaluate the expression as an [ANY-maze protocol item]

Description

A number of Events, Actions, and built-in variables in ANY-maze procedures require an extra value or 'parameter' to be specified. For example, the *Animal enters zone* event requires the zone to be specified; the *Set speaker volume* action requires the speaker to be specified, and the *Duration of sequence* built-in variable requires the sequence to be specified.

Usually, you'll set this parameter by simply selecting it from a drop-down list on the item itself, but there will be occasions when you wish to use the result of an expression to specify the parameter.

For example, the following section of a procedure uses a variable, rather than the zone directly, to specify the parameter to an event (You can tell that the :

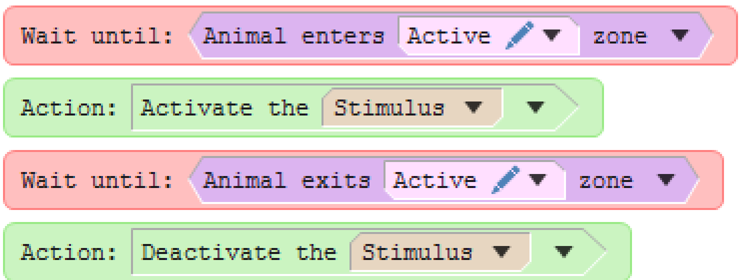


Figure 1. This procedure uses a variable called 'Active' as the parameter to the Animal enters zone *event. The pink colour and the presence of the button show that it's a variable rather than an ANY-maze protocol item, which would be brown.*

For the example in figure 1 to work, the 'Active' variable *must* previously have been assigned to something that ANY-maze can interpret as a zone. This could be the zone itself, or alternatively, some text that is the name of the zone. Either way, ANY-maze knows that a zone is required and will attempt to evaluate this as the name of a zone. If it can't, then this warning will be generated.

More information

All available zones (and other items from the protocol list) are listed under the *Variables* tab of the procedure editor, under *ANY-maze protocol items*.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- User-defined variables
- Built-in variables

Warning: The parameter to the variable was invalid

Description

A number of built-in variables in ANY-maze procedures require an extra value or 'parameter' to be specified. For example, the *Distance from zone* variable requires the zone to be specified, and the *Duration of sequence* variable requires the sequence to be specified.

Usually, you'll specify this parameter by simply selecting it from a drop-down list on the item itself, but there will be occasions when you wish to use the result of another expression to specify the parameter.

For example, the following section of a procedure uses a user-defined variable, rather than the zone directly, to specify the parameter to the *Animal is in zone* variable:

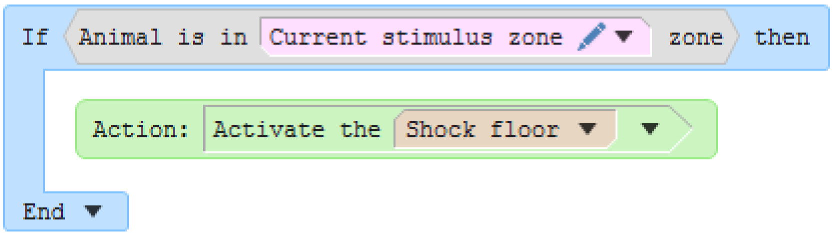


Figure 1. This procedure actually uses a user-defined variable called 'Current stimulus zone' as the parameter to the Animal is in zone *built-in variable, rather than using a zone directly. The variable must previously have been assigned to a zone.*

For the example in figure 1 to work, the 'Current stimulus zone' variable *must* previously have been assigned to something that ANY-maze can interpret as a zone. This could be the zone itself, or alternatively, some text that is the name of the zone. Either way, ANY-maze knows that a zone is required and will attempt to evaluate this as the name of a zone. If it can't, then this warning will be generated.

More information

If this warning is generated, then the built-in variable will be evaluated as #N/A (if it's a numeric variable), False (if it's a True/False variable), or empty text (if it's a text variable). You can tell which type of variable it is by its shape - numeric variables are lozenge-shaped (with rounded ends), True/False variables have angled ends, and text variables are rectangular.

You can test whether a numeric expression in a procedure is equal to #N/A using the Is undefined *function, which is described under* *Other functions*.

All available zones (and other items from the protocol list) are listed under the *Variables* tab of the procedure editor, under *ANY-maze protocol items*.

When the procedure editor is in 'Simple' mode, the Variables *tab is not shown. You can switch the procedure editor between* Simple view *and* Full view *using the buttons on the ribbon bar.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Built-in variables
- User-defined variables

Warning: Division by zero

Description

There are two mathematical operators available to a procedure which involve dividing one number by another: divide (*/*) and modulus (*Mod*). If the right-hand side of either of these expressions evaluates to zero, this warning will be generated.

Example

In the following statement, the 'Division by zero' warning could be generated if the animal has never entered the centre zone. In this case, the number of visits will be zero, causing the procedure to try and divide the value in the 'Total' variable by 0.

Figure 1. This statement may result in a 'division by zero' warning, if the animal has never entered the centre zone.

More information

If this warning is generated, the result of the */* or *Mod* operator will be set to #N/A.

You can test whether a numeric expression in a procedure is equal to #N/A using the Is undefined *function, which is described under Other functions.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Maths functions and operators
- Other functions

Warning: The repeat count is invalid.

Description

When a Repeat statement is used to repeat a set number of times, the parameter specified for the statement must be an _integer_ value that is zero or greater.

You can use any numeric expression as the repeat count, but it must evaluate to a positive whole number.

If the result of an expression is used as the count, it's not possible for the procedure editor to work out what value that expression will have when you write the procedure, so ANY-maze can only check the value when the procedure is actually run. In this case, if the expression results in an invalid count (for example, the value is less than zero, or the value is undefined), then a warning will be displayed in the Test details report and the count used will be zero (i.e. the contents of the Repeat statement will not be run).

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Loop statements

Warning: The index to the array variable was not a valid value

Description

When a user-defined variable is set up as an array consisting of a number of values, these individual values can be accessed using the 'index' of the value in the array. For example, an array called 'Count' which is set up to have 4 values can provide access to the individual values in the array using Count(1), Count(2), Count(3), and Count(4).

When using an expression in a procedure to access a variable's index in this way, you can enter specific numeric values, or alternatively, the procedure can use the results of other numeric expressions for the indexes.

If a specific numeric value is used for the index, then the procedure editor can detect whether this value is valid when the procedure is checked for errors - if invalid, then an *Error* will be displayed in the list of errors and warnings at the bottom of the procedure editor and the procedure will not be valid.

However, if the result of an expression is used as the index, it's not possible for the procedure editor to work out what value that expression will have when it is actually run, so ANY-maze can only check the value when it is actually used in the procedure. In this case, if the expression results in an invalid index for the variable, a warning will be displayed in the Test details report and the value retrieved from the variable will be set to #N/A (if the type of expression returned should be numeric), False (if it's a True/False value), or empty text (if it's a text expression).

Example

The following loop initialises all the values of a four-value array variable using a counter. However, if you look closely at the way the loop works, it will actually run through the loop 5 times - so the last time through, it will be trying to access Tone Frequency (5), which is an invalid index for an array with 4 values.

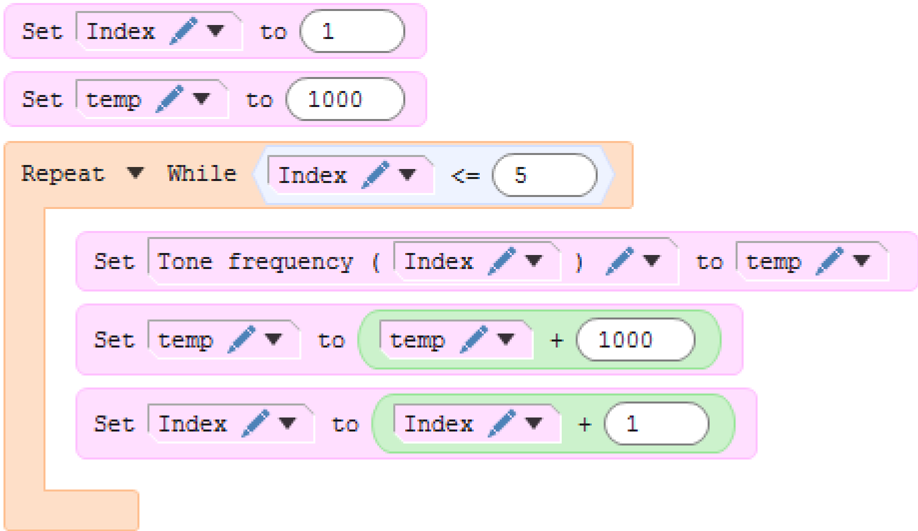


Figure 1. This procedure uses a loop to initialise an array of four tone frequencies. However, this will generate an error because the last time through the loop, the value of the 'Index' variable will be 5, and there are only 4 values in the array.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- User-defined variables

Warning: The selector value is invalid.

Description

When the Selector Value Text variable is used to retrieve the name for a given value, the value must be a valid value for this selector. The valid range of values is between 0 and the maximum possible value for the selector.

If the second parameter to this function is not a valid value for the selector, this warning will be generated, and the built-in variable will evaluate to '#N/A'.

This is text containing '#N/A', and not *the undefined numeric value #N/A, since the variable does not return a numeric value.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Built-in variables

Warning: A negative time value was used in a 'Time elapsed' event

Description

A procedure can use the Wait until statement to wait for a number of possible Events, one of which is the *Time elapsed* event. When this event is used, you need to enter the time to wait for as part of the event. This time can either be entered as text (for example, '30min' or '100ms') or can be the result of another numerical expression; if it's a numerical expression that is used, ANY-maze will assume that the value is in seconds.

If the expression results in a negative number, this warning will be generated, since a procedure cannot wait for a negative time.

More information

If this warning is generated, then the value used for the time elapsed will be 0. This means that the procedure will wait for 0 seconds to elapse, so will effectively drop straight through this *Wait until* statement without waiting at all.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- The Wait until statement
- Events available to a procedure

Warning: The parameter to the function was not a positive numeric value

Description

The *Log* and *Ln* mathematical functions both act on a value that must be positive. If they are given a negative value, or zero, the result is undefined.

When used in a procedure, if the expression used in the *Log* or *Ln* function is zero or negative, this warning will be generated and the function will return #N/A.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Mathematical functions and operators

Warning: The parameter to the function was a negative value

Description

The *Sqrt* mathematical function acts on a value that must be greater than or equal to zero. If it is given a negative value, the result is undefined.

When used in a procedure, if the expression used in the *Sqrt* function is negative, this warning will be generated and the function will return #N/A.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Mathematical functions and operators

Warning: The expression resulted in a numerical overflow

Description

For certain Maths functions and operators, it's possible that passing large numbers to them will cause them to overflow (i.e. the result would be a number that's too big for the computer to store!). This applies to the *Pow* function, and the + and *x* operators.

If a procedure is passed numeric expressions that would cause these to overflow, this warning will be generated.

More information

Numbers in an ANY-maze procedure are 'double-precision' floating-point numbers, and can have any value up to approximately 1x10^308. So it's quite unlikely that you'll encounter this error under normal conditions! If this error is generated by a procedure, then the result of the expression which would overflow will be set to #N/A.

You can test whether a numeric expression in a procedure is equal to #N/A using the Is undefined *function, which is described under Other functions.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Maths functions and operators
- Other functions

Warning: The number of items in the list passed to 'Initialise array' did not match the number of indexes in the array

Description

A procedure can use the *Initialise array* function to set up an array variable using a single statement, rather than having to use a loop to set each index separately. The values are entered as a comma-separated list of values, and the number of values must be equal to the number of values in the array. If it isn't, this error will be generated.

Example

This example shows how an array of 5 values is initialised by passing 5 comma-separated values to the *Initialise array* statement. If the number of comma-separated values in the list were not equal to 5, this warning would be generated.

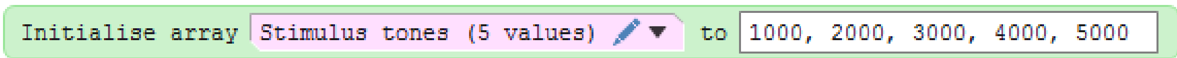


Figure 1. Using the Initialise array *statement to set up the values of an array variable.*

More information

If this warning is generated, the procedure will initialise what values it can, and continue running. If not enough values are specified, these array indexes will be left as they are; any extra values will be ignored.

For example, if the initialisation in figure 1 above used '1000, 2000, 3000' then the array would be initialised to contain the values 1000, 2000, 3000 as its first three values, and the last two would be left with their previous values. (If this is the first time the array is used, then the last two values would be 0). If you use '1000, 2000, 3000, 4000, 5000, 6000' then the last value would be ignored, and the array values would be initialised to 1000, 2000, 3000, 4000 and 5000.

The Initialise array *statement can be used at any point in a procedure, and doesn't necessarily have* *to be used at the start.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Other statements
- User-defined variables

Warning: The sizes of the source and destination arrays passed to 'Copy array' did not match

Description

The *Copy array* statement copies the contents of one array variable into another array, overwriting what is already there. This warning is generated if the sizes of the arrays are not the same, i.e. they don't have the same number of values.

Example

The following example shows the *Copy array* statement in use:

Figure 1. Copying the contents of an array variable into a different array variable.

More Information

If this warning is generated, the procedure will still continue running and will copy what it can of the array. If the array being copied from is larger than the array being copied to, then the extra values will be ignored and not copied. If the array being copied from is smaller than the one being copied to, then the extra array indexes will still contain their previous values.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Other statements
- User-defined variables

ANY-maze help topic H0489

Warning: Could not set the location of the movable zone, because this zone is not set up for the current stage to 'The location will be set by a procedure'

Description

A procedure can set the position of a movable zone using the *Set location of movable zone* action. However, in order to do this, you need to specify that 'The location will be set by a procedure' for the stage in which the test is running. If this is not set, then the procedure will generate this warning when it encounters after the *Set location of movable zone* action.

More information

To specify that the location of a movable zone will be set by a procedure, select in turn each of the stages in the protocol list that the procedure will run in. For each movable zone, the stage will contain an option for the *Location of the ... zone during this stage*.

Selecting this option will allow you to specify that the zone's location will be set by a procedure, by selecting the *The location will be set by a procedure* option.

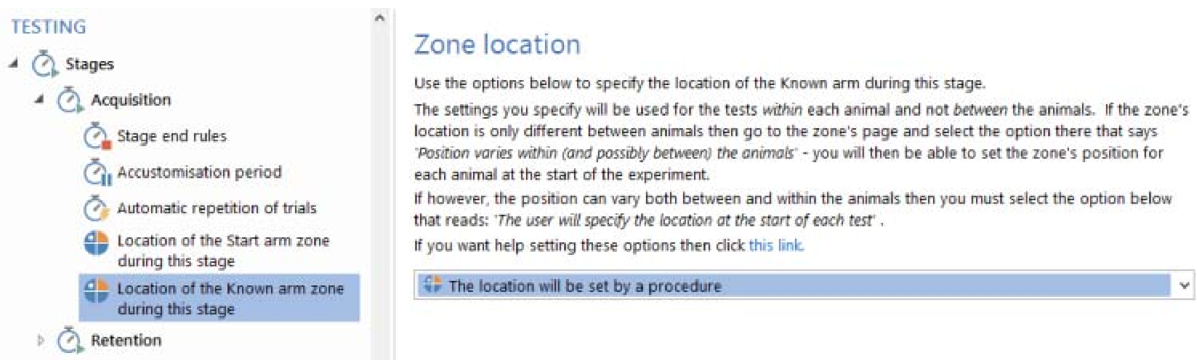


Figure 1. Selecting the option for a stage to allow a procedure to set the location of a movable zone.

If a procedure encounters this action, for a movable zone that is *not* set to be changed by a procedure, then the statement will simply be ignored and the position of the zone will not be set.

When you insert the Set location of movable zone *action into a procedure, the procedure editor should display a message warning you that this will only work if the zone in the test's stage is set to 'The location will be set by a procedure'.*

Note also that the *Set location of movable zone* action can **only** be used in a procedure *before* the test starts - i.e. in the shaded area at the top of the procedure. You can't change the location of a movable zone *after* the test has started.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Specifying positions of movable zones
- Actions available to a procedure

Warning: The procedure tried to set the location of the zone, but this does not represent a valid position for this zone

Description

A procedure can set the position of a movable zone using the *Set location of movable zone* action. This action specifies the movable zone whose location is to be set, and the zone position to set it to.

If the position is not a valid position for the zone, then this warning will be generated in the Test details report and the position of the zone will not be set. This will mean that the zone itself will effectively not exist for this test, and ANY-maze won't be able to generate results for the zone - for example, counting entries into the zone, time spent or distance travelled in the zone, etc.

If other movable zones are dependent on the position of this zone, ANY-maze won't be able to determine the location of those zones either - so there will be no results generated for these dependent zones either.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Specifying positions of movable zones
- Actions available to a procedure

Warning: The procedure tried to determine the location of '...', but this is not a valid movable zone

Description

If your procedure has any movable zones, a procedure can find out the location of the zone using the *Location of movable zone* built-in variable. This variable needs to be told which movable zone to retrieve the location of, and usually you'll just use the drop-arrow next to the zone name to select which movable zone you require the location of.

However, there may be circumstances under which you use the result of another expression to determine the zone, e.g. you might have the zone you're interested in stored in a user-defined variable . If this is the case, and ANY-maze can't determine which zone the expression or variable refers to (or if the expression *is* a zone, but it's not a *movable* zone), then this warning will be generated.

Example

This section of procedure uses a user-defined variable within the *Location of movable zone* variable, when checking the location of a movable zone.

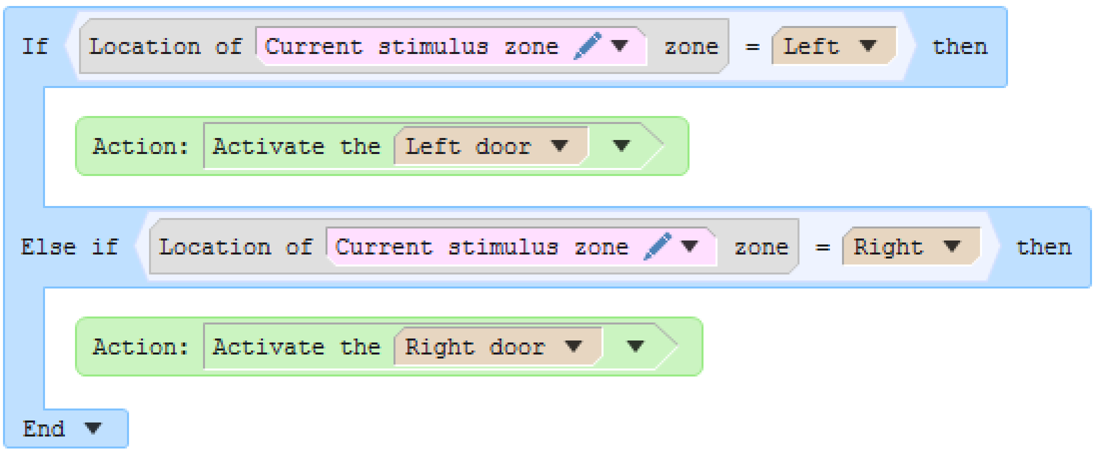


Figure 1. This procedure uses a variable called 'Current stimulus zone' as the parameter to the Location of movable zone *variable, rather than using the zone directly. The variable must previously have been assigned to a zone.*

This will work correctly as long as the 'Current stimulus zone' has previously been set to a movable zone (or text that is the name of a movable zone); if not, then this warning will be generated.

More information

If this warning is generated, the expression will return '#N/A' as the location of the zone.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Specifying a movable zone
- Built-in variables
- User-defined variables

Warning: The procedure tried to access the treatment for this test, but the treatment has not been set

Description

A procedure can ask ANY-maze which treatment is being used for the current test, using the *Treatment* built-in variable.

However, it's possible that the treatment has not yet been specified for this test (since it's possible to start running tests before entering any treatment information on the Experiment page). If the treatment information has not yet been entered, then ANY-maze obviously can't determine the treatment when the test is run.

If this is the case, then using the *Treatment* built-in variable within a procedure will result in this warning being generated, and the built-in variable will evaluate to '#N/A'.

This is text containing '#N/A', and not *the undefined numeric value #N/A, since the treatment is not a numeric value.*

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Built-in variables

Warning: The timer resolution set for the procedure was invalid

Description

Within an ANY-maze procedure, there is a time resolution of approximately 50ms. This will usually be sufficient, but if you need to wait for elapsed times that are less than this (for example, to toggle the state of an I/O output switch), then you will need to use the *Set timer resolution* action to alter the timer resolution for this procedure.

Valid values for the timer resolution are between 1ms and 60s; this error is generated if you try to set the timer resolution to anything outside of that range.

If you pass a numerical value without any units to the Set timer resolution *action, ANY-maze will assume that this value is in seconds.*

Example

If you need your switch to be toggled for 10ms, you'd need to change the resolution to 10ms or less, as follows:

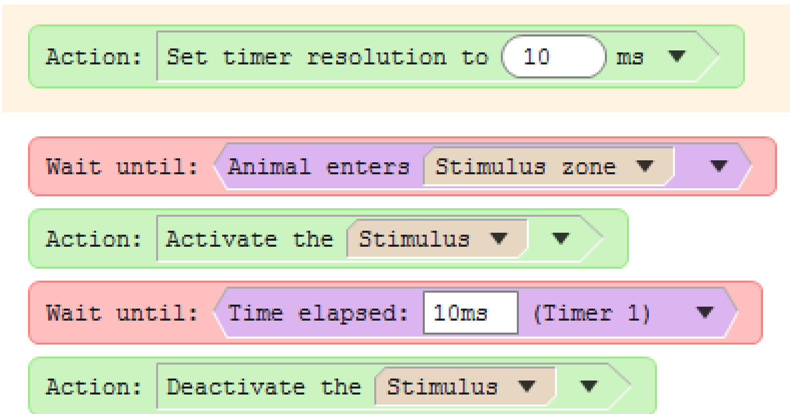


Figure 1. This procedure alters the time resolution before the start of the test, and then uses it to ensure a small time interval for the toggling of an output switch.

Note that changing the timer resolution in this way only affects the procedure that it is used in, and not any other procedures. However, it does mean that the procedure needs to process its statements much more often, using more of the computer's processing power, so it's not a good idea to alter the resolution unless you really need to (or are advised to do so by ANY-maze technical support).

More information

If you pass an invalid timer resolution to the *Set timer resolution* action, it will be ignored and the timer resolution will be left at its previous valid value.

See also:

- Errors and warnings while writing a procedure
- Errors and warnings while running a procedure
- Actions available to a procedure

Error: The action is no longer supported

Description

The following actions are no longer available to ANY-maze procedures:

- Turn the white noise generator on
- Turn the white noise generator off

If you have used any of these actions in a procedure, they will be removed when the procedure is loaded, and the procedure will no longer run. To find out how to work around this, please see below.

Turn the white noise generator on/off

White noise in ANY-maze can be used either as a short-term stimulus within a test, or as a continuous background noise to mask other sounds in the lab - for example, the noises created by other equipment, noises in the room next-door, etc. To be used as a background noise, one of your computer's speakers must be selected as the 'white noise generator' using the *Setup devices* button on the I/O Page. (For more details on how to do this, please see the help topic on Soundcards).

If you want to use white noise as a stimulus during a test, the best way to do this is to set up one of your computer's soundcards as an I/O Device in the protocol, and then use one of the soundcard's speakers as an I/O port. The procedure can then turn on the white noise as follows:

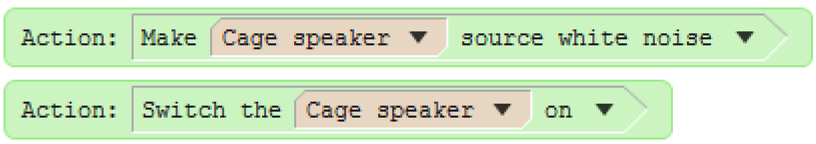


Figure 1. This procedure will set the speaker to play white noise, and then turn it on.

See also:

- Using soundcards as I/O audio devices
- Playing white noise from a soundcard
- Actions available to procedures

- Errors and warnings while running a procedure

Error: The built-in variable is no longer supported

Description

The following built-in variables are no longer available to ANY-maze procedures:

- Analogue output minimum limit
- Analogue output maximum limit

If you have used any of these built-in variables in a procedure, they will be removed when the procedure is loaded, and the procedure will no longer run. To find out how to work around this, please see below.

Analogue output minimum limit / Analogue output maximum limit

The limit for these ports is no longer available as a built-in variable in ANY-maze because if you are using multiple apparatus, there may actually be different ports from different devices set up for this output.

Rather than asking ANY-maze for the limits for this output, you will need to determine which physical ports are used by this analogue output (you can do this from the Port to use sub-element of the output on the Protocol page) and work out what voltages are appropriate to use, based on which ports of which devices are physically connected. You can find out the range of the port(s) by plugging in the devices containing the port(s) and looking on the I/O page - select the device in the sidebar on the left-hand side of the page, then select the *Analogue Outputs*. The port information will show the output voltage range of the port.

Alternatively, you can ignore the limits and simply attempt to set the value of the analogue output from within the procedure - if the value you try and set is outside the valid range, it will be clipped to the minimum or maximum of the output's range.

See also:

- An introduction to analogue outputs
- Setting up an analogue output
- Analogue output actions
- Built-in variables
- Errors and warnings while running a procedure

Error: The procedure could not run the sub-procedure

Description

When writing procedures it's possible to take out bits of a procedure and put them into a sub-procedure. This reduces procedure complexity and means that part of a procedure that is used multiple times only needs to be written once.

This error means that ANY-maze has failed to run such a sub-procedure. This may be because the parameter passed to this statement does not represent a valid sub-procedure (for example, the statement is passed a variable, but the variable's content cannot be evaluated as a subprocedure).

See also:

- Re-using parts of procedures
- Errors and warnings while running a procedure

Debugging procedures

Introduction

The flexibility allowed by procedures can result in some complex logic, and they can become a bit like mini computer programs.

Because of this potential complexity, you should take steps to make sure that a procedure is working correctly before using it in a real experiment. We strongly recommend that you **always test your procedures** before relying on them!

When the procedure isn't running quite as you expect, what can you do to work out what is going wrong? ANY-maze provides you with some tools to help out with this:

- Error reporting
- Generating a debug output log
- Outputting text to the display during the test

Error reporting

If any errors or warnings occur when a test is run, a yellow information bar will appear at the top of the ANY-maze screen, informing you of this fact. You'll also find the (test debug output) button is highlighted; if you click on this button, then the Test debug output window will open so you can view the error(s) and/or warning(s) that occurred.

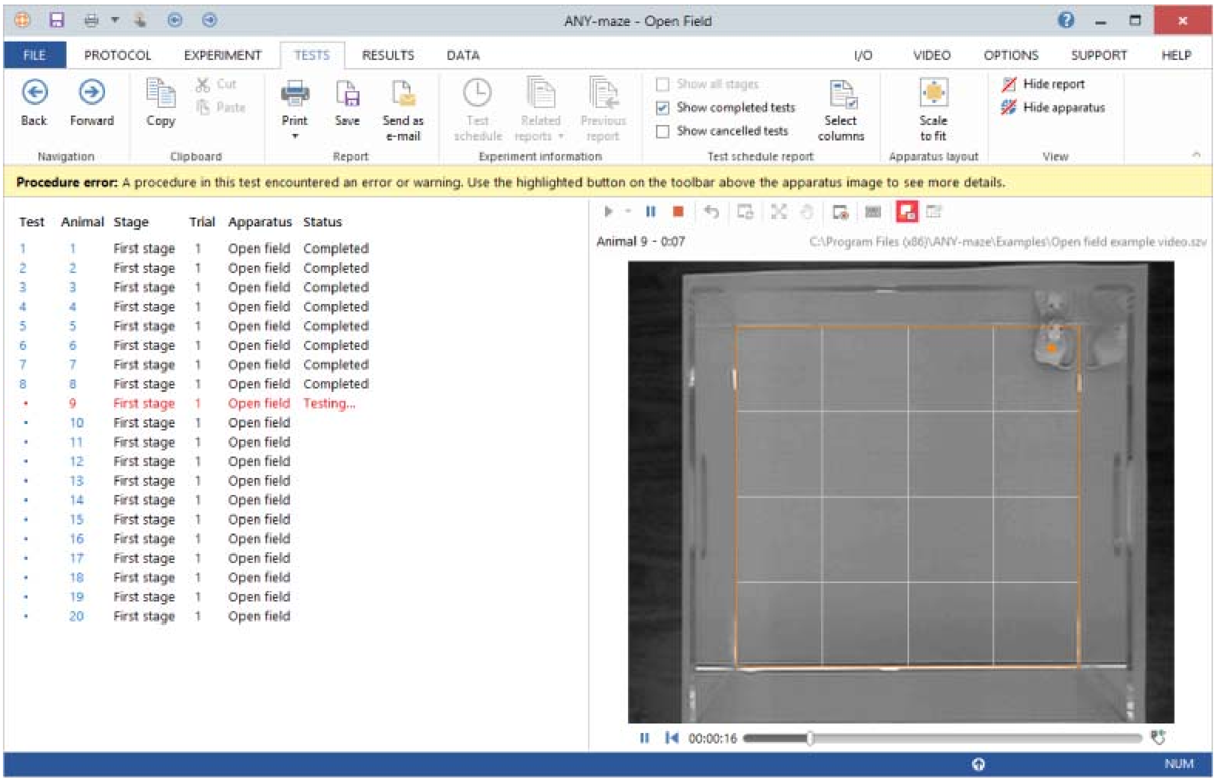


Figure 1. The information bar that appears if a procedure encounters any errors or warnings during the test.

You'll also get a message box displayed at the end of the test. Any errors and warnings will be listed, alongside the name of the procedure and the test clock at the time the error occurred, in the Test details report for the test.

Generating a debug output log

On the Tests page, just above the video picture of the apparatus, is a series of buttons (including the (Start test) button) – if your protocol includes at least one procedure, then you see a 'procedure debug' button (see figure 1) which, when clicked, will cause the debug output window to open. Procedures can write to this window using the *Output text to debug window* action, which can very useful as it allows you to see what's happening inside your procedure as it runs.

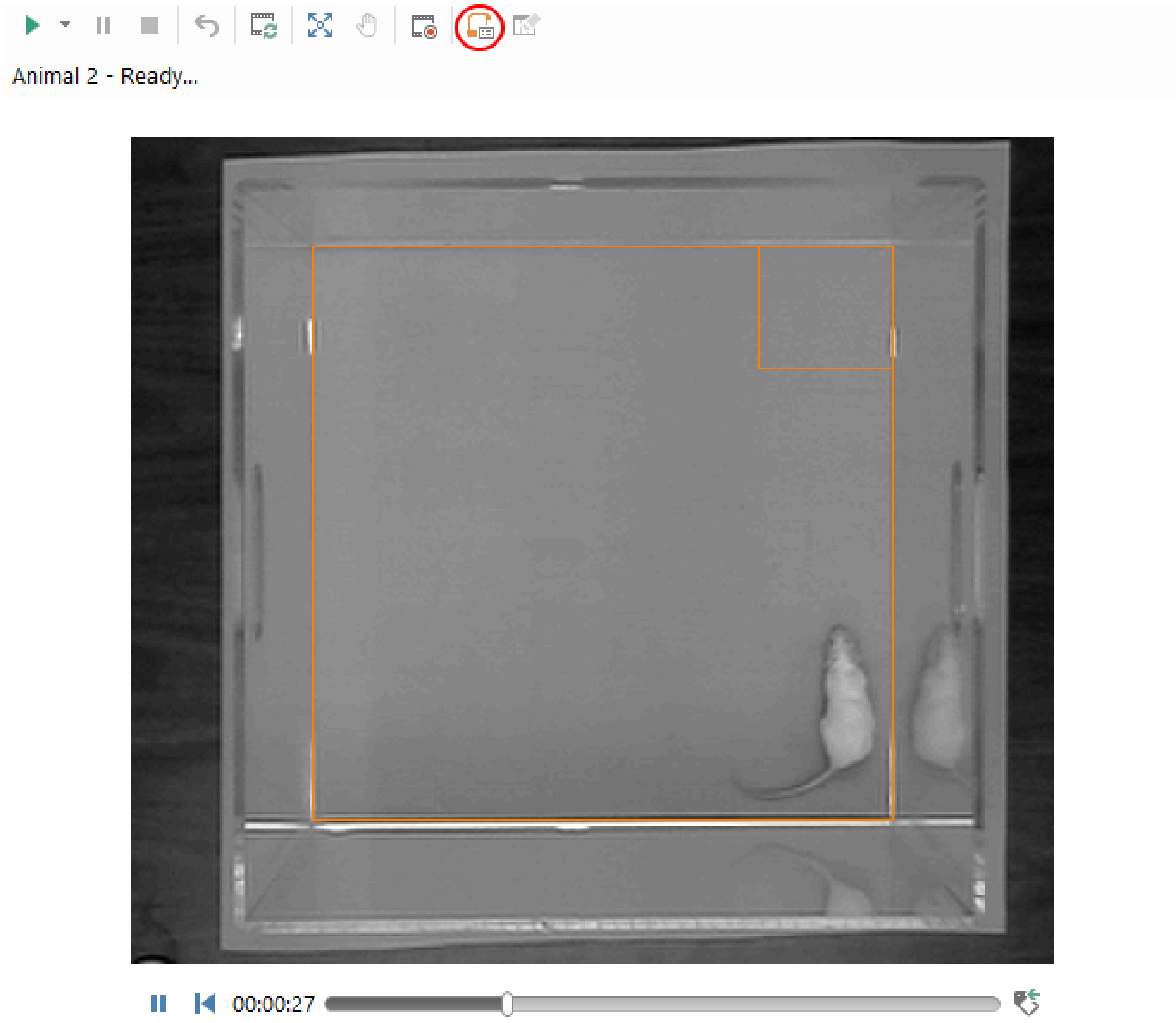


Figure 2. The video picture of the apparatus on the Tests page, *showing the test debug button.*

For example, the following procedure outputs text to show how far through the procedure it has got. If you run a test and press the lever and you DON'T see 'Lever pressed' shown in the debug output, then you'll know that the reason the animal isn't getting a pellet is because the *Wait until Lever activated* is not working for some reason (and not because the pellet dispenser isn't working).

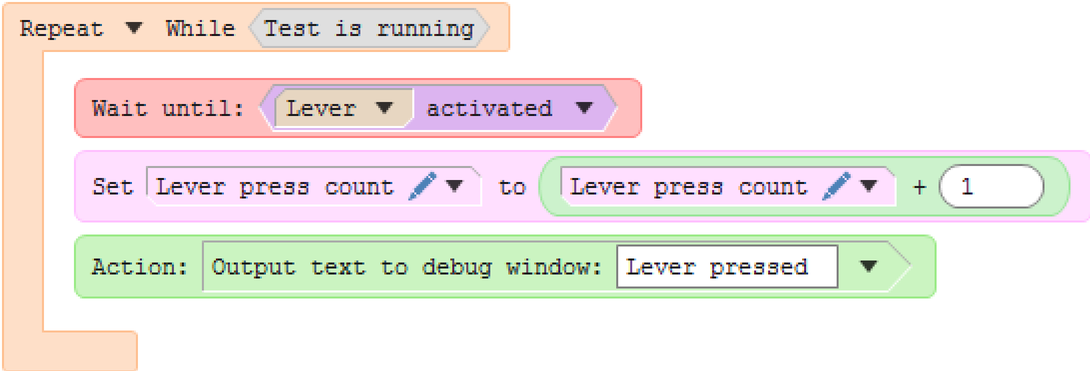


Figure 3. A procedure that uses the Output text to debug window *action to keep track of when the animal presses the lever.*

Running this test will show text in the debug output window:

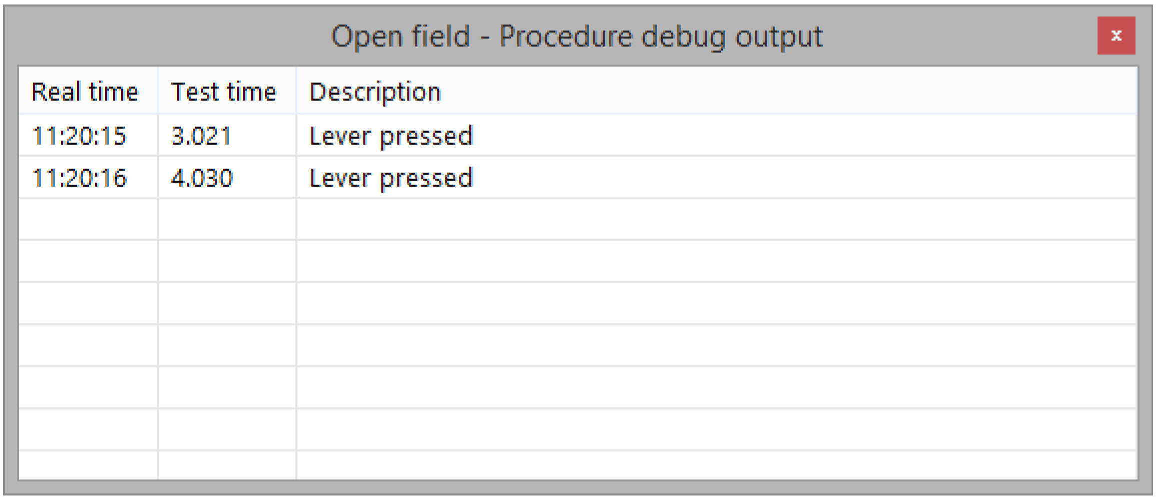


Figure 4. The debug output window, showing some output from the procedure from figure 2.

This window will scroll automatically as new text is added to the end of the list. The debug output does not get cleared when a test ends – when you want to clear it. you have to do so manually, by clicking the clear test debug button (which is shown next to the button).

Outputting text to the display during the test

A procedure can use the *Output text on display* action to display any text that it wants on the screen.

To stop displaying the text, use the *Remove text from display* action.

Any text displayed on the screen in this way will also be included in any video that is recorded during the test - See Recording videos of tests.

While this can be a useful way to see what is going on during the test, using the debug output window is usually better, because you see a history of what has happened during the test, and the output sticks around until the window is explicitly cleared.

See also:

- Actions available to procedures

Procedures and the end of a test

Overview

The lifetime of a procedure in ANY-maze is not necessarily the same as the test for which it runs. A procedure may run through all its statements and then finish, while the test continues running; alternatively, if a procedure is still running when the test ends, the procedure will be stopped.

More information

A procedure may still be running when a test ends, usually because the procedure includes some kind of loop using a Repeat statement or Go to statement. If it's still running, then the procedure will continue executing statements until it reaches a Wait until statement, at which point it will stop running. This is because the procedure knows that no further events can occur after a *Test ends* event, so it will just exit. The exception to this is the *Test continuation* event, which is described below.

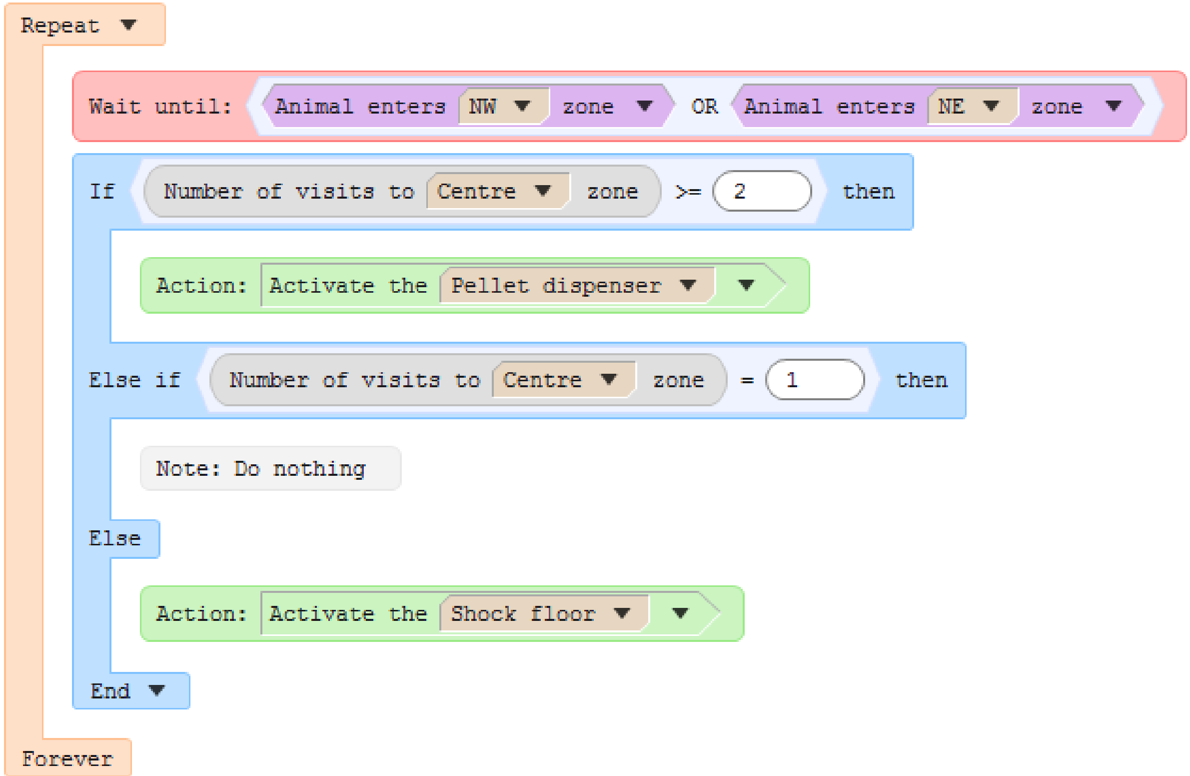


Figure 1. If this procedure was currently executing the If *statement when the test ended, it would carry on evaluating the conditions, and take the appropriate action. Control would then loop back to the* Wait until *statement, at which point, the procedure would stop running as it knows that the test has ended.*

Waiting for the *Test ends* event.

There may be circumstances under which you want to take specific action *only* at the end of the test. For example, you might have a radial arm maze that opens and closes doors in a particular sequence, or according to the animal's movements. When the test ends, you want to open all doors of the maze to leave it in a 'known' state. So how would you achieve this, if the procedure ends when the test ends?

When the test ends, the procedure will receive a *Test ends* event. If that's the event that the procedure is currently waiting for, then it will process that *Test ends* event and run any subsequent statements. Only if it reaches another *Wait until* statement (or there are no more statements in the procedure) will the procedure finally stop. So the following procedure could be used to close the doors of our radial arm maze:

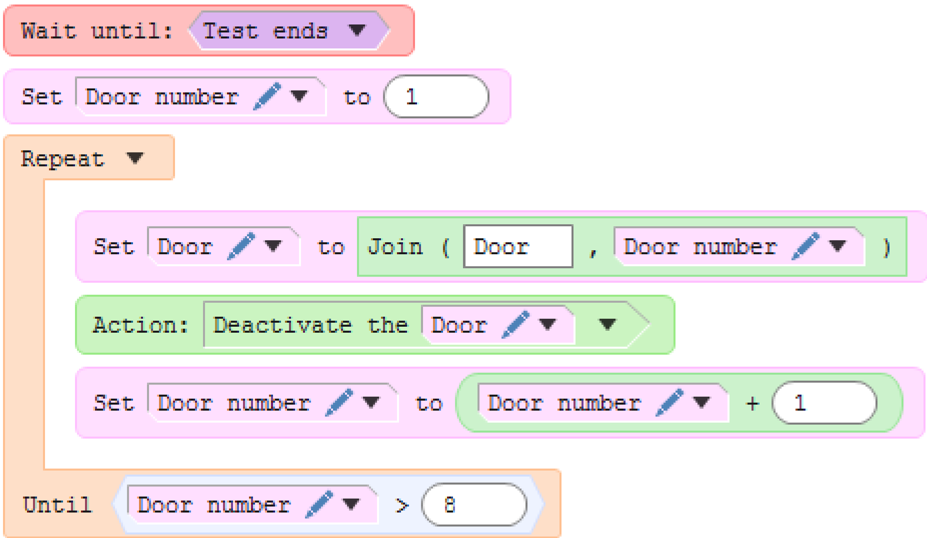


Figure 2. When the test ends, this procedure will run the statements after the Wait until *statement. After repeating the loop 8 times, the procedure will drop out and end anyway.*

Note the use of the 'Join' function to create some text for 'Door 1', 'Door 2', etc. These are the names of the output switches that control the doors of the radial arm maze.

Note that when you create a procedure that waits for the *Test ends* event in this way, it is best to write it as a separate procedure, rather than adding these statements to an existing procedure. This will guarantee that the processing of the *Test ends* event will always occur.

To see why this is necessary, consider the two figures above. Imagine that we appended the procedure in figure 2 to the procedure in figure 1, i.e. the *Wait for Test ends* event was immediately after the end of the *Repeat* statement in figure 1.

What would happen at the end of the test? Well, it would depend on exactly which statement the procedure was executing when the test stopped. The chances are quite high that it is in the first *Wait until* statement, waiting for the animal to enter a zone. If this is the case, the *Test ends* event is received, the procedure looks to see if it's the event it's waiting for (which it isn't), and then, because the test has finished and the procedure is in a *Wait until* statement, the procedure will end. The actions to close the doors are never run.

Even if the procedure is currently executing the *If* statement or the *Repeat* statement when it receives the *Test ends* event, the result will be the same because the statements will always get back to the *Wait until* statement that is waiting for the zone entry.

Keeping the procedure separate, as in figure 2, works because that procedure is *only* waiting for the

Test ends event - it can't get stuck waiting for something else when the test actually does end.

Test continuation

There are some circumstances under which a procedure may decide that it's right to end the test, but you want to manually override this and force the test to continue. For example, you might have a water-maze and a procedure that ends the test when the animal enters the platform zone. Sometimes, the animal might just swim straight over the zone and exit the other side, in which case you might not want this to count as the zone being found; however, the test has already ended because the procedure thinks that the animal entered the zone. In this case, you'd want to tell ANY-maze to resume running the test, allowing it to continue tracking until the animal *really* found the platform.

If you want to allow a test to be manually continued after a procedure has ended it, then you need to use the *End the test, but allow continuation* action rather than the basic *End the test* action. This will allow you 10 seconds after the test has finished, to press the ▶ (Start test) button to force the test to continue.

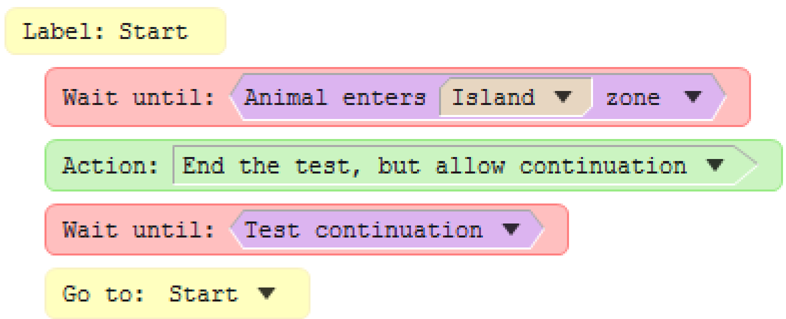


Figure 3. This procedure ends the test, but allows you to force it to continue if necessary.

Note that the procedure in figure 3 differs from the one described in the first walkthrough example, in that it also loops back to the start of the test, in case you *do* force continuation - otherwise the procedure will never go back to waiting for the animal to enter the island again.

The waiting for the *Test continuation* event has the same restriction as for the *Test ends* event that we saw above - i.e. it's best to wait for this event in a separate procedure. An exception to this is the procedure in figure 3 above - since it waits for the *Test continuation* event *immediately* after ending the test, it will always be the next event that the procedure receives, so the procedure in this example will continue running.

See also:

- Actions available to procedures

Preventing a test from starting until an I/O device is ready

Overview

Experiments often rely on the use of external devices, and sometimes need these devices to be in a certain state before the experiment starts. The most common example is probably that of temperature - apparatus with thermal plates might need the temperatures of these plates to be correct before the test should start, or the water in a water-maze might need to be at a certain temperature. ANY-maze allows you to prevent a test from being started until certain conditions have been met.

Details

Two actions are avaiable to ANY-maze procedures which can control whether or not a test can be started:

- *Prevent test start*
- *Allow test start*

These actions can only be used in the shaded area at the top of a procedure, since they are only applicable *before* the test has started.

The *Prevent test start* action will simply cause the (Start test) button to be disabled. You must specify some text when you use this action; this text will be displayed above the apparatus image to explain why the test cannot be started.

Once the relevant condition has been met, you can use the *Allow test start* to remove this text and enable the (Start test) button.

Note that Temperature controllers already have a way of specifying a start temperature (and preventing a test from starting if that temperature is not reached) - see Specifying a temperature controller's start temperature for details on how to do this.

Example

The following example shows a procedure for an experiment where the animal can activate or deactivate some levers. The test must start with the levers in the 'inactive' position; this simple procedure will prevent the text test from starting until the levers have been restored to their inactive positions.

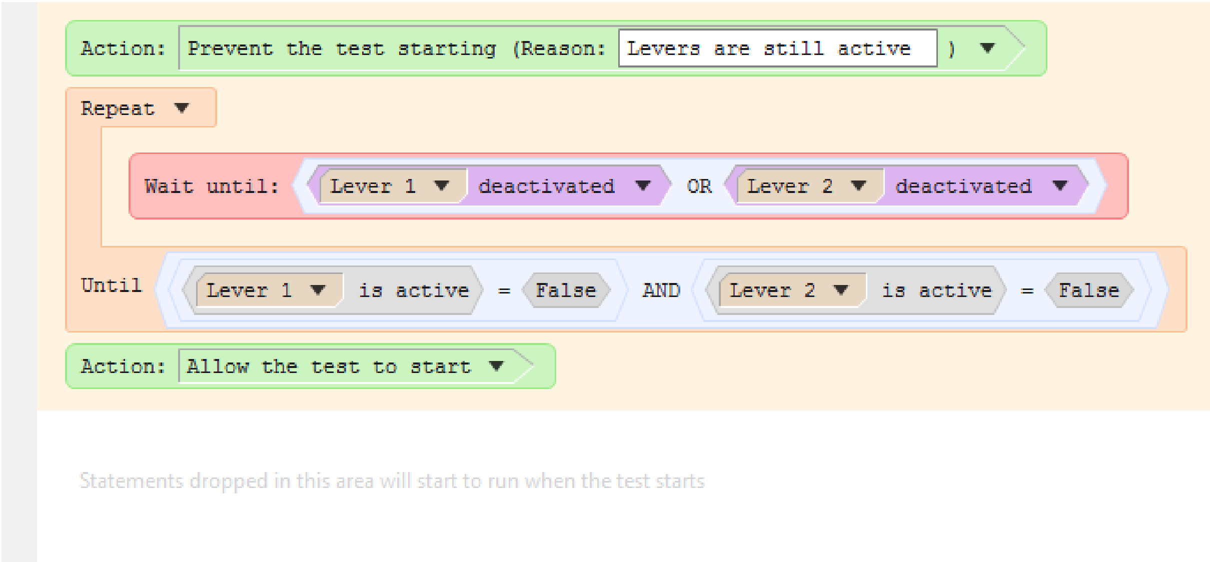


Figure 1. This procedure will prevent the test from starting until the two levers are in the correct position.

Note that the statements in this procedure are all in the shaded area at the top of the procedure, i.e. they are run *before* the test starts.

Sharing and re-using procedures

Introduction

Procedures can be reused in different experiments, or even shared between labs. This is done by *exporting* a procedure from one experiment, and then *importing* the procedure into a different experiment.

Obviously, there are some potential problems with this - if you're trying to share a procedure between two open-field experiments, for example, you might run into difficulties if one experiment has called a zone the 'centre' zone, and the other has called it the 'middle' zone.

This topic describes how to import and export procedures, and gives some advice on how to deal with potential problems that might arise.

Exporting a procedure

Exporting a procedure involves saving it to an XML file on your computer. Each individual statement in the procedure is saved with its own XML 'tag'.

Any references to items in the protocol (e.g. zones, points, sequences etc.) are made by name. So in the case of the water-maze example that is shipped with the ANY-maze software, the resulting XML will look a bit like this (some of the XML has been removed for clarity):

```
<Procedure Name="Found platform" Type="Action">
  <Statement Type="Wait">
    <Condition>
      <pn:4800,0,0><s:l,4800><s:l,5300><s:l,5800><i>
        <EventID>ZONEENTRY</EventID>
        <Zone>
          <Ordinal Type="Zone">Island</Ordinal>
        </Zone>
      </Event>
    </Condition>
  </Statement>
  <Statement Type="Action">
    <Fire>
      <ppn:3500,0,0><s:l,3500><i>
        <Effect>STOPTEST</Effect>
        <ExpressionParam>
          <Ordinal Type="TestEndReason">Found platform</Ordinal>
        </ExpressionParam>
      </Action>
    </Fire>
  </Statement>
</Procedure>
```

Importing a procedure

Importing a procedure involves taking an existing XML or text file and loading it into the procedure editor. The file must contain a specific format that the procedure editor can understand.

Providing the file contains a valid format, a brand new procedure will be created and the file will be loaded into it.

What happens if there are problems?

There are a number of possible problems that can occur when trying to import a procedure from an XML file. If the XML itself is invalid (i.e. it doesn't describe a procedure, or has become corrupted in some way), then you'll be warned with a message and nothing will be imported.

Even if the XML itself is valid, there are still some potential issues. For example, in the XML file above there's a zone called 'Island' (it's been highlighted) - so ANY-maze will attempt to find a zone with that name and link the procedure up to it. However, if the protocol you're importing the procedure into has used the name 'Platform', ANY-maze won't be able to determine which zone to use.

If this happens, you'll be prompted to choose the equivalent items - see Making sure the protocol items in a procedure match your protocol).

Importing procedures with variables

If a procedure contains user-defined variables, then new variables with those names will be created for the newly-imported procedure.

If a procedure is loaded that uses a *shared* variable (i.e. one that is shared between all procedures), and a shared variable already exists with the same name, then the new procedure will simply use the existing variable.

However, if a procedure contains a variable that is *not* shared, but has the same name as an existing shared variable, then the name of the imported variable will be adjusted to be unique (e.g. a variable called 'Count' might be renamed to 'Count (2)'). This is because variable names need to be unique within any given procedure.

See also:

- User-defined variables
- Making sure the protocol items in a procedure match your protocol

Making sure the protocol items in a procedure match your protocol

Introduction

When you're importing a procedure from an XML file - perhaps an example file that you've been sent by ANY-maze technical support, or a procedure that one of your colleagues has created - the procedure might have been exported from an experiment file where the names of the protocol items are different to yours. For example, an exported procedure might contain an I/O input called 'Lever', whereas your protocol has called it 'Switch'.

When this happens, ANY-maze can't import the procedure as it is, so it will ask you how to deal with these items that it doesn't know about. This topic describes how this works.

Details

When you're importing a procedure, and the protocol items in the procedure don't match those in your protocol, ANY-maze will display a window asking you to choose the equivalent items in your protocol.

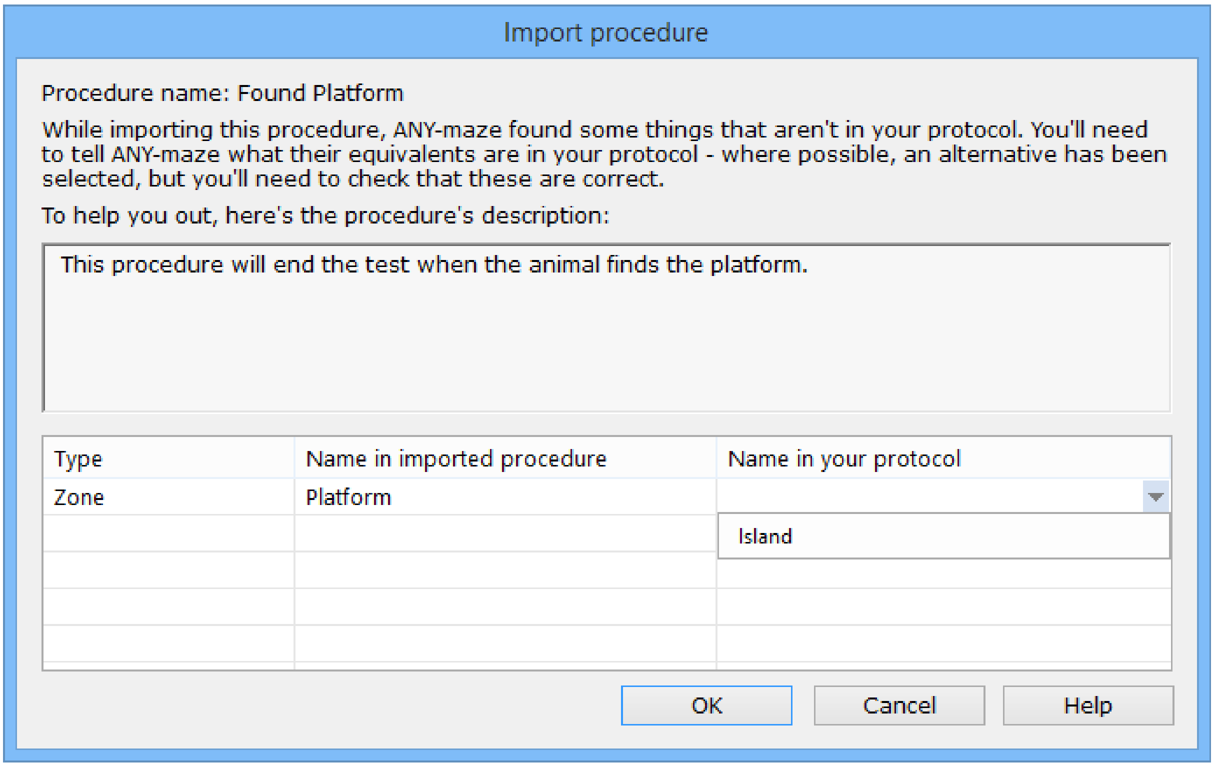


Figure 1. The window used to specify protocol items for an imported procedure.

If the names of the items in the imported procedure already match items of the same type in your protocol, you won't see this window.

The top of the window will show the procedure name, and any notes that were saved with the procedure. Hopefully, these should explain what the procedure does in enough detail to allow you to make choices in the list below it.

The window also shows a list of all the items in the procedure that ANY-maze doesn't know about. This applies to all elements of the protocol that can be referred to by a procedure - zones, sequences, I/O inputs and outputs, etc. For each item, it will list the type of item, and the name of this item in the procedure. The last column allows you to choose the equivalent item in your protocol, from a list of all those available.

If there's only one item of this type in your protocol, then ANY-maze will choose this as the default. Make sure you check these though, as it may simply be that your protocol doesn't yet contain the equivalent item. You don't *have* to select an item in this third column; however, if you don't, the resulting procedure will contain an empty space where that item ought to be. This will be flagged as an error, and will need to be resolved before you can use the procedure.

At any time, you can click the *Cancel* button to cancel the procedure import, or *OK* to import the procedure with as many replacements as you have specified.

Making sure the protocol items in a procedure match your protocol

Introduction

When you're importing a procedure from an XML file - perhaps an example file that you've been sent by ANY-maze technical support, or a procedure that one of your colleagues has created - the procedure might have been exported from an experiment file where the names of the protocol items are different to yours. For example, an exported procedure might contain an I/O input called 'Lever', whereas your protocol has called it 'Switch'.

When this happens, ANY-maze can't import the procedure as it is, so it will ask you how to deal with these items that it doesn't know about. This topic describes how this works.

Details

When you're importing a procedure, and the protocol items in the procedure don't match those in your protocol, ANY-maze will display a window asking you to choose the equivalent items in your protocol.

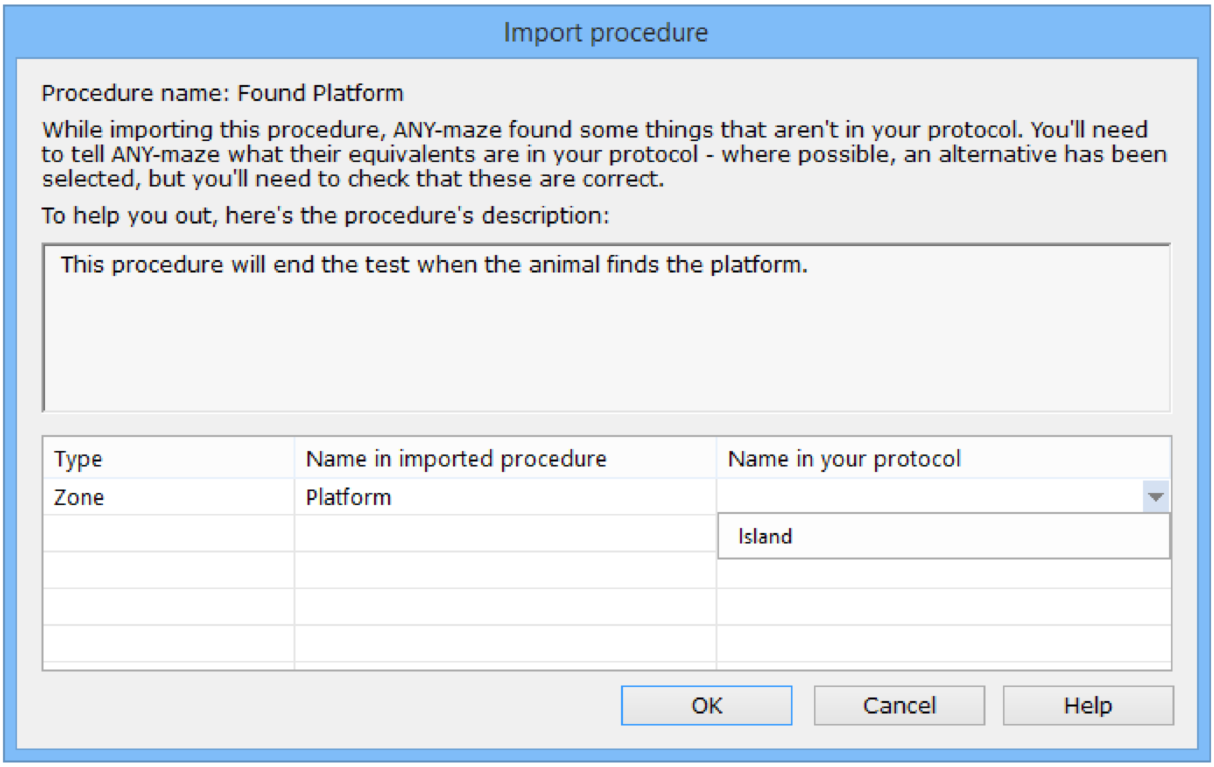


Figure 1. The window used to specify protocol items for an imported procedure.

If the names of the items in the imported procedure already match items of the same type in your protocol, you won't see this window.

The top of the window will show the procedure name, and any notes that were saved with the procedure. Hopefully, these should explain what the procedure does in enough detail to allow you to make choices in the list below it.

The window also shows a list of all the items in the procedure that ANY-maze doesn't know about. This applies to all elements of the protocol that can be referred to by a procedure - zones, sequences, I/O inputs and outputs, etc. For each item, it will list the type of item, and the name of this item in the procedure. The last column allows you to choose the equivalent item in your protocol, from a list of all those available.

If there's only one item of this type in your protocol, then ANY-maze will choose this as the default. Make sure you check these though, as it may simply be that your protocol doesn't yet contain the equivalent item. You don't *have* to select an item in this third column; however, if you don't, the resulting procedure will contain an empty space where that item ought to be. This will be flagged as an error, and will need to be resolved before you can use the procedure.

At any time, you can click the *Cancel* button to cancel the procedure import, or *OK* to import the procedure with as many replacements as you have specified.

Procedure measures

Unlike almost all other measures in ANY-maze, procedure measures do not update to reflect post-test changes in the procedure definition. Therefore, procedure measures are always based on the procedure that was in force when the test was performed.

Procedures don't have any measures defined themselves, but you can create your own measures by using *result variables*. These are essentially numeric values that your procedure can set as it runs, and these values can then be included in ANY-maze result analysis. For more information, see Creating and using result variables.

Once a procedure has been set up to use a result variable, a number of measures will be available for analysis:

If the result variable is set up to be noted 'only once, at the end of the test', then the following measure will be available for it:

- Value

If the result variable is set up to be noted 'every time it is changed by the procedure' or 'only when explicitly set', then the following measures will be available for it:

- Average value
- Maximum value
- Minimum value
- Sum of values
- Count of values

The following is a description of these measures:

Value

Description	Reports the value of the result variable at the end of the test.
Calculation method	The value of the result variable at the end of the test. If the value is never noted, the value will be reported as 0 (since result variables are initialised to 0 at the start of a test).
Analysis in zones	The value of this result variable, if the animal is in the zone at the end of the test. If the animal is not in the zone at the end of the test, the value will be undefined.

Analysis across time	This measure cannot be analysed across time.
Units	No units defined; this is dependent on how the procedure calculates the value of this result variable.
Notes	This measure is only available if the result variable is set up to be noted 'only once, at the end of the test'.

Average value

Description	Reports the average value of the result variable.
Calculation method	Simple average of the values noted by the procedure during the test.
	If the value is never noted, the average will be undefined.
Analysis in zones	Average of the values reported while the animal is in the zone.
	If the value is never noted while the animal is in the zone, the average will be undefined.
Analysis across time	This measure can be analysed across time. The result is the average of the values reported during the time period.
	If the value is never noted during the time period, the average will be undefined .
Units	No units defined; this is dependent on how the procedure calculates the value of this result variable.
Notes	This measure is only available if the result variable is set up to be noted 'every time it is changed by the procedure' or 'only when explicitly set'.

Maximum value

Description	Reports the maximum value of the result variable noted by the procedure during the test.
Calculation method	The largest value of the result variable noted by the procedure during the test.
	If the value is never noted, the maximum value will be 0 (since result variables are initialised to 0 at the start of a test).
Analysis in zones	The largest value reported while the animal is in the zone.
	If the value is never noted while the animal is in the zone, the maximum value will be undefined.
Analysis across time	This measure can be analysed across time. The result is the largest of the values reported during the time period.
	If the value is never noted during the time period, the maximum value will be

undefined.

Units No units defined; this is dependent on how the procedure calculates the value of this result variable.

Notes This measure is only available if the result variable is set up to be noted 'every time it is changed by the procedure' or 'only when explicitly set'.

Minimum value

Description Reports the minimum value of the result variable noted by the procedure during the test.

Calculation method The smallest value of the result variable noted by the procedure during the test.

If the value is never noted, the minimum value will be 0 (since result variables are initialised to 0 at the start of a test).

Analysis in zones The smallest value reported while the animal is in the zone.

If the value is never noted while the animal is in the zone, the minimum value will be undefined.

Analysis across time This measure can be analysed across time. The result is the smallest of the values reported during the time period.

If the value is never noted during the time period, the minimum value will be undefined.

Units No units defined; this is dependent on how the procedure calculates the value of this result variable.

Notes This measure is only available if the result variable is set up to be noted 'every time it is changed by the procedure' or 'only when explicitly set'.

Sum of values

Description Reports the total sum of the result variable's values during the test.

Calculation method Every time the result variable is noted, the value is added to the ongoing sum.

If the value is never noted, the sum of values will be 0 (since result variables are initialised to 0 at the start of a test).

Analysis in zones The sum of the values reported while the animal is in the zone.

If the value is never noted while the animal is in the zone, the sum will be 0.

Analysis across time This measure can be analysed across time. The result for a time period is the total sum of all values of the result variable that are noted during the time period.

	If the value is never noted during the time period, the sum of values will be 0 (since a user-defined variable is always initialised to 0).
Units	No units defined; this is dependent on how the procedure calculates the value of this result variable.
Notes	This measure is only available if the result variable is set up to be noted 'every time it is changed by the procedure' or 'only when explicitly set'.

Count of values

Description	Reports the total number of times during the test that the result variable was noted.
Calculation method	Each time the result variable is noted, a counter is incremented. If the value is never noted, the count of values will be 0.
Analysis in zones	Counts the number of times the result variable is noted while the animal is in the zone. If the value is never noted while the animal is in the zone, the count will be 0.
Analysis across time	This measure can be analysed across time. The result for a period is the number of times the result variable is noted during the period. If the value is never noted during this time period, the count of values will be 0.
Units	None; this is simply a count of the number of times that a result variable was set by a procedure.
Notes	This measure is only available if the result variable is set up to be noted 'every time it is changed by the procedure' or 'only when explicitly set'.

See also:

- Result variables
- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures

- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Event measures
- Virtual switch measures

Time markers

Introduction

Time markers are specific time points in a test that can be set using an Action. Setting a time marker inserts a specific, named marker point within the set of results for a given test, and that time marker can then be used for analysis.

For example, you may have an animal in an environment where it can move freely between zones. The animal has a sensor attached which monitors some aspect of its brain activity. At some point, it will move into a zone where it is given a stimulus, and you want to measure the value of the sensor at the point that it received this stimulus (i.e. at the point where it moves into the zone). In fact, you would probably want to start monitoring the sensor for a short time *before* the animal enters the zone, and see what effect the stimulus has on its brain activity.

So how could you achieve this, given that you have no idea when the animal is actually going to enter the zone? Well, this is what *Time markers* are for.

You would set up a procedure to wait until the animal entered the zone in question, and then use an action to set a time marker at this point. This time marker can then be used in the rest of the protocol in one of two ways:

- For analysis of time periods
- As a point for measurement of a baseline value for a signal input

Creating a time marker

You can create a time marker in one of three ways:

- Click the *Create a time marker* button on the ribbon bar. This will take you to the Time marker settings window, where you should enter the name of the time marker (for example, 'Stimulus starts') and click OK.
- Select the procedure editor's *Actions* tab, open the *Analysis* group, and drag the *Set a time marker* action into your procedure at the point that you require it to be set. Once you've done this, click on the button to edit the name of the time marker.
- On the Variables tab, right click anywhere and select *Create a time marker* from the resulting menu. Again, this will open the Time marker settings window.

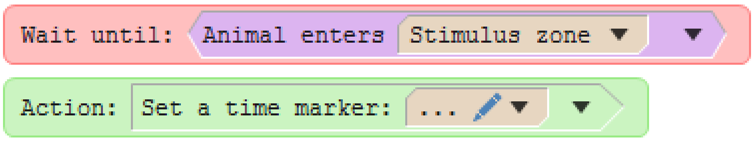


Figure 1. A procedure that will set a time marker.

When the procedure editor is in 'Simple' mode, the Create a time marker *button is not shown.*

Once you've set up a time marker in a procedure, you can use it in the protocol for analysis of a time period, or with a signal input.

Using time markers for analysis of time periods

Once you've used the procedure editor to create a time marker, you can change to the Protocol page and select *Analysis across time* in the protocol list. Right click to create a new time period, and select *This time period is based on the following time marker*. You'll find in the drop-down list the time marker that you just created in the procedure editor, and you can choose when you want your time period to start based on the time marker (you can start the time period on, before, or after the time marker that you set).

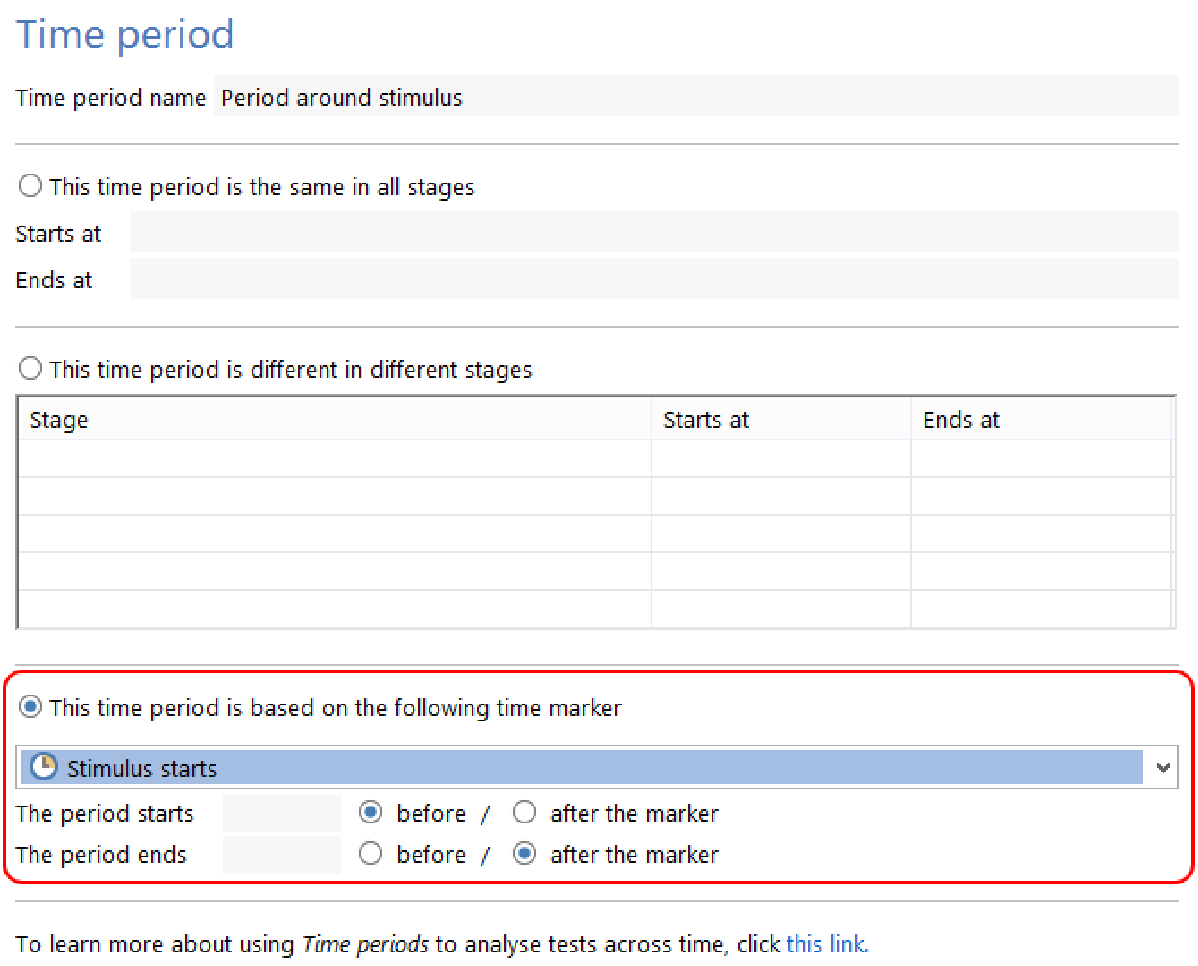


Figure 2. Using the time marker on the Protocol page, to start a time period.

For more details, see Specifying the start and end of a time period that is based the occurrence of a time marker.

Using time markers with signal inputs

In the example in figure 1 above, if you were looking to measure the change in the signal *after* the stimulus started, you wouldn't care about the actual value of the signal input beforehand (since its baseline value is likely to be slightly different across the animals in your test). Rather, you would be looking for the *difference* in value between the input after the stimulus, and before it.

But when does the baseline value for the signal get measured? ANY-maze allows you to use the average value of the signal input, up to the point defined by your time marker. To set this up, select the *Signal inputs* item in the protocol list, and select the signal input that you're interested in. Ensure that *Baseline is signal's average value until the following time marker* is checked, and then select the time marker from the drop-down list below it:

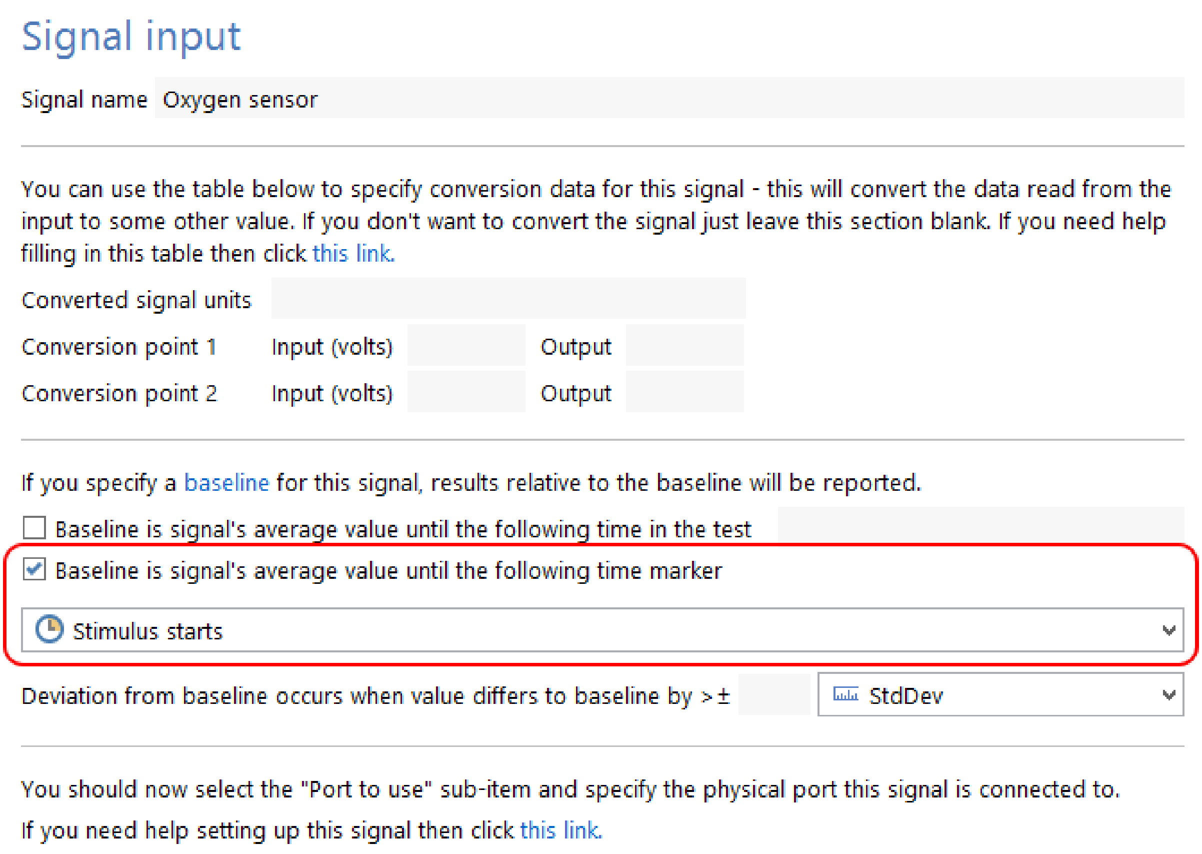


Figure 2. Using the time marker on the Protocol page to determine the baseline

for the signal input.

For more details, see Setting up a signal's baseline.

Understanding the changeover from Events and Actions to Procedures

Introduction

ANY-maze version 5.10 introduced Procedures as a replacement for Events and Actions. Procedures, which are described in detail in the Introduction to procedures topic, are more flexible than Events and Actions, as well as being simpler to set up and understand.

All new experiments that are created in versions 5.10 and above automatically use Procedures instead of Events and Actions, but what about existing experiments, or new experiments into which you load existing protocols? These are questions that this topic addresses.

- When are Procedures used rather than Events and Actions?
- Can I switch back to use Events and Actions in an experiment that is using Procedures?
- When are Events and Actions used rather than Procedures?
- Can I switch to use Procedures in an experiment that is using Events and Actions?
- How do I switch to use Procedures in an experiment that is using Events and Actions?

When are Procedures used rather than Events and Actions?

1. Procedures are always used in new experiments created in version 5.10 of ANY-maze and above.
2. Procedures are always used in new experiments created in version 5.10 of ANY-maze and above into which you load a protocol that includes no Events or Actions.

Can I switch back to use Events and Actions in an experiment that is using Procedures?

No. An experiment that is using Procedures, even if you've not actually added any procedures to it, will always use procedures; you can't change back to use the old Events and Actions system.

When are Events and Actions used rather than Procedures?

1. Events and Actions are used in experiments created in versions prior to 5.10.
2. Events and Actions are used in new experiments (created in version 5.10) into which you load a protocol (created using a version prior to 5.10) that includes Events or Actions.

Can I switch to use Procedures in an experiment that is using Events and Actions?

Yes. You can sometimes switch from using Events and Actions to using Procedures, even in experiments or protocols that were created in versions prior to 5.10. The rules about this are:

1. The experiment cannot have been started - i.e. it must not include any performed tests.
2. Any Events and Actions defined in the protocol will be deleted.

How do I switch to use Procedures in an experiment that is using Events and Actions?

1. Open the experiment.
2. Switch to the Protocol page.
3. Select the *Events* group in the protocol list.
4. If the experiment has not been started then you will see a button titled *Change this protocol to use Procedures*. Click it (if the experiment has been started, then the button simply won't be displayed).
5. A message will be displayed warning you that the change to use procedures will cause any Events and Actions defined in the protocol to be deleted, and that the change to procedures is irreversible. If you want to continue, then click *Yes*.
6. Any Events or Actions will be removed from the protocol and it will change to use Procedures - so you'll see a *Procedures* group in the protocol list rather than an *Events* group.

See also:

- An introduction to procedures
- An introduction to events and actions

Variable settings

Introduction

The 'Variable settings' window is opened when you create a new variable for a procedure by clicking on the *Create variable* button, or when editing a variable using the button.

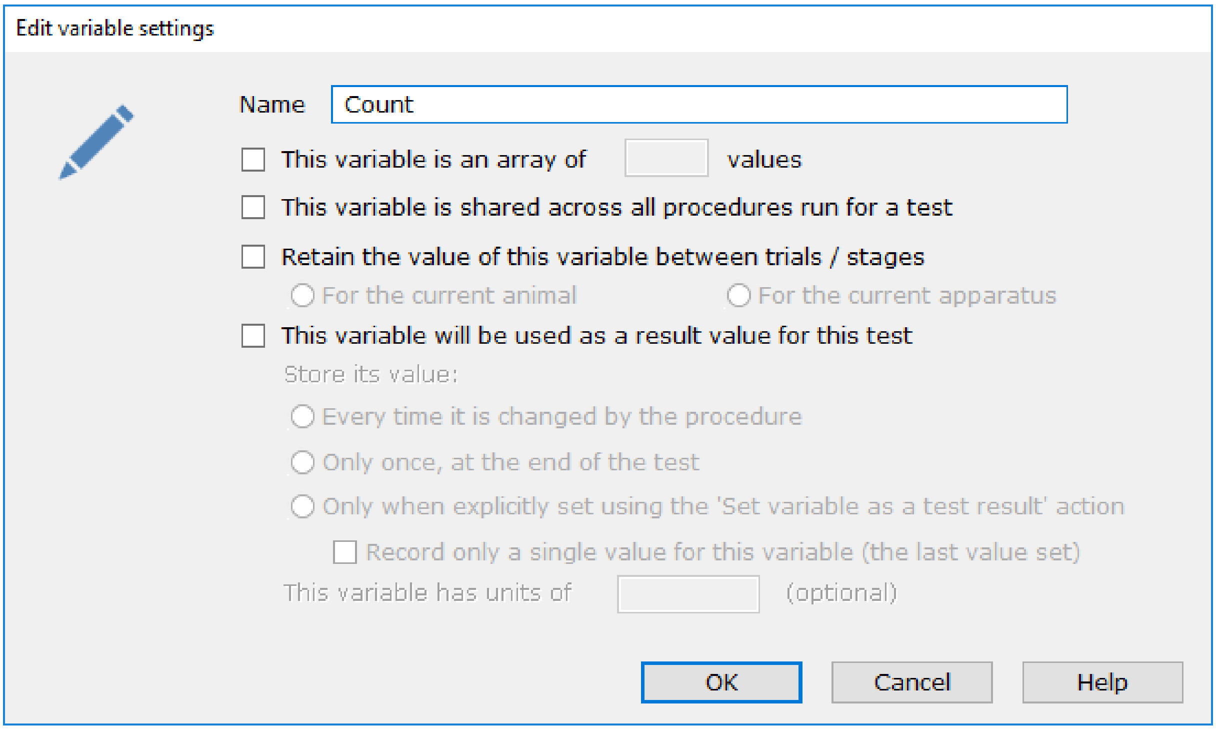


Figure 1. The variable settings window.

Details

The following controls are available on the variable settings window:

- Name
- This variable is an array of ... values

- This variable is shared across all procedures run for a test
- Retain the value of this variable between trials / stages
- This variable will be used as a result value for this test

Name

Enter the name of the variable. The name must be unique amongst all variables used by this procedure (either variables that are only used in this procedure, or those that are shared between procedures and are therefore automatically available to this procedure). ANY-maze won't let you click the *OK* button to save the variable unless the name you've entered is unique.

There is no limit to the number of characters you can use for a variable name, and it can contain any alphanumeric characters, including spaces. You'll usually want to keep the name fairly short, while still being meaningful.

If you edit the name of an existing variable, then ANY-maze will update everywhere in the procedure that you've used this variable to reflect the new name.

This variable is an array of ... values

Specify whether you want the variable to be an 'array' of values, rather than a single value. This can be useful if you want to store multiple values for something - for example, an array of tone frequencies that you want to play out of a speaker in a certain order.

For more details, see User-defined variables.

This setting is off by default.

This variable is shared across all procedures run for a test

Usually, you'll only need a variable to store some kind of internal value for the procedure it's created for. However, there are circumstances where you might want to share this state across multiple procedures that are running for the same test - for example, you might have one procedure that sets a variable's value, and then another that checks the value.

For more details, see User-defined variables, or the worked example using I/O to see an example.

This setting is off by default.

Retain the value of this variable between trials / stages

A variable will usually hold some internal state for an individual test - i.e. a specific trial for an animal. By default, a variable is reset at the start of each trial. If the variable is used as a numeric value, its value will be set to 0; if it's a True/False variable, it will be set to False, and if it's a text variable, it will contain '0'.

There are some circumstances under which you may need to know some information about the animal's previous trial - for example, you may decide on the location of a movable zone based on the zone that the animal visited last in its previous trial. In this case, you would turn on this option to store the value of the variable at the end of a test, *For the current animal,* and therefore make it available to the next trial for the same animal. Note that this storage of variables will apply even across stages - so, for example, if a variable's value is stored in the last trial of a *Training* stage, that variable will retain the same value at the start of the animal's first trial for its subsequent *Treatment* stage.

Similarly, you can select to retain the value of the variable *For the current apparatus* - this means that you can store some information about the state of the apparatus between tests in that apparatus (for example, if you need to record which area of the apparatus an animal finished in, so you can determine where the next test in that apparatus should start).

For more details, see User-defined variables.

For an example of how to retain the state of an I/O element between an animal's trials, see here.

This setting is on by default.

This variable is not *shared between animals, only between different trials for the same animal.*

This variable will be used as a result value for this test

Variables are useful within a procedure, to maintain some internal state and help in the running of the procedure, but what if you're using a variable to calculate some value that you'd like to use in your analysis?

In this case, you would set up the variable to be used as a 'result variable'; the name of the variable will then appear in the Results, reports and data section of the protocol list, or in lists of available measures on the Results page.

Once you've checked this option, you need to choose when this result variable's value will actually be noted:

- every time it is changed by the procedure
- only once, at the end of the test
- only when explicitly set using the *Set variable as a test result* action

The middle of these options ('only once...') will simply make a note of the value of the variable when the test ends. Since there's only one value, the only measure available for analysis will be 'value'.

The first and last of these options will potentially result in the variable's value being noted several times during a test. This will make more measures available - maximum, minimum, average, etc. See Procedure measures for more details of the result available. If you're setting the variable explicitly using an action, but you're only interested in a single value for this result variable, you can check the option *Record only a single value for this variable* - this will only use the *last* value that the procedure sets, and will make this available in results as a single 'value' measure.

An optional unit can be specified for this variable. This has no effect other than to be given alongside the result variable's name in reports and on the Results page and Data page.

For more details on the use of result variables, see Result variables.

See also:

- User-defined variables
- Result variables
- Procedure measures

Time marker settings

Introduction

Time markers are named time points inserted into a test's results by a procedure, using the *Set a time marker* action. They can then be used to define Time periods (or to define Signal inputs' baselines).

For full details, see the Time markers topic.

Detail

The only thing you need to give a time marker is a name. This needs to uniquely identify the marker point, in case there is more than one marker point set during a test.

Once you've created a time marker, this marker can be reused by multiple procedures if required.

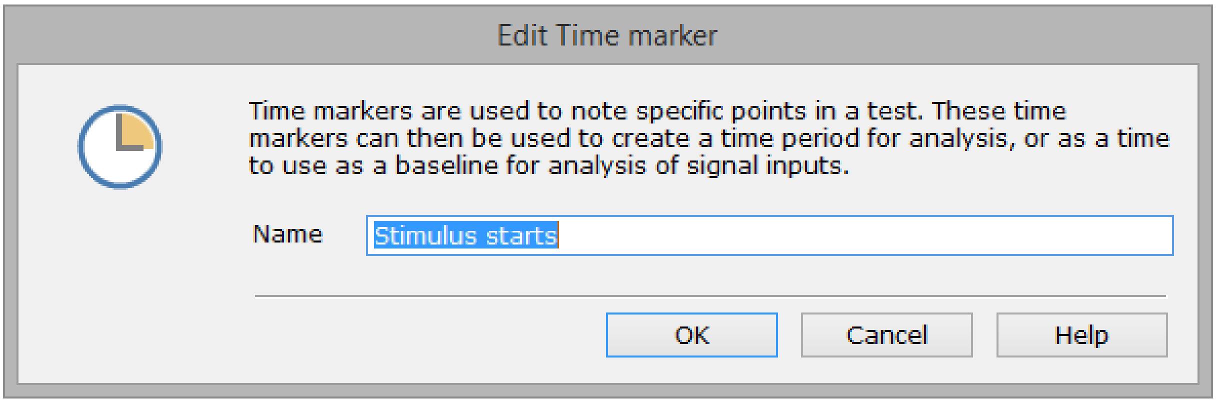


Figure 1. The Time marker settings *window.*

Limitations

The name of a time marker is limited to 64 characters, and must be unique. If the name you have entered is not unique, ANY-maze will prevent you saving the time marker using the *OK* button. If you edit the name of an existing time marker, the name will be updated everywhere that it is used in your procedures.

Test end reason settings

Introduction

Test end reasons are set by a procedure using the *End the test (specifying reason)* action. They can be used to filter tests that end for a specific reason, or can be used as an independent variable in the analysis of results.

Detail

The only thing you need to give a test end reason is a name. This needs to be carefully considered, as not only is it noted as the reason for the end of the test, but can also be used in the analysis of results.

Once you've created a test end reason with a name, this reason can be reused by multiple procedures if required.

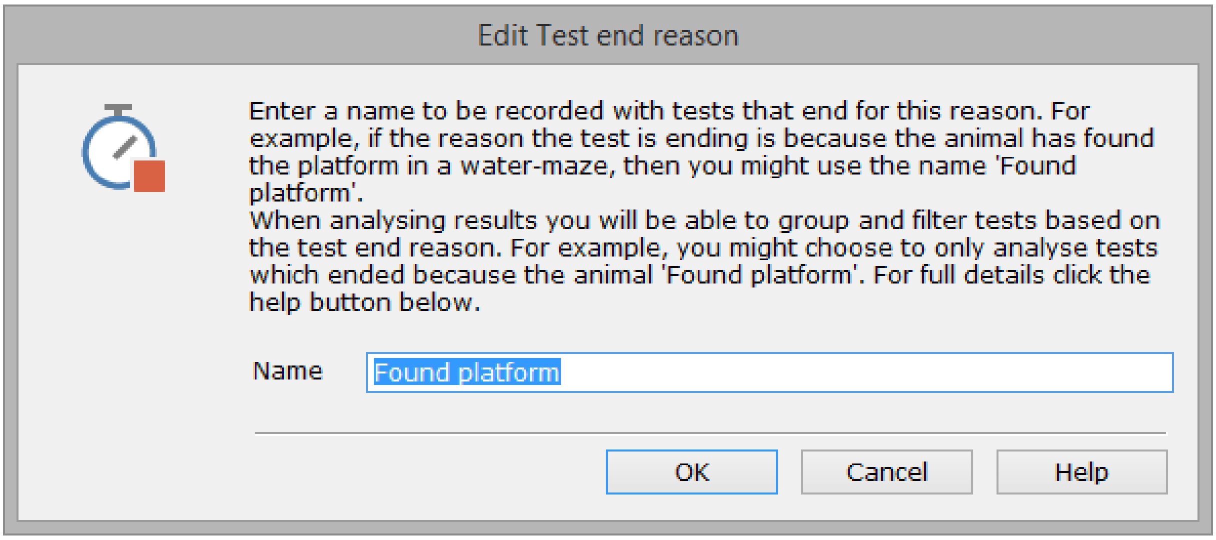


Figure 1. The Test end reason settings *window.*

Limits

The name of a test end reason is limited to 64 characters. It must be unique within this protocol; ANY-maze won't let you save the test end reason using the *OK* button if the name is not unique. If you edit the name of an existing test end reason, the name will be updated everywhere that it is used in your procedures.

Import procedure

Overview

When importing a procedure in ANY-maze, the *Import procedure* window will be displayed. Here you can navigate to the folder that contains the procedure you want to import, and select the procedure file.

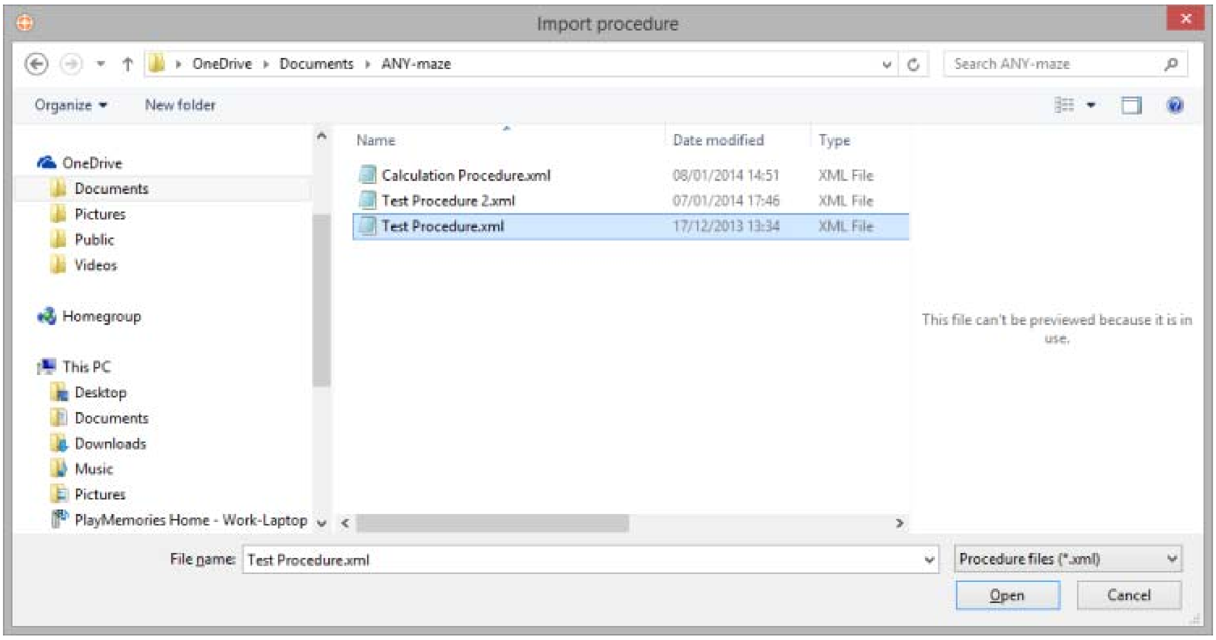


Figure 1. The Import procedure *window.*

Details

You will probably be familiar with this window already, as it's based on the standard 'open file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging *the bottom right corner with the mouse.*

Files list	The files list shows the folders and procedure files stored in the current location. You can click a file to select it, in which case, clicking the *OK* button will open the file, or you can simply double click a file to open it without having to click OK. You can also open a folder, so you can see what's inside it, by double clicking the folder in this list.
File name	You can use this field to enter the name of the file to open; however, it's usually much easier just to click a file in the *Files list*.
Type of file	You can use this list to choose what types of files are shown in the *Files list*. By default, the list shows XML files, but you can change it to show text files, or simply all files irrespective of their type. Note, however, that ANY-maze can only open valid procedure files.

See also:

- Sharing and re-using procedures

Export procedure

Overview

When you export a procedure in ANY-maze, the *Export procedure window* will be displayed. Here you can provide a name for the procedure, and select where on your computer or network you'd like it to be saved.

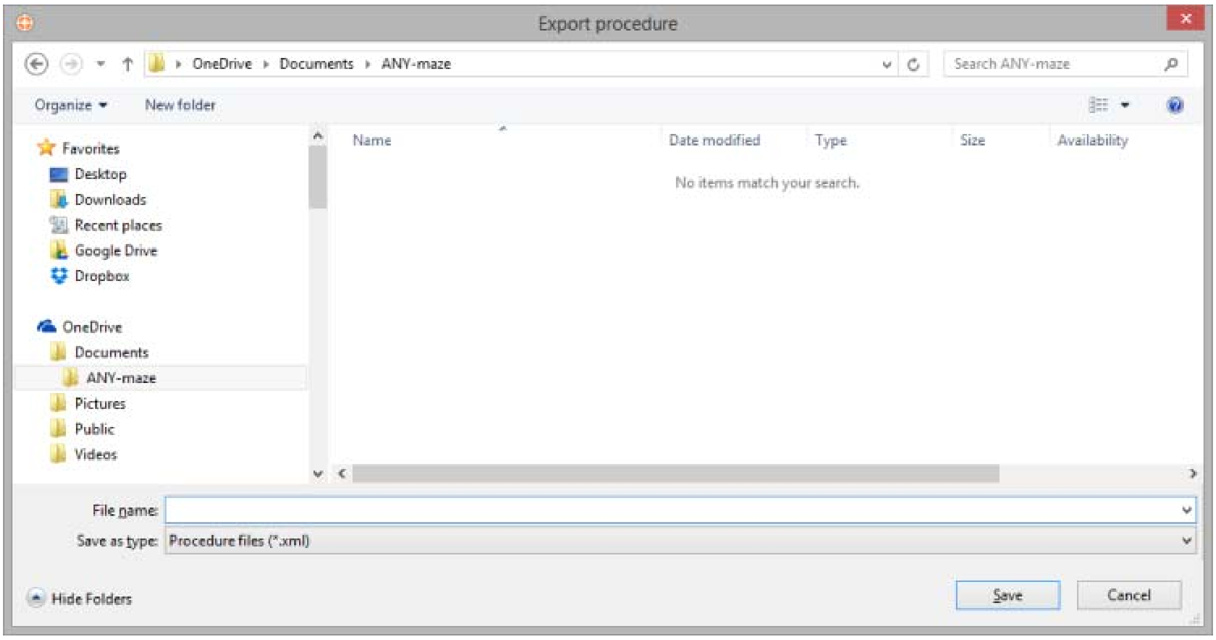


Figure 1. The Export procedure *window.*

Details

You will probably be familiar with this window already, as it's based on the standard 'save file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging *the bottom right corner with the mouse.*

Files list The files list shows the folders and procedure files stored in the current location. You can double click a folder to open it, and you can click on a file to transfer its name to the *File name* field.

File name Use this field to enter a name for the procedure file. ANY-maze will automatically fill in the procedure's name, assuming you've given it one, but you can delete this and type in something else if you wish to.

*File names can be up 255 characters long but cannot contain any of the following characters: \ / : * ? " < > |. If the procedure's name includes any of these characters, then they'll be replaced by a space when the file name is copied to this field.*

Type of file Procedures are always exported in XML format, so you can ignore the file types listed.

See also:

- Sharing and re-using procedures

Events and actions

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Events and actions are used to detect specific situations during a test (events) and to take some action when they occur. For example, you could use an event to detect entry into the Island zone in a water-maze and define an action to end the test when this occurs. A protocol can include any number of events and actions.

For more information about events and actions, refer to the topics below.

- An introduction to events and actions
- Setting up an event
- Setting up an action
- Editing an event or action
- Deleting an event or action
- Event measures

An introduction to events and actions

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Introduction

Events and actions provide a method by which you can detect specific situations that occur during a test (events) and take some action because of them. For example, you could:

- Detect when the animal enters a zone, and end the test when this occurs (see figure 1);
- Detect when the animal is immobile for a certain amount of time, and pause the ANY-maze video recorder; unpausing it when the animal becomes mobile again;
- Detect when the animal activates an input switch, and turn on a switch output for a certain duration - perhaps to activate a shocker.

Figure 1. The steps that would occur in a water-maze, where entry into an island zone is to end the test.

Detecting events

The intention of ANY-maze *events* is to detect when a particular situation occurs during a test. To this end, ANY-maze allows you to base an event on a wide range of different things, including,

- The actual performance of the test; e.g. the test starting.
- Time; for example, a set duration since the test started or a specific time of day.
- Activity and inactivity of the animal in the apparatus as a whole and also in individual zones.
- Mobility and immobility of the animal in the apparatus as a whole and also in individual zones.
- Entry, exit, permanence, etc. in a zone.
- The occurrence, non-occurrence, etc. of a sequence
- The occurrence, duration, etc. of a 'key behaviour' either in the apparatus as a whole or in an individual zone.
- The activation, deactivation, etc. of an input switch either in the apparatus as a whole or in an individual zone.

Performing actions

The purpose of an *action* is to do something when an event occurs. What the action actually does is defined by you, and could be:

- To end, pause, or unpause the test.
- To start, stop, pause, or unpause the ANY-maze video recorder.
- To add a label to the video of the test (so you can quickly find the event in the recording).
- To play a sound or to turn the ANY-maze white noise generator on or off.
- To turn a switch on or off.
- To activate or reset another event.

In fact, a single action can cause any number of these things to occur - so it might add a label to a video, play a sound, *and* activate a switch.

Detecting multiple events

An obvious question about the events/actions model used in ANY-maze is to ask why the system includes these two elements separately - why not simply have a single item which, when triggered, performs the action?

In fact, there are two reasons why the system works this way. First, it allows you to define a number of different events, any of which can trigger a single action - e.g. the action will occur if event A or Event B or event C occurs. And secondly, it allows you to define multiple events, all of which must occur in order to trigger an action - e.g. the action will only occur if event A and Event B and event C occur.

For example, you might want to end the test if the animal enters a certain zone but only if it does this after 1 minute of the test has passed. In this case, you could create an event that would be triggered 1 minute after the test start and another that would be triggered when the animal enters the zone. You would then create an action to end the test but set it to occur only if BOTH of the events were 'active' (i.e. they had both been triggered).

In fact, you might spot a problem with this example - if the animal enters the zone during the first minute, then the action WON'T be triggered because the '1 minute' event is not active; but when the end of the first minute occurs, the action WOULD be triggered because now both events would have occurred. In other words, the test would end after one minute provided the animal had visited the zone during that minute - and this is NOT what we intended.

The solution to this problem is related to how events are *reset*. Basically, after an event has been triggered, it is said to be *active* and it remains in this state until it's *reset* which can be done either automatically, by the system, or 'manually' by an action. In the case where an event is reset automatically, the system first processes all the effects of the event and then resets the event so it's no longer active. In contrast, a manual reset event has its effects processed and is then left in an active state - it's only reset if some action explicitly resets it.

Let's see how this can be used to fix the problem with the above example: What we would want to do is to set the 'Animal enters the zone' event to be automatically reset, but the '1 minute' event to be manually reset (in fact, we'll never reset it). Thus, if the animal enters the zone during the first minute of the test the 'Animal enters the zone' event will become active; however, nothing will actually happen because the '1 minute' event will be inactive. Having determined that there are no actions to perform, ANY-maze will simply reset the 'Animal enters the zone' event back to an inactive state.

Now, after 1 minute the '1 minute' event will become active, but it won't trigger the action either, because the 'Animal enters the zone' event won't be active - and because it's a manual reset event, it'll be left in the active state. What will happen now if the animal enters the zone is that both events will become active, and this WILL trigger the action which will end the test.

In fact, this would be much easier to handle with procedures - so if you have a set-up like this, you're better off using procedures instead of events and actions. It's possible to change a protocol that uses events and actions to use procedures instead - for more details, see Understanding the changeover from Events and Actions to Procedures.

Setting up an event

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Introduction

For a general introduction to events, see An introduction to events and actions.

To add an event to a protocol, click the *Add item* button in the ribbon bar and select *New event* from the menu which appears.

The New event menu option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the event's name.
2. Define the trigger for the event
3. Specify whether the event is reset automatically
4. Specify which stages the event will actually be used in

What next?

After completing these steps, you should consider whether you want to include any actions in the protocol.

See also:

- Adding elements to a protocol
- Editing an event or action
- Deleting an event or action
- Event measures

Event name

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

In brief

Enter a name for the event in the *Event name* field on the event's settings page. You must make an entry, but it can be anything you like.

Details

Event names should identify the situation that triggers the event - for example, 'Island entry' or 'Lever press'.

If an event triggers an action that ends the test, then the *event*'s name will be reported as the test end reason - you should bear this in mind when naming such events.

Limits

You can enter anything you like up to a maximum of 32 characters.

Defining an event's trigger

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

In brief

To define an event's trigger you should click the link titled *Help me define the trigger for this event.* This will open the Event trigger wizard which will lead you a step at a time through the process of defining the trigger for the event.

Details

Full details of the steps necessary to define an event's trigger are given in the Event trigger wizard.

The Event trigger wizard

The Event trigger wizard helps you to define the situation which will trigger an *event* - in other words, you use it to specify what has to happen in a test for the event to occur.

For example, in a water-maze, you might want to end tests when the animal finds an island. In this case you will need to create an 'event' for when the animal enters the island zone. In fact, you'll also need to create an action that will end the test when this event occurs.

The Event Trigger Wizard takes you a step at a time through the stages necessary to define an event:

- To move through the wizard, simply click the *Next* > button.
- If you want to go back to a previous step, perhaps to alter an answer you gave, then you can click the < *Back* button.
- You can cancel the wizard at any time, without creating a rule, by clicking the *Cancel* button.

Event trigger wizard : Trigger type

An event's trigger is based on a particular situation that has to occur. (Here, I'm using the term *situation* rather loosely because a rule might be based around something which *doesn't* occur, for example the animal fails to enter a particular zone). Anyhow, the point is that you first need to choose what this situation is going to be.

As there are lots of possible situations, they're shown here in groups - for example, *Triggers which relate to activity in the island zone*.

- To see the detailed situations, inside a group simply click the group.
- To choose the individual situation you want to use, simply click it - it'll then be shown highlighted.
- You won't be able to click the *Next >* button until you've selected something here.

If you want to define more than one situation, then you should simply create another event and define its trigger to be based on the second situation - you can continue in this way, creating as many events as you like.

Event trigger wizard : Number of occurrences

In this step, you need to specify how many times the situation you chose in the previous step has to occur.

If you select the option for *More than once,* then a field will appear where you can enter the actual number, which can be any value from 1 to 999.

Event trigger wizard : Apply to trials

In this step, you need to specify which trials the event will be applied to. (Note that the stages it's applied to are defined directly in the protocol and not in this wizard.)

So, for example, you might only want the event to be applied to the first two trials in the first stage of your experiment. In this case, here, you would specify *Apply the event to the first 'n' trials in the stage*, and then enter 2 as the number of trials, and in the protocol itself, you would select just the first stage.

Event trigger wizard : Duration

In this step, you need to specify how long the situation you chose in the previous step has to last. For example, if you specified that the animal has to remain in a zone for certain amount of time, you should enter the amount of time here.

The value you enter can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 50 milliseconds to 24 hours.

Event trigger wizard : Time since test start

In this step, you need to specify how long after the start of the test the event should be triggered. For example, entering '60' would cause the event to trigger one minute after the start of the test.

The value you enter can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 50 milliseconds to 24 hours.

Event trigger wizard : Time until test end

In this step, you need to specify how long before the programmed end of the test the event should be triggered. For example, entering '120' would cause the event to trigger two minutes before the programmed end of the test.

The value you enter can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 50 milliseconds to 24 hours.

Event trigger wizard : Time interval

In this step, you need to specify the interval at which the event should trigger. For example, entering '60' would cause the event to trigger once every minute.

The value you enter can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 50 milliseconds to 24 hours.

Note that the event will always trigger once, but thereafter it will only actually trigger repeatedly (at the specified interval) if it is reset. Therefore it's normal to set this type of event to reset automatically, although you could choose to explicitly reset it as part of an action's effects.

Event trigger wizard : Random time interval

In this step, you need to specify a range of times at which the event should trigger. For example, entering a minimum of 60s and a maximum of 90s would cause the event to trigger a random time between 60 and 90 seconds - for example, 72s. As this event triggers repeatedly, it will occur again after another period of between 60 and 90 seconds - perhaps 61s this time.

The values you enter can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 50 milliseconds to 24 hours. The maximum time must be greater than the minimum.

Note that you can't enter a minimum of 0. The reason for this is that such a minimum would mean that the event could, in theory, trigger at the same time as the 'previous' event (i.e. 0 seconds after it). Clearly, this is impossible to actually achieve, so there must be at least a short delay between the events.

Note that the event will always trigger once, but thereafter it will only actually trigger repeatedly (at the specified interval) if it is reset. Therefore it's normal to set this type of event to reset automatically, although you could choose to explicitly reset it as part of an action's effects.

Event trigger wizard : Time interval

In this step, you need to specify how long after the event you specified in the previous step this event should trigger. For example, if you enter '60' then this event will be triggered 1 minute after the other event.

The value you enter can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 50 milliseconds to 24 hours.

Event trigger wizard : Time of day

In this step, you need to specify the time of day when the event should be triggered.

You should specify the time as HH:MM:SS using the 24 hour clock. For example, 22:30:00 will trigger the event at half past ten in the evening.

Event trigger wizard : RPM

In this step, you need to specify the how fast or slow the rotary encoder should be turning for the event to be triggered.

For an event where the speed needs to exceed a certain value, the event will trigger when the encoder changes from turning slower to turning faster than the value you enter here. To retrigger the event, the encoder will need to slow down to below this value and then speed up again and pass the value.

Likewise, for an event where the speed needs to fall below a certain value, the event will be triggered when the encoder changes from turning faster than the value you enter to turning slower.

You should specify the speed in revolutions per minute (RPM). The minimum value is 1 and the maximum value is 999, although be aware that some encoders won't be accurate at such high speeds.

Event trigger wizard : Number of occurrences in a specific time

In this step, you should specify the number of times that the situation you chose in the previous step has to occur, and within what time limit. For example, if the situation is that a key must be pressed, then you could specify that it must be pressed three times within 30 seconds.

You should specify the number of times the situation should occur by entering a number from 1 - 230; for the time limit, you can use any of the units d (days), h (hours), min (minutes), or s (seconds). You can mix units and also specify decimal values. So, for example, the period of eight and a half minutes could be entered as '8.5min' or as '8min 30s'. If you don't specify any units, then seconds will be assumed.

You can enter any value from 1 second to 99999 seconds (approx. 28 hours).

Event trigger wizard : Value of a signal or sensor

In this step, you should specify the value that the signal or sensor must have in order to trigger the event. The units will depend on the signal or sensor you have chosen, and are shown next to the field where you enter the value.

Event trigger wizard : Duration that a signal or sensor must be above or below a certain value

In this step, you should specify how long the signal or sensor's value must remain at, above or below a certain value in order to trigger the event.

You can use the following units of time: ms (milliseconds), s (seconds), min (minutes), h (hours). You can mix units, so entering 2min 30s is the same as 150s. If you don't enter any units. then seconds is assumed.

You can enter any duration from 50 milliseconds to 24 hours.

Event trigger wizard : Time limit for a signal or sensor to increase or decrease by a certain value

In this step. you need to specify the period within which the signal or sensor must change value in order to trigger the event. For example, you could specify that a weight sensor must change by 5g in 5 minutes.

You can use the following units of time: ms (milliseconds), s (seconds), min (minutes), h (hours). You can mix units, so entering 2min 30s is the same as 150s. If you don't enter any units, then seconds is assumed.

You can enter any duration from 50 milliseconds to 24 hours.

Event trigger wizard : Finished

In this final step of the Event trigger wizard, you should review the rule you've defined to check that it's correct.

If you want to edit the rule, then simply use the *< Back* button to work backwards through the steps of the wizard to the appropriate page.

Specifying whether an event is reset automatically

In brief

When an event is triggered, it becomes *active*, and it remains in this state until it is *reset*. You can choose whether an event is reset automatically (i.e. by the system) or manually (i.e. by an action). To do this, simply check or un-check the box on the event's settings page.

Details

When an event is reset automatically, ANY-maze changes it back to an inactive state after processing all the effects of its activation. For example, if you have an auto-reset event that is triggered by the animal entering a zone then, when the animal enters the zone, the following steps will occur:

1. The animal enters the zone.
2. The event is triggered - it becomes active.
3. ANY-maze checks all actions to see whether any of them depend on the event.
4. Any actions which depend on the event are triggered and their effects are processed (perhaps a switch output is activated).
5. The event is reset - it becomes inactive again.

In general, you will probably want to use auto-reset events because they are repeatedly triggered whenever their triggering situation occurs. In the above example, this means that *every* time the animal enters the zone, the switch output will be activated.

In contrast, a manually reset event will remain active after it's been triggered, and therefore even if the triggering situation occurs again, the event *won't* be triggered and any actions that depend on it *won't* occur. Thus a manual reset event can be used as a *one-shot* event, i.e. it'll trigger just once.

Of course, that's not to say that a manual reset event can never be reset; it can be - but this is done as a result of some other situation occurring.

For example, imagine you want to activate a switch if the animal enters zone B within 30 seconds of entering zone A. In this case, you could use the following events and actions:

- Event : Zone A entry. Manual reset.
- Event : 30 seconds after Zone A entry. Auto reset.
- Event : Zone B entry. Auto reset.
- Action : When '30 seconds after Zone A entry' event occurs, reset 'Zone A entry' event.
- Action : When 'Zone A entry' AND 'Zone B entry' events are both active, activate the switch.

Thus when the animal enters zone A, the 'Zone A entry' event becomes active; 30 seconds later, the event will be reset. If while this event is active the animal enters zone B, then the switch will be activated. Note that if the animal entered zone B and then zone A, nothing would happen - because the 'Zone B entry' event auto-resets, so the subsequent entry into zone A won't trigger the switch output because the zone B event won't be active.

As you can see, this ability to choose whether to automatically or manually reset events allows you to create complex rules for controlling your tests.

In fact, this would be much easier to handle with procedures - so if you have a set-up like this, you're better off using procedures instead of events and actions. It's possible to change a protocol that uses events and actions to use procedures instead - for more details, see Understanding the changeover from Events and Actions to Procedures.

Choosing the stages in which to process an event

In brief

You should use the list of stages on the event's settings page to select those stages in which you want the event to be processed. If you un-check a stage, then the event will be entirely ignored in all trials in that stage; it'll be like the event simply doesn't exist.

Details

In general, you will probably want ANY-maze to process an event in all the stages of an experiment - indeed, this is the default setting used by the system. However, there may be experiments in which the event is not required all the time.

For example, you might wish to shock the animal in some situation during a *Training* stage, but not during a *Retention* stage.

Setting up an action

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Introduction

For a general introduction to actions, see An introduction to events and actions.

To add an action to a protocol, click the *Add item* button in the ribbon bar and select *New action* from the menu which appears.

The New action *menu option will be disabled until the protocol includes at least one event - this is because actions are triggered by events, and with no events, an action would simply never be used.*

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the action's name.
2. Specify which event, or events, can trigger the action
3. Specify the actual effect of the action

What next?

After completing these steps, you should consider whether you want to include any calculations in the protocol.

See also:

- Adding elements to a protocol
- Editing an event or action
- Deleting an event or action

Action name

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

In brief

Enter a name for the action in the *Action name* field on the action's settings page. You must make an entry, but it can be anything you like.

Details

Generally, action names should identify the effect the action has; however, they're only actually used to label the action - the system doesn't use them itself.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying which event, or events, can trigger an action

In brief

Select the event, or events, that can trigger the action by selecting them in the list shown on the action's settings page. You can choose any number of events, whether or not they're used by other actions too.

Details

If you choose more than one event, you will also need to specify how the events should be combined in order for the action to be triggered. The two options are to AND the events or to OR them.

As you'd expect, ANDing multiple events means that the action will only occur if *all* the events are 'active' at the same time. To be precise, actions are triggered when an event becomes active - so, if the action is triggered by more than one event, all the events except one must already be active when the non-active event becomes active.

ORing events simply means that the action will be triggered when *any one* of the events becomes active - in this case, the state of any of the other events is ignored.

Specifying the effect of an action

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

In brief

You should select the effect, or effects, of an action using the list shown on the action's settings page. You can choose any number of different effects from the various groups available.

Details

An action's effect is performed when the action is triggered. The action can have a single effect (e.g. to end the test) or multiple effects (e.g. to play a sound and to activate a switch output).

The effects are shown in groups. Some groups, such as *Test control*, are always present while others, such as a group to control a switch output, depend on what other elements are defined in the protocol.

Specifically, the possible groups are:

Test control	The options in this group allow you to pause, unpause, or end the running test.
Video recorder control	The options in this group allow you to start, stop, pause, or unpause the video recorder. There's also an option to add a label to the video recording. Note that, with the exception of *Start,* these options will only actually have an effect if the test is being recorded - see What to record while testing. The start option will actually begin recording the test (if the test is already being recorded, it will have no effect). The name and location of the video file and the quality of the recording are set in the same way as they would be if the *Automatically record videos of all tests* option was being used - you'll find full details in the What to record while testing topic.
X switch output	The options in this group allow you to activate or deactivate the relevant switch.
X speaker	The options in this group allow you to switch a speaker on or off, or change what the speaker plays.
X temperature controller	The options in this group allow you to dynamically alter the temperature of the temperature controller.

X lighting controller	The options in this group allow you to dynamically alter the light level of the lighting controller.
X syringe pump	The options in this group allow you dynamically change the running state, direction, flow rate, or volume of the syringe pump.
X shocker	The options in this group allow you administer shocks and dynamically alter their duration and/or intensity.
Sounds	The options in this group allow you to turn the ANY-maze white noise generator on or off, or play a *Test action* sound. The specific sounds played are defined in the Sounds section of the ANY-maze Options page.
X event	The options in this group allow you to activate, deactivate, or reset the relevant event. Deactivating an event will prevent it from being used until it is activated again (i.e. it's as if the event doesn't exist); resetting an event will result in it not being 'active' when it is checked to see if the event has occurred.
Send an alert message	This option allows you to enter the text of a message and then specify how the message should be delivered when the action is triggered. The options include: Display on screen; Send as an E-mail; Send as an SMS message to a cell phone.
Run a program	This option allows you specify a program that should be run when the action is triggered. In fact, you can enter any Windows command line.

Changing what a speaker plays

The action described here is for use in protocols created in version 4 of ANY-maze. In version 5, Events and actions have been replaced by procedures, *for which there are similar, but more flexible, actions available.*

Introduction

During a test, you can use an Action to alter what a speaker plays. Specifically, you can change the sound the speaker plays, the volume it plays at, and the duration that the sound lasts. This is done using the *Changing what a speaker plays* window shown in figure 1.

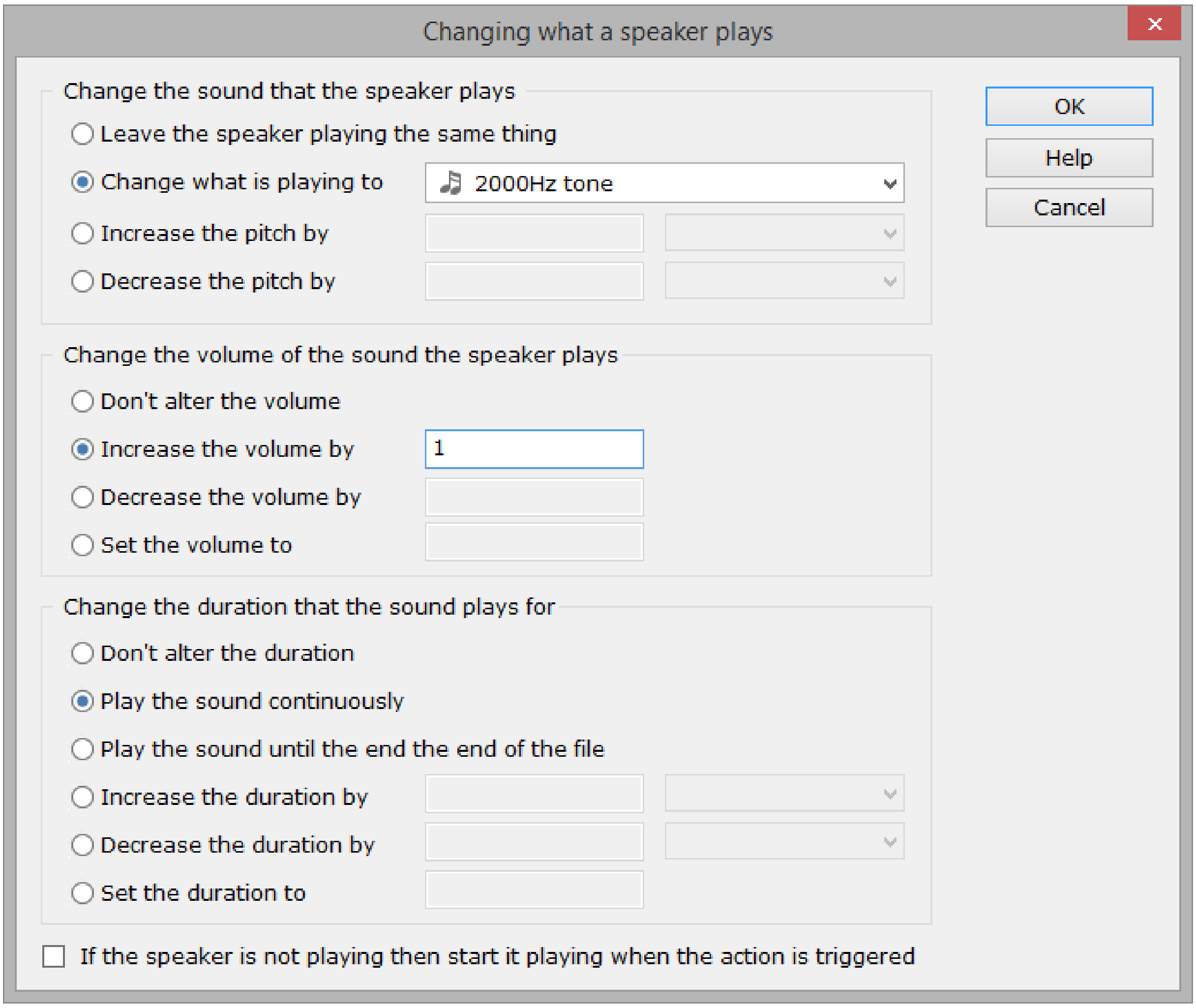


Figure 1. The Changing what the speaker plays *window, which is used to specify how an Action should alter the speaker.*

This window can be used to make the action do one or more of the following:

- Change the sound the speaker plays
- Change the volume
- Change the duration the speaker plays for
- Start the speaker playing

Changing the sound being played

There are two ways in which you can change the sound played - you can simply select a completely different sound, or you can alter the pitch of the sound. Note that altering the pitch will only actually make a difference if the speaker is set to play a 'Tone'.

If you change the sound being played to a sound file, and the file you select is the same as the file that is already being played, then the file will start playing again, from the beginning.

If you increase or decrease the pitch of a tone, and the new pitch would be outside ANY-maze's frequency range (65Hz - 32000Hz), then the new tone will be *clipped* to be within this range.

Changing the volume

You can alter the volume to be louder, quieter, or to be a specific value. In ANY-maze, the volume of a speaker is specified as a percentage, where 0% is silence and 100% is as loud as the speaker can play. So to increase or decrease the volume, you should specify the *amount* the volume should change by. For example, specifying an increase of '10' will *add* 10 to the current volume; thus a volume setting of 60% will become 70%. Note that the value you specify DOES NOT alter the volume by the specified percentage - if it did, then 10 would change 60% *by* 10% which would yield a new setting of 66%.

If the result of a change in volume would create a setting outside the range 0% - 100%, then the new setting will be *clipped* to be within this range.

Changing the duration the speaker plays for

Important: Changes to the duration will NOT cause the duration of a playing sound to be set to the new duration; rather, the sound will start playing AGAIN and will last for the new duration.

You alter the duration that a sound plays for either by a certain percentage of the current duration (for example, you could increase the duration by 50%) or by a certain number of milliseconds. Alternatively, you can specify a fixed duration.

Entering a fixed duration of 0 will cause a sound to play forever. Note that decreasing the duration will never cause the new duration to become 0; rather, the minimum duration will be 2ms.

Entering a fixed duration of 1, when the sound that the speaker plays is a file, will cause the file to play once and then stop. Entering 0 will cause the file to repeat forever. Any other value will cause the file to play for that duration, repeating if necessary until the duration has elapsed.

Other than with the special values of 0 and 1, you can use any of the units d (days), h (hours), min (minutes), s (seconds), or ms (milliseconds) with your entry. You can mix units and also specify decimal values. So, for example, the period of one and a half minutes could be entered as '1.5min' or as '1min 30s'. If you don't specify any units, then seconds will be assumed (unless you enter 0, or you enter 1 when the speaker is playing a file).

You can enter any value from 1 millisecond to 1 day.

Starting the speaker playing

If the speaker is already playing when the action is triggered, then it will continue playing but will use the new settings. In the case of a change to the sound itself or to the duration, this will mean the sound starts playing again from the beginning; in the case of a change to the volume, the change will occur without causing playback to restart.

If the speaker is not playing when the action is triggered, then the new settings will usually just be saved, so that next time the speaker does play it will use them. However, you can choose to have the new settings applied and then have the speaker start playing using them.

Changing the temperature of a temperature controller

The action described here is for use in protocols created in version 4 of ANY-maze. In version 5, Events and actions have been replaced by procedures, *for which there are similar, but more flexible, actions available.*

Introduction

During a test, you can use an Action to alter the temperature of a temperature controller. This is done using the *Alter the temperature of a temperature controller* window shown in figure 1.

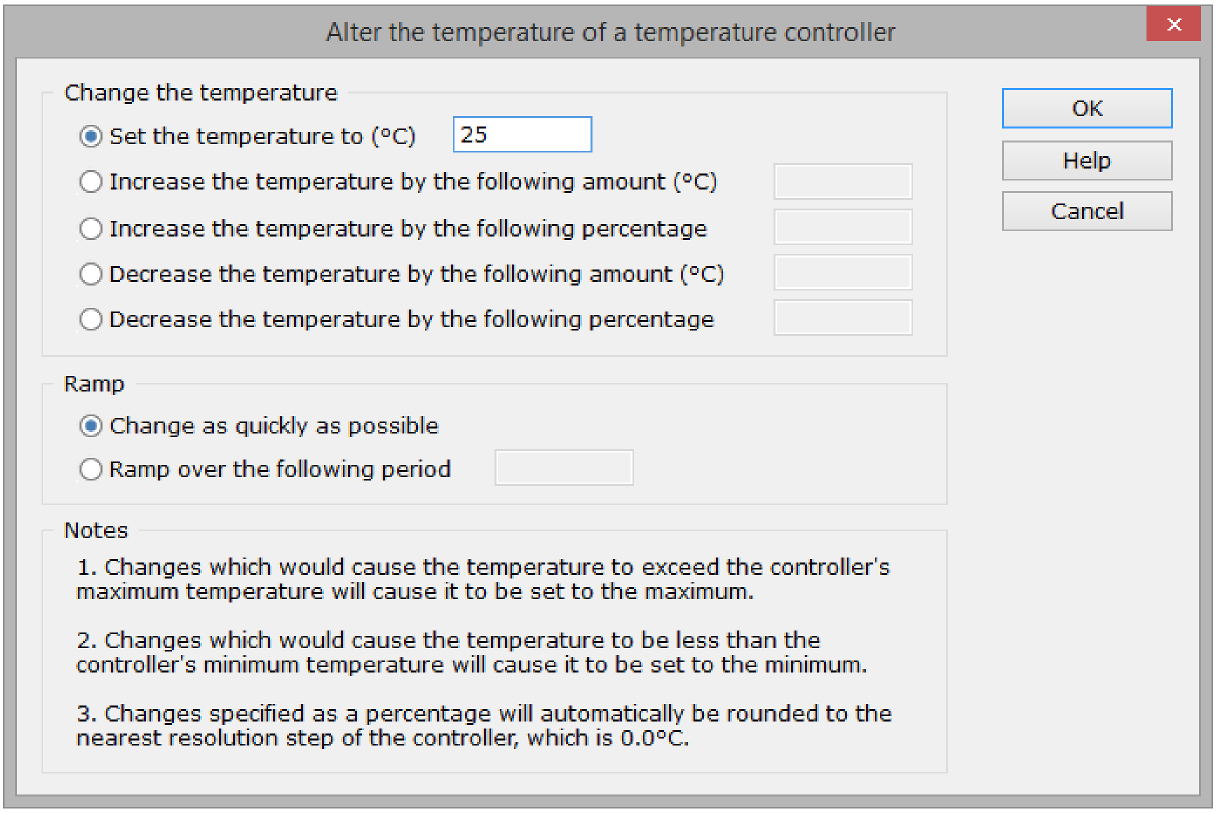


Figure 1. The Alter the temperature of a temperature controller *window, which is used to specify how an Action should alter the temperature controller.*

Specify the new temperature

There are three options for how the temperature should change:

- The temperature can be set to a specific value.
- The temperature can be set to increase or decrease from the current value by some specific number of degrees.
- The temperature can be set to increase or decrease from the current value by a certain percentage.

In all cases, the temperature will never be set to a value greater than the maximum temperature or less than the minimum temperature, as defined in the protocol.

When the temperature change is specified as a percentage, the value will be rounded to the nearest resolution step of the controller.

When the temperature is set to increase or decrease (rather than being set a specific value), the change will be applied to the currently programmed temperature and not the temperature that the device is actually at. For example, imagine that the controller has been programmed to be 20°C (and is at 20°C); when it is told to increase its temperature by 10°C, this will start to happen - now imagine that it has reached 25°C when it is told to decrease by 10°C. When this happens, the *programmed* temperature would be 30°C (20°C + 10°C) so the decrease will reduce it back to 20°C again - it *won't* change it to 15°C (which would be 10°C less than the temperature it is actually at).

Specifying a temperature ramp

Temperature controllers can't change their temperature instantly, so any change will always *ramp* up or down from the current value. You can choose whether you want this ramp to occur as quickly as possible, in which case the ramp speed will depend on the controller, or to ramp over some period that you specify. For example, you might specify that the temperature should change by 10°C over the course of 5 minutes, in which case the temperature would change by 1°C every 30s (which will almost certainly be much slower than the maximum speed the device could achieve).

As mentioned in the previous section, temperature increases or decreases are applied to the currently *programmed* temperature and not the temperature that the device is actually at, and this can have an effect on how ramp durations are interpreted: For example, imagine that a device is changing temperature from 20°C to 30°C over the course of 10 seconds; but after just 5 seconds, it is told to decrease temperature by 10°C over 10 seconds - what will happen?

Well, first of all, we know that the actual temperature will be 25°C (because it was changing from 20°C to 30°C over 10 seconds, and we're 5 seconds through this). We also know, from the section above, that the system will decrease the temperature from the *programmed* value (30°C), so the new programmed temperature will be 30-10 = 20°C. That leaves one question; how long should the ramp take?

The programmed ramp duration is 10s, so it would seem logical that the temperature will drop from 25°C to 20°C over 10s, i.e. a ramp of -0.5°/s. However, the action specified that the temperature should drop by 10°C over 10s, which implies a ramp of -1.0°C/s; so what will actually happen is that the temperature will drop from 25°C to 20°C at -1°C/s, i.e. over a period of 5 seconds. The key point to understand is that the *slope* of the programmed ramp will be used, and not the *duration*.

Changing the light level of a lighting controller

The action described here is for use in protocols created in version 4 of ANY-maze. In version 5, Events and actions have been replaced by procedures, *for which there are similar, but more flexible, actions available.*

Introduction

During a test, you can use an Action to alter the light level of a lighting controller. This is done using the *Alter light levels* window shown in figure 1.

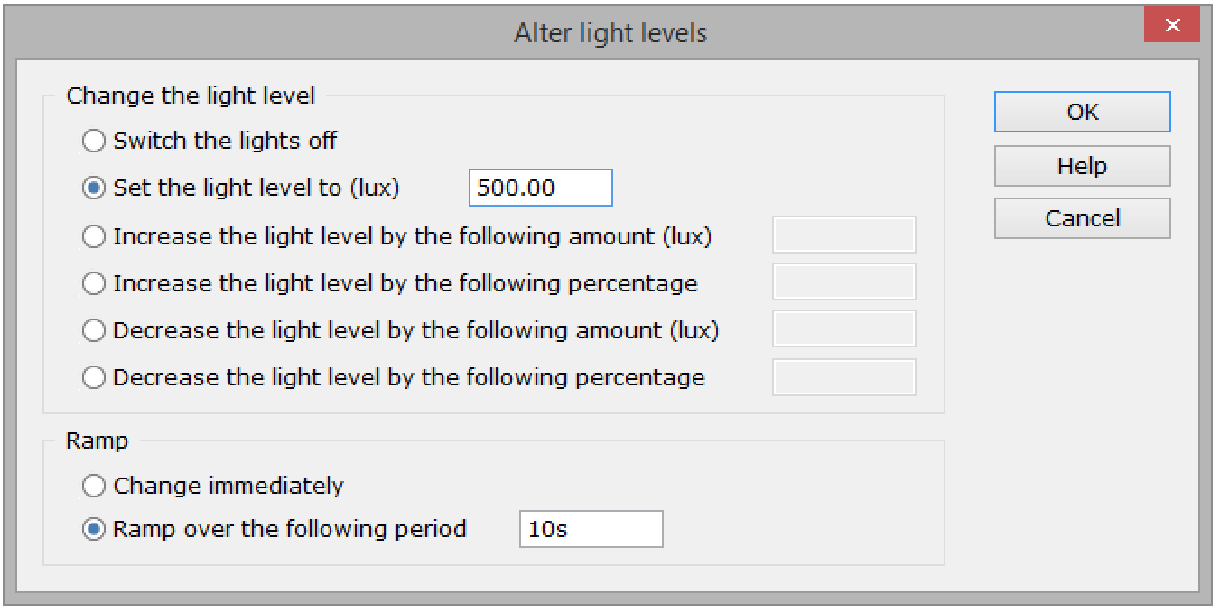


Figure 1. The Alter light levels *window, which is used to specify how an Action should alter the lighting controller.*

Specify the new light level

There are three options for how the light level should change:

- The level can be set to some specific value, in lux.

- The level can be set to increase or decrease from the current value by some specific number of lux.
- The level can be set to increase or decrease from the current value by a certain percentage.

When the level change is specified as a percentage, the value will be rounded to the nearest resolution step of the controller.

When the light level is set to increase or decrease (rather than being set a specific value), the change will be applied to the currently *programmed* light level and not the light level that the device is actually at. For example, the controller might be programmed to be at 200 lux when an action sets it to increase by 100 lux over 10 minutes. Two minutes later, another action might set it to decrease by 100 lux. In this case, it will still be ramping from 100 lux to 200 lux and will be currently be at 120 lux - however, the *programmed* level is 200 lux, so the decrease of 100 will be applied to this and the controller will be changed to 100 lux.

Ramping the change in light level

A lighting controller can alter the light level instantly, but there may be occasions when you want it to ramp from the current level to the new level over the course of time. In this case, you can specify the ramp duration and the controller will alter the level from its current value to the new value over this period. For example, specifying that a controller should change from 0 lux to 1000 lux over 5 seconds will bring the lighting up slowly, rather than switching it on instantly.

As mentioned in the previous section, light level increases or decreases are applied to the currently *programmed* level and not the level that the device is actually at, and this can have an effect on how ramp durations are interpreted: For example, imagine that a device is changing light level from 200 lux to 300 lux over the course of 10 seconds; but after just 5 seconds, it is told to decrease the level by 100 lux over 10 seconds - what will happen?

Well, first of all, we know that the actual level will be 250 lux (because it was changing from 200 lux to 300 lux over 10 seconds, and we're 5 seconds through this). We also know, from the section above, that the system will decrease the light level from the *programmed* value (300), so the new programmed level will be 300-100 = 200 lux. That leaves one question; how long should the ramp take?

The programmed ramp duration is 10s, so it would seem logical that the light level will drop from 250 lux to 200 lux over 10s, i.e. a ramp of -5 lux/s. However, the action specified that the level should drop by 100 lux over 10s, which implies a ramp of -10 lux/s; so what will actually happen is that the level will drop from 250 lux to 200 lux at -10 lux/s, i.e. over a period of 5 seconds. The key point to understand is that the *slope* of the programmed ramp will be used, and not the *duration*.

Changing the running state, direction, flow rate or volume of a syringe pump

The action described here is for use in protocols created in version 4 of ANY-maze. In version 5, Events and actions have been replaced by procedures, *for which there are similar, but more flexible, actions available.*

Introduction

During a test, you can use an action to alter the running state, direction, flow rate or volume of a syringe pump. This is done using the *Adjust syringe pump* window shown in figure 1.

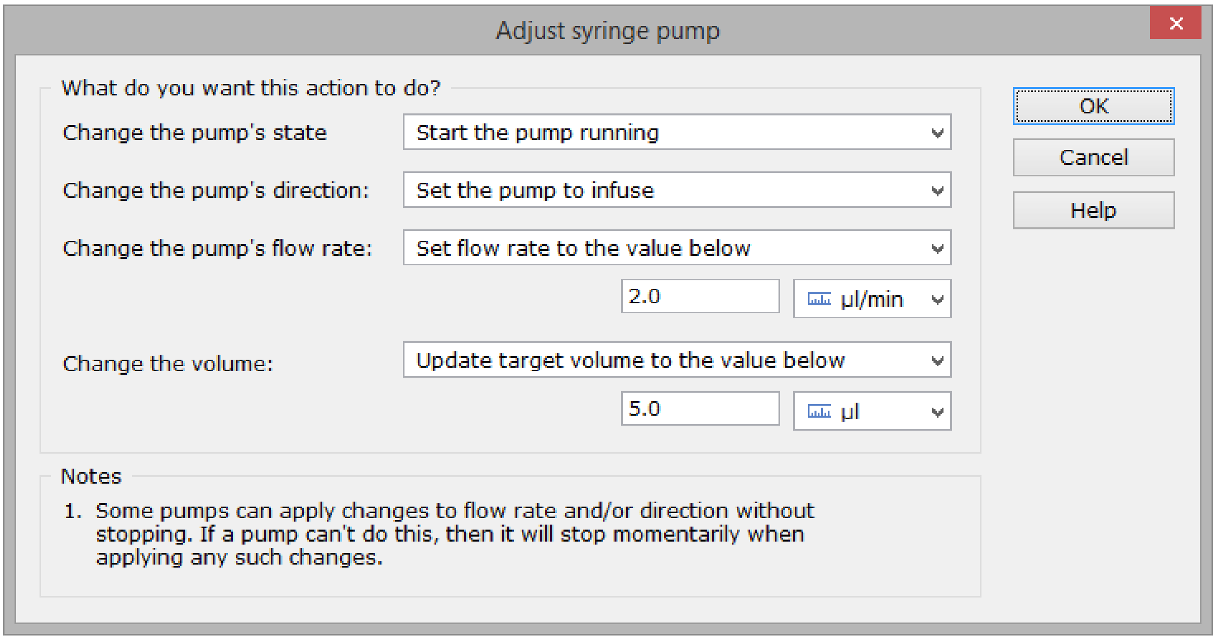


Figure 1. The Adjust syringe pump *window, which is used to specify how an Action should alter the syringe pump.*

Changing the pump's state

There are four options for changing a pump's state (i.e. whether it is running or not):

- *No change* - As you'd expect, this leaves the pump in the same state.
- *Start the pump running* - This will cause the pump to start running in whatever the current direction is, and at whatever the current rate is. If you also specify changes to the direction and/or rate (see below), then these will be applied before the pump starts. If the pump is already running, then this action will have no effect. Note that a pump may not actually start; for example, it might be stalled (at the end of its travel), in which case it will start and immediately stop again.
- *Stop the pump* - This causes the pump to stop. If the pump is already stopped, then it has not effect.
- *Toggle the pump's state* - If the pump is running, this will stop it. If it is stopped, this will start it. Note that the same considerations as are listed above for starting a pump apply when a pump is toggled from stopped to started. Note that the new state of the pump will be the opposite of its actual state, which may not be the last state you set it to. For example, imagine you set a pump to run, and then two minutes later you set it to toggle - you'd expect the pump to stop, right? This is indeed what would happen, unless the pump has stalled during the two minutes; in which case, it will be stopped and so toggling it will try to start it again.

Changing the pump's direction

There are four options for changing a pump's direction:

- *No change* - As you'd expect, this leaves the pump set to the same direction.
- *Set the pump to infuse* - This will set the pump's direction to infuse (liquid will come out of the syringe). If the direction is already set to infuse, then this will have no effect.
- *Set the pump to withdraw* - This will set the pump's direction to withdraw (liquid will flow into the syringe). If the direction is already set to withdraw, then this will have no effect.
- *Reverse the pump's direction* - If the pump is currently infusing, this will set it to withdraw; if it is currently withdrawing, this will set it to infuse.

You *can* change the direction of a pump when it is running. Some pumps can change direction without stopping, in which case this is what will happen; others require that the pump is stopped, the direction is changed and the pump is started again. If this is the case of the pump being used, then it will be stopped momentarily so the direction change can be applied (the stop is usually so short that you won't notice it).

If you change the direction of a pump that is stopped, then the new direction will be used the next time you start the pump.

Note that not all pumps support two directions (some only infuse and don't withdraw). If you try to alter the direction of a single-direction pump, then it will simply ignore you.

IMPORTANT: When you change the direction of the pump, the volume will always be reset. For example, imagine you specified that the pump should infuse 3ml at a rate of 1ml/min. Then, after 2 minutes, you change the direction to 'withdraw'. At this point, the pump will have infused 2ml, so will have 1ml remaining. But because you have changed the direction, the volume will be reset to 0 and the pump will begin to withdraw 3ml at a rate of 1ml/min. Of course, you can alter the volume at the same time as you change the direction (see below for options for altering the volume), so if you only want to withdraw 1ml, you could alter the volume accordingly.

Changing the pump's flow rate

There are only two options for changing the flow rate:

- *No change* - As you'd expect, this leaves the flow rate as it is.
- *Set flow rate to the value below* - If you select this option, then some fields will be enabled where you will enter the flow rate and units. For example, you might specify 3.5 with units of μl/min.

You should take care to ensure that you specify a flow rate that your pump supports; for example, setting the flow rate to 0.1 nl/hour probably won't work on many pumps (which simply don't support such low rates). If you do specify an unsupported rate, then the flow rate won't change.

You can always change the flow rate of a pump that is running. If the pump supports flow rate changes while running, then the rate will change immediately. If not, then the pump will be stopped, the rate change will be applied and the pump will start again. This usually occurs so quickly that you won't even notice that it happened.

If you change the flow rate of a pump that is stopped, then the new rate will be used when the pump is next started.

Changing the volume

There are four options for changing the volume:

- *No change* - As you'd expect, this makes no change to the volume that will be infused (or withdrawn).
- *Update target volume to the value below* - If you select this option, then some fields will be enabled where you will enter the new volume and its units. For example, you might specify 5.0 with units of ml. The important point about this option is that it takes into account the volume already infused (or withdrawn). For example, imagine you set up a syringe pump to infuse a volume of 4ml at a rate of 1ml/min. So after 1 minute, 1ml will have been infused and the remaining volume will be 3ml. But now you specify that you want to *Update the intended volume* to 6ml; so what will happen is that the intended volume, rather than being 4ml as you original specified, will be 6ml; but 1ml will still have been infused, so there will be a remaining volume of 5ml still to go. An important aspect of this option is that it can cause the pump to stop. For example, imagine if

instead of updating the intended volume to 6ml, we updated it to 0.5ml; but 1ml has already been infused - so the intended volume has already been delivered (indeed it has been exceeded, but there's not much we can do about that) so the pump will stop.

- *Reset the target volume to the value below* - If you select this option, then some fields will be enabled where you will enter the new volume and its units. For example, you might specify 5.0 with units of ml. The important point about this option is how it differs to the previous one. In this case, you are telling ANY-maze that you want to infuse (or withdraw) the entire specified volume starting now. So the volume that has already been infused (or withdrawn) is ignored (although it is of course recorded in the test's results) and the volume is 'reset'. For example, imagine you specified that the pump should infuse a volume of 4ml at a rate of 1ml/min. So after 1 minute, 1ml will have been infused and the remaining volume will be 3ml. At this point, you *Reset the intended volume* to 6ml - so 6ml MORE will now be infused (the 1ml already infused will just be ignored).
- *Clear the volume* - Clearing the volume means that the pump will continue to run without giving any consideration to the volume that is being infused (or withdrawn). The pump will only stop when it is either explicitly stopped (by an Action), the test ends, or the pump stalls (it reaches the end of its travel).

Note that changes to the pump's direction always reset the 'current' volume to zero and begin infusing (or withdrawing) the *intended* volume again.

See also:

- An introduction to syringe pumps
- Setting up a syringe pump

Changing the duration and/or intensity of shocks

The action described here is for use in protocols created in version 4 of ANY-maze. In version 5, Events and actions have been replaced by procedures, *for which there are similar, but more flexible, actions available.*

Introduction

During a test, you can use an action to alter the duration or intensity of the shocks that a shocker will administer. This is done using the *Adjust shocker* window shown in figure 1.

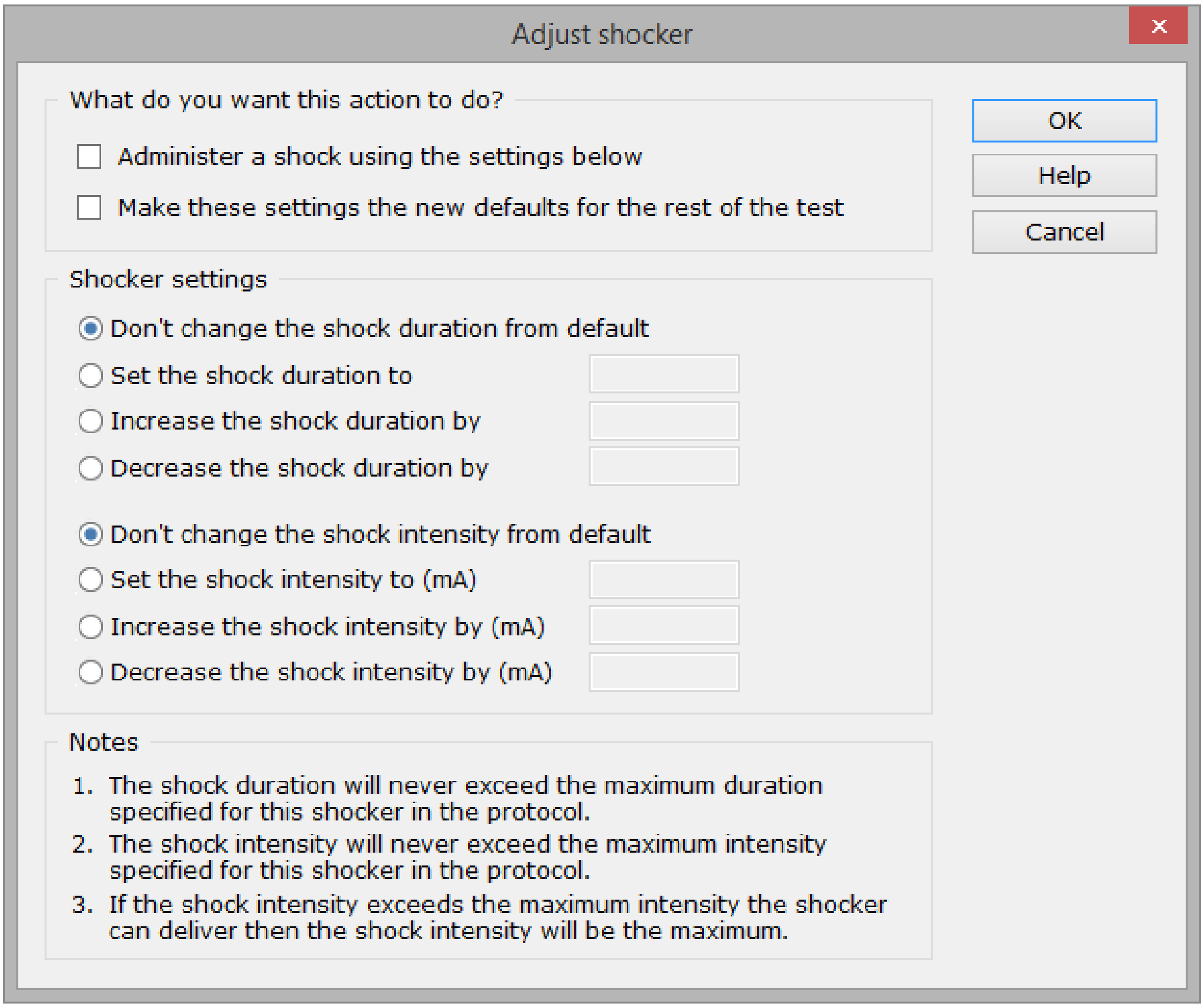


Figure 1. The Adjust shocker *window, which is used to alter the intensity and/or duration of the shocks administered by a shocker.*

Changing the shock duration

When you include a shocker in your protocol, one of the things you specify is the duration of the shocks that the shocker will apply. In many cases this will suffice, but imagine you wanted to create an experiment in which the duration of the first shock was 0.5s but the duration of the second shock was 1.0s; how would you do that?

The answer would be to define an action that would alter the 'default' shock duration (which you specified when you added the shocker to the protocol) to the new duration you want to use.

In fact, there are three options you can use:

- Set the duration to some new value, for example 1.0s
- Increase the duration by some value, for example 0.5s
- Decrease the duration by some value, for example 250ms

There is also a fourth option, which is to leave the duration unaltered - this is useful when all you want to do is change the shock intensity without changing the duration.

In all three cases, you can specify the duration in seconds or milliseconds; for example, 1.2s and 1200ms would both be valid (and identical) values.

It is important to remember that when you set up a shocker in the protocol, you also specify an absolute maximum duration that a shock can last - for example 5s. This maximum will always be respected, so if you set an action to increase the shock duration by 1s and the action occurred multiple times, then it might be that the duration would in theory be longer than 5s - however, the system would simply apply a 5s shock as that's the maximum you set.

Changing the shock intensity

In the same way that you can alter the duration of the shock applied, you can also change the intensity. Here you have the same types of options as for duration:

- Set the intensity to some new value, for example 2.0mA
- Increase the intensity by some value, for example 0.1mA
- Decrease the intensity by some value, for example 0.2mA
- Leave the intensity unaltered - useful when you just want to change the duration

In this case, the intensity values are always in milliamps, so you don't need to specify the units.

Like for the shock duration, when you set up the shocker you specify an absolute maximum shock intensity, and the system will always respect this no matter what level the shock would theoretically reach.

Choosing how the updated shock duration and/or intensity values are used

As mentioned at the start of this topic, when you add a shocker to the protocol, you specify the duration and intensity of the shocks that it should apply. As we have seen, you can use the *Adjust shocker* window to specify different values, but how should the values actually be used?

Well, there are two options:

- You can simply have the shocker administer a shock using the new values at the time that the action occurs. So, for example, let's imagine that the default shock duration and intensity (as specified in the *Shocker* item in the protocol) are 1s/2mA, and you have specified in this window that the duration should be set to 2s and the intensity to 3mA, then you can just have a 2s/3mA shock administered now. But the default duration and intensity will STILL be 1s/2mA, so if you use an action to simply *Apply a shock* (rather than *Adjust the shock*), then the shock applied will still be 1s/2mA.

- The second option is to use the new values to update the defaults. So, using the data from the previous example, this would ALTER the default values to 2s/3mA, thus any future shock applied would be at these levels. Note that choosing JUST this option will NOT administer a shock; it will only update the defaults.

You can also choose both options - so then the new values would update the defaults and then a shock would be applied using these values.

To understand this difference, imagine that you set the intensity to 'increase by 1mA' and that the duration is not to change. Then, you select the option to apply a shock. So when this action occurs, the current default intensity (let's say it's 2mA) will have 1mA added to it, and a shock of this intensity (3mA) will be applied. If the action occurs again, the same thing will happen - a 3mA shock will be applied. Now you also select the option to update the defaults. In this case, when the action occurs for the first time, the same thing will happen as before; but the new 3mA intensity will then be saved as the new default, so the second time the action occurs, the default intensity will be 3mA and we'll add 1mA to THAT, so a 4mA shock will be administered; and the third time the action occurs, a 5mA shock will be administered, and so on.

See also:

- An introduction to shockers
- Setting up a shocker

Sending an alert message

Introduction

The action effect *Send an alert message* can be used to notify you when an Event occurs. For example, you could create an event to detect when the animal starts moving and use an alert message to send your mobile phone an SMS message to tell you that the animal has 'woken up'. The options for this are set using the *Set up an alert* window shown in figure 1.

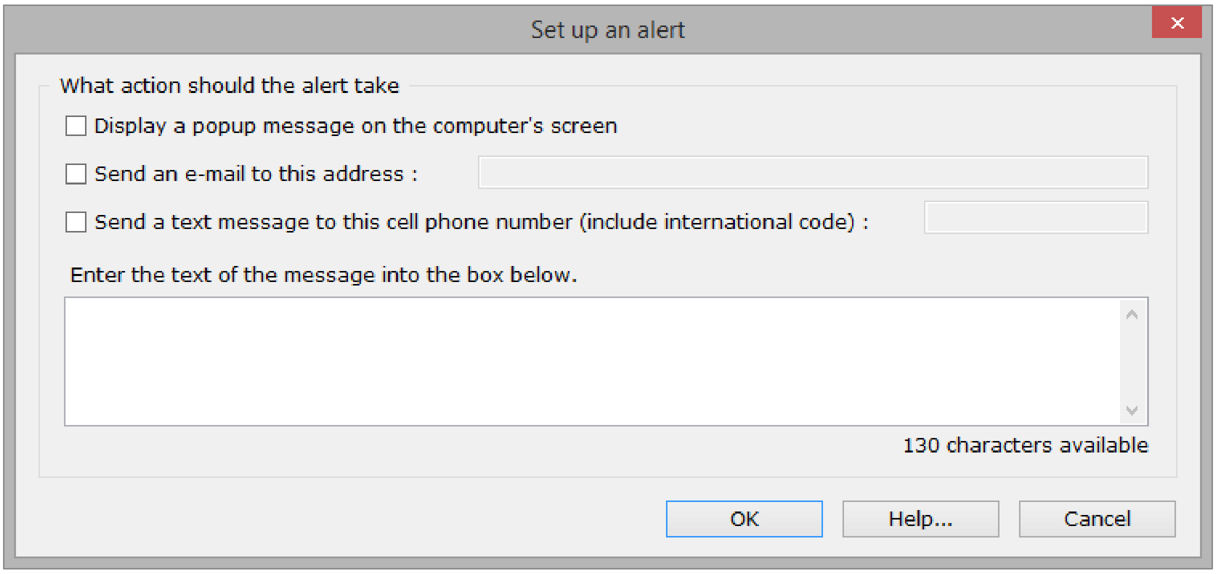


Figure 1. The Set up an alert *window, which is used to set the options for an action to generate an alert message.*

Details

There are three ways that an alert message can be 'sent' and you can choose any combination of them:

- A message on the computer screen

- E-mail
- SMS message

Message on screen

A message on the screen is the simplest alert, but of course it assumes you'll be near your computer. The message will pop up on top of all open windows and, if you've specified an Alert sound, a sound will play.

E-mail

To send an e-mail, you simply need to enter an e-mail address. Your computer doesn't need to have any e-mail software installed, although of course it does need to be connected to the internet. You can enter more than one e-mail address if you wish; just separate multiple addresses with commas.

SMS message

Perhaps the best way to receive an alert is as an SMS message to your mobile phone. In this case, you need to enter your phone number, including the international dialling code and excluding any leading zeros. For example, if you are in the UK (international code 44) and your number is 0777-1234-567, then you would enter 447771234567. In this case, you can't enter multiple numbers, although you can work round this by creating two identical actions with different phone numbers.

In order to send SMS messages, your computer must be connected to the internet.

Message text

The message text must be short (less than 130 characters) and if you plan to send it as an SMS message, then it should exclude accented characters and most symbols. If you do enter a character that can't be sent in an SMS message, it will simply be replaced with a dash.

Running a program

Introduction

When an action is triggered by a procedure, you can choose that one of the action's effects should be to *Run a program*. In this case, you will be asked to enter the program's *Command line, Parameters* and *Run options* using the *Run program* window shown in figure 1.

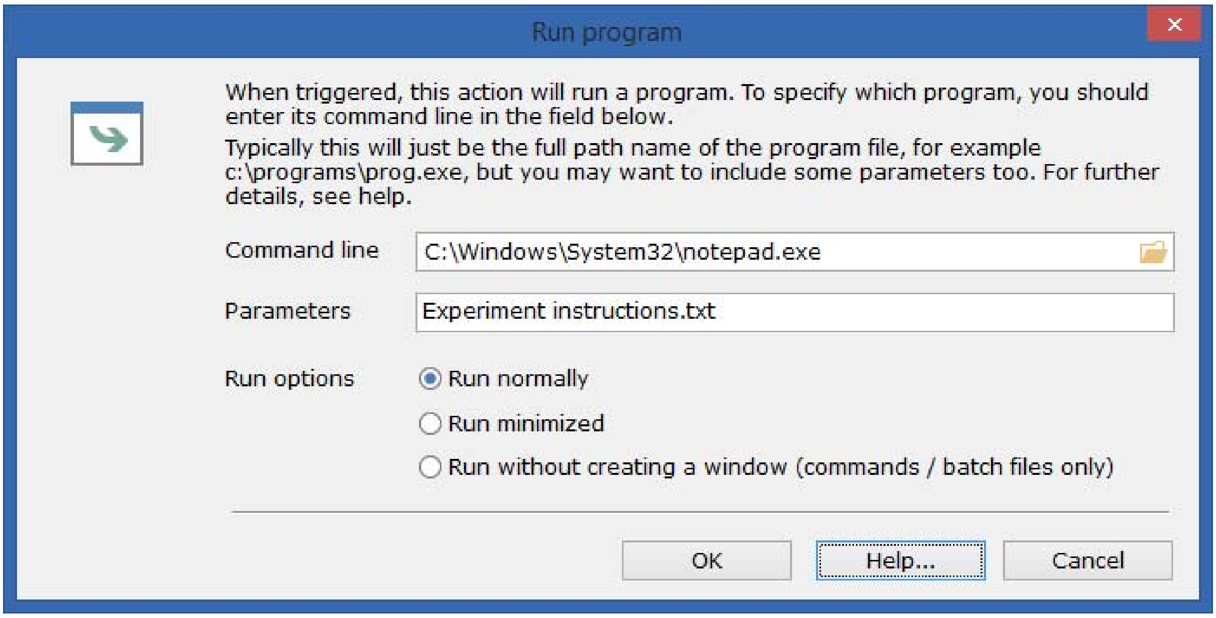


Figure 1. The Run program *window, which is used to run an external program or Windows command line.*

Details

The command line is usually just the *path name* of the program you want to run, for example:

```
c:\windows\notepad.exe
```

The easiest way to enter this is to use the (browse) button to select the relevant program file.

As well as specifying the program to run, you can also specify *parameters* such as:

```
mynotes.txt
```

In this case, the command will cause the notepad program to open the file mynotes.txt

If you want the action to perform multiple commands, then you can write a Windows batch file and then have the command line run that. For full details about Windows command lines, batch files, etc., refer to your Windows documentation.

The *Run options* allow you to specify how the program will be displayed:

Run normally	The program will be run in a normal window, in the same way that it would appear if you started it by double-clicking on its icon. This usually means that the program will open on top of ANY-maze. If the program is a command or batch file, a new console window will open in which the command/batch file will run. In this case, when the command or batch file ends, the console window will close - this usually means you briefly see a window appear and disappear, so in this case it is better to use one of the other options.
Run minimized	The program will be run in a minimized window - so it will typically just appear as a new running program in the Windows task bar. ANY-maze will continue to be the top-most, active window.
Run without a window	This option only works if the program to run is a command or batch file (if it is a normal Windows program, then this option is the same as *Run normally*). In this case, the command or batch file will run without opening a console window. This is useful when the command or batch file runs quickly, as it effectively just runs in the background.

Browsing for a program file to run

Introduction

When specifying the program that the *Run a program* Action should run, it's usually easiest to select the *Browse* button and then browse for the program file that you want.

Details

The *Browse* window is a standard 'File open' window - simply navigate to the file you want, select it, and then click the *OK* button - the command to run the program will automatically be written to the Run program window's *command line* field.

Editing an event or action

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Introduction

You can edit absolutely everything related to an event or an action at any time, whether before, during or after an experiment has been performed. However, as events and actions are only actually processed during a test, any changes you make will only affect the tests performed *after* you make the change.

See also:

- Editing the elements of a protocol
- Setting up an event
- Setting up an action
- Deleting an event or action

Deleting an event or action

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Introduction

Generally speaking, you can delete an event or an action at any time, whether before, during or after an experiment has been performed. However, as events and actions are only actually processed during a test, any changes you make will only affect the tests performed *after* the deletion.

Restrictions

There is one restriction on deleting events: If an event can trigger an action which ends a test, then ANY-maze will record the test end as being caused by the *event* (not the *action* as you might expect). As a result of this, ANY-maze will not allow you to delete an event which has caused at least one test to be ended - because if you did delete it, the system would have no way of knowing what caused the test to end.

This restriction is actually very easy to work round - simply edit the event so that it is no longer applied in any stages of the experiment. The effect of this will be that no new tests will use the event, but the event will continue to exist - thus allowing any test (or tests) which already ended because of it to continue to reference it.

See also:

- Editing the elements of a protocol
- Setting up an event
- Setting up an action
- Editing an event or action

Event measures

Events and actions have been replaced by procedures, *but they are documented here as they can still be used in legacy experiments.*

Unlike almost all other measures in ANY-maze, event measures do not update to reflect post-test changes in the event definition. Therefore, event measures are always based on the event definition that was in force when the test was performed.

ANY-maze will score the following measures for each event defined in the protocol:

- Number of events
- Latency to first event

Number of events

Description	Reports the number of times an event occurred.
Calculation method	Each time the event occurs, a counter is incremented.
Analysis across time	This measure can be analysed across time. The result for a period is the number of times the event occurred during the period.
Units	None
Notes	Event measures only report events which were detected while a test was being performed. Unlike almost all other measures, an event measure will not update to reflect post-test changes in the event definition.

Latency to first event

Description	Reports the latency to the first occurrence of an event.
Calculation method	When the event occurs, the test time is noted as the latency. If the event doesn't occur during the test, then the result is undefined.
Analysis across time	This measure can be analysed across time. The result for a period is the time within the period that the event first occurred. If the event doesn't occur during the time period, then the result is undefined.
Units	Seconds
Notes	Event measures only report events which were detected while a test was being performed. Unlike almost all other measures, an event measure will not update

to reflect post-test changes in the event definition.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Virtual switch measures

Treatment groups

Introduction

For a general introduction to protocols, see An introduction to Protocols.

Use the *Treatment groups* element of the protocol to specify whether you want to place your animals into groups, and if you do, how ANY-maze should assign the animals to the groups.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Choosing whether to use treatment groups
2. Choosing how ANY-maze should assign the animals to the groups
3. Specifying whether to run experiments blind

What next?

After completing these steps, you should consider whether you want to record your own Animal IDs within ANY-maze, or whether you'll just use ANY-maze's animal numbers.

See also:

- Editing elements of a Protocol

Choosing whether to use treatment groups

In brief

By default, ANY-maze will expect the animals in an experiment to be organised into Treatment groups. This is usually what you'll want, as in most experiments animals are indeed organised in this way. However, you may be setting up a protocol for experiments in which you won't be testing different groups of animals - in this case, you can simply switch off the use of treatment groups entirely.

You can switch treatment groups on and off using the option in the Treatment groups settings.

Details

Using treatment groups in ANY-maze provides the following benefits:

- ANY-maze will allow you to set up the groups and specify the number of animals in each one.
- The system can automatically assign animals to the groups, or you can do this manually.
- ANY-maze can run the experiment blind, which means it will code the groups and hide the meanings of the codes from the experimenter.
- When analysing results, ANY-maze will be able to organise the results by treatment group. Thus, reports can show the data for each group separately, graphs can show different series for each group, and 'Treatment group' can be used as an independent variable in statistical analysis.

Unless your animals simply won't be in divided into groups, then **we recommend that you use treatment groups.**

You can change between using and not using treatment groups at any time. If you switch treatment groups off, then ANY-maze won't remove any treatment groups you may have already entered; it will just ignore them. This means that if you change your mind and switch treatment groups back on again, all the treatment data will still be there.

This option may be selected and disabled in 'Ugo Basile Operon' protocol mode, if the protocol has been set to automatically randomize the allocation of animals between treatment groups. This choice will have been made when adding an Operon to the protocol.

See also:

- Choosing how to assign animals to treatment groups

- Specifying whether to run experiments blind

Choosing how ANY-maze should assign animals to treatment groups

In brief

ANY-maze generally expects animals to be divided into distinct groups which it calls *treatment groups,* because this is the most common situation. You can change this so that the system doesn't use treatment groups, but assuming you are using them, then there's a question of how the animals should be assigned to the groups.

For example, imagine you have 20 animals in 4 groups; the system could put the first 5 animals in the first group, the next 5 in the second, etc. or it could put the first animal in the first group, the second animal in the second group, etc. There are, of course, other possibilities too.

The options in the Treatment group settings allow you to choose exactly how this assignment is performed. It's important to understand that the animals will always* be tested in order, in other words animal 1 will be tested first, then animal 2, then animal 3 etc.

** The test order may differ if animals will be tested more than once, i.e. in multiple trials (see Specifying the order in which animals will be tested) and can also be changed at the time tests are performed.*

Details

There are four options for assigning animals to treatment groups. They are:

Manual assignment — This option will prevent ANY-maze from assigning the animals to any treatment groups (they'll all be recorded as *Unassigned*); instead, you will need to specify the group for each animal individually which you can do using the *View animals* option on the Experiment page. This option is useful when none of the others meet your needs.

This option may be disabled in 'Ugo Basile Operon' protocol mode, if the protocol has been set to automatically randomize the allocation of animals between treatment groups. This choice will have been made when adding an Operon to the protocol.

Block assignment — Assigning animals in blocks will assign the first 'n' animals to the first treatment group, the next 'n' animals to the second group etc., where 'n' is the size of the group. For example, if you have 4 groups (A, B, C and D) each consisting of 5 animals then *block assignment* would place the first 5 animals in group A, the next 5 in group B, etc., i.e. A A A A A B B B B B C C C C C D D D D D.

Cyclic assignment — Assigning animals in a cyclic fashion will assign the first animal to the first treatment group, the second animal to the second group etc. until each group includes one animal. It would then cycle back to the start again. For example, if you have 4 groups (A, B, C and D) each consisting of 5 animals then *cyclic assignment* would place the first animal in group A, the next in group B, the next in group C, etc., i.e. A B C D A B C D A B C D A B C D A B C D.

Random assignment Assigning animals in a random fashion will, not surprisingly, simply assign the animals to the treatment groups in a random order. For example, if you have 4 groups (A, B, C and D) each consisting of 5 animals then the system might place the first animal in group C, the second in group A etc., e.g. C A D C B D A A B D C B A D C B C D A B.

You can change the assignment system you wish to use at any time, although it will only affect new animals added to the experiment, it won't change the assignment of any existing animals. Nevertheless, you can always change the assignment of an animal using either the Animal details report or the *View animals* option on the Experiment page.

Specifying whether to run experiments blind

In brief

Selecting the *Run experiments blind* option in the Treatment groups settings causes ANY-maze to *hide* the actual treatments the animals receive from the experimenter. Although you might think that this is unnecessary in an automated system, you may still find it useful if you plan to use keys to record behaviours which the system can't score automatically.

Details

Choosing to run an experiment blind will have the following effects:

- The treatment groups you enter will be given random, single letter codes.
- Until all the tests in the experiment are complete, the system will only refer to the treatments using their codes - i.e. the actual names of the treatments will be hidden from the experimenter.

Because the treatments are coded, the experimenter won't know which is which - but of course *someone* will have to know this, in order to label the treatments with the appropriate codes.

The actual coding which has been applied can be seen by selecting the *Reveal treatment coding* button on the Experiment page. When running a blind experiment, you will normally want to ask a colleague to do this, so as to find out what the codes are, and to label your treatments appropriately.

You can change this option at any time; if you switch it off, ANY-maze will start to show you treatment names as well as codes (although it won't be able to 'un-randomise' the codes).

Animal ID

Introduction

Normally, ANY-maze will refer to animals by a simple sequential number - so Animal 1, Animal 2, Animal 3 and so on. However, it's likely that you already have a system for identifying animals, perhaps based on their cage numbers, or the experimental groups they're part of.

Clearly then, it might be desirable to use your own IDs in place of ANY-maze's animal numbers and the *Animal ID* element of the protocol allows you to do this.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Specify whether you'll be using your own IDs or not
2. Choose whether or not to verify IDs before each test
3. Specify the name to be used for the animal ID

What next?

After completing these steps, you should consider whether you want ANY-maze to record the weight of your animals before their tests.

See also:

- Experiment page

Specifying whether you'll be using your own Animal IDs

In brief

Use the button on the Animal ID settings page to specify whether you want to use your own animal IDs or not.

Details

Normally, ANY-maze will refer to each animal by a sequential number; i.e. Animal 1, Animal 2, etc. If you select this option, then you will be able to enter your own ID for each animal (for example, 'C1A1') and ANY-maze will then refer to the animal using your ID instead.

Entering Animal IDs

If you choose to use your own animal IDs, then you will of course need to actually tell ANY-maze what the ID of each animal is. The easiest place to do this is on the Experiment page, where you can use the *View animals* button to see a spreadsheet listing all the animals in the experiment. This spreadsheet will include a column for the animal ID, where you can simply enter whatever the IDs are.

How ANY-maze uses Animal IDs

If you choose to use your own animal ID, then ANY-maze will *always* use your ID to refer to the animals. So if, for example, you have specified that ANY-maze's animal 1 has an ID of 'C1A1' then ANY-maze will only refer to the animal by this ID and never as animal 1. For example, you will see messages such as 'Ready to test animal C1A1', or 'Results for animal C1A1' and so on.

Of course, ANY-maze will only actually know what the animal IDs are if you enter them (see previous section). So, until you specify that animal 1 has an ID of 'C1A1' (or whatever), the system won't know how to refer to it. To overcome this, it uses a # sign in front of its animal number.

For example, imagine you have three animals in your experiment; ANY-maze will simply number them animals 1, 2 and 3. You then enter an ID of 'C1A1' for animal 1 and 'C1A2' for animal 2, but you don't enter an ID for animal 3. ANY-maze will then refer to the animals as 'C1A1', 'C1A2' and '#3'.

A side effect of this is that it is not a good idea to use a # sign as the first character of your own numbering system.

Choosing whether or not to verify IDs before each test

In brief

If you choose to use your own animal IDs then you can also, optionally, specify that the ID must be verified before every test. What this means is that prior to each test, you will need to 'scan' a barcode or implantable chip that encodes the animal's ID.

This option can be switched on using the button on the Animal ID settings page

Details

For example, let's imagine that you have an animal with ID '1234-5678' and that this ID is encoded in a chip that is implanted under the animal's skin (like the chips used to 'tag' pets). If you choose the option to *Verify the animal's ID before each test,* then when ANY-maze is ready to test this animal, it will display a status message reading:

> Animal 1234-5678 - Animal ID not verified yet, scan the animal's barcode or chip

You would then use a scanner to read the chip; this would send the code '1234-5678' to ANY-maze which would confirm that this is indeed the ID of the animal to test and the test status would change to:

> Animal 1234-5678 - Ready...

On the other hand, if you scanned the wrong animal then ANY-maze would display a message to tell you that the animal ID is not verified and you would need to resolve the situation - presumably by finding the correct animal to test. By the way, the message that's displayed when you scan the wrong animal ID tells you the ID of the animal that you *did* scan, which can be very useful when trying to understand what's gone wrong.

Clearly to use this feature you need a barcode reader or chip scanner; details of compatible devices can be found here.

Specifying the name to be used for the animal ID

In brief

If you choose to use your own animal IDs, you can optionally specify what you want ANY-maze to call the IDs by entering it into the field on the Animal ID settings page

Details

By default, ANY-maze will call animal IDs 'Animal ID', but you might already have a name for them - for example 'Tail number' - and it would probably make things much clearer for you if the IDs were referred to using the terminology you're already familiar with.

To change the name, just delete the default entry of 'Animal ID' and enter your own name. You can enter anything you like up to 32 characters, although it's best to try to use a short name as it will be used as a column heading in a spreadsheet.

You must enter something.

You can edit the name at any time.

Barcode readers or chip scanners that can be used to verify animal IDs

Introduction

ANY-maze includes an option that forces you to verify the ID of an animal before every test (for details about this option see this topic). Verification is performed either by reading the animal's ID from a barcode or by scanning a chip implanted in the animal. This topic describes the barcode and 'chip' options that ANY-maze supports.

Using barcodes to verify animal IDs

If you use barcodes to record animal IDs, then the barcode will typically be printed on a label and the label will be fixed to the animal's home cage. Any barcode format can be used; the only requirement is that your barcode reader supports it.

To read barcodes, you will obviously need a barcode reader. You should use what's known as a 'keyboard-wedge' style reader (almost all readers are of this type). These readers typically connect to your computer via USB, and simply send scanned data to the computer as if it had been typed in at the keyboard.

To make sure that ANY-maze can differentiate data that comes from the reader from data which is actually typed in, you MUST program the reader so that it sends a PREFIX character before, and a SUFFIX character after, every scanned code (this is a common feature available in most readers). Specifically the PREFIX should be the character code 3 (ASCII ETX) and the SUFFIX should be the character code 4 (ASCII EOT).

That's all you have to do - provided the reader sends these prefix and suffix characters, you will be able to use barcodes to verify animal IDs.

Using implantable 'chips' to verify animal IDs

You can use any implantable 'chip' with ANY-maze, as the chip simply encodes some value which you will then use as the ID for your animal. For example, you might buy (or program) a chip which encodes the value 1234-5678, and if you implanted this chip in an animal then you'd simply tell ANY-maze that this animal's ID is '1234-5678'. The important part isn't so much the chip, it's the chip 'scanner' (reader).

Some scanners operate very much like a barcode reader, that's to say they simply send data to the computer (usually through a USB port) as if the data came from the keyboard. This type of scanner will work with ANY-maze, provided that the scanner can be programmed to send a PREFIX character before the scanned data and a SUFFIX character after it (just like the barcode option described above). Again, the PREFIX character MUST be set to be the character code 3 (ASCII ETX) and the SUFFIX must be set to be the character code 4 (ASCII EOT).

If your scanner does not offer these features, then you may still be able to use it, provided that it can output the scanned data as 'serial data'. In this case, the scanner will either connect to the PC via an old-fashioned COM port, or more usually, via a USB port which has been configured as a 'virtual COM port' (the scanner documentation should describe this in detail). If you have a scanner that works in this way, then you need to program it so that it simply sends each scanned code followed by a carriage return character (ASCII CR), through the serial port. The data sent should not include any additional prefix or suffix characters.

In this case, you will also need to set the scanner's serial communication parameters to be:

- 9600 baud
- 8-bit
- No parity
- 1 stop bit
- No flow control

Finally, within ANY-maze, you should select Advanced support on the Support page and enter the number of the COM port to which the scanner is attached in the field labelled *Chip scanner COM port*.

Recommended devices

For barcode readers we have no specific recommendations. Most barcode readers should work, but do be sure to check that the reader can send ASCII code ETX as a prefix character and ASCII code EOT as a suffix.

For 'chips', we recommend the transponders (chips) and scanners from BMDS. We have tested ANY-maze with the BMDS DAS-6008 interface, but other interfaces should also work. More information can be obtained direct from BMDS - inquiry@bmds.com

Animal weights

Introduction

It is common practice to weigh the animals in an experiment before every test. ANY-maze allows you to record these weights within the system, which has the benefit that you can then analyse them in the same way as any other measure - for example, to check whether there's a significant difference in weight between your treatment groups.

In fact, if you use the ANY-maze Animal scales to weigh the animals, ANY-maze will prompt you to weigh them before you test them and then automatically record the weights, making animal weighing a simple and streamlined process.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Specify whether you'll be recording animal weights and if so when you'll be weighing the animals
2. Choose whether ANY-maze will prevent tests from starting until an animal has been weighed
3. Specify how you'll be weighing the animals
4. Specify the name to be used for the animal weight field

What next?

After completing these steps, you should consider whether you want ANY-maze to calculate treatment doses based on the animals' weights.

See also:

- Weighing the animal before a test
- The ANY-maze Animal scales

Specifying whether you'll be recording animal weights and if so when you'll be weighing the animals

In brief

Use the options on the Animal weights settings page to specify whether you want to weigh your animals, and if so, *when* you want to weigh them.

Choosing whether to weigh your animals

You should use the buttons to specify whether you want to weigh your animals. The options are:

- *Don't record animal weights*
- *Record a single weight for each animal for the entire experiment*
- *Record multiple weights for each animal*

The first of these is the default, and means that ANY-maze won't record your animals' weights.

The second option means you will record just a single weight for each animal. When you'll weigh the animal is something you'll specify using the next set of options on the page (see below), so you don't need to worry about that when answering this question, the key point is that you'll only have a single weight for the animal for the entire experiment. This is typically what you'd choose when your experiment will be performed over a short period of time.

The third option means you'll be able to record multiple weights for each animal. This is typically what you'll choose when you experiment lasts for a long enough period of time that the animals' weights may change.

Choosing when to weigh the animals

You should use the buttons to specify when in the experiment you want to weigh you animals. The options are:

- *Weigh all the animals before starting the stage (or stages) selected below*
- *Weigh each animal before its first trial in the stage (or stages) selected below*
- *Weigh each animal before every trial in the stage (or stages) selected below*

Below these options is a list of all the stages in the experiment.

If you choose the first option then you will need to weigh all the animals in the experiment before you perform *any* of the trials in the specified stage (or stages). This is useful when you need to weigh the animals some time before testing them, for example, in order to calculate their treatment doses (because the treatment needs to be administered some time before testing begins).

The second option means you'll weigh each animal immediately before its first trial in the selected stage or stages. If you specified that you'll only record a single weight for each animal then you will only be able to choose a single stage, otherwise you'll be able to choose multiple stages. You can select any stages, so for example, in a water-maze experiment you might want to weigh the animals before their first probe trial, but not before their first training trial. Weighing animals immediately before their tests has the benefit that you are already handling the animal, so it's quick and easy to pop the animal onto the scales before placing it in the apparatus.

The third option is much like the second, except that in this case you'll weigh the animal before *every* trial in the stages, or stages, that you select. This is typically only required when there is a significant period between the trials of the stage.

Converting a protocol in which you used to record animal weights in a field

In previous versions of ANY-maze, you could record animal weights by setting up an appropriate field. You may therefore already have an experiment or protocol in which you've set up a field to record the animal's weight. In this case, if you select one of the above options to record animal weights, ANY-maze will detect that you may already be recording the weight in a field and prompt you to select any existing 'weight' field, using the window shown below. It will then automatically convert the field (and any data you have already entered) to use the new Animal weights setting you have chosen.

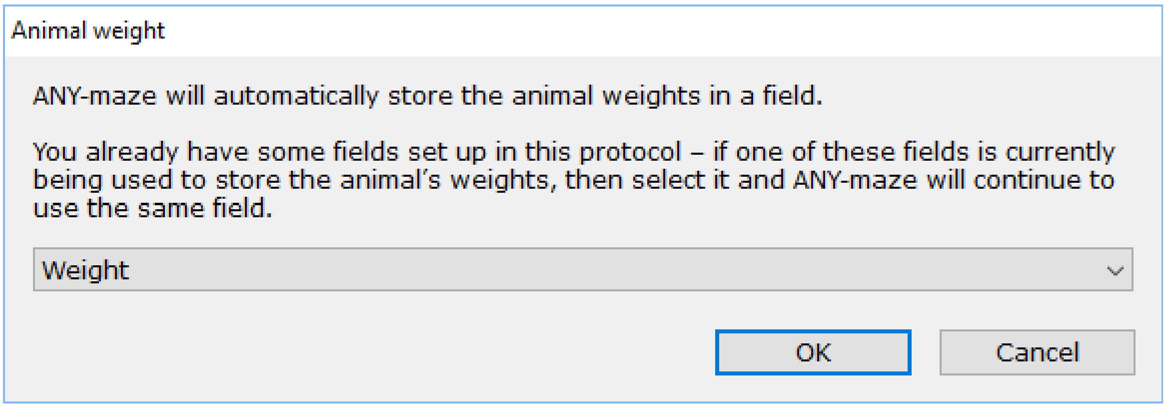


Figure 1. This window is displayed if there are already any numeric fields defined in your protocol - you can specify which field to use (or tell ANY-maze that none of the existing fields are used to measure animal weights).

Editing your choice

You can change these options at any time, even during an experiment, however there can be some repercussions:

- If your change means that some weights that are currently recorded are no longer required, then these weights will be permanently discarded. However, a warning will be displayed first to confirm this.
- If your change means that some new weights are required, then you will need to enter these weights. However, this can be problematic, as, if you didn't weigh the animals at the correct times, you won't know the weights to enter.

See also:

- Weighing the animal before a test
- The ANY-maze Animal scales

Choosing whether to prevent testing if the animal has not been weighed

In brief

If you choose to record your animals' weights then you can use the *Prevent tests from starting until the animal has been weighed* options to ensure that a weight is always recorded before the test is begun.

Details

If you select the option to *Prevent tests from starting until the animal has been weighed* then, before a test can be performed, ANY-maze will require that you either manually enter the animal's weight, or that you weigh the animal using the ANY-maze Animal scales, as in figure 1, below.

Test	Animal	Stage	Trial	Apparatus	Weighing status	Testing status
1	1	First stage	1	Plusmaze	Weighed: 228.0g	Completed
•	2	First stage	1	Plusmaze	Weigh the animal	Waiting for the animal to be weighed
•	3	First stage	1	Plusmaze		
•	4	First stage	1	Plusmaze		
•	5	First stage	1	Plusmaze		
•	6	First stage	1	Plusmaze		

Figure 1. The test on animal 2 can't be started until the animal has been weighed.

Editing your choice

You can change your choice at any time, even during an experiment. There are no restrictions or repercussions.

See also:

- Weighing the animal before a test
- The ANY-maze Animal scales

Specifying how you'll be weighing the animals

In brief

If you choose to record your animals' weights then you will need to specify how you'll actually weigh the animals, either manually, or using the ANY-maze Animal scales.

Details

There are two options for how you will weigh the animals:

- *Using the ANY-maze Animal scales*
- *By entering the weights into ANY-maze manually*

The simplest method is to use the ANY-maze Animal scales, as the scales integrate into ANY-maze. At the appropriate moment in the experiment you will be prompted to place the animal on the scales, the animal will be weighed and you'll then be prompted to remove the animal from the scales - the animal's weight will be recorded automatically.

Alternatively you can weigh the animals manually. Again, ANY-maze will prompt you at the appropriate moment in the experiment, to enter the animal's weight. You will need to weigh the animal and then type in the weight.

Full details about weighing the animals during an experiment can be found here.

Editing your choice

You can change the way in which you weigh animals at any time, even during an experiment. There are no restrictions or repercussions.

See also:

- Weighing the animal before a test
- The ANY-maze Animal scales

Specifying the name to be used for the animal weight

In brief

If you choose to record your animals' weights then ANY-maze will automatically create a field called 'Animal weight' in which to store the data. You can change the name of this field, perhaps to match the terminology you use in your lab, using the option on the Animal weights settings page of the protocol.

Details

By default, ANY-maze will call the field in which it stores the weight of the animal 'Animal weight', but you might want to use a different name - for example, just 'Weight'. To change the name, simply delete the default entry of 'Animal weight' and enter your own name. You can enter anything you like up to 32 characters, although it's best to try to use a short name as it may be used as a column heading in a spreadsheet.

You must enter something. You can edit the name at any time.

ANY-maze requires that animal weights are always entered in grams.

Treatment doses

Introduction

In many experiments, animals are administered a treatment at a dosage that depends on their body weight. For example, you might give an animal 10mg of the treatment per kilogram of body weight.

As you may know, you can record your animals' weight in ANY-maze, so if ANY-maze knows the animals' weight, it means it can easily calculate its treatment dose for you - in my example where the dose is 10mg/Kg, a 200g animal would receive 2mg.

However, if you plan to inject your animal, then knowing that you want to administer 2mg of drug, while helpful, isn't the complete answer to how much the animal's dose should be, as that will depend on the concentration of the solution you'll be injecting. In my earlier example, if the solution is 500mg/litre then you'd need to inject 4ml. ANY-maze can perform this calculation too.

The Treatment doses page of the protocol is where you can specify whether you want ANY-maze to calculate dose information and if so, how you want it to do it.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Specify whether you want ANY-maze to calculate treatment doses, and if so when you'd like it to do so
2. Choose whether ANY-maze will report the dose in milligrams or millilitres
3. If the dose should be reported in millilitres, specify the concentration of the treatment

By the way, you'll not be entering the actual treatment dose here (such as 10 mg/kg), rather what you're doing at this point is telling ANY-maze when, and how, you want the specific doses for the animals to be calculated. The dose for each treatment group is something you will specify when you enter the treatments on the Experiment page.

What next?

After completing these steps, you should consider whether you want to include any fields to record additional information about your animals or tests.

See also:

- Weighing the animal before a test

Specifying whether you want ANY-maze to calculate treatment doses, and if so when you'd like it to do so

In brief

Use the options on the Treatment doses page to specify whether you want ANY-maze to calculate treatment doses and if so, before which tests in which stages.

Details

You should use the buttons to specify whether you want ANY-maze to calculate treatment doses and if so when in the experiment it should do this.

- *Don't calculate dose information*
- *Calculate dose information before starting the stages selected below*
- *Calculate dose information for the first trial of the stages selected below*
- *Calculate dose information for every trial in the stages selected below*

Below these options is a list of all the stages in the experiment.

The first option is the default, and simply means that ANY-maze won't try to calculate treatment doses at all.

As you may notice, the other options are very similar to the options for when you want to weigh the animals. This is not a coincidence as, you need to weigh an animal in order for its dose to be calculated. In fact, you must select an option here which matches with the times in the experiment when you'll weigh the animals. For example, if you will weigh the animals before every test in the first stage, but not in any other stages, then you could choose to calculate their dose before the first trial of the first stage, or before every trial of the first stage, but you couldn't choose to calculate their dose in any other stages (as their weight will not be known). You don't need to worry too much about this, as ANY-maze will warn you if you choose to calculate the animals' doses at a time when ANY-maze will not know their weights.

Calculating dose information before starting a stage or stages

This option is useful when you want to dose the animals some time before starting *any* of the tests in the stage, for example, you might need to administer the treatment 24 hours before the test. In this case the Test schedule report will include entries for you to weigh all the animals before starting the tests in the stage, as in figure 1.

Animal	Stage	Trial	Weighing status	Dose	Treatment	Testing status
1	First stage	Weigh	Weigh the animal		B	
2	First stage	Weigh			A	
3	First stage	Weigh			A	
4	First stage	Weigh			B	
1	First stage	1	-			Waiting for the animal to be weighed
2	First stage	1	-			
3	First stage	1	-			
4	First stage	1	-			

Figure 1. An example of the Test schedule report for an experiment in which the animals are being dosed before they start the First stage. (The experiment has been limited to just 4 animals to aid clarity.)

Calculating dose information for the first trial of a stage or stages

This option is useful when want to dose the animals just a short time before you start the test. It's quite common to run experiments in which you dose the animals at, say 10 minute intervals, then wait perhaps 30 minutes and start testing the first animal. So, at time zero you'd dose animal 1, at time 10 minutes you'd dose animal 2, at time 20 minutes you'd dose animal 3, and at time 30 minutes you'd start testing animal 1 and dose animal 4. Then you'd progress through the experiment testing and dosing animals every 10 minutes. Figure 2 shows an example of a Test schedule report for this type of scenario.

Animal	Stage	Trial	Weighing status	Dose	Treatment	Testing status
1	First stage	1	Weighed: 193.6g	0.194mg	B	Ready
2	First stage	1	Weigh the animal		A	
3	First stage	1			A	
4	First stage	1			B	
1	First stage	2	Not applicable			
2	First stage	2	Not applicable			
3	First stage	2	Not applicable			
4	First stage	2	Not applicable			

Figure 2. An example of the Test schedule report for an experiment in which the

animals are being dosed before the first trial in the First stage. (The experiment has been limited to just 4 animals to aid clarity.)

Calculating dose information for every trial in a stage or stages

This option is much like the previous one, except that a dose will be calculated before every trial in the specified stage (or stages). This would most usually be used if the trials are separated by a long period of time, for example you test the animals once a day over 6 days, and before each test you administer the treatment.

Editing your choice

You can change these options at any time, even during an experiment.

Choosing whether ANY-maze will report treatment doses in milligrams or millilitres

In brief

ANY-maze will show the dose to be administered to each animal as part of the Test schedule report. It can show this either as a mass to administer in milligrams, or as a volume to administer in millilitres; this is where you can choose how you'd like the dose shown.

Details

If you choose to have ANY-maze show the mass to administer, in milligrams, then there's nothing more for you to do. However, if you want ANY-maze to calculate the volume to administer (in millilitres) then you'll also need to specify how ANY-maze will know what the concentration of the treatment is.

Editing your choice

You can change these options at any time, even during an experiment. Note, however, that if you have administered a specific mass to some animals and half way through the experiment you switch over to administering a volume, then ANY-maze will show ALL the animals as receiving a certain volume. For this reason it's generally not a good idea to change this option once an experiment has started.

Specifying the concentration of the treatments that you will use

In brief

If you specify that ANY-maze should calculate the volume of the treatment that you should administer to the animals, then you will be asked how ANY-maze should determine the concentration of the treatments.

Details

There are three options for how ANY-maze will determine the concentration of the treatments:

- *The volume administered should be approximately the same for all treatment groups*
- *The concentration will be different for each treatment group*
- *The concentration will be the same for all treatment groups*

The volume administered should be approximately the same for all treatment groups

This option is the one you're most likely to want to use, as it ensures that the volume administered (usually injected) will be roughly the same for all the animals, irrespective of the treatment they receive.

The easiest way to understand this option is through an example. Let's imagine you have two treatment groups, one will receive 1mg/kg of your drug and the other will receive 5mg/kg. If you made up a solution in which you had dissolved 100mg of your drug in 1 litre of water, then a 100g animal in the first group would need to be injected with 1ml of the solution and a 100g animal in the second group would need to injected with 5ml. Clearly these are not similar volumes and the fact that one group is being injected with 5x the volume of the other group, may confound your results.

So, rather than making up a single solution of your drug, you might choose to make two, one containing 50mg/litre and the other containing 250mg/litre. Now if you injected a 100g animal from the first group with 2ml of the first solution, the animal would receive 0.1g of the drug (i.e. 1mg/kg of its body weight), and if you injected a 100g animal from the second group with 2ml of the second solution, the animal would receive 0.5mg of the drug (i.e. 5mg/kg of its body weight), exactly as we wanted - and both animals would have received the same volume.

If you want to work in this way, then you can simply tell ANY-maze what the target volume you want to inject is (2ml in the above example), and what the approximate weight of the animals is (100g in the example). It will then calculate the concentrations for the drug solutions and show these to you on the Experiment page - see figure 1.

Treatment	Name	Dose mg/kg	Concentration mg/l	No. animals	Notes
Treatment 1	Saline			8	
Treatment 2	Drug X 1 mg/kg	1.000	50.000	8	
Treatment 3	Drug X 5 mg/kg	5.000	250.000	8	
Treatment 4					

Figure 1. When you want to administer approximately the same volume to all the treatment groups, ANY-maze will calculate the appropriate concentration of each treatment and show it to you in the Treatments spreadsheet on the Experiment page.

Note that this does require that all the animals are roughly the same weight. For example, if in the example above, the animal from the first group actually weighed 150g (not 100g) then we'd have needed to inject it with 3ml of our drug solution. Clearly we could change the solution to be 75mg/litre and then the animal would receive 2ml, but if we had another animal in the group that weighs 100g, it would then need to be injected with 1.33ml!

The point is that all the animals need to be roughly the same weight and then they'll all receive roughly the same volume. For example, if the animals in your experiment actually vary in weight from 90g to 120g and you told ANY-maze that you want to inject them with 5ml and that their average weight is 105g, then they would actually receive something between 4.3ml and 5.7ml, but that's probably close enough to 5ml in all cases that the volume injected would not be sufficiently different between the animals to confound your results.

The concentration will be different for each treatment group

This option is much simpler, as you just *enter* the concentration of the treatment for each treatment group. You do this on the Experiment page - see figure 2.

Treatment	Name	Dose mg/kg	Concentration mg/l	No. animals	Notes
Treatment 1	Saline			8	
Treatment 2	Drug X 1 mg/kg	1.000		8	
Treatment 3	Drug X 5 mg/kg	5.000		8	
Treatment 4					

Figure 1. When you want to use a different concentration of your treatment for each treatment group, then ANY-maze will include a 'Concentration' column in the Treatments spreadsheet on the Experiment page, where you can enter the

concentrations.

The concentration will be the same for all treatment groups

The final option is simpler still. You'll just use a single concentration for all the treatment groups.

Specifying the volume to inject for control groups

If you choose either the second or third option, then there is a small issue to address relating to control groups. Let's imagine that you intend to treat some of your animals with 1mg/Kg of drug, some with 5mg/kg of drug and some with saline. So ANY-maze will calculate the volume to administer for the two drug groups based on the concentration of the solution (which you'll have specified) and the animals' weights. But the saline animals will have no 'concentration' for the solution, so what volume should they receive? A simple answer would be to just specify some fixed volume, perhaps 3ml. This would work, but it would mean that ANY-maze would then tell the experimenter to inject a volume it has calculated, based on the animal's weight, for each animal in the drug groups (so these would be varying numbers, as the animals' weights will vary) and *exactly* 3ml for all saline animals. If you're trying to run the experiment blind, then the fact that all the animals in one group are being treated with an identical volume, irrespective of the animal's weight, would be a bit of a giveaway that these animals are the controls.

To address this you need to enter the volume you'd like to administer to the control animals, 3ml in my example, and the weight of animal that should receive this volume - for example 100g. Then ANY-maze will adjust the volume based on the animals' actual weights - so an animal weighing 120g would receive 3.6ml. This would mean that the control animals would *not* all receive an identical volume (unless they all happen to have identical weights) and this would make it much less likely that the experimenter could use the treatment volume to identify the control animals.

Fields

You can use fields to record additional information about animals and tests, such as animals' weights, or test light levels.

The advantages of recording this information in ANY-maze are that all the data related to an experiment is recorded in one place, and more importantly, that you can use the data recorded in fields as independent or dependent variables in statistical analysis.

For more information about fields, refer to the topics below.

- An introduction to fields
- Setting up a field
- Editing a field
- Deleting a field

ANY-maze help topic H0560

An introduction to fields

Introduction

It's common to record additional information in an experiment beyond just the test's results - for example, you're likely to want to record the weights of the animals you use.

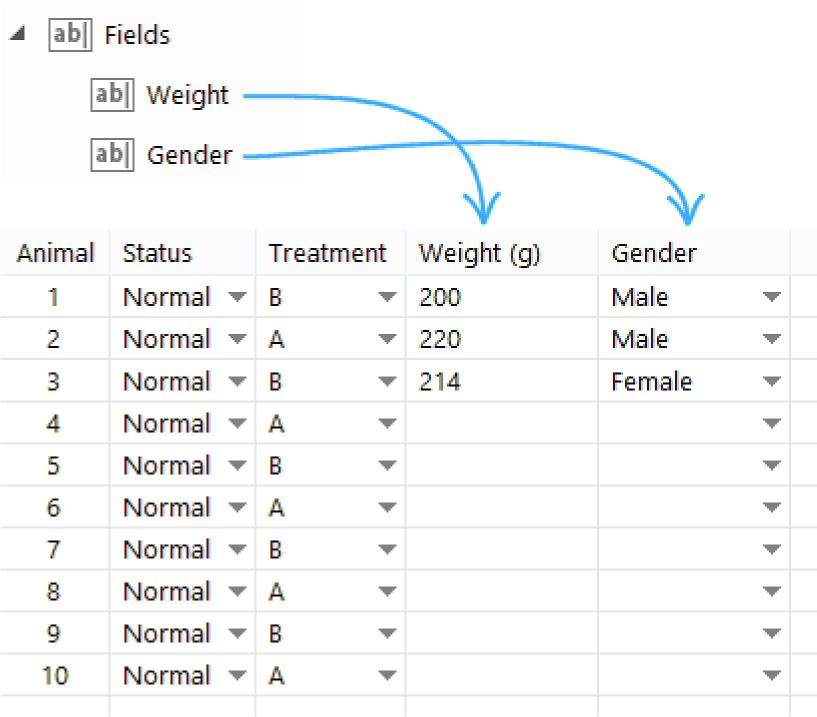
Obviously, this information can easily be recorded on paper and doesn't need to be entered into ANY-maze. Nevertheless, there are some major benefits to recording such additional information in the system:

1. Numeric data, such as animals' weights, can be analysed within ANY-maze. For example, you could use an ANOVA to check that there's no significant difference between the weights of the various treatment groups in an experiment.
2. Data which can have one of a number of specific values, such as gender, can be used as an independent variable in analysis. Thus, for example, you could analyse results between male and female animals; indeed you could use a 2-way ANOVA to analyse both gender and treatment groups.
3. At a purely practical level, recording data within ANY-maze reduces paperwork and keeps all the information relating to an experiment in one place.

Creating fields

To record additional information such as that described above, ANY-maze uses *fields*. Essentially, a field is just a place in ANY-maze where you can enter some information - so, for example, you could create one field to record the animals' weights and another one to record their genders.

Here, it's important to understand that creating a field in a protocol just tells ANY-maze that you want it to include somewhere for you to record this information in an experiment - it doesn't actually mean that you're recording a weight or a gender in the protocol; see figure 1.



Animal	Status	Treatment	Weight (g)	Gender
1	Normal	B	200	Male
2	Normal	A	220	Male
3	Normal	B	214	Female
4	Normal	A		
5	Normal	B		
6	Normal	A		
7	Normal	B		
8	Normal	A		
9	Normal	B		
10	Normal	A		

Figure 1. Creating fields in the protocol causes ANY-maze to include locations to record the specified data in the experiment.

There are three different types of field in ANY-maze - Text, Numeric, and Choice.

Text fields Text fields can accept any data you want to record; however, they can't be used in any analysis. They're useful because they allow you to record data within ANY-maze which would otherwise need to be written on paper.

In versions of ANY-maze prior to V5.1, text fields were used to record Animal IDs. *However, this is now done in its own section in the protocol list - the* ***Animal ID settings page****. All older protocols that used a text field for the animal ID will automatically use this new setting instead.*

Numeric fields Numeric fields can accept any numeric value. A numeric field can be analysed as a dependent variable and can also be included in calculations.

Choice fields Choice fields accept one of a number of different choices such as 'Male' and 'Female'. Choice fields can be used as dependent or independent variables in analysis.

Entering data into fields

As mentioned above, creating a field just tells ANY-maze you'll want to record some information in an experiment - the information itself is entered into either:

- The Animals spreadsheet
- The Animal details report
- The Test details report

Field entries are always optional; that's to say that although you might include a field to record animals' weights, you don't actually have to enter any weights if you don't want to.

Setting up a field

Introduction

For a general introduction to fields, see An introduction to fields.

To add a field to a protocol, click the *Add item* button in the ribbon bar and select *New field* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the field's name.
2. Specify whether the field will be used to record data about animals or tests.
3. Specify the field's type.
4. For a choice field, enter the possible choices.
5. For a numeric field, optionally enter the units.
6. For a numeric field, optionally enter the number of decimal places.

What next?

After completing these steps, you should consider whether you want to include any Virtual switches in your protocol.

See also:

- Adding elements to a protocol
- Editing a field
- Deleting a field

Field name

In brief

Enter a name for the field in the space titled *Field name* on the field's settings page. You must make an entry, but it can be anything you like.

Details

Field names are the only way you will know what's intended to be recorded in the field, so you should make sure the name is concise and meaningful.

You don't need to include words like 'Animal's' or 'Test's' in the field name because ANY-maze will only display animal fields in pages related to animals, and test fields in pages related to tests. For example, you don't need to call a field 'Animal's weight'; just 'Weight' will be fine.

For numeric fields, you shouldn't include the units in the field name, as these are entered separately.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying whether a field will be used to record data about animals or tests

In brief

Fields can either record data about animals or about tests. This division might seem a little odd if your experiments usually only include one test per animal, but consider an experiment where each animal is tested 6 times - you'll have 6 values for a *test field* such as 'water temperature', but only one value for an animal field such as 'gender'.

Select the field type by clicking the appropriate button on the field's settings page.

Details

Although it's usually very obvious whether a field relates to an animal or a test, there are circumstances when it might not be as clear as it seems. For example, you would normally expect to record an animal's weight in an *animal field*. But imagine a situation where you intend to test the animals 6 times over the course of three months. In this case, their weights may differ from one test to another, and you may wish to record their weights as a *test field*.

Specifying a field's type

In brief

There are three types of fields in ANY-maze - *Text, Numeric* and *Choice*. In general, you should use *Numeric* or *Choice* fields in preference to *Text,* as they will allow ANY-maze to use the entries in its analysis. A *Numeric* field can be used to record any number, a *Choice* field can be used to record a single choice from a range of options, while a *Text* field can be used to record anything at all.

Set a field's type using the list shown on the field's settings page.

Details

Text fields	Text fields can accept any data you want to record; however, they can't be used in any analysis. They're useful because they allow you to record data within ANY-maze which would otherwise need to be written on paper.
Numeric fields	Numeric fields can accept any number from -999999 to 999999. If you want to record fractional numbers, then you can specify the number of decimal places that you want the numbers represented to; alternatively, you could change the units you're measuring in (for example, to record 5.5 grams, use 5500 milligrams). The values you enter into a numeric field can be analysed as a dependent variable and can also be included in calculations.
Choice fields	Choice fields accept one of a number of different choices, such as 'Male' and 'Female'. The possible choices are specified as part of the field's definition, and the user will have to choose one of them (or nothing) when making an entry into the field. In other words, if the possible choices are 'Male' and 'Female', the field will only accept these two entries - it won't accept 'Masculine' for example. This is important, because a choice field can be used as a dependent or independent variable in analysis - therefore if the user could enter, for example, both 'Male' and 'Masculine' then the system would treat them as two separate groups.

See also:

- Specifying the possible choices for a choice field

Specifying the possible choices for a choice field

In brief

A choice field is used when the field can only contain one of a predetermined set of values. For example, 'Gender' would be a choice field with the choices being 'Male' and 'Female' (and possibly 'Unknown').

You specify the choices for such a field by entering them as a *comma-separated list* in the space titled *Choices* on the field's settings page - for example: Male,Female

Details

You must enter at least one choice for a choice field. The different choices are entered as a single list with commas between the different possibilities. You can include spaces with the commas if you like; they're just ignored. Below are some examples of valid choice lists:

- Yes,No
- Male, Female
- 1-3 months, 4-9 months, 10-20 months, 21 months or more

Take care to avoid a choice which contains a comma, as it will be considered to be two choices.

When you make an entry into a choice field in the experiment, ANY-maze will automatically complete your entry when it uniquely matches a choice. For example, if you typed 'm' into a 'Male, Female' field, then ANY-maze would complete the entry with 'Male' because none of the other possibilities begins with an 'm'. However, if you typed '1' into a field whose choices were those given in the third example above, then the entry wouldn't be automatically completed because two possibilities begin with a '1'. However, if you then typed a '0', the entry would only match the choice '10-20 months' so this would automatically be completed. You may wish to bear this in mind when you are defining the names of the possible choices.

Restrictions

No single choice can contain more than 80 characters, and the total length of all the choices can't exceed 1200 characters.

Specifying the units of a numeric field

In brief

If you wish to, you can specify the units for a numeric field by making an appropriate entry on the field's settings page. It's best to use the unit's SI abbreviation rather than the full name - for example, 'm' rather than 'metres'.

Details

Specifying units of a numeric field is optional; you can just leave them blank if you want to.

If you do specify units, then ANY-maze will show them in brackets after the field's name, and will quote them in analysis and in reports.

Restrictions

You can enter anything you like up to a maximum of 8 characters.

Specifying the number of decimal places for a numeric field

In brief

If you wish to, you can specify the number of decimal places for a numeric field by making an appropriate entry on the field's settings page. This value specifies how many decimal places will be used when displaying any values entered for this field.

Details

Specifying the number of decimal places for a numeric field is optional; you can just leave this blank if you want to. If you leave it blank, then a value of 0 decimal places will be used (i.e. all the values for this field will be displayed as whole numbers).

If you do specify a value, then whenever ANY-maze displays a value for this field, it will be displayed to the specified number of decimal places. Wherever ANY-maze uses the value in a calculation (for example, as a Result), then the value will be rounded to this number of decimal places.

Restrictions

You can enter any number between 0 and 5.

Editing a field

Introduction

You can edit absolutely everything related to a field at any time, whether before, during or after the experiment. However, although there are no restrictions on what you can edit, there are a few things to be aware of.

Points to note

1. If you change the type of a field, then any entries that have already been recorded for the field will be deleted - ANY-maze will warn you first. For example, if you create a *Text field* called gender and enter the gender of three animals and then change the field's type to *Choice,* then the entries you made for the three animals will be discarded.
2. If you change what a field is recorded for (i.e. for animals or for tests), then entries that you've already made for the fields will be deleted.
3. When you make an entry for a choice field, the entry is actually recorded as the *index* of the choice. For example, if you created a field to record animal gender as either 'Male' or 'Female' and you then recorded an animal as 'Male', what would actually be stored would be the index of the choice - i.e. zero. This means that if you changed the choice 'Male' to 'Masculine' then all the animals would automatically be updated appropriately. However, you should take care with such edits, because if for example you changed the choices from 'Male, Female' to 'Feminine, Masculine', then all the 'Male' animals would suddenly appear as 'Feminine'!
4. If you reduce the number of choices available for a choice field, then any data recorded for choices which no longer exist will be deleted. For example, if you set up a choice field to record animals' ages as '3-6 months, 7-12 months, > 12 months' and you removed the choice '> 12 months', then any animals which were recorded as '> 12 months' would simply have their age discarded.

See also:

- Editing the elements of a protocol
- Setting up a field
- Deleting a field

ANY-maze help topic H0569

Deleting a field

Introduction

You can delete a field at any time, whether before, during or after an experiment has been performed.

To delete a field, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Points to note

If you delete a field, then any entries in the experiment which have been made for it will also be deleted.

Restrictions

There are no restrictions on deleting fields.

See also:

- Deleting protocol elements
- Editing a field

Virtual switches

You can use virtual switches to determine on/off type events based on other information ANY-maze already knows. For example, a virtual switch could be used to determine when the animal is moving faster than a certain speed - when it is, the 'switch' will be on; when it isn't, the 'switch' will be off.

Virtual switches can also be used to determine when two or more on/off type events occur. For example, when two behaviours detected using keys are *both* occurring, or alternatively, when *either* of them occurs.

For more information about virtual switches, refer to the topics below.

- An introduction to virtual switches
- Setting up a virtual switch
- Editing a virtual switch
- Deleting a virtual switch
- Virtual switch measures

An introduction to virtual switches

Introduction

As the name implies, a virtual *switch* is something which is either on or off. But why 'virtual'? Well, a virtual switch doesn't really exist; it's just something you create to determine when other things (which do exist) are in a state *you* want to consider as being on or off.

For example, ANY-maze can determine the speed of the animal as it moves about during a test. Let's say that you would like to know when the animal is moving 'fast'. Well, obviously, this requires that you decide what 'fast' actually means. So let's say that you choose a speed of 1 m/s; when the animal is moving faster than this, it's 'fast', and when it's moving slower than this, it isn't 'fast'. So, we have a switch (on when fast, off when not fast) and it's based on something ANY-maze already knows - the speed of the animal. What we have is a virtual switch!

- Thresholding instantaneous values
- Grouping on/off values
- Virtual switches and zones
- Using virtual switches in events

Thresholding instantaneous values

As the example in the introduction shows, a virtual switch can be used to *threshold instantaneous* values within ANY-maze (in the example, we used the value of 1 m/s to *threshold* the animal's speed).

In this context, I am using the term *instantaneous* to refer to something which has a value at an instant in time - like the animal's speed. Compare this to something like the distance the animal has travelled; this has no value at an instant, but rather it has a *cumulative* value over a period of time.

In the example, we used a virtual switch to apply a threshold to the animal's speed, such that the switch was 'on' when the animal was moving faster than the threshold value. But what if you wanted to know when it was moving slower? Well, you can do that too - in fact, you have three types of threshold: *greater than a value, less than a value* and *between two values.* So, for example, you could use three virtual switches to classify the animal's speed as:

- Fast - Moving faster than 1.0 m/s
- Medium - Moving between 0.3 m/s and 1.0 m/s
- Slow - Moving slower than 0.3 m/s

As this classification shows, you can create multiple virtual switches to threshold the same value.

Although the above description makes it sound like a 'greater than a value' virtual switch will always turn on when the value is greater than the threshold and turn off when it is less, this isn't necessarily the case because you can optionally include some *hysteresis*. This is explained fully in the help topic Using hysteresis with a virtual switch, but in brief, the idea is that you specify a upper threshold and a lower threshold. To turn the switch on, the value must be greater than the upper threshold and to turn it off, it must be less than the lower threshold. And what's the benefit of that? Well, it avoids rapid on/off switching when the value is very close to the threshold, which could create artificially high values for the 'count' of switch activations, and that might be a problem in some applications.

There are a number of instantaneous values that can be thresholded using virtual switches in ANY-maze, and we will gladly add more if requested. For now, the list is:

- The speed of the animal.
- The distance of the animal from a zone (when outside the zone). (See note 1)
- The distance of the animal from the border of a zone (when inside the zone). (See note 1)
- The distance of the animal's head from a zone (when outside the zone). (See note 2)
- The distance of the animal's head from the border of a zone (when inside the zone). (See note 2)
- The distance of the animal from a point. (See note 3)
- The distance of the animal's head from a point. (See note 2)
- The relative length of the animal.
- The value reported by a signal.
- The value reported by a sensor. (See note 4)
- The speed of rotations of a rotary encoder.
- The value reported by a VALUE type plug-in.

Notes:

1. Exactly how ANY-maze determines the distance of the animal from a zone depends on whether zone entries *(sic)* are set to use the entire area of the animal or the animal's centre point (see Choosing how ANY-maze should detect entries into a zone for details). If entries are based on the entire area of the animal, then the calculation of the distance from the animal to the zone will also be based on the entire area of the animal - specifically, the system will use the distance from the point on the animal's edge that is closest to the zone border; on the other hand, if zone entries are based on the centre of the animal, then the distance to the zone will also be based on the centre of the animal - i.e. the distance to the zone will simply be the distance from the centre to the nearest part of the zone.
2. Only available if head tracking is switched on.
3. You can specify how ANY-maze calculates the distance of the animal from a point using an option in the Point's settings in the protocol. Specifically, the distance is either the shortest distance from any point on the animal's edge to the point, or it is

the distance from the centre of the animal to the point.

4. Only sensors which are set to be read continuously throughout the test will be listed.

Grouping on/off values

Another way to turn a virtual switch on and off is to use other on/off items within ANY-maze. For example, you could specify that a switch should be on when either a photobeam is broken, or a key on the keyboard is pressed.

This may seem a little esoteric at first, but let's consider a real example. Imagine you are observing a mother and some pups and you are using ANY-maze to score 6 behaviours using keys on the keyboard of your computer. Three of these behaviours you classify as 'maternal' and three you classify as 'general' (i.e. not benefiting the pups). You run your tests and you press the keys when each behaviour is occurring.

Now you want to see how much time the mother spent exhibiting maternal behaviour. You might choose to simply sum the time of the three individual maternal behaviours, but what if the mother could exhibit more than one at the same time (for example simultaneously suckling and cleaning a pup). In this case, if you sum the total for each behaviour you would get an erroneous result.

Instead, you should create a virtual switch, call it 'Maternal behaviour' and specify that it is on when any of the three maternal behaviour keys is pressed. The total 'on' time of *this virtual switch* will tell you the amount of time the mother was exhibiting maternal behaviour.

In this example, we used one way of grouping on/off items - we said the virtual switch is on when *any* of the constituent keys was on. However, you can also specify that a virtual switch should be on only when *all* the constituent items are on - for example, this could be used to find out how often two maternal behaviours were being exhibited at the same time.

Any combination of the following items can be used to switch a 'grouped' virtual switch on and off:

- Whether the animal is in a specific zone
- Whether or not the animal is oriented towards the centre of a specific zone
- Whether or not the animal is oriented towards a specific point
- Whether the animal is mobile
- Whether the animal is freezing
- Whether the animal is rearing
- Whether a key (being used to manually score behaviours) is pressed
- Whether an on/off input is active (for example, a switch or photobeam)
- Whether a movement detector is sees the animal as moving
- Whether an output switch is active
- Whether a syringe pump is running
- Whether a shocker is active

- Whether another virtual switch (whether a 'threshold' or 'group' type) is active
- Whether an ON/OFF type plug-in is active

3Virtual switches and zones

Virtual switches are automatically analysed both for the apparatus as a whole and also within zones. This means, for example, that a virtual switch which 'thresholds' the animal's speed would allow you to see how much time the animal spent moving 'fast' not just in the entire apparatus, but also in each zone individually.

This also applies to switches that 'threshold' the distance from a zone. Thus, you could see how far the animal is from one zone when it is in other zones. For example, imagine you have divided your apparatus into two zones; one is well-lit, the other less so. In the middle of the apparatus, you have a zone with a novel object in it. You could then look at the distance from the novel object (using a virtual switch), and assess whether it is different when the animal is in the well-lit zone to when it is in the poorly lit one.

3Using virtual switches in procedures

Like many protocol elements in ANY-maze, you can use virtual switches in procedures. This means that a procedure can do such things as turn on a sound if the animal is moving faster than a certain speed; start the video recorder whenever the animal is closer than a certain distance to a zone; or end the test if the combined sum of time 'on' of a number of keys is above a certain value.

Of course, there are many more options than just these, and you can find out more about how to use virtual switches in this way, in the Procedures section of the help.

See also:

- Setting up a virtual switch
- Editing a virtual switch
- Deleting a virtual switch
- Virtual switch measures

Setting up a virtual switch

Introduction

For a general introduction to virtual switches, see An introduction to virtual switches.

To add a virtual switch to a protocol, click the *Add item* button in the ribbon bar and select *New virtual switch* from the menu which appears.

The New Virtual switch menu option will be disabled until the protocol includes at least one apparatus element.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the virtual switch's name.
2. Specify what will cause the virtual switch to turn on.
3. If you set the switch to 'threshold' a measure, choose which measure it should be.
4. If you set the switch to 'threshold' a measure, specify the upper and lower threshold limits.
5. If you set the switch to group other on/off items, specify which items they should be.

What next?

After completing these steps,you should consider whether you want to include any calculations in your protocol.

See also:

- Adding elements to a protocol
- Editing a virtual switch
- Deleting a virtual switch

Virtual switch name

In brief

Enter a name for the virtual switch in the *Virtual switch name* field on the virtual switch's settings page. You must make an entry, but it can be anything you like.

Details

Virtual switch names should specify what it is they will detect. For example, if the switch will determine when the animal is moving at a speed greater than 1 m/s, you might call the switch 'Moving fast' or 'Speed > 1m/s'.

The name of the virtual switch will be included in the names of the switch's results, so to avoid having very long result names, you should try to keep virtual switch names reasonably short.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying what will cause a virtual switch to turn on

In brief

You should use the drop-down list on the virtual switch's settings page to specify what will cause the switch to turn on. There are 5 options available to you - three are used to *threshold* an instantaneous value, and two are used to group other on/off items.

Details

The options available are:

When the measure selected below is greater than a certain value

Use this option to threshold the instantaneous value of a measure, such that the switch is on when the value of the measure is greater than (or equal to) a certain value. You will have to specify the value in the upper limit field. You will also be able to optionally enter a lower limit, so as to add hysteresis to the switch.

When the measure selected below is less than a certain value

Use this option to threshold the instantaneous value of a measure, such that the switch is on when the value of the measure is less than a certain value. You will have to specify the value in the lower limit field. You will also be able to optionally enter an upper limit, so as to add hysteresis to the switch.

When the measure selected below is between certain values

Use this option to threshold the instantaneous value of a measure, such that the switch is on when the value of the measure is between two values. You will have to specify the lower limit (which the value must be greater than or equal to) in the lower limit field and the upper limit (which the value must be less than) in the upper limit field.

When all of the items selected below are active

Use this option to create a virtual switch which will be on when a group of other on/off type items are active. In this case, ALL the items will need to be active simultaneously for the virtual switch to be 'on'.

When any of the items selected below are active

Use this option to create a virtual switch which will be on when a group of other on/off type items are active. In this case, if ANY of the items is active, the virtual switch will be 'on' and will remain 'on' until all the items are inactive.

See also:

1. Specifying the measure to 'threshold' in a virtual switch
2. Specifying threshold limits of a virtual switch
3. Specifying the on/off items that control a virtual switch

Specifying the measure to 'threshold' in a virtual switch

In brief

If you select one of the 'threshold' options for what will cause a virtual switch to turn on, then the main list box on the virtual switch settings pane will show a list of the measures which can be thresholded. You will need to choose one of them.

Details

A 'threshold' type virtual switch will be on when the instantaneous value of some measure is *above, below,* or *between* certain values (depending on the switch type you chose).

In this case, ANY-maze will show a list of all the measures which have *instantaneous* values, and you will need to choose one of them. Here, *instantaneous* is being used to describe a value which has a measurable value at an instant in time - for example, the speed of the animal (contrast this to the distance the animal has travelled, which is a *cumulative* value).

Specifying threshold limits of a virtual switch

In brief

If you select one of the 'threshold' options for what will cause a virtual switch to turn on, then you will need to specify that actual threshold values in the *Upper limit* and *Lower limit* fields.

Details

Upper limit

If you have specified that the virtual switch will be on when the value of a measure is greater than a certain value, then you should enter that value in the *Upper limit* field. In fact the switch will be on if the value is greater than OR EQUAL TO this value. This is usually such a subtle difference as to be unnoticeable. However, it becomes important if you create multiple threshold virtual switches with the same thresholds so as to classify a value. For example, if you specify that when the animal is moving at a speed greater than 1m/s it is 'Fast', and when it is moving at less than 1m/s it is slow, then it is important that at a speed of exactly 1m/s the animal is not classified as both 'Fast' and 'Slow' - it won't be; it will be 'Fast'.

If you specify that the virtual switch will be on when the value of a measure is between two values, then you should enter the larger value here. The switch will only be on when the value of the measure is LESS THAN this value (and GREATER THAN OR EQUAL TO the value in the *Lower limit* field).

If you specify that the virtual switch will be on when the value of a measure is less than a certain value, then you can *optionally* enter an upper limit in this field so as to create some hysteresis.

Lower limit

If you have specified that the virtual switch will be on when the value of a measure is less than a certain value, then you should enter that value in the *Lower limit* field. The switch will be on if the value of the measure is LESS than the value you enter.

If you specify that the virtual switch will be on when the value of a measure is between two values, then you should enter the smaller value here. The switch will only be on when the value of the measure is GREATER THAN OR EQUAL TO this value (and LESS THAN the value in the *Upper limit* field).

If you specify that the virtual switch will be on when the value of a measure is greater than a certain value, then you can *optionally* enter a lower limit in this field so as to create some hysteresis.

Using hysteresis with a virtual switch

Introduction

A 'threshold' virtual switch can be created such that it will turn on when some measure is greater than or equal to a certain value. By implication, this means that the switch will turn off when the measure is less than that value. But having a single on/off value can lead to rapid switching when the measure is very close to the threshold and this can yield artificially high 'activation counts' for the switch. This problem can be overcome using hysteresis.

Details

The best way to describe the problem that hysteresis is designed to overcome is by an example. Imagine that you might create a virtual switch to determine when the animal is moving 'fast' and you choose to set the switch to turn on when the animal is moving at 1.0 m/s or faster.

Now let's analyse what will actually happen when the animal is moving at a speed very close to 1.0 m/s. Let's suppose its speed is 0.5 m/s; 2 seconds later it is 0.95 m/s and then it gets a little faster and is moving at 1.05 m/s, so the switch turns on. But a moment later, it is moving at 0.95 m/s again so the switch turns off; a moment later it's at 1.0 m/s so the switch turns on again and then it gets faster, reaching 1.5 m/s - this is represented graphically in figure 1.

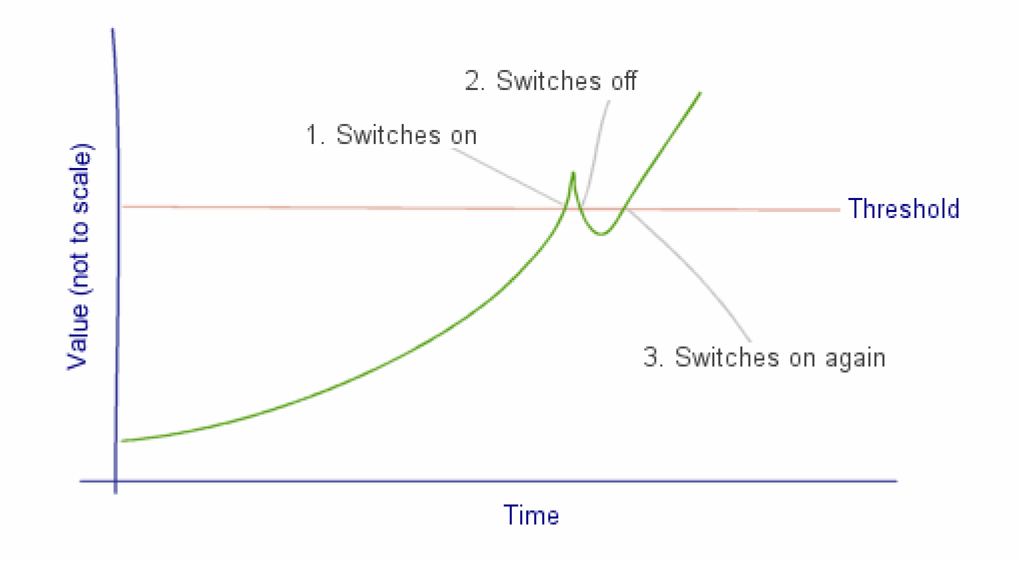


Figure 1. With a single threshold, two switch activations are detected as the value oscillates slightly around the threshold value.

So during this sequence we will have detected two switch activations, which is *strictly* correct, but if you wish to analyse the number of bouts of moving 'fast' you would probably want to consider this sequence as being one such bout. Furthermore, this problem could be much worse than described above, and a little 'noise' at the switching value could yield very large counts.

To overcome this we need to add a little *hysteresis* to our switch. This works by setting *two* thresholds, one to turn the switch on and another to turn it off. Let's say, for example, that we set the 'on' threshold to be 1.0 m/s and the 'off' threshold to be 0.90 m/s. This means that to turn the switch on the animal's speed needs to be 1.0 m/s or greater, but once the switch is on it won't turn off again until the speed drops below 0.90 m/s, thus small oscillations in speed close to the 'on' threshold *won't* cause the rapid switching we saw in the example above. This is shown graphically in figure 2.

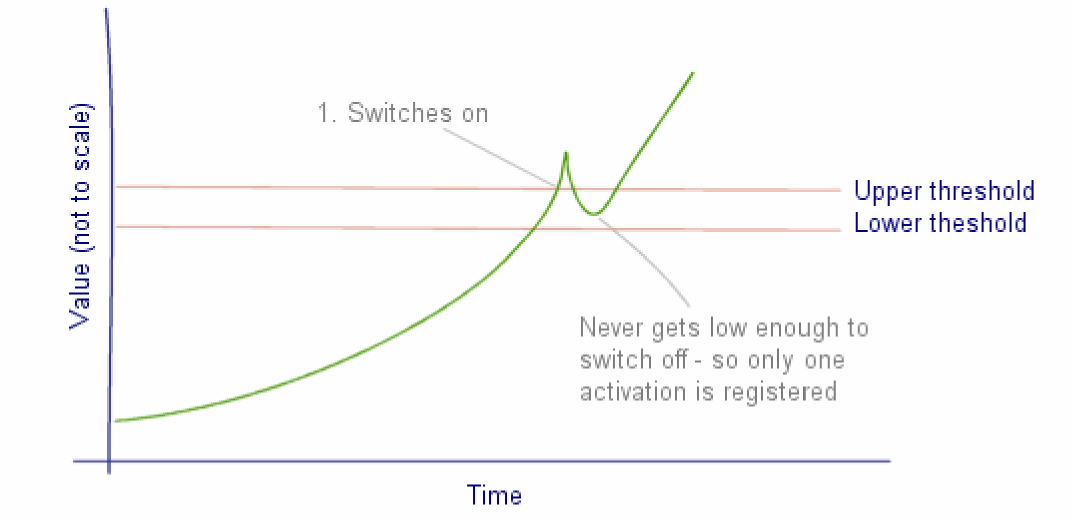


Figure 1. With two thresholds, we introduce some hysteresis into the switch. Now the oscillations are not enough to deactivate the switch, so only a single activation is detected.

So by using hysteresis, we can make our switch more stable, but how do you actually *set* the hysteresis for a virtual switch? The answer is that when creating a 'greater than threshold' type of virtual switch, you can optionally enter an 'off' threshold in the *Lower limit* field. This should be less than the 'on' threshold (which you enter in the *Upper limit* field) but only by a small amount; around 10% is usually fine, although this will depend on your conditions.

You can also apply hysteresis to a 'less than threshold' type of virtual switch. In this case, the 'on' threshold will be the value you enter in the *Lower limit* field, so you should enter the 'off' threshold in the *Upper limit* field. A value about 10% greater than the 'on' threshold should be about right.

The other type of threshold virtual switch is one that is on when the measure is *between two values*. In this case, you can't apply any hysteresis to the switch, as you will have to enter both an upper and a lower limit anyhow.

Specifying the on/off items that control a virtual switch

In brief

If you specify that a virtual switch will be on when other 'on/off' type items are either all on or any of them are on, then you will need to specify which items these will actually be.

Details

When you select that a virtual switch will be on when other on/off items are on, then ANY-maze will show a list of all the available on/off items and you will be able to simply select those you want to include in the switch.

You can include as many items as you like, and they can be of diverse types (so they needn't all be keys, or all photobeams).

If you include just a single item, then the virtual switch will essentially be the same as that item, making it rather pointless, but ANY-maze won't complain.

If you fail to select any items, then the virtual switch will never be activated; in this case, ANY-maze will warn you that this is the case.

See also:

- Specifying what will cause a virtual switch to turn on

Editing a virtual switch

Introduction

You can edit absolutely everything related to a virtual switch at any time, whether before, during or after an experiment has been performed.

Details

Any edits you make to a virtual switch will immediately be reflected in the results for all tests. Thus, for example, if you change the upper limit for a 'threshold' type virtual switch, all the tests you have already performed will be re-analysed using the new threshold. Of course, all tests you perform from this point on will also use the new threshold value.

However, if a virtual switch has been used in a procedure, any changes to the virtual switch will, necessarily, only influence future tests.

See also:

- Editing the elements of a protocol
- Setting up a virtual switch
- Deleting a virtual switch

Deleting a virtual switch

Introduction

You can delete a virtual switch at any time, whether before, during or after an experiment has been performed.

To delete a virtual switch, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting virtual switches.

See also:

- Deleting protocol elements
- Editing a virtual switch

Virtual switch measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a virtual switch:

- Number of activations
- Time active
- Latency to first activation
- Latency to first deactivation
- Distance travelled before 1st activation
- Distance travelled while active
- Longest activation
- Shortest activation
- Average activation duration
- Frequency of activations

Most measures are available for the apparatus as whole (i.e. irrespective of where the behaviour occurred), and also for each defined zone*. For example, if a protocol includes two zones called 'Left' and 'Right', and a virtual switch has been defined for 'Moving fast', then the following sets of measures will be available:

- The measures listed above for moving fast anywhere in the apparatus.
- The measures listed above* for moving fast in the left zone.
- The measures listed above* for moving fast in the right zone.

**Except for 'Average activation duration' and 'Distance travelled before 1st activation', which are only available for the apparatus as a whole.*

Number of activations

Description	Reports the number of times the virtual switch was activated (turned on).
Calculation method	Counts the number of activations.
Analysis in zones	Counts the number of times the virtual switch was activated while the animal was in the zone. Note that if the virtual switch is active when the animal enters

the zone, an activation will not be counted.

Analysis across time	This measure can be analysed across time. For any time period, the result is the number of activations during the time period.
Units	None
Notes	None

Time active

Description	Reports the total amount of time the virtual switch was active.
Calculation method	Sums the duration of each activation of the virtual switch.
Analysis in zones	Sums the amount of time for which the virtual switch was active when the animal was in the zone. For a particular zone, it's possible for the *Time active* to be non-zero while the Number of activations is zero. This can occur if the animal enters the zone when the virtual switch is active. In this case, the time the virtual switch is active in the zone will be registered, but the activation itself won't be (as it didn't occur in the zone).
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of time that the virtual switch was active during the period. For a particular time period, it's possible for the *Time active* during the period to be non-zero while the Number of activations for the period is zero. This can occur if the virtual switch is already active at the start of the period. In this case the time the virtual switch is active will be registered, but the activation itself won't be.
Units	Seconds
Notes	None

Latency to first activation

Description	Reports the amount of time that elapsed in the test before the virtual switch was activated for the first time.
Calculation method	The value of the test clock at the first activation.
Analysis in zones	The value of the test clock at the first activation that occurred when the animal was in the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds

Notes	None

Latency to first deactivation

Description	Reports the amount of time that elapsed in the test before the virtual switch was deactivated for the first time.
Calculation method	The value of the test clock at the first deactivation.
Analysis in zones	The value of the test clock at the first deactivation that occurred when the animal was in the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

Distance travelled before 1st activation

Description	Reports the distance the animal had travelled in the apparatus up to the moment the virtual switch was first activated.
Calculation method	The accumulated total distance travelled is noted at the time of the first activation.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Metres
Notes	None

Distance travelled while active

Description	Reports the distance the animal travelled in the apparatus while the virtual switch was active.
Calculation method	While virtual switch is active, the distance travelled is accumulated.
Analysis in zones	Calculated by summing the distance the animal travelled while in the zone when the virtual switch was active.
Analysis across time	This measure can be analysed across time. For any time period, the result is the distance the animal travelled during the time period while the virtual switch was active.
Units	Metres

Notes	None

Longest activation

Description	Reports the duration of the longest period for which the virtual switch was continuously active.
Calculation method	The duration of each activation is calculated and the largest value is found.
Analysis in zones	The longest period for which the virtual switch was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest period for which the virtual switch was continuously active during the period.
Units	Seconds
Notes	None

Shortest activation

Description	Reports the duration of the shortest period for which the virtual switch was continuously active.
Calculation method	The duration of each activation is calculated and the smallest value is found.
Analysis in zones	The shortest period for which the virtual switch was continuously active while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest period for which the virtual switch was continuously active during the period.
Units	Seconds
Notes	None

Average activation duration

Description	Reports the average duration for which the virtual switch was held active.
Calculation method	Calculated by dividing the Time active by the Number of activations.
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds

Notes	None

Frequency of activations

Description	Reports the frequency with which the virtual switch was activated.
Calculation method	Calculated by dividing the Number of activations by the *Test duration*.
Analysis in zones	Calculated by dividing the Number of activations in the zone by the *Total time in the zone*.
Analysis across time	This measure can be analysed across time. For any time period, the result is the Number of activations which occurred during the period divided by the period's duration.
Units	Hertz
Notes	None

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures

ANY-maze help topic H0582

Calculations

You can use calculations to derive new measures from those which ANY-maze includes as standard. For example, in a plusmaze it's common to report the percentage of time in the open arms. ANY-maze doesn't include this measure, but it can easily be defined using a calculation.

For more information about calculations, refer to the topics below.

- An introduction to calculations
- Setting up a calculation
- Editing a calculation
- Deleting a calculation

An introduction to calculations

Introduction

Although ANY-maze will automatically calculate a large number of different measures, there may be occasions when you want to calculate a result it doesn't supply as standard.

For example, in a plusmaze protocol, you would probably define an 'Open arms zone' and this would cause ANY-maze to report (amongst other things) the 'Time in the open arms'. However, in the plusmaze it's quite common to report the 'Percentage of time in the open arms' rather than the absolute time, but this is a value which ANY-maze *doesn't* include as standard.

The solution to this, as you'll doubtless have guessed, is to use an ANY-maze calculation. Essentially, a calculation is a way to derive a new measure from those which the system supplies automatically. However, calculations are more powerful than you might at first imagine because they can include *functions* which can calculate results either *across* or *within* an animal's trials.

Calculation formulae

As you'd expect, a calculation is defined using a formula. This can include absolute values, such as 100; the results of measures, such as the 'Time in the open arms'; and the four mathematical operators: plus, minus, multiply and divide.

Here, for example, is a simple calculation formula:

Time in NW zone / 2

ANY-maze calculates the results of a formula by working from the left to the right, so the result of:

20 / 2 + 3

would be 20 / 2 = 10 + 3 = 13. Of course you might have wanted to divide 20 by the sum of 2 and 3, in which case the above formula will give you the wrong answer. To fix it, you need to include parentheses to change the calculation order. For example, the result of the formula:

20 / (2 + 3)

would be 2 + 3 = 5, 20 / 5 = 4. You can *nest* parentheses if you need to - for example, the result of:

20 / (5 - (2 + 1))

would be 10.

Of course, to be useful, a calculation won't just contain numbers but will include the result of at least one measure, or perhaps the result of another calculation. You can easily incorporate the results of almost all the standard measures in calculations. I say almost, because some measures, such as 'First zone entered', yield non-numeric results which clearly can't be used in a calculation.

Functions

As I mentioned in the introduction, calculations can include *functions* which allow you to calculate results both *across* and *within* an animal's trials. For example, imagine you ran a 5-minute test and you'd like to find out what percentage of the total distance travelled by the animal occurred in the first 2 minutes of the test. The formula for this would be:

(ResultForPeriod { Total distance travelled, 0 - 120 } / Total distance travelled) * 100

Here the function *ResultForPeriod* extracts the total distance travelled for just the first 120 seconds of the test (that's what the 0 - 120 bit means). Don't worry if this formula looks a bit complicated, because ANY-maze includes a function wizard which will guide you through the process of choosing and defining a function for inclusion in a calculation.

The ResultForPeriod function extracts a result from *within* a test, but what about those functions I mentioned for results *across* trials? Well, there are a number of them: Count, Sum, Mean, Max, and Min. For each one, you define the first and last trial you want the function to include, and it will calculate the result across the trials. For example:

Sum { Total distance travelled, Acquisition: Trial 1 - Acquisition: Trial 3 }

would sum the distance travelled in an animal's first 3 acquisition trials. Again, the *function wizard* will help you actually create the function, so you needn't worry about the exact syntax used.

ANY-maze has been designed to be extended, and we'll be delighted to add any new functions you might find useful, for free! Just contact ANY-maze technical support.

Calculation results

Once you've defined a calculation, its result can be treated just like any other measure. This means you can include the result in reports, create graphs of the result, and even perform statistical analysis on it. It also means that you can include the result in other calculations, which can occasionally be extremely useful.

Setting up a calculation

Introduction

For a general introduction to calculations, see An introduction to calculations.

To add a calculation to a protocol, click the *Add item* button in the ribbon bar and select *New calculation* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the calculation's name.
2. Specify the number of decimal places for the result.
3. Optionally specify the units of the result.
4. Optionally specify the graph Y axis range.
5. Optionally specify any Named values for use in the formula.
6. Define the calculation's formula.

What next?

After completing these steps, you should check that the protocol's Analysis options are suitable for your protocol.

See also:

- Adding elements to a protocol
- Editing a calculation
- Deleting a calculation

Calculation name

In brief

Enter a name for the calculation in the *Calculation name* field on the calculation's settings page. You must make an entry, but it can be anything you like.

Details

Calculation names should specify the result of the calculation, for example, 'Percentage of time in the Open arms' or 'Total zone transitions'.

Limits

You can enter anything you like up to a maximum of 64 characters.

Calculation decimal places

In brief

In the *Number of decimal places* field on the calculation's settings page, enter the number of decimal places you would like the result of the calculation to be quoted to.

Details

There's little point including large numbers of decimal places in calculation results - indeed, the default value is zero. However, in some cases you may want to include 2 or 3 decimal places, particularly if you are using a calculation to alter the units of a result - for example, quoting times in minutes rather than in seconds.

Results are rounded (rather than truncated) to the number of decimal places you request.

Limits

You can enter any number from 0 - 9.

Calculation units

In brief

In the *Units of the result* field on the calculation's settings page, you can enter the name of the units of the result of the calculation.

Details

You don't have to make an entry, but if you do it will be quoted in reports. Where relevant, I suggest you do include units - preferably using SI symbols, as these will agree with the units ANY-maze uses for the standard measures.

Limits

You can enter anything you like up to a maximum of 8 characters.

Graph Y axis range

In brief

Normally, when ANY-maze displays graphs, it automatically calculates the range for the Y axis based on the data it will plot. However, you may wish to force a graph which is displaying the result of a calculation to have a fixed Y axis range, for example 0 - 100.

You can optionally specify a maximum value for the positive Y axis and/or a minimum value for the negative Y axis using the fields shown on the calculation's settings page.

Details

These values are most useful when the result of a calculation is a percentage, as you can fix the positive Y axis maximum at 100% which makes the graph easier to interpret.

If you don't make an entry for these values, then ANY-maze will automatically calculate the Y axis range when it displays a graph.

If the actual values that will be plotted on a graph exceed the values you enter here, then ANY-maze will, again, automatically calculate a suitable range.

You can use the fact that you can fix the Y axis range for a calculation as a way to override ANY-maze's normal automatic Y axis scaling for the standard measures.

For example, imagine you want to plot graphs of total distance travelled in your experiments. If in a particular experiment the maximum value in the graph is 9m for example, then ANY-maze will automatically set the Y axis maximum to 10. If in a second experiment the maximum value is 17m, then the Y axis maximum will be 20. Now, if you compared the two graphs, they would be a little misleading as the graph for the first experiment would have a larger column than the graph for the second one - of course, the Y axes would be labelled appropriately so the graphs would be 'correct', but you might prefer to have all graphs of total distance travelled plotted with a maximum Y axis value of say, 30 - thus avoiding possible confusion.

As you can probably see, an easy way to do this is to define a calculation whose result is simply the value of the measure you want to plot, in this case 'Total distance travelled'. If you then specify the Y axis maximum for this 'calculation', then a plot of the calculation's results will show exactly the same values as a plot of the measure's results, but with a fixed Y axis.

Limits

For the positive Y axis, you can enter any number from 1 to 999999.

For the negative Y axis, you can enter any number from -1 to -999999.

Using Named values in calculations

In brief

As the name implies, a *Named value* is simply a value which has a name. For example, you'll probably be familiar with the name 'Pi' which has a value of 3.1415926, and you would perhaps find it easier to write a formula such as: (30 * Pi) / 180 rather than: (30 * 3.1415926) / 180, especially as the former also helps to clarify what the formula is for.

In ANY-maze calculations, you can optionally set-up two named values and then use them in the calculation's formula.

Details

The best way to understand Named values is through an example. Imagine you are creating a calculation to sum the time the animal spent in those zones which it entered more than three times. (To keep things simple let's say we have four zones.)

The formula would be:

```
( ( Number of entries to Zone A > 3 ) * Time in Zone A ) +
( ( Number of entries to Zone B > 3 ) * Time in Zone B ) +
( ( Number of entries to Zone C > 3 ) * Time in Zone C ) +
( ( Number of entries to Zone D > 3 ) * Time in Zone D )
```

Here the '>' operator equates to 1 if the value on the left is greater than the value on the right, and 0 otherwise. So if the animal entered Zone A four times, the first part of the formula would be (4 > 3); here the value on the left is greater than the value on the right and so this would equate to 1, which would then be multiplied by the time in Zone A, which would equal the time in Zone A. If the animal entered Zone B three times, then the second part of the formula would be (3 > 3); the value on the left is NOT greater than the value on the right and so this would equate to 0, which would then be multiplied by the time in Zone B, which would equal 0. If you follow through the logic, you'll see that this formula will sum the time the animal spent in zones it visited more than three times.

Now let's imagine you wanted to change the criteria for this formula to sum the time the animal spent in zones it visited more than *five* times. Obviously you'd have to change all of the '3's in the formula to be '5's. This is where a named value would be useful; let's say we created a named value called 'Limit' then we could change the above formula to be:

```
( ( Number of entries to Zone A > Limit ) * Time in Zone A ) +
( ( Number of entries to Zone B > Limit ) * Time in Zone B ) +
( ( Number of entries to Zone C > Limit ) * Time in Zone C ) +
( ( Number of entries to Zone D > Limit ) * Time in Zone D )
```

And we could easily set the value of Limit to be 3, 5 or whatever we want - without having to alter the formula at all.

Setting up a Named value

ANY-maze allows you to define two named values, where each one consists of a name and its value. In the above example we could use just one of the named values, giving it a name of 'Limit' and a value of '3'. (You would simply leave the other named value's fields blank.)

Editing a Named value

You can edit either the name or the value at any time and ANY-maze will immediately update the formula and its result in all tests.

Using a Named value in a calculation's formula

A Named value can be included in a formula by right clicking, or pressing the *Insert* key, and selecting *Named values* from the menu which appears. Note that if you haven't defined any named values then there will be no entry for *Named values* in the menu.

Defining a calculation's formula

In brief

A calculation's formula is entered into the box shown at the bottom of the calculation's settings page. Specifically, you can define a calculation by:

- Entering absolute numeric values.   For example 10.2
- Using the mathematical operators + - * /   For example 20 / 2
- Using the mathematical functions: Abs, Sqrt, Log, Ln, Sin, Cos, Tan, Asin, Acos, Atan.   For example Sqrt 25
- Using the comparison operators = < >   For example 10 > 5
- Setting precedence using parenthesis ( ).   For example 20 / (2 + 3)
- Selecting the value of a measure's result by pressing the *Insert* key and choosing the measure from the menu which appears.
- Including the result of a function by pressing the *Insert* key and choosing *Add function...* from the menu which appears. The function wizard will then help you define the function.

Details

For full details about defining a formula, refer to the sections below:

- The elements of a formula
  - Absolute numbers
  - Mathematical operators
  - Mathematical functions
  - Comparison operators
  - Operator precedence and parenthesis
  - Measure results
  - Named values
  - Functions
- Syntax errors in formulas
- Editing a formula
- External changes which affect a formula

The elements of a formula

A formula can consist of absolute numbers, operators, measure results, and function results.

Absolute numbers

You can include any number in a formula, provided it has no more than 15 digits including a decimal point. For example, 10, 3.14159, 10000000 are all valid numbers. You can't enter negative numbers directly; for example, -10 is illegal, but you can simply use 0 - 10 to create the same result.

Mathematical operators

You can include the four standard mathematical operators + - * / in a formula. As you'd expect, these perform operations of addition, subtraction, multiplication and division respectively.

Mathematical functions

You can include the following mathematical function in a formula:

Abs	Calculates the absolute value of whatever is to the right of it. For example: Abs -20 = 20
Sqrt	Calculates the square root of whatever is to the right of it. For example: Sqrt 81 = 9
Log	Calculates the base 10 logarithm (also called the common logarithm) of whatever is to the right of it. For example: Log 72 = 1.857
Ln	Calculates the natural logarithm of whatever is to the right of it. For example: Ln 72 = 4.277
Sin	Calculates the sine of the number of degrees to the right of it. For example: Sin 30 = 0.5
Cos	Calculates the cosine of the number of degrees to the right of it. For example: Cos 30 = 0.866
Tan	Calculates the tangent of the number of degrees to the right of it. For example: Tan 30 = 0.577
Asin	Calculates the inverse sine in degrees of the value to the right of it. For example: Asin 0.866 = 60°
Acos	Calculates the inverse cosine in degrees of the value to the right of it. For example: Acos 0.5 = 60°
Atan	Calculates the inverse tangent in degrees of the value to the right of it. For example: Atan 1.732 = 60°

You can apply multiple functions, one after the other; for example, Sqrt Abs -81. Here the Sqrt function will be performed on Abs -81, and Abs will be performed on -81, so the result will be: abs -81

= 81, Sqrt 81 = 9.

A mathematical function can be included in a formula by right clicking, or pressing the *Insert* key, and selecting *Maths functions* from the menu which appears.

Comparison operators

You can include the comparison operators = > < in a formula. These operators compare the values on either side of them and create result of 1 if the comparison is true and 0 if it is false.

This is more useful than it might at first appear, because you can *multiply* the comparison result by some other value to implement what is effectively an 'if' statement. For example, the formula ((MyValue > 10) * Result1) + ((MyValue < 10) * Result2) would yield a result of Result1 when MyValue is greater than 10 and a value of Result2 when it is less than 10 (you would need to add another clause to cope with the situation when MyValue is equal to 10).

Operator precedence and parenthesis

There is no operator precedence as such in ANY-maze formulae; rather, formulae are always calculated from the left to the right. This means, for example, that the result of 20 / 2 + 3 is 13, because 20 / 2 = 10 and 10 + 3 = 13.

You can alter this left-to-right calculation order using parenthesis. For example, in the formula 20 / (2 + 3), ANY-maze will first calculate 2 + 3 and then divide 20 by the result.

If parentheses are *nested,* then calculation starts with the inner-most parentheses. For example, in the formula 20 / (5 - (2 + 1)), ANY-maze will first calculate 2 + 1, then subtract the result from 5 and finally divide 20 by the answer; i.e. this would be 2 + 1 = 3, 5 - 3 = 2, 20 / 2 = 10.

Measure results

The result of a measure can be included in a formula by right clicking, or pressing the *Insert* key, and selecting the appropriate measure from the menu which appears - see figure 1.

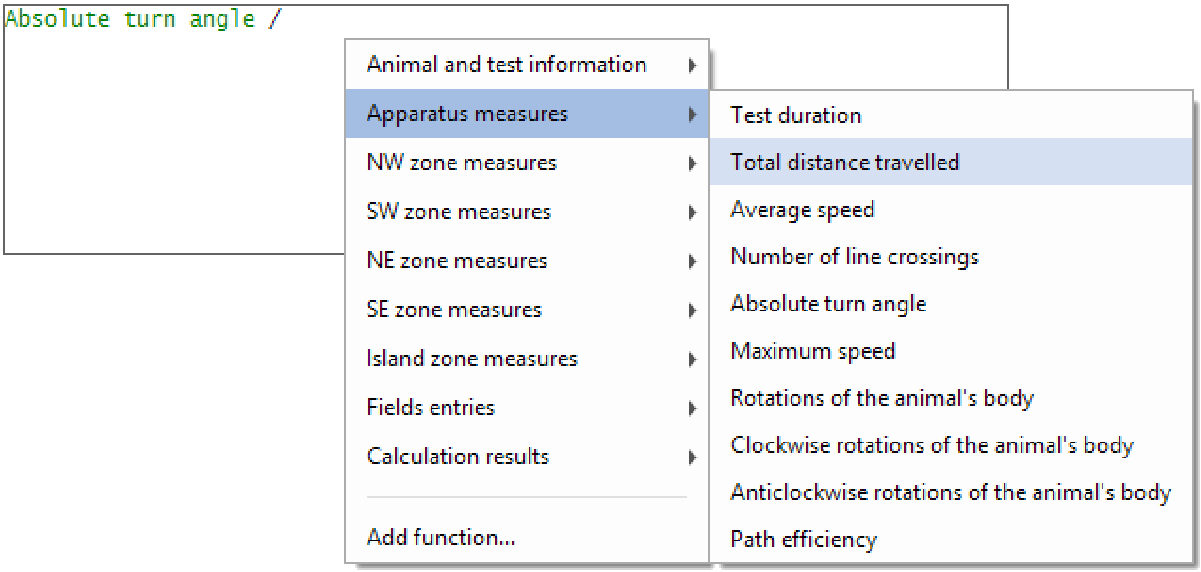


Figure 1. To add a measure result to a formula, press the Insert *key and choose the measure from the menu which appears.*

Measures that don't have numeric results, such as *First zone entered,* won't be included in the list shown. Measures appear in formulae in green.

Named values

A Named value can be included in a formula by right clicking, or pressing the *Insert* key, and selecting *Named values* from the menu which appears. Note: If you haven't defined any named values then there will be no entry for *Named values* in the menu.

Functions

To add a function to a formula, you should press the *Insert* key and choose *Add function...* from the menu which appears. This will cause the function wizard to open which will guide you through the necessary steps.

After adding a function to a formula, its full definition will be shown in blue.

ANY-maze includes the following functions:

Count — Counts the number of trials (from a specific group of trials) for which an animal has a result - i.e. for which the result is NOT undefined. For example, you could use this to find out in how many trials of a stage the animal entered a particular zone. If the animal never enters a zone, then the result for 'Time in

	the zone' will be undefined, so counting how many trials have a result for 'Time in the zone' will tell you in how many trials it entered the zone.
Sum	Calculates the sum of a particular measure across a group of trials. For example, you could use this to find out the total amount of time that the animal spent in a particular zone in all the trials of a particular stage.
Mean	Calculates the mean value of a particular measure across a group of trials. For example, you could use this to find out the mean amount of time that the animal spent in a particular zone in the trials of a particular stage.
Max	Calculates the maximum value of a particular measure in a group of trials. For example, you could use this to find out the maximum amount of time the animal spent in a particular zone in the trials of a particular stage.
Min	Calculates the minimum value of a particular measure in a group of trials. For example, you could use this to find out the minimum amount of time that the animal spent in a particular zone in the trials of a particular stage. Note that this returns the minimum *result* of those trials which actually have a result for the measure. So if, for example, an animal spent 10s and 50s in a zone in the first two trials of a stage and then didn't enter the zone at all in the third trial, then the *Minimum* time in the zone in the three trials is 10s, not zero.
ResultForTrial	Returns the result of a particular measure for a specific trial in a specific stage. This can be useful when you want to calculate the results of one trial in relation to another. For example, to calculate the change in distance travelled between the first trial and subsequent trials in a stage.
ResultForLastTrial	Returns the result of a particular measure for the last trial in a stage. This is useful when a stage can be ended by a Stage end rule, as the index of the last trial will be different for different animals. For example, for one animal the last trial might be trial 4 whereas for another animal it might be trial 6.
ResultForPeriod	Returns the result of a particular measure for just a part of a test. For example, you could use this function to find out the distance travelled by the animal in the first two minutes of a five minute test. If a particular test ended during the period you specify, then its result will be the value up to the end of the test. If a particular test ended *before* the start of the period you specify, then its result will be undefined.

ANY-maze has been designed to be extended, and we'll be delighted to add any new functions you might find useful, for free! Just contact ANY-maze technical support.

Syntax errors in formulas

If you define a formula which doesn't make sense, then ANY-maze will tell you that the formula contains a *syntax error*. This just means that ANY-maze doesn't understand it. For example, entering 10 * / 2 would generate a syntax error.

When a syntax error is detected ANY-maze will highlight, in red, the part of the formula it doesn't understand - this usually makes it very easy to find and correct the mistake. Below are some points to note about the formula syntax:

- Operators + - * / must have an absolute number, measure, or function on either side of them. For example, '10 * 2' is valid; '10 * / 2' is not.
- All opening brackets must have a corresponding closing bracket. For example, '(10 + 2)' is valid; '(10 + 2' is not.
- There must be an operator between any two absolute numbers, measures, or functions. For example, '10 / Test duration' is valid; '10 Test duration' is not.

Editing a formula

You can edit a formula much like you would text, i.e. by simply moving the cursor to the relevant position and then entering or deleting characters. However, you should note that measure names and function definitions act like a single 'character'.

- To move about in the formula, use the left, right, up and down arrows.
- To move the cursor to an arbitrary position, move the mouse pointer there and click.
- To delete a value, operator, measure, or entire function that's *in front of* the cursor, press the *Delete* key.
- To delete a value, operator, measure, or entire function that's *behind* the cursor, press the *Backspace* key.
- To edit a function, move the cursor to the *start* of the function and then press the *Insert* key. Select the *Edit function...* option from the menu which appears.
- Formulae are *self-formatting;* in other words, spaces appear automatically and you can't alter them.
- You can't make selections in a formula like you can in text; nor can you cut, copy or paste.

External changes which affect a formula

As you may realise, it's possible to make alterations to a protocol which will have side effects on formulas. For example, if a protocol includes the formula:

(Time in the open arms zone / Test duration) * 100

then deleting the open arms zone will obviously have a rather dramatic effect on the formula! In these cases, ANY-maze will replace parts of a formula which are no longer valid with ???, so our formula would become:

(??? / Test duration) * 100

The result of such formulae is considered to be undefined, which means that ANY-maze will simply act as if the formula has no result at all.

See also:

- Adding elements to a protocol
- Editing a calculation
- Deleting a calculation

The Function wizard

Introduction

ANY-maze functions allow you to calculate results *across* multiple trials and *within* individual trials.

In fact, functions are not limited to just this type of operation, and ANY-maze could also include many other functions too. If there are additional functions you would find useful, then please contact ANY-maze technical support - we'll be delighted to add them if we can (for free).

Using the function wizard

The function wizard is designed to help you use the ANY-maze functions. It will guide you a step at a time through the processes of defining a function for inclusion in a calculation.

- To move through the wizard, simply click the *Next >* button.
- If you want to go back to a previous step, perhaps to alter an answer you gave, then you can click the *< Back* button.
- You can cancel the wizard at any time, without adding a function to your calculation, by clicking the *Cancel* button.

The Function wizard : Choosing a function

In this step, you should choose the function you want to use. The available functions are listed in alphabetical order, and as you select one, a brief description will be shown below the list.

Functions available in ANY-maze

Count	Counts the number of trials (from a specific group of trials) for which an animal has a result - i.e. for which the result is NOT undefined. For example, you could use this to find out in how many trials of a stage the animal entered a particular zone. If the animal never enters a zone, then the result for 'Time in the zone' will be undefined, so counting how many trials have a result for 'Time in the zone' will tell you in how many trials it entered the zone.
Sum	Calculates the sum of a particular measure across a group of trials. For example, you could use this to find out the total amount of time that the animal spent in a particular zone in all the trials of a particular stage.
Mean	Calculates the mean value of a particular measure across a group of trials. For example, you could use this to find out the mean amount of time that the animal spent in a particular zone in the trials of a particular stage.
Max	Calculates the maximum value of a particular measure in a group of trials. For example, you could use this to find out the maximum amount of time the animal spent in a particular zone in the trials of a particular stage.
Min	Calculates the minimum value of a particular measure in a group of trials. For example, you could use this to find out the minimum amount of time that the animal spent in a particular zone in the trials of a particular stage. Note that this returns the minimum *result* of those trials which actually have a result for the measure. So if, for example, an animal spent 10s and 50s in a zone in the first two trials of a stage and then didn't enter the zone at all in the third trial, then the *Minimum* time in the zone in the three trials is 10s, not zero.
ResultForTrial	Returns the result of a particular measure for a specific trial in a specific stage. This can be useful when you want to calculate the results of one trial in relation to another. For example, to calculate the change in distance travelled between the first trial and subsequent trials in a stage.
ResultForLastTrial	Returns the result of a particular measure for the last trial in a stage. This is useful when a stage can be ended by a Stage end rule, as the index of the last trial will be different for different animals. For example, for one animal the last trial might be trial 4 whereas for another animal it might be trial 6.
ResultForPeriod	Returns the result of a particular measure for just a part of a test. For example,

you could use this function to find out the distance travelled by the animal in the first two minutes of a five minute test. If a particular test ended during the period you specify, then its result will be the value up to the end of the test. If a particular test ended *before* the start of the period you specify, then its result will be undefined.

ANY-maze has been designed to be extended, and we'll be delighted to add any new functions you might find useful, for free! Just contact ANY-maze technical support.

The Function wizard : Choosing a measure

In this step, you need to choose the measure you want the function to apply to. For example, if you want to calculate the distance the animals travelled in the first two minutes of the test, then the measure would be *Total distance travelled*.

As protocols typically include large numbers of measures, they're shown here in groups. To choose the measure you want to use, simply click it with the mouse - it will then be shown highlighted.

You may find that some measures you're used to seeing aren't listed here. This is because ANY-maze only shows measures which can logically be used with the function you chose in the previous step.

For full details about the individual measures, refer to the following topics:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

The Function wizard : Specifying the period of the test

In this step, you need to specify the period of the test that you want the measure's result calculated for.

You can either specify the period by entering both a *From* and a *To* value, with the *To* value being greater than the *From* value, or by selecting a *Time period* defined in the protocol. If you enter a *From* and *To* time, then you can use units of s (seconds), min (minutes) or h (hours) with your entry. You can also enter decimals, so for example, 30 seconds could be entered as '30s' or '0.5min'. You can enter any value from 1s to 24h. If you enter a fractional number of seconds, the fractional part will be discarded.

Also note that:

- When evaluating the result of a particular test, ANY-maze will only include that part of the test that falls inside the period you specify.
- If the test ends during this period, then the function will be evaluated from the *From* time you specify up to the end of the test.
- If the test ends *before* the *From* time you specify, then the result of the function will be undefined.

For exact details of how ANY-maze calculates individual measures for analysis across time, refer to the following topics:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures

- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

The Function wizard : Specifying the trials to include

This step is just designed to make it easier to define a function which will process all the trials in an individual stage, as this will very often be what you want to do. If this is the case for your function, then simply select the second of the two options shown.

If you don't want to include **all** the trials in a stage, or if you want to include trials from more than one stage, then you should select the first of the two options shown in this step.

See also:

- An introduction to stages

The Function wizard : Choosing the stage

As you've chosen to calculate the 'Result for last trial', ANY-maze needs to know which stage you're referring to. For example, you might be using this function to extract the result for the last trial in a training stage; in which case, you would simply need to select the 'Training' stage in the list shown.

See also:

- An introduction to stages

The Function wizard : Choosing the stage

As you've specified that you wanted the result for this function to be calculated for 'All the trials in one stage', then all you need to do in this step is specify which stage it is.

For example, if you wanted the result calculated for all the trials in an 'Acquisition' stage, then you would simply select 'Acquisition' from the list of stages shown here.

See also:

- An introduction to stages

The Function wizard : Specifying the trials

In this step, you need to specify which trials you want the result of the function to be calculated for. You do this by specifying the first and last trials to be included.

For example, if you just want to calculate the result for the first 3 trials of a 6-trial acquisition stage, then you would select 'Acquisition' in the top list and type '1' in the box below it; you'd then select 'Acquisition' in the lower list and type '3' in the box below it.

You can also select trials that span more than one stage. For example, if you have a protocol which includes three stages (let's call them stages 1, 2 and 3 for simplicity), each of which includes 4 trials, then specifying 'Stage 1' and 'trial 3' in the top part, and 'Stage 3' and 'trial 2' in the bottom part would cause the following eight trials to be included in the result of the function:

Stage 1, trial 3

Stage 1, trial 4

Stage 2, trial 1

Stage 2, trial 2

Stage 2, trial 3

Stage 2, trial 4

Stage 3, trial 1

Stage 3, trial 2

- The stages/trials you specify are inclusive
- If you don't specify a *First trial,* then ANY-maze will use the first trial in the 'First stage'.
- If you don't specify a *Last trial,* then ANY-maze will use the last trial in the 'Last stage'.
- You can specify trials which *might* not exist. For example, in a training stage, animals may stop being tested when they achieve a certain goal. Thus, some animals might have 3 trials, others 4 trials and others 5 or 6. If you specified that a function is to be calculated for trials 1 to 4 of the training stage, then an animal which only had 3 training trials won't have a trial 4 - in this case, the animal's result for the function will just include those trials it *does* have, i.e. 1 to 3. On the other hand, you might specify that the function is to be calculated for trials 4 to 6, in which case the result of the function for an animal which had only 3 trials will be undefined.

See also:

- An introduction to stages

ANY-maze help topic H0601

The Function wizard : Specifying the trial

In this step, you need to specify which trial you want the result of a *ResultForTrial* function to return.

You simply need to select a stage and enter a trial number. If you don't enter a trial number, then trial 1 will be assumed.

See also:

- An introduction to stages

The Function wizard : Finished

In this step, you should check the function definition to ensure it's correct, and assuming it is, click the *Finish* button to add the definition to your calculation's formula.

- If you want to alter the definition of the function, then you can simply click the *< Back* button to work backwards through the wizard's steps and change anything you like.
- If you decide you don't want to include the function at all, then just click the *Cancel* button to close the wizard without adding the function to the calculation.

Once you've added a function to a calculation, you can still edit or delete it if you want to. Full details can be found under *Editing a formula* in the Defining a calculation's formula topic.

Editing a calculation

Introduction

You can edit absolutely everything related to a calculation at any time, whether before, during or after an experiment has been performed. If you do this, then the result of the calculation for tests that have already been performed will use the new calculation.

For details about how to edit a calculation's formula, see 'Editing a formula' in Defining a calculation's formula.

See also:

- Editing the elements of a protocol
- Setting up a calculation
- Deleting a calculation

Deleting a calculation

Introduction

You can delete a calculation at any time, whether before, during or after an experiment has been performed.

To delete a calculation, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting calculations.

See also:

- Deleting protocol elements
- Defining a calculation's formula
- Editing a calculation

Analysis across time

Introduction

In many experiments, you are likely to want to look at not only the results for the entire duration of the tests, but also at the results for different parts of the tests - for example, the first minute, the second minute, and so on.

In ANY-maze, there are two ways to perform this type of *Analysis across time*. You can either divide tests into equal length *segments,* or you can create *time periods* covering any arbitrary part of the test. Whichever method you use, you can then view, plot and analyse results between the segments or periods.

- Using equal length test segments to perform analysis across time
- Creating time periods for analysis across time

Using equal length test segments to perform analysis across time

Performing analysis across time using *time segments* is the quickest and simplest method, as you just have to specify the duration of the segments and ANY-maze does everything else.

Specifying the duration of time segments

The length of time segments is set in the *Analysis across time* section of the protocol's *Analysis* element.

The default segment duration is 30 seconds, but you can change it to any value from 1 second to 24 hours. You can enter the duration using the units s, min, and h for seconds, minutes, and hours respectively. You can also use decimals, so for example '1.5min', '1min 30s' and '90s' are all the same duration.

You can alter this value at any time, including after an experiment is complete, and ANY-maze will simply re-analyse your results automatically, which usually only takes a few moments.

Viewing results across time

The key to working with results that are being analysed across time is the *Segment of test* measure. This measure represents the different time segments that the test has been broken into, so when the segment duration is 30 seconds, this measure has values of '0-30 seconds', '30-60 seconds' and so on.

You can use this measure as a grouping measure for text reports, as the x axis value in graph reports, or as an independent variable for statistical analysis. Taking a graph report as an example, this would generate a graph showing time on the x axis and the value of some dependent variable on the y-axis

- as is shown in figure 1, below.

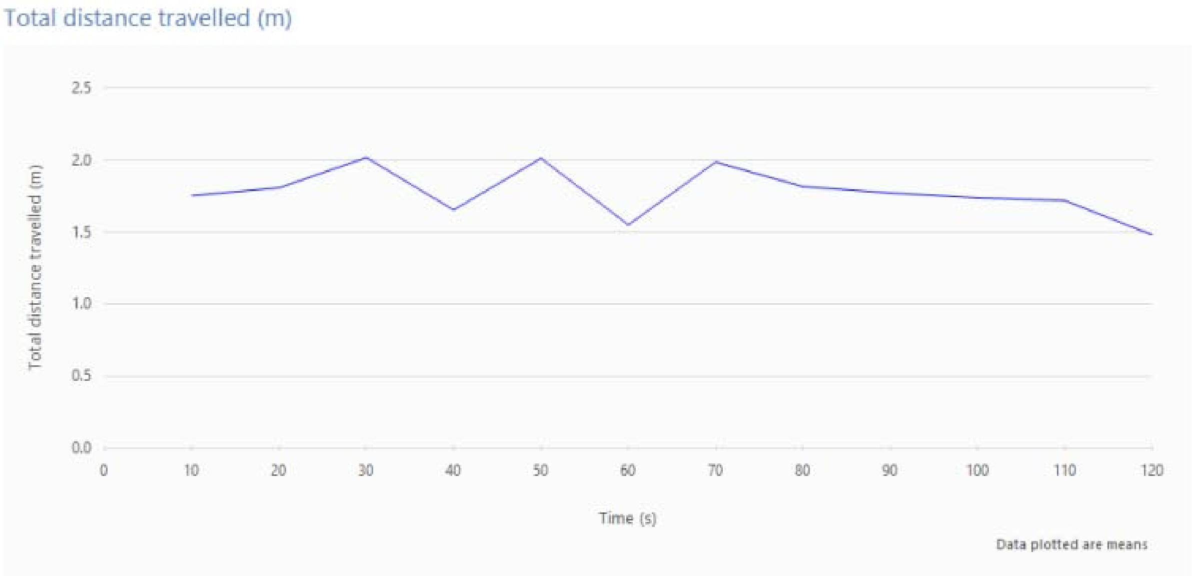


Figure 1. Example of the distance travelled plotted across time using a 10 second time segment duration.

You can also use this measure to show results for tests segments on the Data page - in this case, you simply need to include the measure as one of the columns on the report.

Finally, you can also use the *Segment of test* measure as a *Limit* both in a Results report and on the Data page, such that only a certain segment (or segments) are included in the report. For example, you might want to perform some analysis of the first minute of a 10 minute test, so you would simply limit the report to only include the segments '0-30 seconds' and '30-60 seconds' (assuming your segment duration was 30 seconds).

Creating time periods for analysis across time

Using *time segments,* as described above, is a very easy way to analyse results across time, but it has one problem - the test is always divided into consecutive, equal duration segments. While this is suitable for lots of experiments, there are times when you need to have more control and in those cases, you can create *Time periods.*

As the name implies, a *Time period* is just a period of your test, for example '30-90 seconds', but the difference to a *time segment* is that this period can cover **any** arbitrary part of your test. For example, imagine you have a test in which you want to compare the results for the periods 30-90 seconds and 200-260 seconds - using *time segments* this would not be possible, but using *Time periods* you would simply create one period for 30-90 seconds, and a second one for 200-260 seconds and then perform your analysis using the *Time period* measure as an independent variable (this measure is analogous to the *Segment of test* measure described in the previous section).

For further details about setting up time periods, refer to these topics:

- An introduction to Time periods
- Setting up a Time period
- Editing a Time period
- Deleting a Time period

What next?

After specifying how you want to perform analysis across time, you should review the Analysis options to see whether the default settings are suitable for your protocol.

An introduction to Time periods

Introduction

ANY-maze includes two ways to analyse results across time - *Time segments* which divide tests across time into equal length 'chunks', and *Time periods* which can be used to divide a test in any arbitrary way.

Time segments are described in the main Analysis across time topic, whereas Time periods are described here.

Using time periods to compare two parts of a test

Imagine you have a test in which you want to compare the results for the periods 30-90 seconds and 200-260 seconds. Using *Time periods,* this is very easy to do. First, you would create a new Time period and specify that it starts at 30 seconds and ends at 90 seconds; then, you would create a second time period starting at 200s and ending at 260s. You would then simply analyse your results using *Time period* as your independent variable.

There is no limit to the number of periods you can create in this way, nor are there any restrictions on the period that each one can cover - which means that two time periods can include some or all of the same part of the test. This can be useful if, for example, you want to look at the results of the first minute of your tests and also for the first 5 minutes - clearly the '5 minute' period will include all of the 'first minute' period.

Periods can be different in different stages

Although a time period will usually cover a certain part of a test, for example 60-90 seconds, there may be situations when the period will differ depending on the stage.

For example, in a fear conditioning experiment, you might present an audible stimulus to the animal at time 60 seconds until time 90 seconds - so you could create a time period called 'Stimulus' and set it to start at 60s and end at 90s. But in the second stage of your experiment, the stimulus is presented at time 280s and lasts until time 310s. Clearly, it would be desirable to have a single 'Stimulus' period, so you could easily compare results for the two stages, and this is where having the ability to define a period as differing between stages is useful.

In this example, you would simply specify that the period is different in different stages, and then enter the start and end time for each stage. The period itself would still be a single item, so you could look at results for the 'Stimulus' period and ANY-maze would automatically cope with the fact that for some tests this would mean the results for the period 60-90s, and for others the period 280-310s.

Periods that depend on what the animal does

In fact, time periods can be even more flexible, in that you can specify that the time a period covers should depend on *something the animal does.* For example, you might want to compare the behaviour of your different treatment groups during the first 10 seconds following entry into a specific zone. But of course, different animals will enter the zone at different times, so how could you define when the period should start and when it should end?

In this case, you would specify that the time the period covers depends on a *Time marker*, which can be set by a procedure. Full details can be found in the Time markers topic. In this example, the procedure would wait until the animal enters the zone, and then set a time marker. You can then specify that the *Time period* should start 0s after the time marker, and end 10s after it. You would then be able to look at the results for the time period, and ANY-maze would automatically cope with the fact that the actual time it covers would be different in every test.

See also:

- Setting up a Time period
- Editing a Time period
- Deleting a Time period

Setting up a Time period

Introduction

For a general introduction to time periods, see An introduction to Time periods.

To add a time period to a protocol, click the *Add item* button in the ribbon bar and select *New time period* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the time period's name
2. For a 'fixed' period, specify the start and end of the period
3. For a period that differs between stages, specify the start and end time in each stage
4. For a period that depends on a time marker defined by a procedure, specify the how long before (or after) the time marker the period should start and end

If your protocol was created in a version of ANY-maze prior to V5.1, and uses Events and actions *rather than procedures, you can specify that the time period is based on an* event *rather than a* time marker.

See also:

- An introduction to Time periods
- Editing a Time period
- Deleting a Time period

Time period name

In brief

Enter a name for the time period in the *Time period name* field on the time period's settings page. You must make an entry, but it can be anything you like.

Details

Time period names should either specify the actual period, for example '20-75 seconds', or should specify what is happening during the period, for example 'Stimulus' or '10 seconds after zone entry'.

Limits

You can enter anything you like up to a maximum of 32 characters.

Specifying the start and end of a 'fixed' time period

In brief

For a 'fixed' time period, i.e. one which always covers the same period in all tests, you should simply check the button labelled *This period is the same in all stages* and then enter the start and end times in the field provided.

Details

You can enter the time using units of s, min, h and d for seconds, minutes, hours and days respectively. You can also mix units and use decimals, so for example, '90s', '1min 30s' and '1.5min' all specify the same time. Note however that the resolution is 1s, so even though you can enter a time such as 23.5s, this will simply be considered to be 23s.

Limits

You can enter any times from 0 to 7 days, although the start time must be before the end time.

As mentioned above, the resolution of the time is 1s.

Specifying the start and end of a time period that differs between stages

In brief

For a time period that differs between stages, you should first select the button labelled *This time period is different in different stages* and then enter the start and end times in the table provided.

Details

When you select *This time period is different in different stages*, ANY-maze will automatically fill the table with a list of all the stages in the protocol. You should then enter the start and end times of the period in each stage.

The times can be entered using units of s, min, h and d for seconds, minutes, hours and days respectively. You can also mix units and use decimals, so for example, '90s', '1min 30s' and '1.5min' all specify the same time. Note however, that the resolution is 1s, so even though you can enter a time such as 23.5s this will simply be considered to be 23s.

You don't have to enter times for all the stages if you don't want to. Tests in a stage with no times will simply have undefined results for the time period.

Limits

You can enter any times from 0 to 7 days, although the start time must always be before the end time.

As mentioned above, the resolution of the time is 1s.

Specifying the start and end of a time period that is based on a time marker

In brief

For a time period that is based on a time marker, you should first select the button labelled *This time period is based on the following time marker* and then select the time marker in the drop-down list. You should then specify how long before, or after, the time marker the period starts and ends.

Details

When you select *This time period is based on the following time marker,* ANY-maze will automatically fill the drop-down list with a list of all the time markers defined in the protocol. Simply select the one you want to use - you must make a selection.

If you haven't yet set up any time markers to be used by procedures, then nothing will be listed in the drop-down list and you won't be able to set up a time period based on a time marker.

To define the start of the period, you should enter a time and specify whether the period starts this duration before, or after, the event. You should also specify the end time in a similar way.

The times can be entered using units of s, min, h and d for seconds, minutes, hours and days respectively. You can also mix units and use decimals, so for example, '90s', '1min 30s' and '1.5min' all specify the same time. Note however that the resolution is 1s, so even though you can enter a time such as 23.5s, this will simply be considered to be 23s.

Limits

You can enter any time from 0 to 7 days. The start time must always be before the end time, but they can both be before or after the event, if you wish.

As mentioned above, the resolution of the time is 1s.

Setting up a time marker

For more details on how to set up a time marker using a procedure, see Time markers in the Procedures section of this Help.

Editing a time period

Introduction

You can edit everything related to a time period at any time, whether before, during or after an experiment has been performed - there are no restrictions.

When you do edit a time period, ANY-maze will automatically re-analyse all your tests and the results will then be available - this usually only takes a few moments.

See also:

- Editing the elements of a protocol
- Setting up a time period
- Deleting a time period

Deleting a time period

Introduction

You can delete a time period at any time, whether before, during or after an experiment has been performed.

To delete a time period, simply select it in the protocol list and click the *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting time periods.

See also:

- Deleting protocol elements
- Setting up a time period
- Editing a time period

Analysis options

Introduction

The analysis options are used to control how ANY-maze actually analyses experiment results:

- Setting the level of significance for statistical tests
- Choosing how to manage latency of events which don't occur
- Choosing how to manage division by zero
- Calculation of instantaneous speed
- Removal of 'jumps' in the animal's track
- Changing the maximum value in heat maps
- Setting the options for analysis of heading errors
- Setting the options for analysis of movement towards and away from zones and points
- Specifying how ANY-maze should process overlapping keystrokes when adding additional scoring to tests
- Filtering the protocol elements which are analysed
- Specifying the results that ANY-maze should report when performing experiments in the OPAD cage
- Extending ANY-maze analysis using plug-ins

Setting the level of significance for statistical tests

ANY-maze includes the ability to perform a number of statistical tests (ANOVA etc.) which you can use to assess the results of your experiments. The level of significance (alpha) in these tests is an *analysis option*; it's set to $p \leq 0.05$ in all new protocols but you can change it to any value between 0.01 and 0.50.

Choosing how to manage latency of events which don't occur

ANY-maze can measure the latency of a number of events, such as 'Latency to the first entry to a zone', 'Latency to a key press', and 'Latency to the start of a sequence'.

Of course, there always exists the possibility that a particular event doesn't actually occur in a test - the animal doesn't enter the zone, the key isn't pressed, or the animal never performs the sequence. In these cases, the latency value will usually be reported by ANY-maze as undefined.

Considering a latency to be undefined is strictly correct (as we have no data for it), but it means that the Ns for the latency measure will be reduced, as ANY-maze considers an *undefined* result as a result which simply doesn't exist.

For example, imagine you tested 3 groups of 10 animals in the plusmaze. During the tests, all the animals entered the open arms at some point, except for two animals in the first group which spent the entire test in the closed arms or central area. If you then analysed the 'Latency to first entry to the open arms', you would find that the N for *this measure* for Group 1 would be 8, rather than 10, because for two of the animals, ANY-maze wouldn't have a result at all.

To prevent this reduction in groups' Ns, some researchers choose to consider the latency for events which don't occur to be the test's duration rather than being *undefined*. If you want to, you can set ANY-maze to work in this way by checking the box labelled *Use the test duration as the latency for events which don't occur*.

Choosing how to manage division by zero

Some of the measures that ANY-maze reports are averages, for example, 'Average duration of visits to a zone' or 'Average time to complete a sequence'. Clearly, these values will be calculated by dividing one value by another, for example the total time in a zone by the number of visits to the zone. This however, can cause problems if the denominator is zero - for example, the animal never entered the zone or never completed the sequence.

In these cases, ANY-maze will normally consider the result for the average to be undefined, but as an undefined result is treated as a result which simply doesn't exist, this will cause the group Ns for the relevant measure to be reduced.

For example, imagine you tested 3 groups of 10 animals in the plusmaze. During the tests, all the animal entered the open arms at some point, except for two animals in the first group which spent the entire test in the closed arms or central area. If you then analysed the 'Average duration of visits to the open arms', you would find that the N for *this measure* for Group 1 would be 8, rather than 10, because for two of the animals, ANY-maze wouldn't have a result at all.

To avoid this reduction in the groups' Ns, you can tell ANY-maze to consider the result of an average for which the denominator is zero to be *zero* rather than *undefined*. To do this, simply check the box labelled *Use zero as the result for undefined averages*.

A similar situation exists with the *path efficiency* measure, when measured as the efficiency to the first entry to the zone. This is calculated as the straight line distance between the first and last position, divided by the total distance travelled between those positions. Obviously if the animal has never entered the zone, the path efficiency can't be calculated - again, the N would be reduced by the number of animals who never entered the zone. You can avoid this by telling ANY-maze to use zero as the path efficiency (thus ensuring that N matches the number of animals) by checking the box labelled *Use zero as the result for undefined path efficiency to zones*.

Calculation of instantaneous speed

This option is only included to allow compatibility with results from experiments performed in versions of ANY-maze prior to V4.60. It should **not** be turned on for new experiments.

Prior to V4.60, the instantaneous speed reported by ANY-maze was the *maximum speed,* and this was a little exaggerated. A new method of calculation was introduced in V4.60, and we recommend that you use this new method unless you need to compare instantaneous speed results with experiments created in ANY-maze versions prior to V4.60.

Note that the new method is *always* used when making 'live' measurements for charts, so if this option is turned off, you may notice a difference between the values used on the chart when drawn in 'real-time' during the test, and when viewed post-test.

Removal of 'jumps' in the animal's track

This option is described in detail here.

What next?

After setting the general analysis options, you should consider whether you want to include any charts in the protocol.

Removal of 'jumps' in the animal's track

The option to ignore jumps can be found in the Protocol, on the settings page for Analysis options

Introduction

When tracking in a water-maze experiment, the movement of the water will often result in reflections which will cause movement in the video image. Sometimes, ANY-maze will incorrectly track these reflections rather than the animal. This will result in an obvious 'jump' in the animal's position, where the tracking position jumps to the reflection and then back to the correct position of the animal. This is most obvious on the test's Track plot report, but can also be seen on the Test data report; it will obviously also affect measures which involve a calculation of distance, speed, etc.

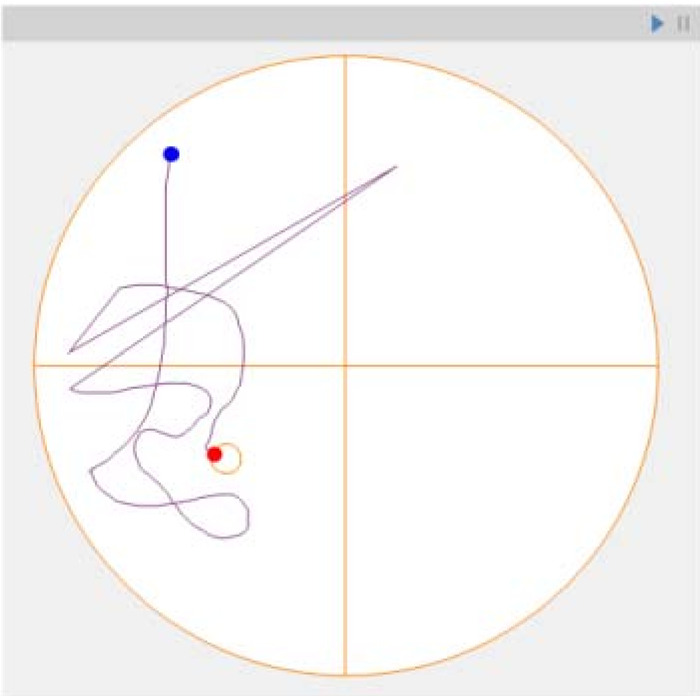


Figure 1. A test's track plot report, showing the obvious jump due to a reflection in the water-maze.

ANY-maze can detect these 'jumps' during analysis, and ignore them in the test's results.

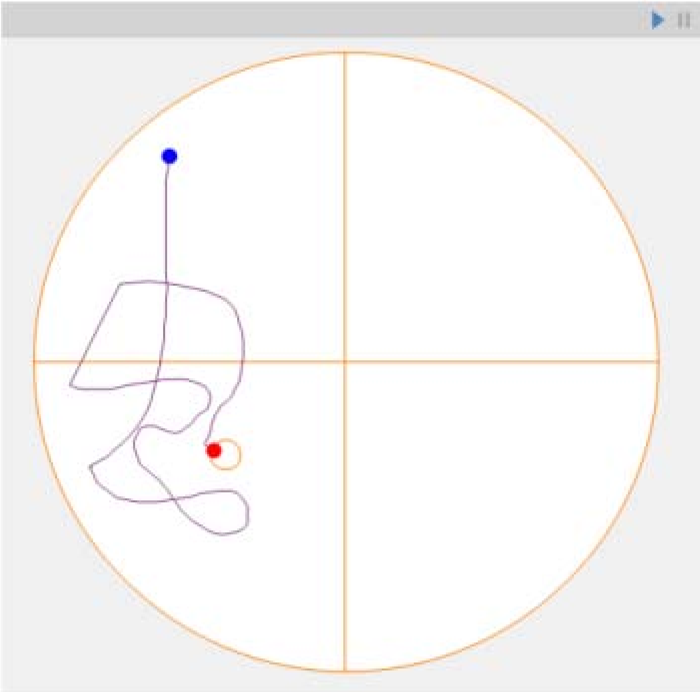


Figure 2. The track plot shown in figure 1, after the jump has been removed.

Note that the way ANY-maze determines whether a point is a 'jump' or not is based on the speed at which the tracking points jump from one to another, and uses the assumption that in water, the animal can't accelerate very fast. So it may be that more than one point is actually removed from the results - firstly, the invalid point due to the reflection; and secondly, the point after that. This is because in both cases, the point shows a speed of movement and acceleration that is considered to be too fast for the water-maze.

*This feature is **only** recommended for use with a water-maze experiment; if turned on, it will increase the time taken for ANY-maze to analyse the test's results.*

Changing the maximum value in heat maps

Introduction

Heat maps use a range of colours to provide a visual representation of how much time the animal spends in different parts of the apparatus during a test, with blue as the shortest time and red as the longest. But what is the 'longest' time? The obvious answer would be 'the longest time the animal spent in a specific area during the test which the heat map shows'. But the problem with that is that you wouldn't be able to compare one heat map with another, as the red area would represent a different time in every test.

- Comparing heat maps for tests within an experiment
- Comparing heat maps across experiments

Comparing heat maps for tests in an experiment

In order to allow you to compare the results of different tests within an experiment, ANY-maze calculates the maximum time spent in any area for *all* tests performed so far in the experiment.

Note that this means that the maximum value can change as tests are performed. For example, consider a test 1 minute long in an open field, where the animal spends about a quarter of the time in a given corner (and the rest of the time moving about elsewhere). The heat map will show the animal's occupancy of this corner in red, where red represents about 15 seconds.

However, on the next test, the animal goes into a different corner and spends most of its time there - say 40 seconds. So the maximum scale for heat maps is now 40s, and this is represented by red on a heat map. This means that the maximum value for the heat map scale for the entire experiment is now 40 seconds, rather than 15. If you were to view the *previous* test's heat map now, the red area would have changed colour (as 15 seconds is no longer the maximum, and is therefore no longer red).

This is the way ANY-maze works by default, and is set by checking *The maximum found in all tests in this experiment*.

Note that the maximum value will be that calculated across *all* trials for *all* animals in the experiment - so on the Results page, if you have applied a filter to only include a subset of the tests, you might not see any tests that actually show a red area for this maximum value. If you have turned on head tracking, it's also the maximum from all heat maps of the centre point *and* the head, to allow comparison of heat maps of the animal's centre with those of its head.

When looking at the *Mean heat map of the group's centre point* on the Results page, the maximum value given on the scale is the maximum of any of the groups' mean values.

Comparing heat maps across experiments

It's all very well to have the heat map maximum 'red' value calculated across all tests within an experiment, but what happens if you need to compare the results with another experiment, in another file?

In this case, you will need to set a specific value for the maximum, and make sure it is the *same value* in both of the experiment files. To work-out what value to use, you'll need to look at the automatically-calculated maximum value in both experiments and note the *highest* value.

Next, on the Protocol page, navigate to *Analysis options* > *Heat maps* and select the option *Set the heat map maxiumum to be: 'The following value'*. Finally, enter the value you noted earlier. This value is a time, and you can enter it using units of s, min and h for seconds, minutes and hours respectively. You can also use decimals, so for example '1.5min', '1min 30s' and '90s' are all the same duration. You can enter any time between 1s and 1d (24h).

You can alter this value at any time, including after an experiment is complete, and ANY-maze will simply re-analyse your results automatically, which usually only takes a few moments.

More details

The scale for a heat map is displayed above the heat map itself, and the minimum and maximum values are labelled. You'll notice that the maximum is prefixed with the '~' symbol; this means that the value is approximate.

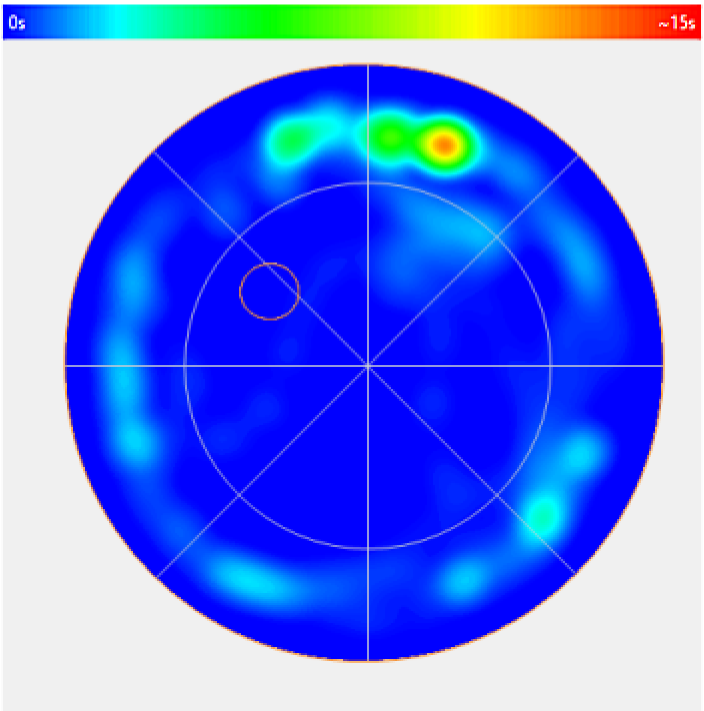


Figure 1. A heat map, showing the scale above it.

The reason that the maximum scale value is only approximate is a result of the way the maximum value is calculated. The heat map is built up by looking at each of the individual tracked positions of the animal, and 'adding to' the colour for each position. In reality, however, the animal isn't at an exact point so the heat map calculation actually averages a number of points around the 'hottest' position. This averaging means we get a better representative picture of the animal's occupancy over the course of the test, but means that the calculation of the maximum value is based on an average value of a number of points, rather than an individual position - which would be exactly accurate, but would not make a very helpful heat map.

What next?

After setting the maximum heat map value, you should review the Options for analysis of heading errors to see whether the default settings are suitable for your protocol.

Options for analysis of heading errors

Introduction

ANY-maze includes a number of measures that relate to the animal's *Heading error* (sometimes also called 'angular deviation'). These measures indicate how far the animal deviated from a direct course to a zone or a point.

Specifically, ANY-maze can calculate the *Initial heading error* to a zone or point and also the *Average heading error*. The options described here are used to define exactly how these calculations are performed.

- Specifying how to determine the animal's initial heading
- Specifying how to analyse headings to zones

Specifying how to determine the animal's initial heading

To calculate an initial heading *error*, it's clearly necessary to define what the animal's initial heading actually is.

ANY-maze provides two methods for doing this. You can either specify that the initial heading is the vector from the animal's first position in the test to its position after a short delay has elapsed, or that it's the vector from the animal's first position in the test to the first position outside a circle of a certain radius centred on the first position in the test.

When using a delay, you should specify the duration you wish to use either in seconds or in milliseconds by using the units *s* and *ms* respectively. You can enter any value from 1ms to 10s.

When using a distance, you should specify the radius of the circle in millimetres - this can be any value from 1 to 9999.

Specifying how to analyse headings to zones

ANY-maze provides two methods to specify a heading to a zone - either the heading to the centre of the zone, or the heading to any point on the zone perimeter.

In fact, using the second of these options is the most *correct* method, but it is computationally intensive and may cause results analysis to slow down significantly. In this case, you may wish to use the heading to the zone's centre as an approximation of the heading to the zone (particularly for small zones where the difference is negligible).

Clearly, if you do use the centre of the zone to determine the heading to the zone, then ANY-maze will have to identify where the centre actually is. It does this using the zone's *centre of mass* (i.e. the mean x, y coordinate of all the points within the zone), but it's important to understand that this point may not actually be within the zone. For example, consider a ring shaped zone - the centre of the zone will be at the centre of the ring, but this point won't lie within the zone.

What next?

After setting the heading error analysis options, you should review the Options for analysis of movement towards and away from zones and points to see whether the default settings are suitable for your protocol.

See also:

- Zone measures
- Point measures

Options for analysis of movement towards and away from zones and points

Introduction

ANY-maze includes the ability to determine for how much of a test (or arbitrary time period) the animal is moving towards or away from a zone or point.

- Defining movement towards or away from something
- Setting the critical angle for movement towards or away from zones and points
- Specifying how to analyse movement towards and away from zones

Defining movement towards or away from something

Analysis of movement towards or away from something isn't quite as simple as it might at first appear. For example, if an animal moves in a direct line from position A to a point at position B, then clearly it would be considered to be moving towards the point; but if its course would miss the point by 2mm, should it still be considered to be moving towards the point?

To resolve this issue, ANY-maze uses a 'critical angle'. A good way to visualise this angle is to imagine that the animal has a flashlight and is pointing it in the direction that it's moving in. The flashlight will illuminate a conical area in front of the animal, and if a zone or point is 'illuminated' then ANY-maze will consider that the animal is moving towards it. In this analogy, the 'critical angle' defines the angle of the flashlight's cone - see figure 1.

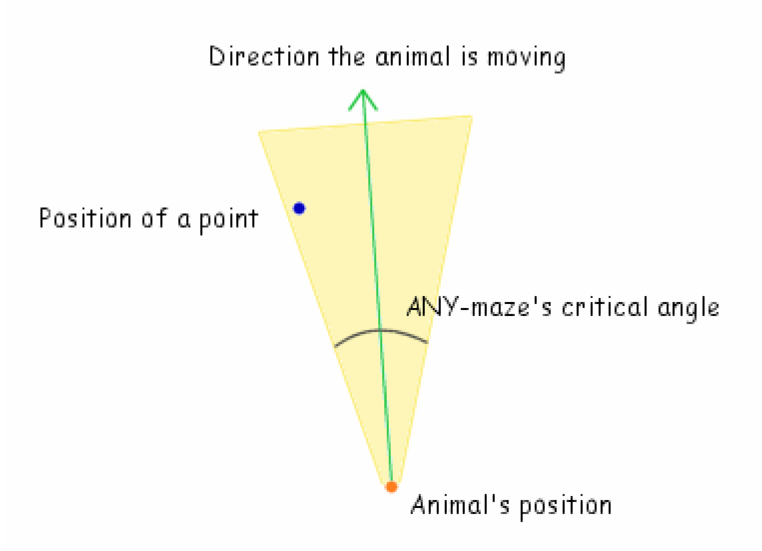


Figure 1. The 'critical angle' can be thought of as describing a cone projected in front of the animal and centred on the animal's direction of movement (like the light from a flashlight). If any point or zone is within the cone, then ANY-maze considers that the animal is moving towards it.

Of course, this still leaves the question of how the system determines movement away from a point or zone. In this case, you can simply imagine that the animal is shining the flashlight behind itself (i.e. the cone projects in the opposite direction) - the animal is then moving away from any point or zone 'illuminated' by this reverse-pointing flashlight.

Setting the critical angle for movement towards or away from zones and points

By default, ANY-maze uses a critical angle of 90°; however, you can alter this to any value from 1 to 180° simply by editing the value shown. Note that you must enter something.

Specifying how to analyse movement towards and away from zones

In the introduction to this topic, I used an example of the animal moving towards a point, but what about movement towards (or away from) a zone? In this case, there are two ways that ANY-maze can operate - it can either consider just the centre of the zone (much like a point) or it can consider the *entire area* of the zone.

As you'd expect, using the entire zone area is usually the best solution. To revert to our flashlight analogy, you can imagine that in this case, ANY-maze will consider that the animal is moving towards a zone if *any* part of the zone is 'illuminated' by the flashlight - see figure 2. Whereas if you use just the centre of the zone, then the animal will only be considered to be moving towards the zone if the zone's centre is 'illuminated'.

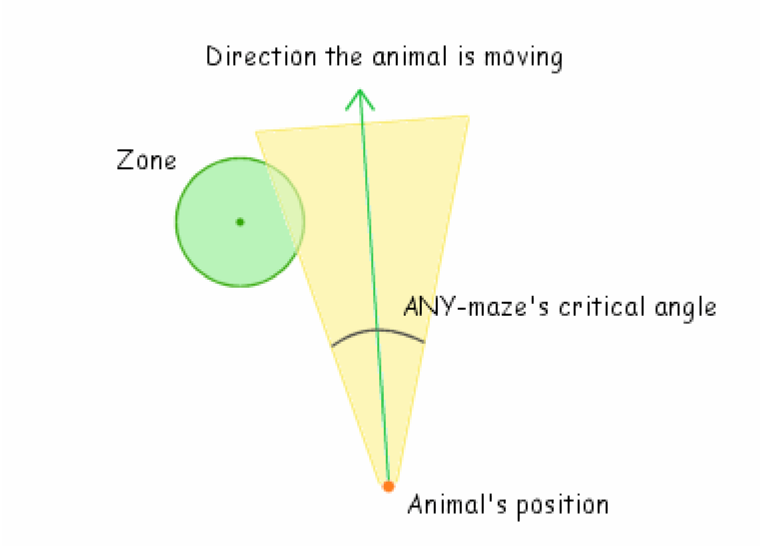


Figure 2. In this example, if the entire zone area was being used to define movement towards a zone, then ANY-maze would consider that the animal is moving towards the zone. However, this would not be the case if just the centre of the zone was being used.

So why would you ever want to use the centre point? The answer is that calculating movement using the entire zone is very computationally intensive, and using this method in long experiments with large zones can cause the analysis to take a substantial period of time - in this case using the centre of the zone offers a quicker alternative.

Of course, using the centre of the zone raises a further question - where *is* the centre of the zone? For a simple circular zone this is obvious, but zones in ANY-maze can be any shape. This issue is resolved by using the zone's 'centre of mass' as it's centre point - this is the mean x, y coordinate of every point within the zone. It's important to understand that this means the centre of the zone may not, itself, be in the zone at all. For example, consider a ring shaped zone - in this case, the centre of the zone will be the centre of the ring, but this point won't lie within the zone.

What next?

After setting these options, you should consider whether you will need to add additional scoring to tests using keystrokes, and how ANY-maze should cope with overlapping keystrokes.

See also:

- Zone measures

- Point measures

ANY-maze help topic H0617

Options for how ANY-maze should process overlapping keystrokes when adding additional scoring to tests

Introduction

ANY-maze allows you to review a test any number of times, and on each review, add some more scoring to the test results using keys.

This feature is described fully in the Adding additional scoring to a performed test topic.

When adding additional scoring to test in this way, a potential problem can arise - the same key can have overlapping key presses in two different test reviews. This topic describes the problem, and the options you have to address it.

The problem

The 'overlapping keystrokes' problem is best described using a diagram - see figure 1, below:

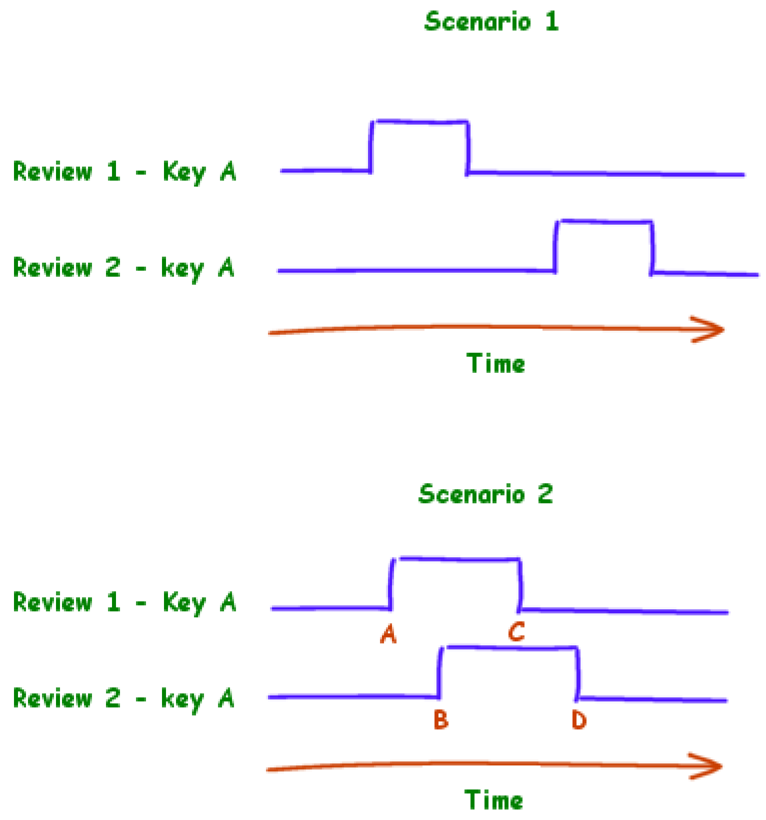


Figure 1. Two scenarios of multiple presses of the same key in two reviews of a test. In scenario 1, the key presses are entirely separate; but in scenario 2, they overlap - making their meaning obscure.

Looking at scenario 1, it's clear that key A was pressed twice, once in the first review and once in the second. ANY-maze will count these presses as two presses, and will show the total time pressed as the sum of the time that the two keys were on, etc. So in this case, everything is as you would expect it to be.

Now look at scenario 2. In this case, how many key presses occurred? It looks like two, but remember this key is scoring a behaviour, and the two reviews are considering the same video - so what's happening here? Did the animal perform whatever behaviour the key is being used to score once (starting at time A and ending at time D), or did it perform the behaviour twice (starting at time A, stopping presumably at some point before time B, starting again, presumably at some point before time C, and finally ending at time D)?

As there is no way for ANY-maze to know which of these options is correct, it offers you the choice of using either of them.

The solutions

The first solution to the problem is to *merge* overlapping keystrokes; thus a key which is ON and which then goes ON again is still just considered ON, although it then has to go OFF the same number of time before it will be considered OFF. So in this case, ANY-maze would consider scenario 2 to consist of ONE key press starting at time A and ending at time D.

The second solution to the problem is to *split* the overlapping keystrokes, so that each key press is seen as a press. In scenario 2, this would cause the system to count TWO key presses, with a notional OFF event just before the second ON event. As it's clearly illogical to say a key went on twice and off three times, the now redundant OFF event related to the first ON event is then eliminated.

Again, this is easier to understand in a diagram - see figure 2.

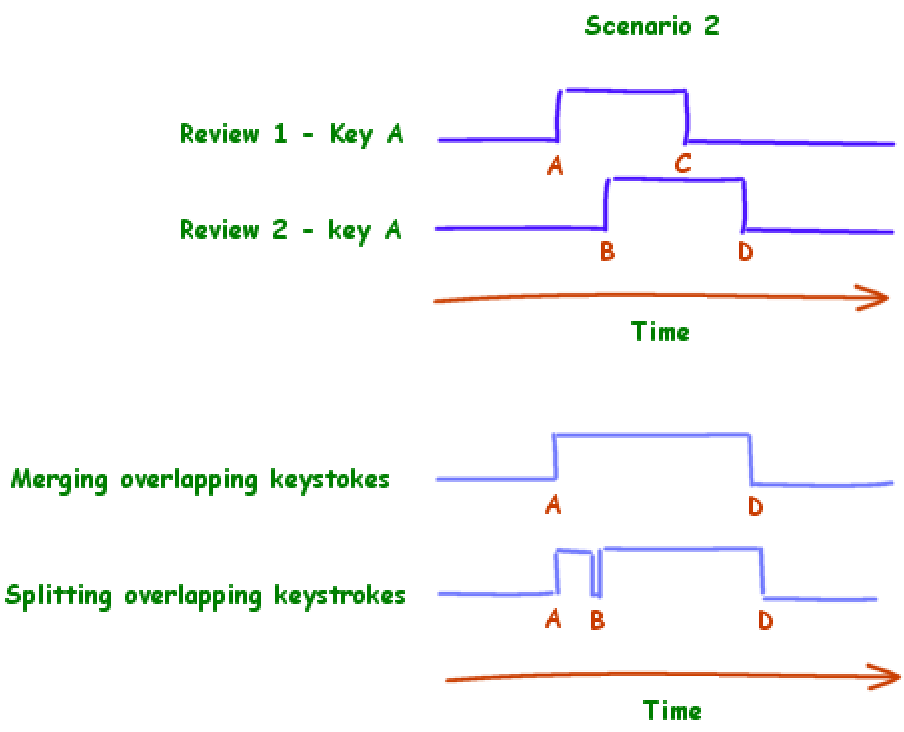


Figure 2. The ways ANY-maze can analyse Scenario 2 (from figure 1). In the first case, the key presses are merged to create a single press; while in the second case, they're split to create two ***separate*** *presses.*

So, in the *Analysis and results* section of the protocol, you can choose whether ANY-maze should 'Merge' or 'Split' overlapping keystrokes.

It's probably worth mentioning that this problem will generally never occur, because you will probably choose to score *different* behaviours on each review of a test - so in the first review you might be using key A, while in the second review you might be using key B. In this case, it doesn't matter at all if the keystrokes overlap, as they relate to different keys and therefore different behaviours, and clearly these *could* occur at the same time.

Filtering the protocol elements which are analysed

Introduction

ANY-maze can analyse a vast number of measures (there are over 300) and it does this whenever you load an experiment which includes any completed tests. The analysis of results goes on in the background while you work, which means that usually, when you want to view any results, the analysis has already been done.

Even though the analysis goes on in the background, it can take some time if you have a large experiment and/or a very complex protocol, and you may have to wait for result analysis to finish before you can view some results. The *Results analysis filter* page of the protocol allows you to limit the areas that you're interested in analysing, in order to speed up result calculation.

Limiting the number of results calculated

To see how limiting analysis can help speed things up, let's consider the measures that ANY-maze calculates for zones - there are over 60 of them and they're calculated for every zone. It may be that you've set up several zones, but you're not interested in analysing what the animal does in each one - you might have created them to be part of a sequence, for example. In this case, you can opt to remove these zones from result analysis, and ANY-maze won't calculate any results for them. This will reduce the number of measures to be calculated by more than 60 for each zone that you remove and consequently reduce the time that analysis takes.

To remove an item completely from result analysis, simply un-check it in the top-most list on the *Results analysis filter* page. This means that ANY-maze won't calculate any results for this item.

Limiting analysis across zones

Some measures, such as those for key presses, virtual switches and I/O items, are not only calculated for the apparatus as a whole, but also for *each zone* in the apparatus (unless you've removed the zone from results analysis as described above, of course).

If you're not interested in the results of some of these items within individual zones, you can un-check them in the bottom list. This will mean that the item will still have results calculated for the apparatus as a whole, but not for each individual zone as well - again, this will reduce the time it takes to analyse an experiment's results.

See also:

- Information measures

- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Specifying the results that ANY-maze should analyse in the OPAD cage

Introduction

When tests are performed in an OPAD cage, ANY-maze will report some OPAD-specific results; specifically, the *Temperature when contact is broken* and *Temperatures of interest*. The *Analysis of OPAD results* element in the protocol is used to define exactly how these results are generated.

- Temperature when contact is broken
- Temperatures of interest
- Editing the OPAD analysis options

Temperature when contact is broken

The *Temperature when contact is broken* essentially reports the temperature of the OPAD cage thermal elements when the animal broke contact - which usually occurs because the thermal stimulus had become too aversive. Of course, an animal might make and break contact multiple times - the result reported is the average for all the occasions when the animal broke contact.

Selecting the 'breaks' of interest

In OPAD tests, you will typically be ramping the temperature of the thermal elements up from around room temperature to some 'hot' value, or down to some 'cold' value. In the case where the temperature is ramping up, you will probably want to know when the temperature became aversively hot, causing the animal to break contact; and in the case where the temperature is ramping down, you would want to know when it became aversively cold.

It would seem that the 'Temperature when contact is broken' is exactly the value you need to report this 'aversive temperature', but it's important to understand that by default, this will report the average temperature of ALL breaks, not just those that occurred when the temperature was ramping towards an aversive condition. To address this, you can specify that the average should only include breaks which occur when the temperature is ramping up and is above some value (or is ramping down and is below some value).

For example, let's imagine that the temperature starts at 25°C, ramps up to 50°C, then drops back to 25°C before going back up to 50°C again, and that this repeats 10 times during the test. In this case, you will probably not be interested in breaks that occur below 30°C (when the temperature is not aversive); nor will you be interested in breaks that occur when the temperature is going back down from 50°C to 25°C. So, you could specify that the 'Temperature when contact is broken' should just include those breaks that occur when the temperature is ramping UP and is above 30°C.

Ignoring short contacts or short breaks

When the animal breaks contact (because the thermal elements have become aversively hot or cold), it may wait a moment and then make contact again - however, it will probably immediately break contact because the element will still be too hot or cold. Thus, you may see a long period of contact (while the device has not reached an aversive temperature), followed by a break, and then followed by a few short contacts. When analysing the 'temperature when contact is broken', you would probably say that the temperature when the first of these breaks occurs is the temperature you want to use; in other words, you would want to ignore the very short contacts that follow it. To achieve this, you simply need to specify that ANY-maze should *Only include a break if the animal has been in contact for at least* a certain period of time - for example, 5 seconds.

OPAD will detect all breaks in contact, even if the break only lasts a very short time. However, this may not be very appropriate; for example, the animal might be in contact for 30 seconds, then move its position very slightly, causing a brief break, and then resume contact for another long period - in this case, you probably wouldn't want this short 'break' to be used in calculating the *Temperature when contact is broken*. To address this, you can specify that breaks of less than a certain duration should be ignored - for example, you could ignore any break that lasts less than 1 second.

Temperatures of interest

As the name implies, a 'Temperature of interest' is a temperature range which interests you. For example, you may have determined that animals typically break contact with the thermal element when it is at around 40°C, but when you give them a certain drug, this increases to 50°. This being the case, you might decide to create two 'temperatures of interest' - one for the range 35-45°C, and another for the range 45-50°C.

For each temperature of interest that you define, ANY-maze will report:

- Time in contact
- Number of times contact broken
- Number of times contact made
- Number of licks

To specify a temperature of interest you simply need to enter a temperature range in °C, for example from '40' to '45'. A range must cover at least 1°C, so for example you can't specify a range from '40' to '40'. You can enter as many ranges as you like, and they can overlap, so specifying one range from 40-50 and another from 45-50 is OK.

To remove a range, just delete the From and To values.

Editing the analysis options

The options for both the *Temperature when contact is broken* and *Temperatures of interest* can be edited at any time, including after tests have been performed. Any changes you make will immediately cause the experiment's data to be re-analysed, and the results will then update automatically.

Extending ANY-maze using plug-ins

Plug-ins are only available in licensed copies of ANY-maze.

Introduction

ANY-maze *plug-ins* are small pieces of software that can be used to extend the abilities of ANY-maze. For example, a plug-in might read data from a heart rate monitor and feed this data into ANY-maze, thus you would be able to see information about the animal's heart rate during the test. But, better still, this information would become *part of ANY-maze*, meaning that you'd be able to see how the heart rate differed when the animal was in different zones and/or how it differed across the duration of the test - and you'd be able to use ANY-maze's statistical reports to analysis these results.

Another use for a plug-in might be to simply add some analysis that ANY-maze doesn't provide. For example, you might be interested in determining abrupt changes in the animal's speed - such as would be seen in a 'darting' type of behaviour. ANY-maze doesn't (at least, currently) perform any analysis that would quantify this behaviour, but a plug-in could be written which would detect such 'darts'. Again, you could then use the existing facilities of ANY-maze to analyse 'darts' in different areas of the apparatus, to see how 'darts' vary during the duration of the test, etc.

An important thing to understand about plug-ins is that they're designed to be written by third-parties. Thus, the manufacturer of the heart-rate monitor (in the above example) might write a plug-in to provide the heart rate data to ANY-maze; or a student might write a plug-in to detect the 'darting' behaviour.

Clearly then, if someone wishes to write a plug-in, they'll need detailed information about how to do this and this documentation is provided in the ANY-maze plug-ins topic. But, in brief, a plug-in is a Windows DLL (dynamic link library), which will usually be written in C/C++. The DLL needs to export just one routine through which ANY-maze and the plug-in will communicate. The code required to actually make the DLL into a plug-in is very simple and even quite an inexperienced programmer shouldn't have any difficultly writing it. Of course, the code that performs the actual *function* of the plug-in (monitoring heart-rate, detecting darts, or whatever) may be far more complex!

Details

As mentioned above, a plug-in is just a 'DLL' file. What this means isn't very important, but what is important is that to install a plug-in you simply need to copy the 'DLL' file into the same folder as the ANY-maze program file (`ANY-maze.exe`) By default, this is `C:\Program Files (x86)\ANY-maze` on a 64-bit operating system, or `C:\Program Files\ANY-maze` on a 32-bit operating system.

When ANY-maze starts up, it automatically checks all DLLs in this folder to see whether any of them are plug-ins and if any of them are, it adds a *Plug-ins* entry to the Analysis and results element of the protocol - see figure 1.

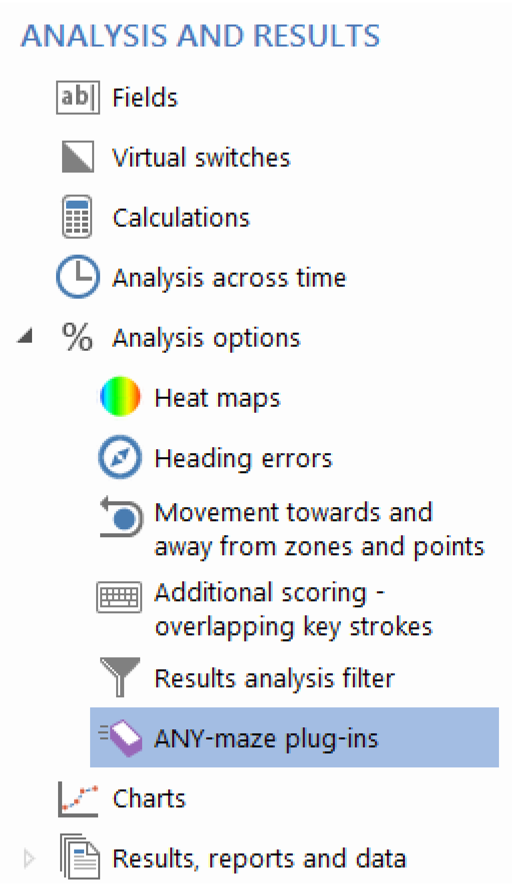


Figure 1. When ANY-maze detects a plug-in, it automatically includes the Plug-ins *item in the* Analysis and results *element of the protocol.*

Selecting the plug-ins item will show a list of the plug-ins that were detected, together with links to view the plug-ins' help and to alter the plug-ins' settings - see figure 2.

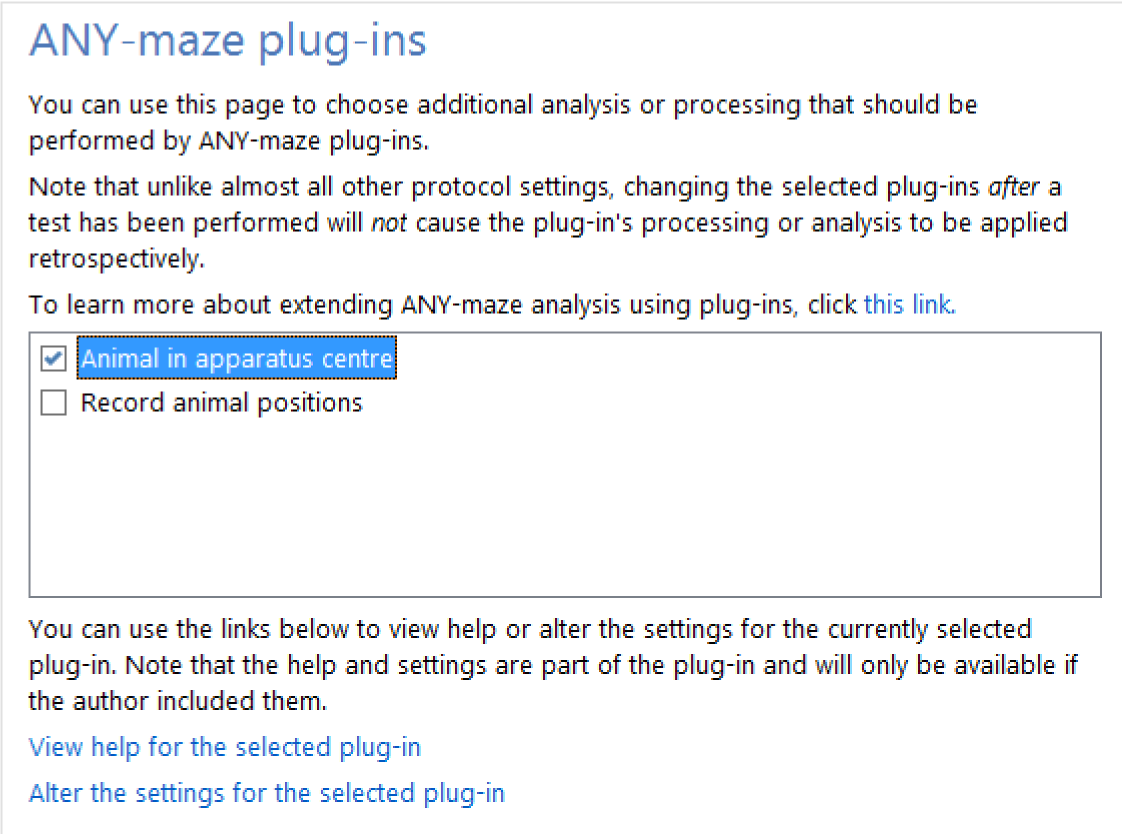


Figure 2. The plug-ins detected by ANY-maze are listed, and you can simply choose which of them you would like to use in your protocol. The links below the list access the Help and Settings for the selected plug-in.

Viewing a plug-in's help

A plug-in author can optionally include help with the plug-in, which you can view by selecting the plug-in in the list and then clicking the appropriate link. The help message should describe what the plug-in is designed to do, what the results it generates mean (assuming it does generate some results), and what settings, if any, it includes.

Altering a plug-in's settings

A plug-in can optionally include a window where the user can alter the plug-in's settings. For example, the demo plug-in provided with ANY-maze determines when the animal is in the centre of the apparatus. In this case, the plug-in includes one setting, which is the maximum distance the animal can be from the centre for it to be considered to be 'in the centre'.

In the heart-rate monitor example given above, the settings might be used to set which channel of a heart rate monitor the plug-in should capture data from.

Activating a plug-in

By default a plug-in, even if installed on your computer, won't be used by a protocol. If you want to use the plug-in in a protocol, you simply need to check the box next to the plug-in's name - see figure 2, above.

It's very important to understand that a plug-in analyses data during a test. This means that you can't activate a plug-in after a test has been performed and expect to see the result of the plug-in's analysis in your results. This is unlike almost every other protocol element.

See also:

- ANY-maze plug-ins
- Writing a plug-in
- Plug-in API reference

Charts

Charts are used to show a plot of some value across time. A wide range of values can be plotted including such things as the animal's speed, its distance from a zone, the value being read from some I/O device, etc.

Charts can be viewed while a test is running, thus providing live feedback of the value being plotted, and can be viewed, copied, saved, or printed post-test.

For more information about charts refer to the topics below.

- An introduction to charts
- Setting up a chart
- Editing a chart
- Deleting a chart
- Printing, copying and saving charts
- Setting up chart state channels

An introduction to charts

Introduction

Charts show a plot of some value across time. For example, the chart in figure 1, below, shows a plot of the animal's speed in a water-maze test.

Figure 1. An example of a chart - in this case, the chart is plotting the speed of the animal in a water-maze test.

Charts can be displayed both post-test and also live (i.e. during a test), which provides a great way to get visual feedback of the magnitude of some measure that interests you.

Setting up a chart is extremely simple - you just add it to your protocol, give it a name, and select the value you want it to show. You can, of course, choose to specify things like the x-axis width, the trace colour, etc. but these will be set to default values, so you only need to set them if you want to.

Live charts

Plotting a chart during a test is a great way to get visual feedback of some value, as can be seen in the example in figure 1. But that chart shows just 30 seconds of data; what about a chart of a test that lasts for hours, or even days?

When you create a chart, you can specify the width of the x-axis - this defaults to the duration of the test, but you can select any period you want, for example 1 minute. During the test, when the trace reaches the right-hand edge, the chart begins to scroll automatically - thus you can always see the *most recent* 1 minute (or whatever width the x-axis has). In fact, you can adjust the x-axis width dynamically (so you could increase the width from 1 minute to 5 minutes, or even 12 hours, if you wanted to) and you can drag the chart so you can see the data that has scrolled off the left-hand side.

These actions can be performed using the mouse, in a simple and intuitive way.

Multiple charts

You're not limited to charting a single value; you can in fact chart any number of different values simultaneously. The charts are simply stacked, but they all share the same x-axis and they all scroll together - see figure 2 for an example.

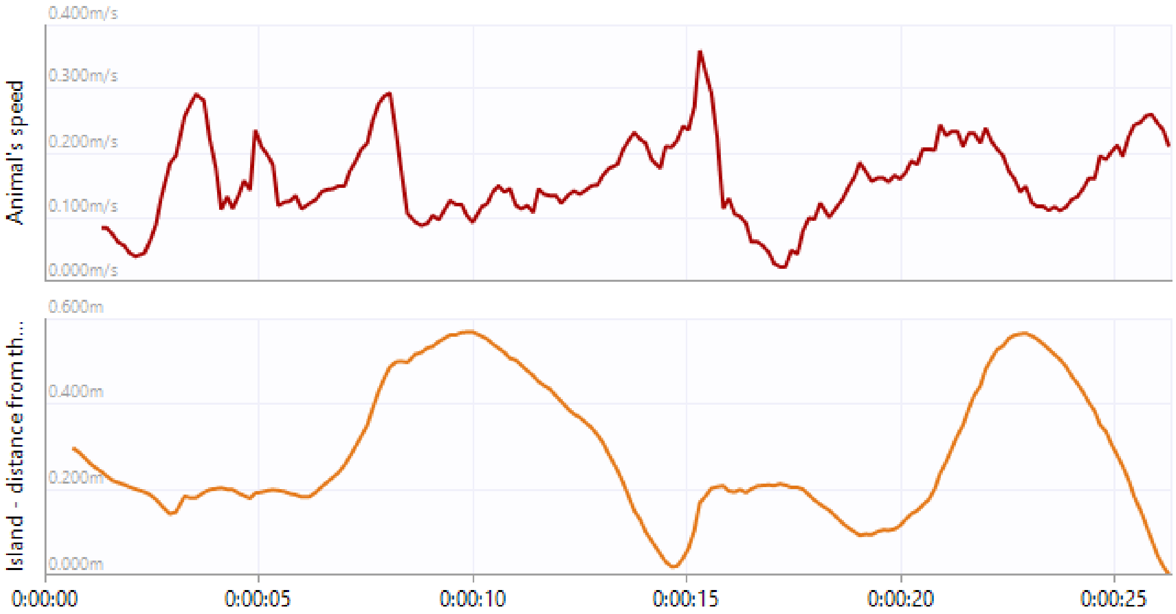


Figure 2. You can plot multiple values - each one is shown in a different chart, although all the charts share the same x-axis.

Because the x-axis of all the charts is the same, it makes it very easy to see dependencies in the values being plotted.

State channels

In addition to plotting a value, a chart also be set up to shade its background to show certain 'states'. For example, in a chart showing the animal's speed, you could add a *state channel* to indicate whether or not the animal is in a particular zone. This way, you'll have a visual indication as to whether there is any correlation between the animal's speed and its presence in the zone. For more details on this, see Setting up chart state channels.

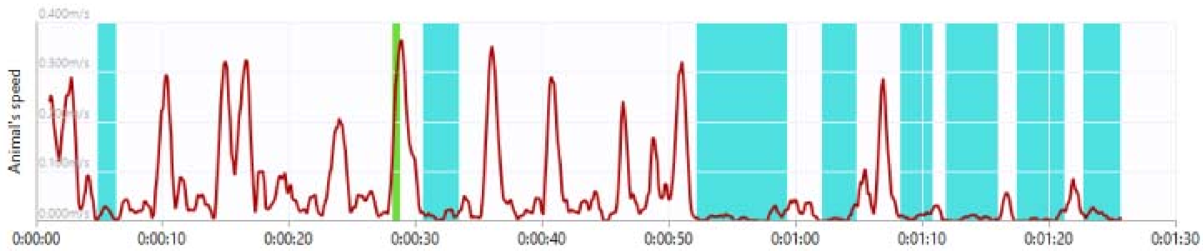


Figure 3. A chart with two state channels: a green background indicates when the animal was in the 'Centre' zone, and a blue background indicates when it was immobile.

Viewing charts post-test

After tests are complete, you can view their charts by going to the Test details report. Selecting the *Related reports* button in the ribbon bar will allow you to select *Test charts* from a list.

You can include different charts post-test to those that are shown *during* the test - for example, you might choose to have just one important value plotted during tests, but post-test, you might like to be able to view the plots of five different values.

As with almost everything in ANY-maze, you can alter the charts after tests have been performed - so you can chart a value that wasn't set to be plotted when you performed the test, and you can alter the characteristics of charts such as their axis and trace colours.

Outputting charts

Charts can be copied to the clipboard or saved to files and, in the case of the Test charts report, printed as well.

When you copy or save charts that show multiple values, like those in figure 2, you have the choice of copying or saving just one specific chart or all the charts in the set.

See also:

- Setting up a chart
- Editing a chart
- Deleting a chart

- Printing, copying and saving charts
- Setting up chart state channels

Setting up a chart

Introduction

For a general introduction to charts, see An introduction to charts.

To add a chart to a protocol, click the *Add item* button in the ribbon bar and select *New chart* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the chart's name.
2. Select the data value that the chart will actually display.
3. Optionally set the trace colour
4. Optionally set the default x-axis width for the chart when viewed post-test
5. Specify whether the chart should be displayed 'live'
6. If the chart is to be displayed 'live', set the live display options
7. Decide whether to use 'state channels' to change the background colour of the chart based on the state of other conditions during the test.

What next?

After completing these steps, you should review the standard reports to see whether the default settings are suitable for your protocol. You may also want to add some results reports of your own.

See also:

- Adding elements to a protocol
- Editing a chart
- Deleting a chart

Chart name

In brief

Enter a name for the chart in the *Chart name* field on the chart's settings page. You must make an entry, but it can be anything you like.

Details

Chart names are displayed as the y-axis label of the chart, so the name should describe the data that is being plotted.

It's a good idea to keep the name as short as possible, because the y-axis may not be very long. When a name won't fit the axis it is 'clipped off' and '...' is added to it, so for example a name like 'Distance to the platform zone' might be shown as 'Distance to...'. For this reason, it's best to make the start of the name specify what the value is; so 'Platform distance' might be a better name, as even if it were clipped to 'Platform dist...', you would still know what it means.

Limits

You can enter anything you like up to a maximum of 32 characters.

Chart data to display

In brief

You can choose the data that a chart will display from the *Data to display* drop-down list on the chart's settings page.

Details

By default, the data to display is 'None'. If you leave it like this, the chart simply won't ever be displayed.

The list of data that can be displayed by a chart depends to a very large extent on the other things that are included in the protocol. For example, for every zone that you include in the protocol, you will be able to display:

- In zone - whether the animal is in the zone or not
- Distance from the zone - the distance from the animal to the zone when the animal is outside the zone
- Distance to border - the distance from the animal to the border of the zone when the animal is inside the zone

Other values that can be displayed include:

- The speed of the animal
- Distance from a point
- Whether a key is active or not
- Whether an on/off input is active or not
- The sequence of on/off input activations - based on the inputs' index values
- The value of a sensor
- The speed (RPM) of a rotary encoder
- Whether an output switch is active or not
- The volume of the sound that a speaker is playing
- The brightness of lights controlled by a lighting controller
- Whether a shocker is active or not and what the shock current is
- Whether a virtual switch is active or not

We are always adding to this list, so if there's a value you need that's not included, just let us know -

and if it's possible, we'll include it (for free).

Chart trace colour

In brief

ANY-maze will automatically choose a colour for the trace of a chart, but you can change it simply by clicking the *Colour of the trace* box on the chart's settings page.

Details

When you click the *Colour of the trace* box, a colour selection window opens, where you can choose any colour you want for the trace. The colour box will then be shown in the colour that you chose.

Setting the default x-axis width for the chart when viewed post-test

In brief

By default, when a chart is displayed post-test, the x-axis will display the entire period of the test. If you wish, you can change this so that the default width will be some other period, by selecting the appropriate option on the chart's settings page and entering the period you wish to use.

Note that when you actually *view* a chart, you can alter its x-axis width - so the value you specify here is just the default value; that's to say, the value shown when the chart is opened.

Details

Usually when you view a chart post-test, you will probably want to see the entire test displayed across the graph - this makes it easy to review what happened in the test, and you can then 'zoom in' on any areas that interest you (by moving the mouse over them and rolling the mouse wheel).

However, in very long tests - for example, tests that last for days - this might be inappropriate, as the graph's trace may just be an almost 'solid block'. In these cases, you might want to specify that the default width is something less, perhaps 6 hours, and then you could scroll the graph to review the entire test's data.

Setting whether a chart should be displayed live

In brief

To have a chart displayed live, you simply need to select the option to *Display the chart in real-time during tests* shown on the chart's settings page.

Details

The act of including a chart in the protocol means that the chart will automatically be included in the Test charts report of all the tests in the experiment. However, if you also want the chart to be displayed 'live' during a test (which provides an excellent form of visual feedback for the value being plotted), then you need to specify this explicitly by selecting this option.

If you choose to show the chart live, there are a number of additional options available to you which are described in the topic Settings used when a chart is displayed live.

Settings used when a chart is displayed live

In brief

If you specify that a chart should be displayed live during tests, then on the chart's settings page, you will be able to set the following options:

- Whether to show the real-time value on the chart
- The default x-axis width
- The default y-axis maximum and minimum

Showing the real-time value

When this option is selected, the actual value being plotted is shown on the graph as in figure 1, below.

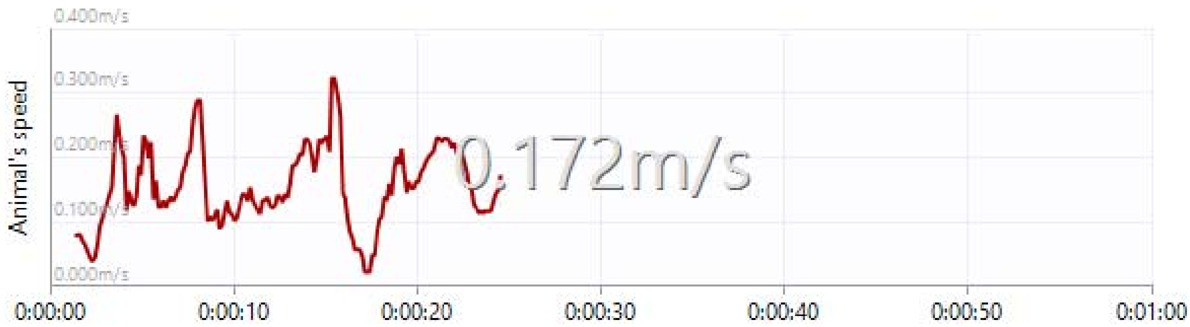


Figure 1. When a chart is displayed live, you can choose to have the real-time value of the data shown on the chart's surface.

When a chart is displaying a *value*, such as a speed or a distance, then it is this value that is shown on the chart (as in figure 1). When a chart is displaying a *state,* such as whether the animal is in a zone or not, then the state's COUNT is shown on the graph - so this would show how many times the animal has entered the zone.

The default x-axis width

By default, ANY-maze will set the width of the x-axis of a 'live' chart to the programmed duration of the test. If you wish, you can change this to some other value, simply by entering the width in this field. The value you enter should be a period of time using units of s (seconds), min (minutes), or h (hours). Leave the field blank for ANY-maze to use its default.

Note that this value is the *default* width for the x-axis; you can always adjust the width dynamically to some other value while the test is running.

The default y-axis maximum and minimum

ANY-maze will usually set the y-axis of a chart dynamically based on the data being plotted. For example, if the maximum value in the data being plotted is 9.26, then the y-axis will run from 0 to 10.

In some cases, the automatic scaling of the y-axis may not be appropriate; for example, you might be plotting a temperature and you know that the value will never go below 20°C. In this case, having the axis minimum set to 0 would be inappropriate, and you could simply enter 20 as the minimum value to fix this.

Any values that you enter will be used by ANY-maze as the new *default* maximum and/or minimum, but if the data goes outside the values you specify, then ANY-maze will automatically adjust the scale to fit. So, if the temperature you expected to never go below 20°C *did* go down to 18°C, then ANY-maze would adjust the y-axis minimum so that a value of 18°C would fit.

ANY-maze help topic H0630

Setting up chart state channels

Introduction

In addition to plotting a value, a chart can also be set up to shade its background to show certain 'states'. For example, in a chart showing the animal's speed, you could add a *state channel* to indicate whether or not the animal is in a particular zone. This way, you'll have a visual indication as to whether there is any correlation between the animal's speed and its presence in the zone.

An example of state channels can be seen in the following chart from an open field experiment, which has been set to show state channels for the animal being in the centre zone (green) and the animal being immobile (blue):

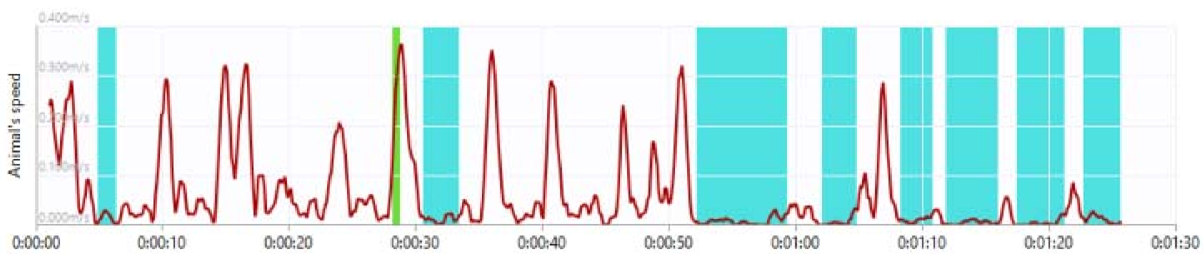


Figure 1. The background of the chart is shown in green when the animal was in the 'Centre' zone, and blue when the animal was immobile.

Details

When you add a chart to the protocol list, it will automatically include a sub-element called *State channels*. (If you can't see this element, you might need to expand the chart item by clicking on the arrow to its left).

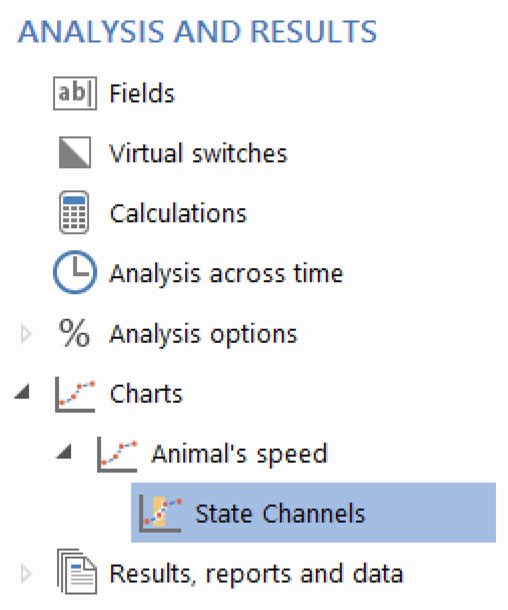


Figure 2. The State channels *item appears under the chart in the protocol list.*

Selecting the *State channels* sub-element will cause the settings pane to show a list of all the possible states that the chart can include. You simply need to select the states that you wish to display - you can select up to four. You can also change the colour used to indicate each state using the colour boxes at the bottom of the page.

Once you've set up the state channel(s) for a chart, the background of the chart will be shown in the specified colour whenever the state is active. If more than one 'state' is active at the same time, then the background will be shown as horizontal stripes of the colours of all the active states.

Editing a chart's state channels

You can edit anything about a chart's state channels either before, during or after an experiment has been performed.

Deleting a chart's state channels

It's not possible to delete a chart's *State channels* sub-element. If you don't want your chart to display any state channels, just de-select all the items in the state channels list.

Editing a chart

Introduction

You can edit absolutely everything related to a chart at any time, whether before, during or after an experiment has been performed - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a chart
- Deleting a chart
- Setting up chart state channels

Deleting a chart

Introduction

You can delete a chart at any time, whether before, during or after an experiment has been performed.

To delete a chart, simply select it in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting charts.

See also:

- Deleting protocol elements
- Editing a chart

Printing, copying and saving charts

Introduction

It's very easy to save or copy charts; you just need to right click on the chart and select an appropriate option from the menu that appears.

You can't print 'live' charts (those shown while a test is running), but you can print the Test charts report simply by clicking the *Print* button shown in the main tool bar.

Copying charts

To copy a chart, right click on it and select either *Copy this chart* or *Copy all charts* from the menu that appears. Selecting the former option will copy just the chart you have clicked on, whereas selecting the latter option will copy all the charts in the set.

Charts are copied to the clipboard as both bitmap and vector graphics. Which one will be used depends on the program you paste the chart into; for example, pasting into Microsoft Word will use the vector image, which can be scaled to any size and will still display and print sharply.

Saving graphs

Saving a graph is much like copying it; you right click on the graph and select either *Save this chart to a file* or *Save all charts to a file* from the menu that appears.

When saving a chart, you can choose the format that the chart is saved in. The options available are 'Bitmap', 'Gif', 'JPEG', 'TIFF', 'PNG', and 'Enhanced metafile'. The first four formats are all raster graphics formats, while the latter two are vector formats, which means that the saved chart can be scaled and retain its sharpness. For further information, refer to the save chart window.

Results, reports and data

Unlike the other elements of a protocol, which are narrowly targeted at a specific aspect of an experiment, the *Results, reports and data* element encompasses all those aspects which relate to report generation. Specifically, this includes: defining results reports; specifying the measures that are included in the Test schedule report; specifying the results that are shown on the Animal details and Test details reports; specifying what data is included on the Data page; and defining Report sets.

For further details, refer to the topics listed below:

- An introduction to results, reports & data
- Test schedule report settings
- Animal and test details reports settings
- The Test data report settings
- Data page settings
- Setting up Results reports
- Setting up Report sets
- Information measures

An introduction to results, reports & data

Introduction

The *Results, reports and data* element is used to define various report-related aspects of a protocol. Specifically, these are:

- Defining results reports in the protocol
- Selecting what's shown in the standard reports
- Selecting what's shown on the Data page
- Creating report sets

Defining results reports in the protocol

In many experiments, before you even start testing, you will know exactly what results you will want reported - for example, in a plusmaze you would probably want to know, at least:

- The numbers of entries into the open arms
- The numbers of entries into the closed arms
- The percentage of time in the open arms
- The total distance travelled in the test

Of course, you could easily use the Results page to analyse each of these measures in each plusmaze experiment you perform. However, a more efficient method would be to define a report which performs this analysis as part of the 'plusmaze' protocol. This would mean that for *any* plusmaze experiment, you would be able to instantly view these results without having to select the appropriate measures, report type, options, etc. each time.

Defining a report in the protocol doesn't prevent you from using the Results page to perform additional analysis of your data; rather, it just makes it easy to access 'standard' results that you think you'll always want to see. Of course, if in your protocol there isn't such a group of 'standard' results, then you simply don't need to include any results reports at all.

To actually define a results report in the protocol, you first have to add a report element to the protocol and then add any number of *report sections* to it. Each section will show the results for a measure, or group of measures, in a particular format.

For example, in a plusmaze protocol, you might create a report which consists of two sections: The first showing a graph of 'Percentage of time in the open arms' for each treatment group, and the second showing an Analysis of variance of the same data - as is shown in figure 1.

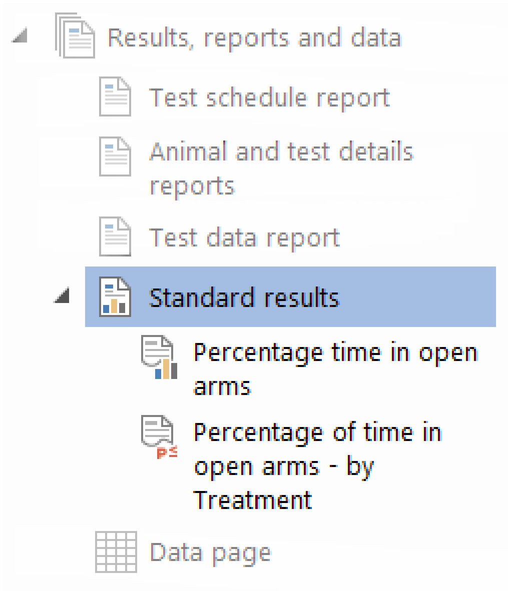


Figure 1. A results report containing a graph and a statistics section - the report could include any number of additional sections to show other measures or formats.

Of course, you would probably want to include other measures in this report, but I'm sure you get the idea.

Having defined a report in your protocol, you'll find that it appears in a drop-down list labelled *Stored reports* in the ribbon bar of the Results page - see figure 2. Simply selecting an item from the list displays the appropriate report, which can then be viewed, printed, copied, saved, or even sent by e-mail.

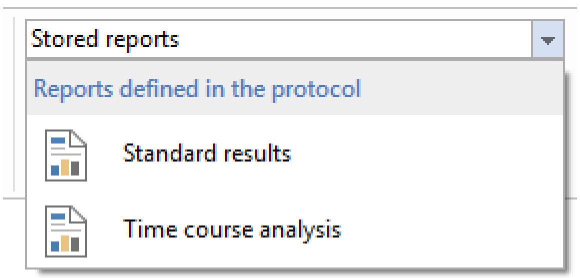


Figure 2. Once defined, a results report is created by selecting its entry from the list of stored reports that's shown in the ribbon bar at the top of the Results page.

Selecting what's shown in the standard reports

As well as results reports, ANY-maze includes a number of *standard* experiment management reports such as the Test schedule report and the Animal details report. These reports are *standard* in the sense that they're automatically defined by ANY-maze and you can't remove them. However, you can control what they show.

For example, in a water-maze protocol, you may use a procedure that causes tests to end when an animal finds an island zone. Of course, some animals may not find the island and so their tests would end because the test duration (perhaps 5 minutes) has expired. Thus, you could have two *test end reasons*: 'Found island' and 'Test duration', and you would probably find it useful if this information was included on the Test schedule report. On the other hand, in a plusmaze protocol, including the 'Test end reason' on the Test schedule report would be pointless because *all* tests would end because of the 'Test duration'.

Clearly, then, you might want to choose the information that's shown on the *standard* reports, and you can do this by selecting either the *Test schedule report*, *Animal and test details reports,* or *Test data report* items which are listed under the *Results, reports and data* element of the protocol - see figure 3.

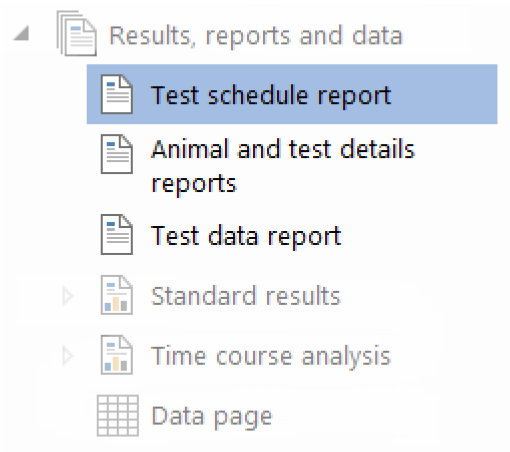


Figure 3. You can use the Test schedule report, Animal and test details reports, *and* Test data report *items to choose what's shown in these reports.*

Selecting what's shown on the Data page

The Data page shows the results of an experiment in a spreadsheet format, with each row showing a single test (or a segment of a test) and each column showing an individual result - for example, Test duration or Total distance travelled. The principal use of this page is to transfer data to other programs, usually to carry out more sophisticated statistical analysis than ANY-maze performs.

Clearly, then, you may know when you design your protocol exactly what information you will want to transfer, and therefore exactly what rows and columns you would like to have shown on the Data page. If this is the case, then you can specify this information using the *Data page* item listed under the *Results, reports and data* element of the protocol.

Of course, this doesn't mean you *have* to choose the information displayed on the Data page as part of the protocol; rather, it just means you *can* do this if you anticipate that you'll want the same data shown for all the experiments. Furthermore, even if you do specify the data shown, it won't stop you altering it using the options included on the Data page itself.

Creating Report sets

On completing experiments, you're probably in the habit of gathering up all the results, reports, notes etc. and filing them together - this way, if you want to refer back to the experiment, everything is in one place. Of course, with time, you'll perform lots of experiments - and at the end of a year, you may have a couple of files labelled *Water-maze experiments*, or whatever.

An obvious advantage of a computerised system such as ANY-maze is that you can record everything on the computer, and as each experiment is stored in its own file, you can easily build up an electronic archive of your experiments. Well, almost! There are a few reasons why such an electronic archive might not be as good as it sounds.

- Different individuals in the lab might keep their ANY-maze files in different folders - this makes it easier for each person to manage their own work, but it will mean your 'archive' isn't all in one place.
- To refer to a file, you need to have access to the ANY-maze computer which might be inconvenient if someone's performing an experiment.
- To actually view an experiment's results, you need to know how to use ANY-maze - this might be a problem if you would like colleagues who don't actually use the system to be able to refer to the experiment's results.

Of course, none of these problems arise if you simply maintain a 'paper archive' such as I described above. But a paper archive takes up space, is located in a single place where people have to go to refer to it, and documents can be removed or even lost - what's really needed is an electronic archive without the shortcomings I described above, and this is the principal (although not exclusive) use of ANY-maze *report sets*.

A report set is, as the name implies, a set of reports related to a single experiment. For example, in a plusmaze experiment, a report set might consist of:

- The Protocol report - which defines the methodology used.
- The Test schedule report - which shows details of the sequence in which the tests were performed.
- The Animal and Test details reports - which show full details of the tests performed on each individual animal, together with any notes the experimenter made
- Some results reports (see above) - which show the results of the experiment, perhaps

including graphs and statistical analysis.

This full set of reports can, in a single step, be created and saved as a group of related 'web pages' together with an index page. What's more, the experiment's index page can be added to a master index of all your plusmaze experiments.

By saving report sets to one of your lab's file servers, you can create an easily- accessible archive of your experiments:

- Because the reports are saved as 'web pages' (i.e. in HTML format), anyone who can access the computer where they're saved can read them without needing to know how to use ANY-maze.
- As all experiments are added to a master index, you only need to go to this index to access the data for any experiment you, or anyone else, has ever performed.
- Any number of people can access the reports, without the risk of them being removed, edited, or lost.
- If the reports are saved to a file server, then nobody needs to use the ANY-maze computer to read them.
- You can easily use ANY-maze to update a report set with new analysis.

As well as *publishing* report sets in this way, you can also send an individual experiment's report set by e-mail. This can be a great way to share results with a co-worker as all the data is included in an easily accessible format.

See also:

- The Tests page
- The Results page
- The Data page

Test schedule report settings

Introduction

The Test schedule report has been designed to help you manage the running of tests in an experiment. As such, the information that you might want displayed will often depend on the exact nature of your protocol.

For example, in a protocol in which you will be using more than one piece of apparatus for simultaneous testing of your animals, you will almost certainly want to include the 'Apparatus' in the report - whereas if you only have a single piece of apparatus, this information would be rather pointless.

Choose the information to show

To choose the information to show in the report is very simple:

- Select the *Result, reports and data* element in the protocol list.
- Select the *Test schedule report* item - this item is automatically included in the protocol and you can't delete it.
- Select the columns you want shown using the list displayed in the Settings pane.

The list of data you can include in the report is shown in groups and subgroups, you simply need to click a group or subgroup title to open it.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One

	deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Select same as	This options opens a sub-menu that lists other reports defined in the protocol. If you'd like the measures in this report to be the same as those in one of the other reports, just select the appropriate entry in the list. This saves you having to repeat the same selections again and again.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Although you might be tempted to include lots of columns, this isn't necessarily a good idea because it will probably mean the report is so big that you have to scroll it from side to side to see everything. In general, I'd advise you to add as few columns as possible, and then alter them as you gain experience using the protocol.

What next?

After setting the options for the test schedule report, you should consider what information you would like displayed in the animal and test details reports.

See also:

- The Test schedule report

Animal and test details reports settings

Introduction

The Animal details report and the Test details report act as unified locations where you can see all the information relating to an individual animal or test and, of course, this will include the test's results - assuming it's been performed.

However, as you have probably noticed, ANY-maze can analyse literally hundreds of different results and this obviously leads to the question of which results should actually be included in these reports.

In most protocols, there will be a few *key* results, such as the 'Percentage of time in the open arms' in a plusmaze, or the 'Time to find the island' in a water-maze. You will probably find it useful if these, at least, are shown in the Animals and Test details reports. That's not to say that you can only include a limited number of results - you can include as many as you like - but you probably won't want to make the reports unwieldy by including lots of data.

Choosing the results to display in the reports

To actually choose the results to include in the reports, you should:

- Select the *Result, reports and data* element in the protocol list.
- Select the *Animal and test details reports* item - this item is automatically included in the protocol and you can't delete it.
- Use the list displayed in the Settings pane to select the measures whose results you want included.

The list of measures is shown in groups and subgroups and you simply need to click a group or subgroup title to open it.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to

	view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Select same as	This options opens a sub-menu that lists other reports defined in the protocol. If you'd like the measures in this report to be the same as those in one of the other reports, just select the appropriate entry in the list. This saves you having to repeat the same selections again and again.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

What next?

After setting the options for the animal and test details reports, you should consider what information you would like displayed in the test data report.

See also:

- The Animal details report
- The Test details report

The Test data report settings

Introduction

The Test data report shows a spreadsheet that lists 'events' that occurred during a test in chronological order. For example, it can show the time of each zone entry and each zone exit.

In fact, the report can include a wide range of different 'events', not just zone entries and exits - including such things as changes in the position of the animal, keys being pressed, sequences being completed, and so on.

You will usually want to view a subset of all the possible events, because including them all tends to make the report both very large and also difficult to interpret.

Time	Centre position X	Centre position Y	Speed	In NW	In SW	In NE	In SE	In Island
0:00:00.000	#N/A	#N/A	#N/A	0	1	0	0	0
0:00:00.500	225	356	#N/A	0	1	0	0	0
0:00:00.840	224	359	#N/A	0	1	0	0	0
0:00:01.570	227	374	0.068m/s	0	1	0	0	0
0:00:01.840	231	374	0.068m/s	0	1	0	0	0
0:00:02.010	238	378	0.068m/s	0	1	0	0	0
0:00:02.260	250	383	0.141m/s	0	0	0	1	0
0:00:02.590	269	381	0.233m/s	0	0	0	1	0
0:00:02.760	279	378	0.233m/s	0	0	0	1	0
0:00:02.920	291	375	0.270m/s	0	0	0	1	0
0:00:03.260	315	365	0.309m/s	0	0	0	1	0
0:00:03.430	328	358	0.333m/s	0	0	0	1	0
0:00:03.590	342	348	0.430m/s	0	0	0	1	0
0:00:03.760	356	337	0.404m/s	0	0	0	1	0
0:00:03.930	366	325	0.357m/s	0	0	0	1	0
0:00:04.100	376	313	0.357m/s	0	0	0	1	0

Figure 1. An example of the Test data report, showing the animal's position, speed, and whether or not it was in certain zones.

Choosing the events to include in the report

To actually choose the events to include in the report, you should:

- Select the *Result, reports and data* element in the protocol list.
- Select the *Test data report* item - this item is automatically included in the protocol, and you can't delete it.
- Use the list displayed in the Settings pane to select the events you want included.

You will find that events shown in this list change, depending on the context of the protocol. For example, if your protocol doesn't include any sequences, then there won't be an entry for 'Sequence completions', but if you add a sequence to the protocol, then this entry will appear.

If you right click in the event list, then a menu containing the following options will appear:

Select all	Selects all of the events in the entire list. Although this might appear to be an easy way to view all the events that occurred during the test (and it is), you should be aware that it will usually cause the report to become extremely long and hard to understand.
Deselect all	Deselects all events in the entire list. This is useful if you just want to reset the list and start over selecting the events to include.

Choosing the format of times shown in the report

Each event that occurs during a test is given a time in milliseconds since the start of the test. On the report these times can be shown either in hours, minutes and seconds (for example, 00:10:22.152 being 10 minutes, 22.152 seconds since the test start) or simply in seconds (for example, 622.152 being 622.152 seconds since the test start). Usually, the hours, minutes and seconds format is more intuitive, but if you copy the report to Excel, you will find that it won't understand this format and in this case, you should use the 'seconds' only format instead.

In some cases, you may want to see when events occurred in real-time (i.e. the actual time as shown on a clock). This can be useful if, for example, you want to tie together the events recorded in ANY-maze with data that was simultaneously recorded by some other system. In this case, you can choose to have the real-times of the events shown using the 24 hour clock in HH:MM:SS.sss format, for example 13:25:12.961 (being 12.961 seconds after 1:25pm).

Choosing how the animal's position is shown in the report

During tests, ANY-maze records the animal's position in pixels and this is how it will usually show the positions in the Test data report. However, you can choose to have the positions shown in millimetres, rather than pixels, if you prefer. In either case the origin is the top-left of the video picture.

What next?

After setting the options for the test data report, you should consider what information you would like displayed on the data page.

See also:

- The Test data report

Data page settings

Introduction

The Data page shows the results of an experiment in a spreadsheet format, with each row showing a single test (or a segment of a test) and each column showing an individual result - for example, Test duration or Total distance travelled. The principal use of this page is to transfer data to other programs, usually to carry out more sophisticated statistical analysis than ANY-maze performs.

You may know when you design your protocol exactly what information you will want to transfer, and therefore exactly what rows and columns you would like to have shown on the Data page. If this is the case, then you can specify this information in the protocol.

Of course, this doesn't mean you *have* to choose the information displayed on the Data page as part of the protocol; rather, it just means you *can* do this if you anticipate that you'll want the same data shown for all the experiments. Furthermore, even if you do specify the data shown, it won't stop you altering it using the options included on the Data page itself.

Choosing the data shown on the Data page

To actually choose the measures (columns) and tests (rows) shown on the Data page, you should:

- Select the *Result, reports and data* element in the protocol list.
- Select the *Data page* item - this item is automatically included in the protocol, and you can't delete it.
- Use the two lists displayed in the Settings pane to select the measures and tests whose results you want included.

Selecting the measures to include

You can use the *measures list* to select the individual measures you would like shown on the data page - these will be the spreadsheet's columns. The list shows the measures in groups and subgroups, you simply need to click a group or subgroup title to open it.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option

	selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Select same as	This options opens a sub-menu that lists other reports defined in the protocol. If you'd like the measures in this report to be the same as those in one of the other reports, just select the appropriate entry in the list. This saves you having to repeat the same selections again and again.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Selecting the tests to include

To impose limits on the tests (rows) that are included in the Data page spreadsheet you simply need to select items in the list of limits at the bottom of the data page settings.

The list shows items in groups, such as *Trial,* and by clicking the group title, you can actually select individual items - for *Trial,* for example, the items might be 'Training trial 1', 'Training trial 2', 'Training trial 3' and 'Retention trial'. If you only want the data for the 'Retention trial' included in the spreadsheet, then you select this item.

You can select more than one item in a group. For example, selecting 'Training trial 1 and 'Training trial 2' would limit the tests in the spreadsheet to those in Training trial 1 and Training trial 2. You can also make selections in different groups, for example, under *Trial* you might select 'Retention trial' and under 'Test day of the week' you might select 'Wednesday'. This would limit the report to Retention trials performed on Wednesday - I'm sure you get the idea.

As you'd expect, if you make no selections at all, then ALL the tests in the experiment will be included in the spreadsheet.

Excluding retired animals

Animals that are *retired* from an experiment stop being tested, but their results are retained. If you would like to exclude the tests performed on any retired animals from the Data page spreadsheet, then simply check the box labelled *Don't include retired animals in the data table.*

What next

After completing these steps, your protocol is finished!

See also:

- Transferring data to other programs
- The Data page
- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Setting up a results report

Introduction

For a general introduction to results reports, see An introduction to Results, reports & data.

To add a results report to a protocol, click the *Add item* button in the ribbon bar and select *New report* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the report's name.
2. Optionally limit the report to only analyse a subset of an experiment's data.
3. Specify whether any retired animals should be included in the report's analysis.
4. Add at least once section to the report.

See also:

- Adding elements to a protocol
- Editing a report
- Deleting a report

Report name

In brief

Enter a name for the report in the *Report name* field on the report's settings page. You must make an entry, but it can be anything you like.

Details

You should avoid making report names too long - although it might be tempting to use a name which describes everything included in the report, you should instead try to use a more succinct name such as 'Standard results', or 'Time course analysis'.

Limits

You can enter anything you like up to a maximum of 64 characters.

Limiting the data analysed in a report

In brief

In the list shown on the report's settings page, you can optionally select limits to the data that will be included in the report. The list is divided into groups, and you should click a group title to view the entries inside it.

To impose a limit, tick one or more items - the report will only include those tests which match the limits you impose.

Details

In some reports, you may not wish to analyse all the tests performed in the experiment. For example, imagine an experiment in which you trained animals to perform some task. You then treated the animals and you want to compare their post-treatment performance in the task between various treatment groups. In this case, you wouldn't want to include analysis of all the training trials in your results.

To impose limits on what's included in a report, you simply need to select items in the list of *Results to plot*. This list shows items in groups, such as *Trial,* and by clicking the group title, you can actually select individual items. For *Trial,* for example, the items might be 'Training trial 1', 'Training trial 2', 'Training trial 3' and 'Test trial'. If these were the entries for the example I described above, then you'd want to select just the 'Test trial' entry, which you'd do by clicking the box next to it.

You can select more than one item in a group; for example, selecting 'Training trial 1' and 'Training trial 2' would limit the tests in the report to those in Training trial 1 or Training trial 2. You can also make selections in different groups; for example, under *Trial* you might select 'Test trial' and under *Test day of the week* you might select 'Wednesday'. This would limit the report to Test trials performed on Wednesday - I'm sure you get the idea.

As you'd expect, if you make no selections at all then ALL the tests in the experiment will be included in the report's analysis.

Specifying whether retired animals should be included in a report's analysis.

In brief

Animals that are *retired* from an experiment stop being tested, but their results are retained. If you would like to exclude the results for any retired animals from the report's analysis, then simply check the box labelled *Don't include results for retired animals in the report* on the report's settings page.

Details

You might retire animals from an experiment if they fail to achieve some type of training goal, if they become ill, or if they cease to match some type of selection criteria. Retiring an animal, unlike deleting an animal, doesn't automatically remove it from results reports; rather, it signals to ANY-maze that the animal should no longer be tested.

For example, imagine you are training animals to find an island in a water-maze. You might decide that the animals will have a maximum of 6 training trials and if, after this, they still can't find the island, then you'll simply remove them from the experiment. Although you could delete the animals that don't achieve the goal, it would be better to retire them, as you could then include their results in any analysis performed of the training itself. However, if you wanted to compare the animals' performance in their last training trial and a trial performed after some type of treatment, then you clearly wouldn't want to include any retired animals.

To exclude retired animals from a report, you simply need to select the option labelled *Don't include results for retired animals in the report.*

Adding a section to a report

In brief

Reports are made up of a number of individual sections. Each section performs analysis of a specific measure (or measures), and displays the results in a certain format. You must add at least one section to a report, otherwise it won't include any analysis at all.

To add a section, you should first ensure that the appropriate report is selected in the protocol list, then click the *Add item* button in the ribbon bar and select *New report section* from the menu which appears. You will then be able to choose the type of section you wish to add.

Details

Reports consist of one or more *sections*, each of which defines the measures to be analysed, any appropriate analysis options, and the format in which the results will be presented. There are four different section types you can add to a report:

- Text sections
- Graph sections
- Statistics sections
- Track plot sections

Text sections

A text section displays results in a simple text format. The data is collated into groups (for example, treatment groups) and for each group, the report can show the group N, mean, standard deviation, standard error, and the actual data - see figure 1.

Test duration (s)

Treatment	*N*	*Mean*	*SD* (note 1)	*Data* (notes 2, 3)
Saline	6	9.38	±6.03	19.0, 8.5, 11.3, 3.7, 11.4, 2.4
Compound X 1.0 mg/kg	5	18.72	±11.53	3.1, 30.7, 24.3, 10.3, 25.2
Compound X 5.0 mg/kg	5	33.46	±16.19	43.5, 35.7, 51.9, 26.2, 10.0
Compound X 10.0 mg/kg	5	83.76	±31.34	118.5, 110.3, 85.9, 53.1, 51.0

Notes:
1. SD = Standard deviation.
2. The data analysed has been limited in the following way: Trial = Treated trial.
3. Animals which are marked as retired have been excluded from this report.

Figure 1. An example of a text section showing results for the 'Test duration' measure. The data have been grouped by treatment.

Graph sections

As you'd expect, a graph section shows results in a graphical format. Graphs show the means for individual groups and can optionally include error bars. You can show multiple series on a single graph if you want to - see figure 2.

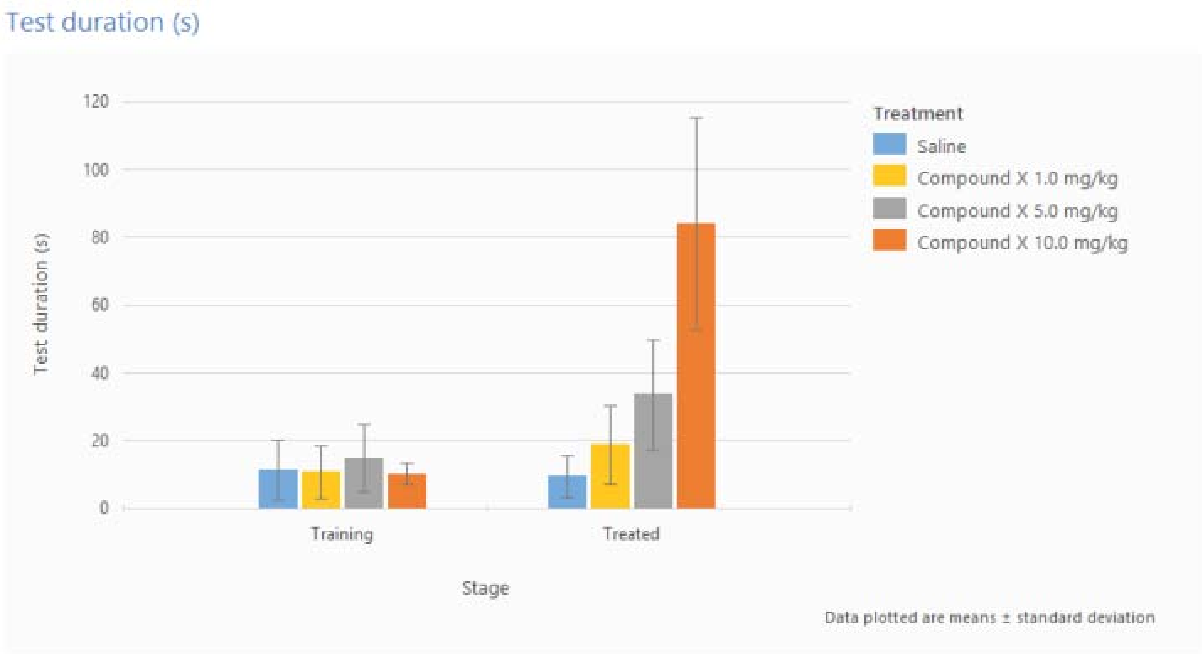


Figure 2. An example of a graph section showing results for the 'Test duration'. The graph shows stages on the x-axis, and uses different series for each of the experiment's treatment groups.

Statistics sections

Statistics sections are used to perform statistical analysis of your results. You can choose up to two independent variables and any number of dependent variables to be analysed and then select an appropriate statistical test from those provided by ANY-maze.

Track plot sections

Track plot sections don't actually perform any analysis as such, rather they show test track plots for different groups of tests - this can be an effective, although subjective, way to visualise test results. For example, showing track plots for different treatment groups in a water-maze can often make it very clear that some groups know where an island is and others don't.

For further details about the different sections, refer to the following topics:

- Setting up a text section of a report
- Setting up a graph section of a report
- Setting up a statistics section of a report
- Setting up a track plot section of a report

There are no limits to how many sections you can include in a report. You can also include the same analysis in more than one section - this can be useful, for example, if you want to show the results both as text and as a graph.

Note that sections appear in the report in the order in which they appear in the report's definition within the protocol. So, for example, if you add a text section and then a graph section, the report will show the text first and then the graph.

In future versions of ANY-maze, you will be able to re-order them if you want to by simply dragging them within the protocol list.

Setting up a text section

Introduction

For a general introduction to report sections, see Adding sections to a report.

To add a text section to a results report, click the ✚ *Add item* button in the ribbon bar and select *New report section* from the menu which appears. On the sub-menu which is then shown, select *New text section*.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Select the measures you want to analyse
2. Select how you want the data grouped
3. Specify what information you want shown

See also:

- Limiting the data analysed in a report
- Adding elements to a protocol
- Editing a report
- Deleting a report

Select the measures you want to analyse

In brief

You should use the list of measures shown on the text section's settings page to select those which you want to include in the text section. The measures are shown in groups; you should click a group's title to open it so you can select the measures inside. You can choose as many measures as you like - each one will be analysed separately in the report.

Details

A text section will show analysis of each of the measures you select, in the order in which they appear in the measures list.

To select a measure in the list, first click the title of its group, for example *Apparatus measures,* and then check the box next to the measure's name, for example *Test duration* - see figure 1.

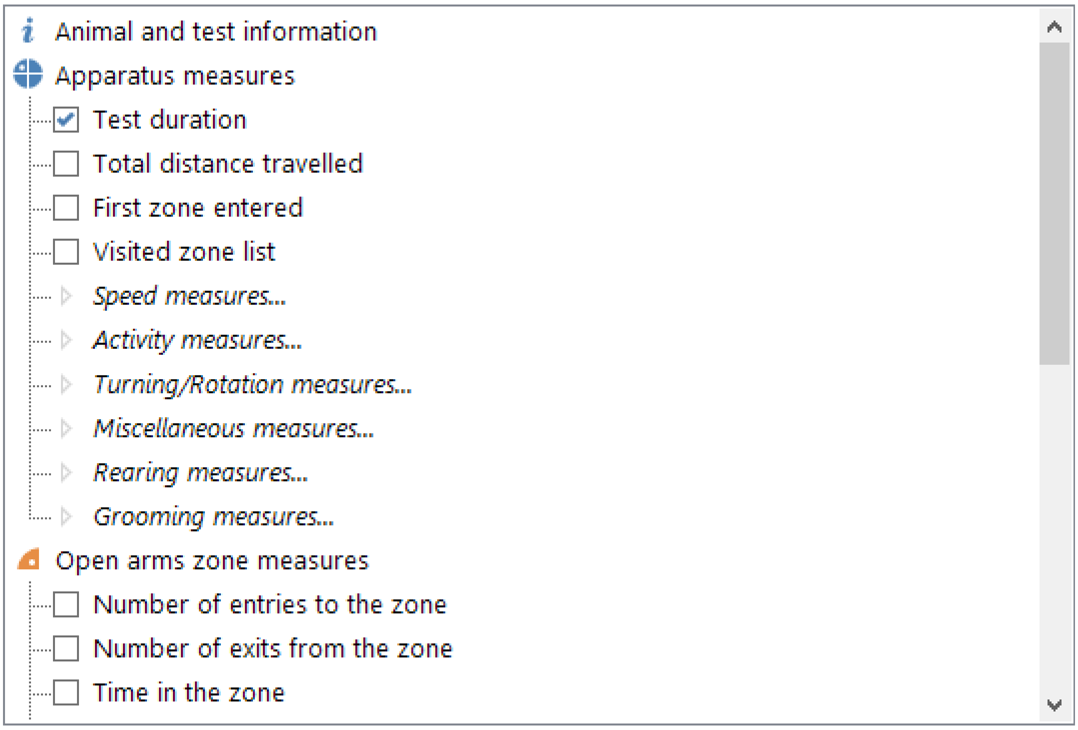


Figure 1. Measures are organised into groups - clicking a title opens the relevant group.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures

- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Select how the data will be grouped

In brief

A text section shows the results of measures for groups of tests. For example, a text section which is grouped by 'Treatment' would show the results for each treatment group in the experiment - see figure 1.

Test duration (s)

Treatment	*N*	*Mean*	*SD* (note 1)	*Data* (notes 2, 3)
Saline	6	9.38	±6.03	19.0, 8.5, 11.3, 3.7, 11.4, 2.4
Compound X 1.0 mg/kg	5	18.72	±11.53	3.1, 30.7, 24.3, 10.3, 25.2
Compound X 5.0 mg/kg	5	33.46	±16.19	43.5, 35.7, 51.9, 26.2, 10.0
Compound X 10.0 mg/kg	5	83.76	±31.34	118.5, 110.3, 85.9, 53.1, 51.0

Notes:
1. SD = Standard deviation.
2. The data analysed has been limited in the following way: Trial = Treated trial.
3. Animals which are marked as retired have been excluded from this report.

Figure 1. A text section grouped by treatment.

You can select up to three *nested* groups, where the nesting causes groups to be further subdivided - see figure 2.

Test duration (s)

Group (note 1)	*N*	*Mean*	*SD* (note 2)	*Data* (notes 3, 4)
Saline				
Male	3	13.90	±4.42	19.0, 11.3, 11.4
Female	3	4.87	±3.21	8.5, 3.7, 2.4
Compound X 1.0 mg/kg				
Male	3	12.57	±10.78	3.1, 24.3, 10.3
Female	2	27.95	±3.89	30.7, 25.2
Compound X 5.0 mg/kg				
Male	2	22.85	±18.17	35.7, 10.0
Female	3	40.53	±13.10	43.5, 51.9, 26.2
Compound X 10.0 mg/kg				
Male	2	80.65	±41.93	110.3, 51.0
Female	3	85.83	±32.70	118.5, 85.9, 53.1

Notes:
1. The data have been grouped by Treatment, then by Sex.
2. SD = Standard deviation.
3. The data analysed has been limited in the following way: Trial = Treated trial.
4. Animals which are marked as retired have been excluded from this report.

Figure 2. A text section grouped by treatment and then by sex.

To select the grouping measures, use the three drop-down lists shown on the text section's settings page.

Details

You must select at least one group. If what you really want is just the raw data without any grouping or analysis, then you should probably use the spreadsheet shown on the Data page.

Specify what's to be shown in the section

In brief

Text sections always show groups' names, Ns, means, and raw data. You can also optionally choose to show the groups' standard deviations and/or standard errors by checking the appropriate options on the text section's settings page

If you wish to, you can also specify that the raw data shown is labelled with animal and/or trial numbers - see figure 1. In fact this is extremely useful because the animal and trial number labels are *links* to the relevant Animal details and Test details reports, respectively.

35.7 (8,1)

Figure 1. Raw data can be labelled with animal and trial numbers. In this case the value is for animal 8, trial 1.

Setting up a graph section

Introduction

For a general introduction to report sections, see Adding sections to a report.

To add a graph section to a results report, click the ➕ *Add item* button in the ribbon bar and select *New report section* from the menu which appears. On the sub-menu which is then shown, select *New graph section*.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Select the measures you want to analyse
2. Specify how you want the graph organised
3. Specify the type of graph you want

See also:

- Limiting the data analysed in a report
- Adding elements to a protocol
- Editing a report
- Deleting a report

Select the measures you want to analyse

In brief

You should use the list of measures shown on the graph section's settings page to choose those that you want to include in the graph section. The measures are shown in groups; you should click a group's title to open it so you can select the measures inside. You can choose as many measures as you like - each one will be analysed separately in the report, i.e. it will appear on a separate graph (or graphs).

Details

A graph section will show a graph (or perhaps graphs) of each of the measures you select, in the order in which they appear in the measures list.

To select a measure in the list, first click the title of its group, for example *Apparatus measures,* and then check the box next to the measure's name, for example *Test duration* - see figure 1.

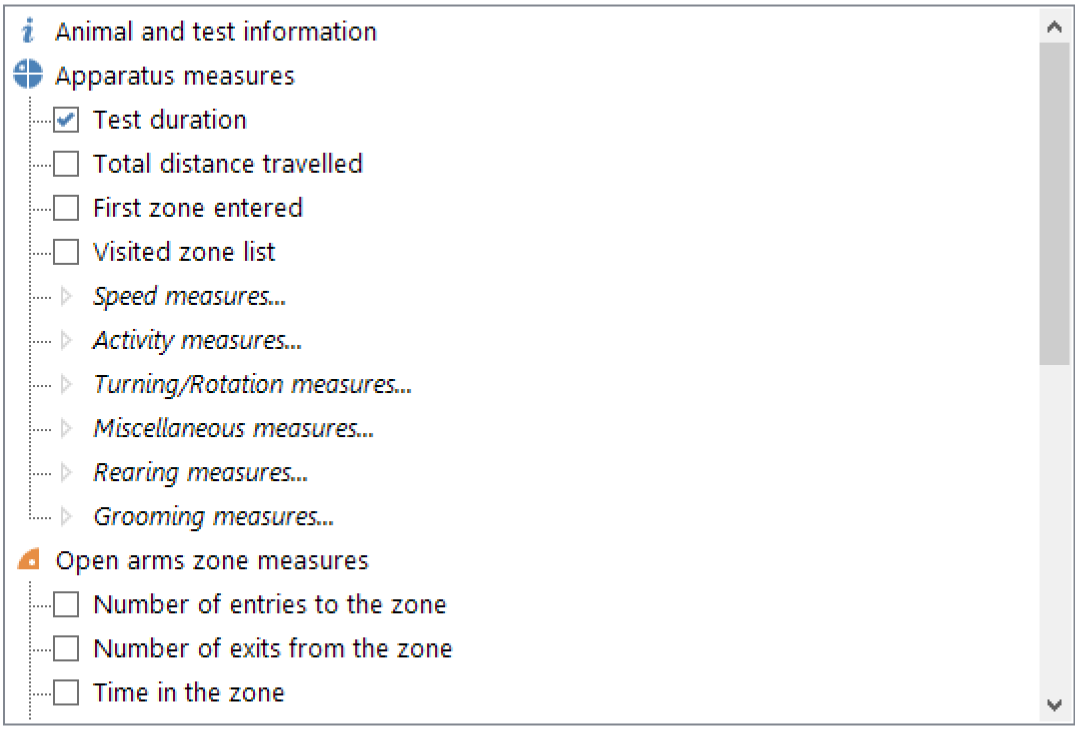


Figure 1. Measures are organised into groups - clicking a title opens the relevant group.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures

- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Specifying how a graph section should be organised

In brief

You should use the *On the x-axis show* drop-down list on the graph section's settings page to select what will be shown on the x-axis of your graph. For example, choosing *Treatment* will cause one column (or point if you use a line graph) to be shown for each of the treatment groups in your experiment.

You can optionally choose to break data into different series and/or across different graphs by making appropriate selections in the other two drop-down lists shown on the same page.

Details

Graphs plot the means of measures for particular groups. At its simplest, a graph will just show the different groups on the x-axis with the measure value on the y-axis - see figure 1.

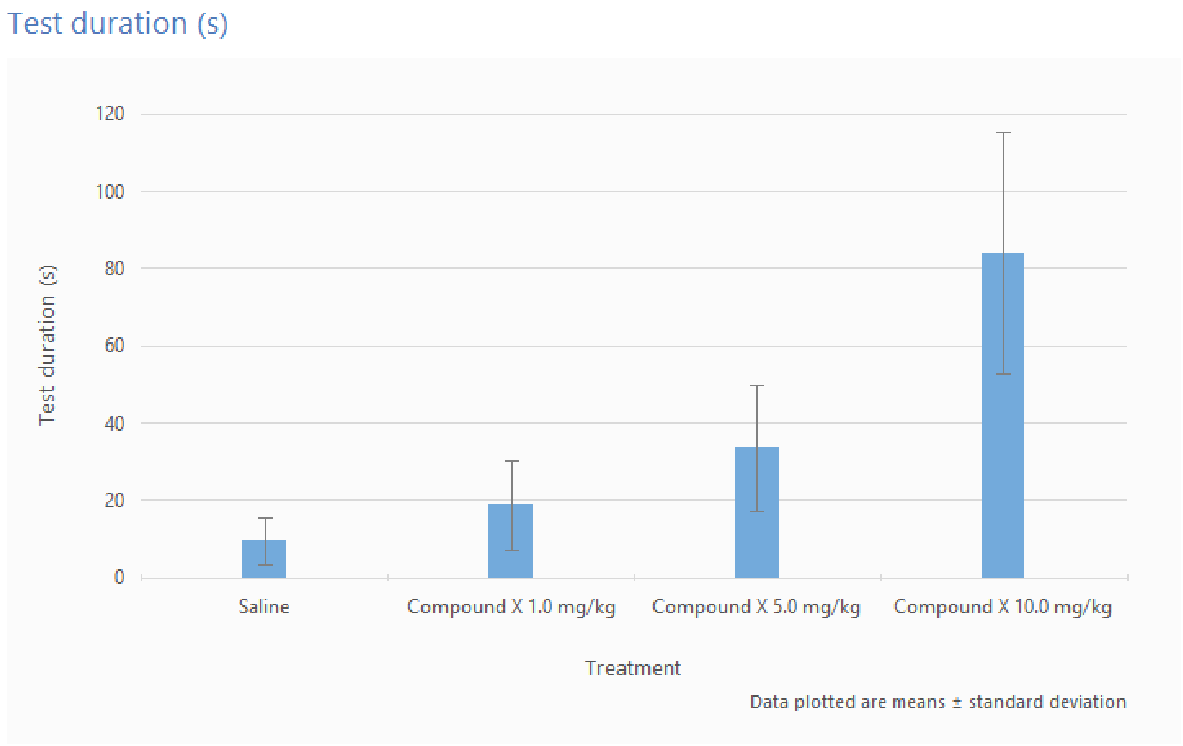


Figure 1. A simple graph showing the mean Test duration for different treatment groups.

However, you might want to split your data into different series too. For example, if your protocol includes different stages, then you might want to show the stages on the x-axis and show different series for each of the treatment groups - see figure 2.

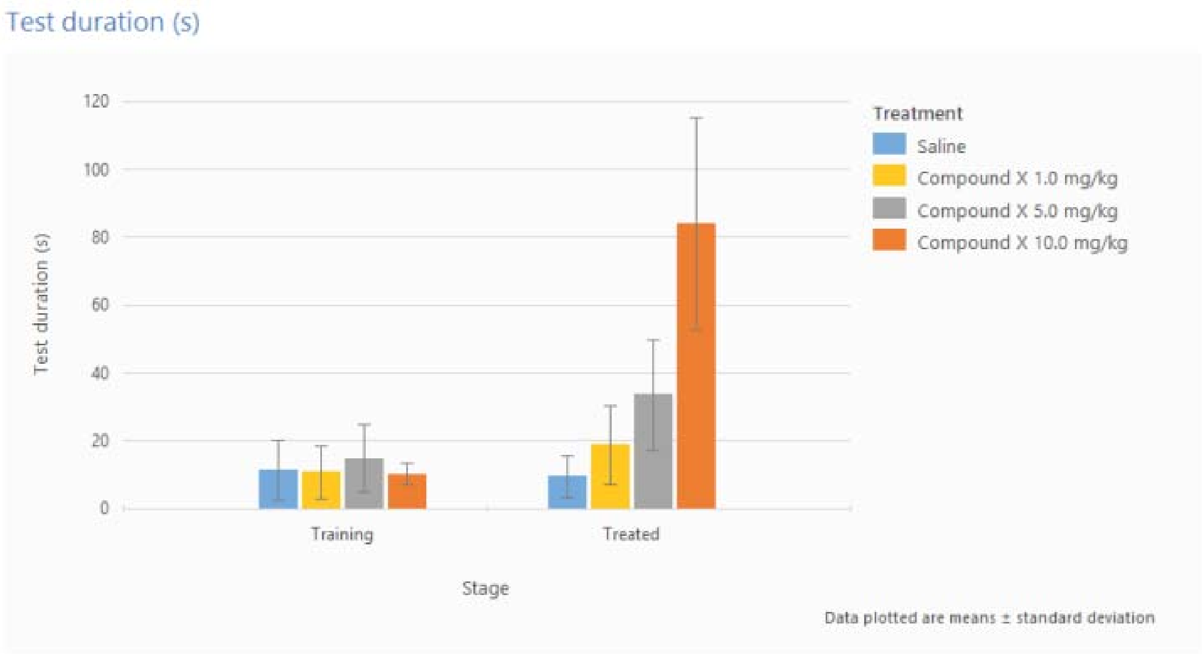


Figure 2. A graph showing the mean Test duration in different stages for different treatment groups.

Alternatively, in this situation, you might prefer to use different graphs, rather than different series on a single graph.

In fact, you can show different graphs *and* include different series on each of them. For example, you might want different graphs for each stage in an experiment, with each graph having a different series for each treatment, and with the x-axis of each graph showing the different trials in the stage.

Specifying the graph type

In brief

ANY-maze can create column graphs, line graphs, or scatter plots. The system will automatically select column graphs if the data on the x-axis is discrete, or line graphs if it's continuous. You can of course change this if you like, using the options on the graph section's settings page.

More details

On column and line graphs, ANY-maze plots mean values and, when you choose what to plot, you can specify that this should include error bars of either the standard deviation (SD) or standard error of the mean (SEM). The error bars can either be positive only or both positive and negative.

On scatter plots, the system shows the individual data points and can optionally include a line indicating the groups' means or medians.

Setting up a statistics section

Introduction

For a general introduction to report sections, see Adding sections to a report.

To add a statistics section to a results report, click the *Add item* button in the ribbon bar and select *New Report section* from the menu which appears. On the sub-menu which is then shown, select *New statistics section*.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Select the dependent variables you want to analyse
2. Select the independent variables you want to use
3. Specify the analysis you want to perform
4. Set optional information you want shown

See also:

- Limiting the data analysed in a report
- Adding elements to a protocol
- Editing a report
- Deleting a report

Select the dependent variables you want to analyse

In brief

You should use the list of measures shown on the statistics section's settings page to choose the dependent variable, or variables, you want to analyse. The measures are shown in groups; you should click a group's title to open it so you can select the measures inside. You can choose as many measures as you like - each one will be analysed separately in the report.

Details

A statistics section will analyse each of the measures you select. The results of the analysis will appear in the order in which they appear in the measures list.

To select a measure in the list, first click the title of its group, for example *Apparatus measures,* and then check the box next to the measure's name, for example *Test duration* - see figure 1.

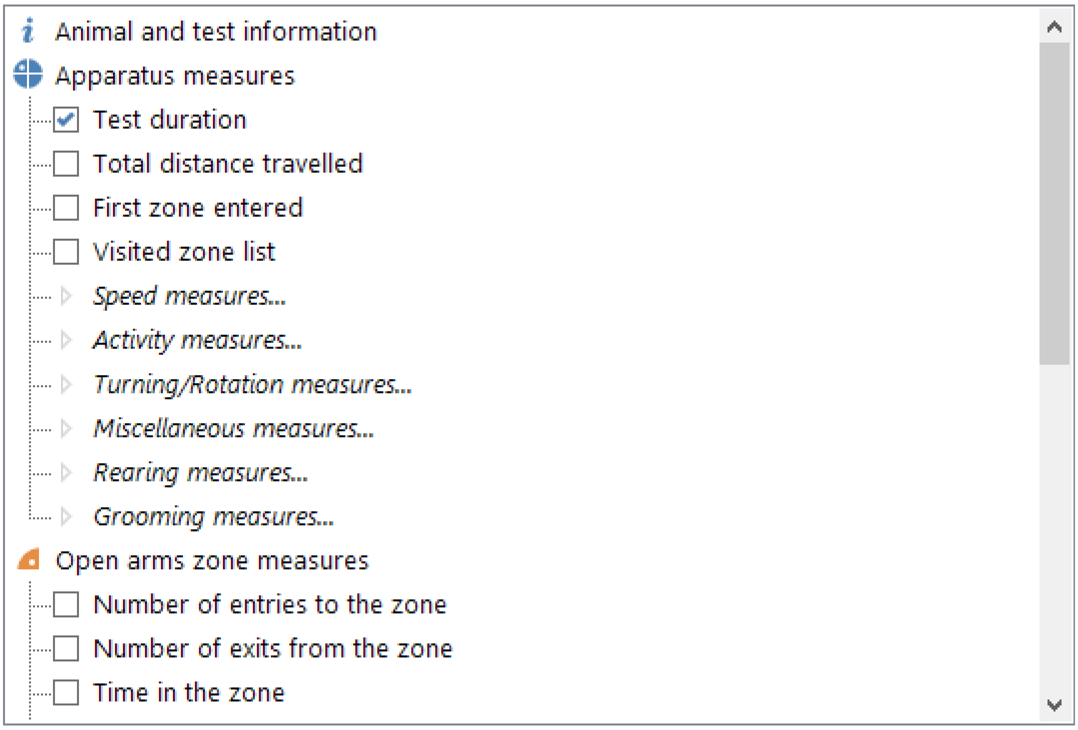


Figure 1. Measures are organised into groups - clicking a title opens the relevant group.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures

- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Select the independent variables you want to use

In brief

You should use the two drop-down lists shown on the statistics section's settings page to select one or two independent variables that you want to use in the analysis. Only measures which can logically be used as independent variables are listed.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Specify the analysis you want to perform

In brief

The first decision to make is whether you want to analyse the dependent variables you have selected using parametric or non-parametric tests. Depending on which of these you choose, ANY-maze will analyse the dependent variables using an appropriate test, for example an ANOVA - for full details, see below.

If you wish to perform *post-hoc* analysis of any statistically significant result (by default, $p \leq 0.05$) then, for parametric data, you can select a suitable test from the *Post-hoc test* drop-down list shown on the statistics section's settings page. If you choose to use Dunnett's test, then you will also need to select the control group for the post-hoc comparisons. For non-parametric data, ANY-maze will automatically select a post-hoc test for you - see table 5 below for details.

Details

ANY-maze includes a number of different statistical tests which it can use to analyse your data. As the system already knows certain things about the variables to analyse, such as whether they represent nominal data, whether there are repeated measures, etc., it is able to choose an appropriate parametric or non-parametric test for itself - full details are given below:

Analysis of nominal data

ANY-maze can perform one-way (but not two-way) analysis of nominal data, such as the first zone entered in a test. To do this, it uses the tests shown in table 1. If the result of this analysis is significant (by default, $p \leq 0.05$) and you've requested *post-hoc* analysis, then the system will employ the tests shown in table 2.

Number of levels of independent variable	Number of levels of dependent variable	Independent samples?	Test
2	2	Yes	Fisher exact test
		No	Cochran Q*
	> 2	Yes	Contingency table
		No	Cochran Q
>2	2	Yes	Contingency table
		No	Cochran Q
	>2	Yes	Contingency table
		No	Not supported

Table 1. Tests used to perform one-way analysis of nominal data in ANY-maze.

**Strictly speaking, this should be analysed using the McNemar test, but the Cochran Q test yields the same answer.*

Number of levels of dependent variable	Independent samples?	Ns are equal	Test
2	Yes		Not supported
	No	Yes	Multi comparisons of data subjected to the Cochran Q test (Marascuilo & McSweeney)
		No	Not supported
> 2	Yes		Not supported
	No		Not supported

Table 2. Tests used to perform post-hoc analysis of nominal data in ANY-maze.

Analysis of continuous data

ANY-maze can perform one-way parametric or non-parametric analysis of continuous data using the tests shown in table 3. For two-way analysis, the tests used are shown in table 4. If the result of this analysis is significant (by default, $p \leq 0.05$) and you've requested *post-hoc* analysis, then the system will employ the tests shown in table 5.

Parametric data?	Number of levels of independent variables	Independent samples?	Test
Parametric	2	Yes	t-test
		No	Paired sample t-test
	>2	Yes	Between subjects ANOVA
		No	Within subjects ANOVA
Non-parametric	2	Yes	Mann Whitney U
		No	Wilcoxon
	>2	Yes	Kruskal Wallis
		No	Friedman

Table 3. One-way parametric and non-parametric tests used to analyse continuous data in ANY-maze.

Parametric	Independent samples?	Test
Parametric	Both IVs independent	2-way between subjects ANOVA
	One IV independent	2-way between/within ANOVA
	Neither IV independent	2-way within subjects ANOVA
Non-parametric	Both IVs independent	2-way Kruskal-Wallis
	Either IV not independent	Not supported

Table 4. Two-way parametric and non-parametric tests used to analyse continuous data in ANY-maze.

Parametric	Comparison to control	Independent samples?	Ns are equal	Test
Parametric	Yes			Dunnett
	No			Bonferroni, Duncan, Fisher LSD, Scheffe, Sidak, SNK, Tukey
Non-parametric	Yes	Yes	Yes	Not supported
			No	Not supported
		No	Yes	Not supported
			No	Not supported
	No	Yes	Yes	Non-parametric analog of the SNK test
			No	Dunn
		No	Yes	Non-parametric analog of the Tukey test for repeated measures
			No	Not supported

Table 5. Post-hoc parametric and non-parametric tests used to analyse continuous data in ANY-maze.

Details as to whether a measure is nominal or continuous can be found under the measure's notes in the following topics:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures

- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Set optional information you want shown

In brief

Statistics sections always show groups' names, Ns, means, and raw data as well as the results of the statistical analysis. If you wish, you can also choose to include the groups' standard deviations and/or standard errors, by selecting appropriate options on the statistics section's settings page.

You can also specify that the raw data shown is labelled with animal and trial numbers - see figure 1. In fact this is extremely useful, because the animal and trial number labels are *links* to the relevant Animal details and Test details reports, respectively.

35.7 (8,1)

Figure 1. Raw data can be labelled with animal and trial numbers. In this case, the value is for animal 8, trial 1.

Setting up a track plot section

Introduction

For a general introduction to report sections, see Adding sections to a report.

Track plot sections show groups of track plots or heat maps for the tests in an experiment. This can act as a good, although subjective, way to visualise an experiment's results.

To add a track plot section to a results report, click the *Add item* button in the ribbon bar and select *New report section* from the menu which appears. On the sub-menu which is then shown, select *New track plot section*.

What you need to do

Select the type of plot you want to show

Use the *Type of plots* drop-down list to specify whether you want track plots or heat maps, and whether you want a separate plot for each animal, or heat maps of the group means. If you've turned on head tracking, you'll also be able to plot the track or heat map of the heads of the animals, as well as their centre points.

Select the way you want the track plots to be grouped

Simply use the three drop-down lists to select how the track plots should be grouped in the report. For example, selecting *Trial* and *Treatment* would cause all the track plots for the first trial to be shown grouped by treatment. Then all the track plots for the second trial would be shown, also grouped by treatment; then the third trial, etc.

You must select at least one grouping measure.

See also:

- Limiting the data analysed in a report
- Adding elements to a protocol
- Editing a report
- Deleting a report

ANY-maze help topic H0659

Editing a results report

Introduction

You can edit any aspect of a results report at any time - there are no restrictions.

See also:

- Editing the elements of a protocol
- Setting up a results report
- Adding a section to a report
- Deleting a results report

Deleting a results report

Introduction

You can delete either an entire results report or an individual section at any time, whether before, during or after an experiment. To do this, simply select the report or section in the protocol list and click the ✕ *Remove item* button in the ribbon bar.

Restrictions

There are no restrictions on deleting results reports, or their sections. However, if you delete all a report's *sections* without deleting the report itself, then the report will not be available on the results page - it'll be as if it doesn't exist at all.

See also:

- Deleting protocol elements
- Editing a results report

Setting up a Report set

Introduction

For a general introduction to report sets, see the *Report sets* section of the An introduction to results, reports & data topic.

To add a report set to a protocol, click the ✚ *Add item* button in the ribbon bar and select *New report set* from the menu which appears.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter the report set's name.
2. Specify the folder which the report set should be written to
3. Specify the reports to include in the report set

Editing a report set

You can edit any aspect of a report set at any time. If you want the changes you make to actually affect any published data, then you will need to re-publish the report set.

Deleting a report set

You can delete a report set from a protocol at any time - just select the report set in the protocol list and then click the ✕ *Remove item* button in the ribbon bar.

Note that deleting a report set from a protocol won't delete any data you've already published - if you wish to delete this, you will need to go to the publication folder and delete it manually.

See also:

- Adding elements to a protocol
- Editing the elements of a protocol
- Deleting elements from a protocol

Specifying a report set's name

In brief

In the *Report set name* field, enter a name for the report set. You must make an entry, but it can be anything you like.

Details

Report set names should specify what's included in the report set - for example, 'Plusmaze experiment details and results' or 'Plusmaze experiment track plots'.

Limits

You can enter anything you like up to a maximum of 64 characters.

Specifying the folder to write a report set to

In brief

In the field titled *Enter or select the folder ...*, enter the full path for the folder where you want the report set to be written. You must make an entry and the folder must already exist.

If you prefer, you can click the button to choose the folder using the Windows explorer. This also allows you to create a new folder if this is necessary.

Details

You can specify any folder, either on your computer or on your network, as the folder to write the report set to. However, if at all possible, it's best to use a folder on a network file server, as this will mean that other users will be able to access the folder - making the report set more widely available. If you want to restrict access to just a certain group of users, then you should be able to do this by setting the folder's *permissions* - ask your network administrator for advice.

There are two other benefits of not storing report sets on the ANY-maze tracking computer. Firstly, the report sets folder can, with time, become quite large, and on an older computer may fill the hard disk - this is less likely to be a problem on a file server. Secondly, if other users access the report sets folder via the network while tests are actually running, then there will be less processor time available for ANY-maze tracking. This won't prevent the tracking from working, but may mean that ANY-maze detects fewer positions per second.

Limits

You can enter any correctly formatted folder name (including UNC names) up to a maximum of 256 characters.

Specify the reports to include in a report set

In brief

Use the list of reports to choose those you want included in the report set. You should also specify whether you want any linked reports to be included in the set.

Details

In general, you will probably want to include at least one Results report in a report set. It's also a good idea to include the Test schedule and Protocol reports, as these will help to provide a fuller picture of the experiment than just the results report alone.

Including linked reports

As you probably know, reports such as the Test schedule report can include links to other, related, reports such as the Animal details and Test details reports. Clearly then, if you choose to include a report which contains such links, it would be useful if the linked reports were included too - this would then allow someone using the report set to jump directly to the linked reports just like you can in ANY-maze.

You can control whether a report set should include such linked reports using the five check boxes shown on the report set's settings pane. Specifically, you can choose whether to include linked Animal details reports, Test details reports, Test track plot reports, Test heat map reports, and Test technical details reports.

Unless you're limited to the amount of space available for storing report sets, then I would suggest that you include all these linked reports in any report sets you create.

Reports which contain graphics

If a report set includes reports which contain lots of graphics - graphs or track plots, for example - then the set may be quite large and could take a few minutes to produce. While this isn't necessarily a problem, it is something to be aware of.

See also:

- An introduction to results, reports & data
- Setting up Results reports
- The Test schedule report

- The Animal details report
- The Test details report
- The Test technical details report
- The Test track plot report
- The Test heat map report
- The protocol report

Information measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze includes the following information measures for each test:

- Test number
- Animal number
- Treatment
- Treatment code
- Stage
- Trial number
- Apparatus
- The reason for test end
- Test date
- Test day of the week
- Test time
- Test time of day
- User
- Test notes
- Animal notes
- Animal lighter/darker than apparatus
- Video file
- Time period
- Segment of test
- Location of moveable zones

Test number

Description	A sequential number assigned to each test when it's completed.
Independent variable	No

Dependent variable No

Notes None

Animal number

Description The animal number of the animal this test was performed on.

Independent variable No

Dependent variable No

Notes Animal numbers are assigned sequentially to animals as they're added to an experiment. These numbers are never reissued, even if an animal is removed from the experiment. For example, if you create an experiment with 10 animals in it, they'll be numbered 1-10. If you immediately delete the 10 animals and then add 10 new animals, the new ones will be numbered 11-20 and the experiment won't have any animals numbered 1-10 at all.

Treatment

Description The name of the treatment given to the animal this test was performed on.

Independent variable Yes

Dependent variable No

Notes None

Treatment code

Description The code of the treatment given to the animal this test was performed on.

Independent variable Yes

Dependent variable No

Notes None

Stage

Description The name of the stage which this test was part of.

Independent variable Yes

Dependent variable No

Notes	All tests must be part of a stage - even if an experiment only contains a single default stage.

Trial number

Description	The trial number within the stage for the animal which this test was performed on.
Independent variable	Yes - see notes.
Dependent variable	No
Notes	Within a stage, an animal can be tested more than once - i.e. have repeated trials. The first test in a stage is the animal's trial 1, the second test is the animal's trial 2, etc. Although the trial numbers will be repeated in different stages, ANY-maze still views them as different when using Trial number as an independent variable. Thus, for example, Trial 1 in an *Acquisition* stage would be seen as different to trial 1 in a *Re-test* stage.

Apparatus

Description	The name of the apparatus that the test was performed on.
Independent variable	Yes
Dependent variable	No
Notes	None

The reason for test end

Description	The reason the test ended. This is either a description of a standard reason (see notes), the name of a procedure that ended the test, or a test end reason created for a procedure. *If your protocol is set up to use Events and Actions rather than procedures, then the test end reason can be the name of the event that triggered an action that ended the test (the name of the event is used because this reflects the reason why the test ended, for example, 'Island entry').*
Independent variable	Yes
Dependent variable	Yes. Nominal data. Can't be analysed across time.
Notes	The standard reasons for ANY-maze to end a test are:

User ended test	The user ended the test by clicking the *Stop test* button.
Test duration	The test ended because the test duration was reached.
End of video	This reason will be used if you're using a video file for the test (rather than tracking the animal 'live'), and the end of this video file is reached.
Lost video signal	The video signal showing the apparatus was lost (for example, the camera was unplugged).
Lost I/O device	An I/O device used in the experiment was lost (i.e. unplugged from the ANY-maze computer).
Out of memory	While running a test, ANY-maze records the test results in the computer's memory - if this memory becomes full, then the test will be ended for this reason. This is very unlikely to occur.
Error saving results	For some reason, ANY-maze is unable to save the results to the experiment file. This is very unlikely to occur.

Test date

Description	The date when the test was *started.*
Independent variable	Yes
Dependent variable	No
Notes	None

Test day of the week

Description	The day of the week (Monday-Sunday) when the test *started.*
Independent variable	Yes
Dependent variable	Yes. Nominal data. Can't be analysed across time.
Notes	None

Test time

Description	The time when the test *started.*
Independent variable	No

Dependent variable	No
Notes	None

Test time of day

Description	The time of day, am or pm, when the test *started.*
Independent variable	Yes
Dependent variable	Yes. Nominal data. Can't be analysed across time.
Notes	None

User

Description	The name of the user who was logged on while the test was performed.
Independent variable	No
Dependent variable	No
Notes	None

Test notes

Description	Any notes recorded for the individual test - see notes.
Independent variable	No
Dependent variable	No
Notes	Only the first 80 characters of the notes are actually used for this measure's value. The notes themselves can be up to 32,000 characters in length.

Animal notes

Description	Any notes recorded for the individual animal this test was performed on - see notes.
Independent variable	No
Dependent variable	No
Notes	Only the first 80 characters of the notes are actually used for this measure's value. The notes themselves can be up to 32,000 characters in length.

Animal lighter/darker than apparatus

Description	Whether the animal is lighter or darker than the apparatus background, as specified in the protocol list under Animal colour.
Independent variable	Yes
Dependent variable	Yes
Notes	None

Video file

Description	The name of the video file from which a test was run. The full path will be displayed on the Data page; the file name only will be shown on the test schedule report.
Independent variable	No
Dependent variable	No
Notes	None.

Time period

Description	A Time period of the test
Independent variable	Yes
Dependent variable	No
Notes	For more information about Time periods refer to An introduction to Time periods

Segment of test

Description	A segment of the test, described as x-y seconds; for example, 30-60 seconds.
Independent variable	Yes
Dependent variable	No
Notes	ANY-maze can break tests down into equal length segments and then perform analysis on them. The length of the segments that tests are broken into is specified in the protocol's Analysis across time element. ⚠ It's very important to understand that when you use the *segment of test* measure, you effectively tell ANY-maze to start analysing segments rather than

tests. This means, for example, that the Data page spreadsheet will include one row per test segment rather than one row per test.

Location of moveable zones

Description	The location of a moveable zone in the test
Independent variable	Yes
Dependent variable	Yes
Notes	There is one measure for each moveable zone in the protocol. You'll find more information about moveable zone here.

See also:

- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Adding elements to a Protocol

In brief

Adding an element to a protocol is very simple, just click the *Add item* button in the ribbon bar and then select the type of element you want to add from the menu which appears.

Details

As well as adding a protocol element using the *Add item* button you can also add one by clicking the right-hand mouse button somewhere in the protocol list and selecting an appropriate option from the menu which appears.

In fact there's even a third way to add an element. If you press the 'Insert' key on the keyboard when a protocol element is selected then a new element of the same type will be added.

You may find that some of the elements on the *Add item* button menu are disabled. This can occur for two reasons:

- When the relevant item requires other elements which aren't present in the protocol yet. For example, you can't add a zone until you've added at least one apparatus item. This is because a zone defines areas within your apparatus but if you have no apparatus clearly you can't have any zones.
- When a test is being performed, you can't add items which would affect the running test. For example, you can't add a new zone while a test is running.

The best strategy when building a protocol is simply to work down the elements shown in the protocol list in the order they're listed.

You can add new elements to a protocol **at any time**, even after an experiment is finished. This is extremely useful as it means you can add new elements to analyse things you hadn't even though of when you designed the protocol. For example, you might create a plusmaze protocol with Open arm and Closed arm zones. You then run an experiment and analyse your results. Later you read an article which reports results for the ends of the open arms and you wonder whether your results would be the same. To find out, you could simply open the existing experiment, add a new 'Ends of open arm' zone and analyse the results!

See also:

- The elements of a protocol
- Editing the elements of a protocol
- Deleting elements from a protocol

Editing elements of a Protocol

In brief

Editing a protocol element is very simple - just select the required element in the protocol list and its current settings will appear in the *Settings* and *Image* panes where you can edit them.

Details

You can edit all aspects of a protocol at almost any time, either before you start the experiment, while you're running it or after you've finished. This can be extremely useful because it not only means that you can always remedy mistakes, but also that you can change the protocol to analyse things you hadn't even though of when you created it.

The only restriction to editing the protocol is that you can't edit some elements *while* a test is actually being performed. This only applies to things which would affect the running test - for example, you can't alter the definition of a zone while a test is being performed.

See also:

- The elements of a protocol
- Adding elements to a Protocol
- Deleting elements from a protocol

Deleting elements of a Protocol

In brief

Deleting a protocol element is very simple - just select the required element in the protocol list and then click the ✕ *Remove item* button in the ribbon bar.

Details

You can also delete an element by right-clicking on it with the mouse and selecting *Delete* from the menu which appears.

In general you can delete almost all the elements of a protocol at any time, although there are some restrictions.

- You can't delete elements which ANY-maze automatically includes in the protocol, such as the Analysis element.
- You can't delete *all* the stages in a protocol - a protocol must always include at least one stage.
- You can't delete a stage which has actually been started - i.e. there's at least one animal that's had at least one trial in the stage.
- You can't delete an apparatus element if at least one test has been performed in the apparatus.
- You can't delete a video source which is used by apparatus in which at least one test has been performed.
- You can't delete elements which have a direct effect on the performance of tests *while* a test is being performed. For example, you can't delete a zone while a test is running.

None of these restrictions should cause you any problems because you can just ignore the elements if you want to.

If your intention in trying to delete an element is to delete the tests related to it, for example, you're trying to delete all the tests in a stage then you should Cancel the tests or Retire or Delete the animals instead.

See also:

- The elements of a protocol
- Adding elements to a Protocol
- Editing elements of a protocol

ANY-maze help topic H0084

Saving and loading Protocols

Introduction

One of the most useful aspects of protocols is that you can save them to a file and then create new experiments based on the saved protocol. For example, let's imagine that you have a plusmaze and a water-maze in your lab. You could easily create two protocols one for each type of apparatus and then simply load the relevant protocol whenever you want to perform an experiment.

- Building a library of protocols
- Saving a protocol
- Loading a protocol
- Editing a saved protocol

Building a *library* of protocols

You can think of saved protocols a little like books in a library. When you want to perform an experiment you go to the library and take out the relevant book. However, rather than removing the book itself, you take a *photocopy* of it. This has the advantage that you can annotate the copy without changing the original in the library. Of course, you might decide that your annotations are so useful that your copy should be added to the library as a new 'original'.

This is exactly how saved protocols work. When you want to create a new experiment based on a protocol you simply load the appropriate protocol into your experiment file. This *copies* the information from the protocol into your file which means that, should you want to, you can alter the protocol in the experiment without altering the original protocol file. Also, just like for the book, you can save an altered protocol thus 'adding' to the library - see figure 1.

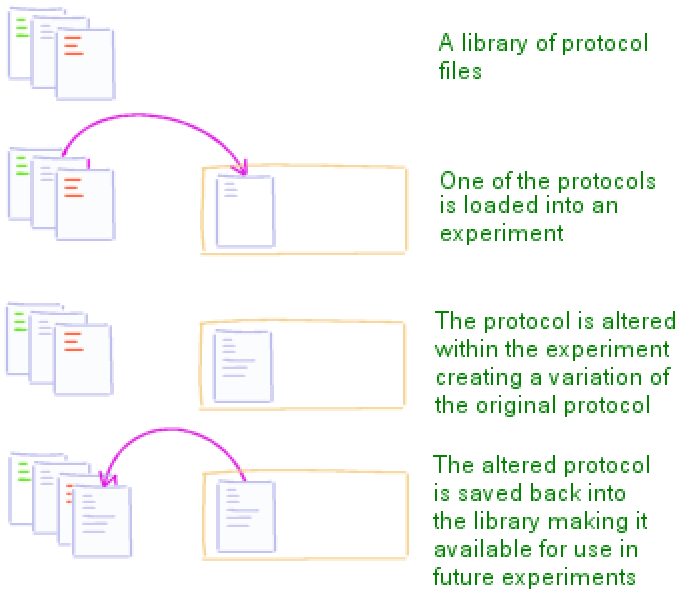


Figure 1. When you load a protocol into an experiment, you load a copy - alterations don't affect the original. If you want to you can save alterations to a new protocol file so the updated protocol can be used in other experiments.

Saving a protocol

It's important to understand that you don't *have* to save protocols. Normally an experiment's protocol is stored inside the experiment's file and is saved whenever you save your experiment.

What you do when you specifically '*save*' a protocol is you extract just the protocol from the experiment and save it on its own. This *saved* protocol isn't actually any use until it's loaded back into an experiment - see figure 2.

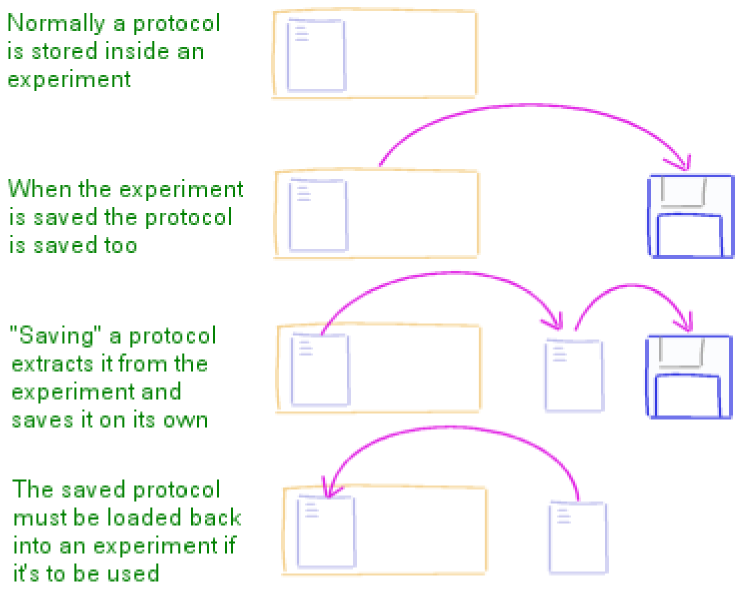


Figure 2. Ordinarily a protocol is saved within an experiment's file. If you save a protocol on its own then it must be loaded into an experiment before it can be used.

To actually save a protocol you should click the *Save protocol* button in the ribbon bar - this will open the Save protocol window where you can specify a name for the protocol file and select where the file should be saved. In fact, ANY-maze will automatically suggest the protocol's name as the file name, although you can change this if you want to.

Loading a protocol

Loading a protocol into an experiment will replace any existing protocol that the experiment contains. Because this could have serious repercussions on the results of any tests, the system won't let you load a protocol into an experiment which has actually been started, i.e. one which includes at least one performed test. However, this is really a minor restriction because you'll most often want to load a protocol into a brand new experiment.

To load a protocol you should click the *Load protocol* button shown in the ribbon bar - this will open the Load protocol window where you can choose the protocol file you want to load.

In fact, as you'll most often want to load a protocol into a new experiment ANY-maze includes some short-cut methods to create a new experiment *and* load a protocol directly into it.

- When you start ANY-maze, or when you select the New experiment option on the File page, ANY-maze shows a list of recently used protocols as a series of 'documents' (see figure 3), clicking any of these documents will create a new experiment and load the selected protocol into it.

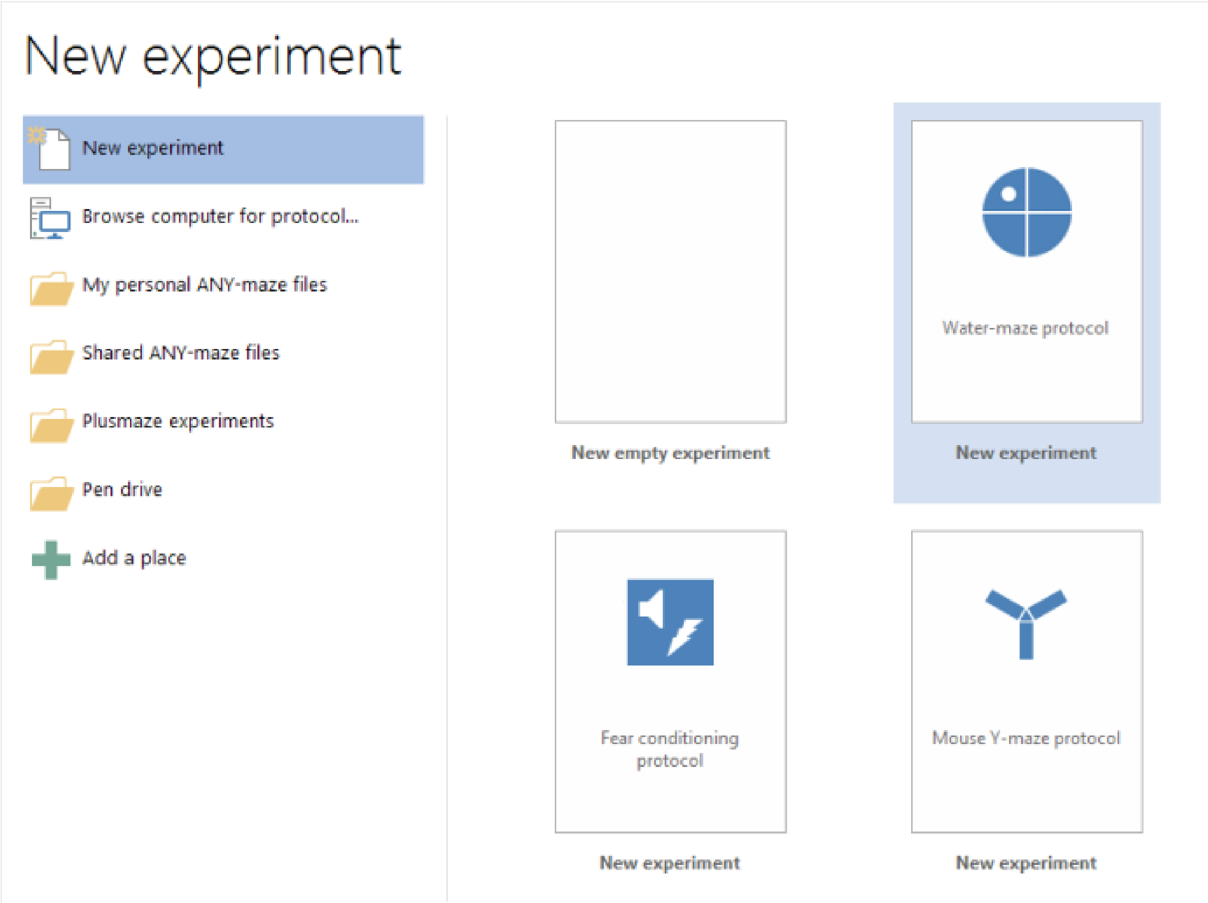


Figure 3. Selecting a protocol document will create a new experiment and load the protocol into it.

- The list of protocol 'documents' just includes recently accessed protocols, if you want to use a protocol that's not listed then you can select the *Browse computer...* option (see figure 3) which will open the Load protocol window, where you can navigate to any protocol file on your computer or network. When you select a file ANY-maze will create a new experiment and load the protocol into it.
- Another short-cut way of creating a new experiment with a pre-loaded protocol is to double click a protocol file in the Window's explorer. This will open ANY-maze, create a new experiment and load the protocol into it - see figure 4.

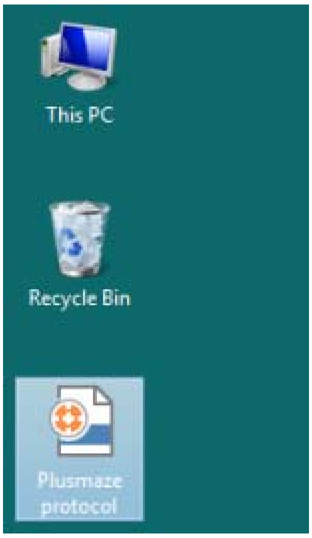


Figure 4. Double clicking a protocol file is a convenient way to open ANY-maze, create a new experiment and load a protocol - all with one action.

In fact it's not strictly necessary to save a protocol to a protocol file in order to load it into an experiment - you can actually load one experiment's protocol directly into another experiment. To do this click the *Load protocol* button in the ribbon bar and in the *Type of file* list in the window which opens select *ANY-maze experiment files* - you'll then be able to select an experiment to load the protocol from. Doing this has no effect on the experiment from which the protocol is loaded.

Editing a saved protocol

Having saved a protocol to a protocol file you may find that you need to edit it. For example, you might need to correct a mistake, or you may wish to add something such as a new zone or calculation.

Editing a protocol file is very simple:

1. Load the protocol - it's usually easiest to just load it into an empty experiment.
2. Make the edits
3. Save the protocol back to *the same file* you loaded it from. This will overwrite the old version with the new one.

See also:

- The Save protocol window
- The Load protocol window

The Save protocol window

Introduction

Use this window to name a protocol file and choose where to save it. You can also elect to protect the file by entering a password.

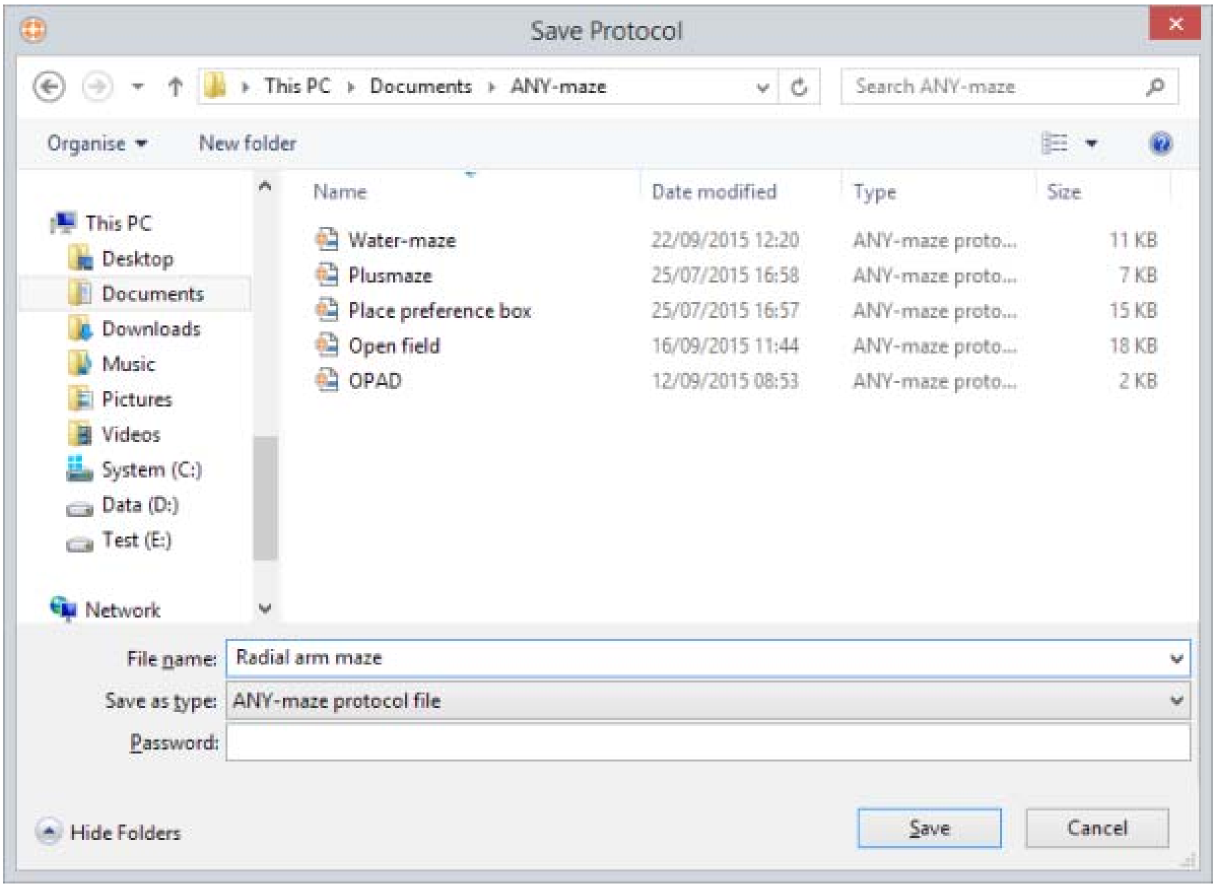


Figure 1. The Save Protocol window.

Details

You will probably be familiar with this window already as it's based on the standard 'Save file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list	The files list shows the folders and protocol files stored in the current location. You can double click a folder to open it and you can click a protocol file to transfer its name to the File name field.
File name	Use this field to enter a name for the protocol file. ANY-maze will automatically enter the protocol's name into this field (assuming you've given it a name) but you can delete this and type in something different if you like. Note: File names can be up 255 characters long but cannot contain any of the following characters: \ / : * ? " < > |.
Type of file	In fact, there is only one choice here, ANY-maze protocol file, so you can ignore this list.
Password	You can use this field to enter a password for this protocol file. Anyone who tries to load the protocol into an experiment will need to enter the password first. In fact, if you enter a password ANY-maze will also encrypt the file so it can't be read by any other software. However, be careful, if you forget the password you won't be able to load the file. If you've chosen to protect your files by default - see the Preferences : My account topic, then your password will already be filled in here - if you don't want to protect this particular file then simply delete the entry so the field is empty.

See also:

- Saving and loading Protocols
- Securing your ANY-maze system and data

The Load protocol window

Introduction

Use this window to select the file which contains the protocol you want to load.

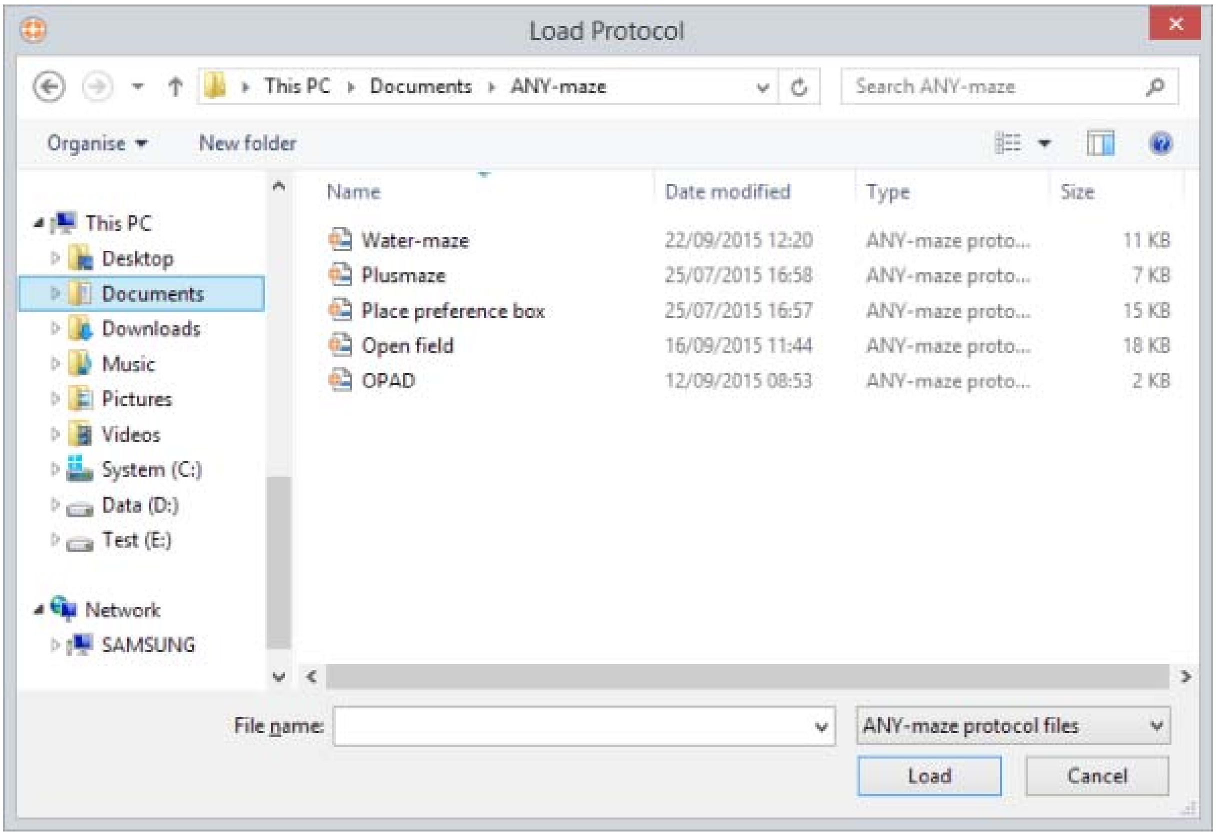


Figure 1. The Load Protocol window.

Details

You will probably be familiar with this window already as it's based on the standard 'Open file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list	The files list shows the folders and protocol files stored in the current location. You can double click a folder to open it or double click a file to load it.
File name	Use this field to enter the name of the protocol file to load. Although you can do this it's usually much easier to simply double click a file in the file list.
Type of file	This determines what types of file are displayed in the file list. It starts off set to 'ANY-maze protocol files' as you will usually want to load a protocol from a *protocol file*. However, you can in fact load a protocol directly from an *experiment file,* which is very useful if the protocol's yet not been saved to a protocol file. If you want to do this then first select 'ANY-maze experiment files' in this list, which will cause the file list to show experiment, rather than protocol, files and then choose the appropriate experiment file from those shown.

See also:

- Saving and loading Protocols

The Protocol report

Introduction

The protocol report describes all aspects of a protocol, and is particularly useful for checking a complex protocol when you've finished creating it. To open the report, simply click the *View protocol report* button in the ribbon bar.

Details

When you open the protocol report it will fill the Protocol page, hiding the Protocol list, etc. - to return to the normal view just click the *Edit protocol* button in the ribbon bar.

You can print the report, copy it to the Windows clipboard, save it in a variety of file formats or send it via e-mail. In fact, you can perform these operations either on the entire report or just a selection - refer to the following topics for more details:

- Printing reports and data
- Saving reports and data
- Copying reports and data
- Sending reports and data by e-mail

The protocol report can also be included as part of a report set. This is usually a good idea as it means anyone who accesses the report set can see how the experiment was performed. Refer to the Report sets topic for further details.

The Experiment page

Overview

The Experiment page in ANY-maze is used to enter data specific to an *individual* experiment. Remember that most of the details that ANY-maze requires in order to perform an experiment will be defined in the protocol and that you can use the same protocol in lots of different experiments.

Specifically, the experiment page is used to enter a title for the experiment, some optional notes and, most importantly, to set up the experiment's *treatment groups* and *animals*.

You'll find full details in the following topics:

- Entering general experiment details
- Setting up treatment groups and animals

See also:

- The Protocol page
- Understanding the relationship between the Treatment groups spreadsheet and the Animals spreadsheet

Entering general experiment details

Introduction

The ANY-maze *Experiment page* is divided into two panes, with the left one being used to record general details about the experiment.

In fact these details usually only consist of the experiment's title and some notes and you don't *have* to enter either if you don't want to. (If you are using ANY-maze with an Ugo Basile Operon then the details may also include some Operon experiment settings.)

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Optionally enter the experiment's title
2. Optionally enter some experiment notes
3. If relevant, enter the Operon experiment settings

Altering this information

You can edit or delete the experiment title and notes at any time - such changes have no repercussions. However, edits to the Operon experiment settings do include some restrictions.

What next?

After completing these steps you should enter details about the experiment's treatment groups.

Experiment title

In brief

In the *Experiment title* field shown on the Experiment page you can optionally enter a title for the experiment.

Details

You can enter anything you like for the experiment's title, although it's obviously a good idea to enter something which will uniquely identify the experiment. The information you enter here is used in the following ways.

- When you save the experiment, ANY-maze will use the experiment title as the *default* file name. Of course, as this is only a default, you can change it if you want to - see Saving experiments for more details.
- You can include the experiment title as part of the header or footer on printed documents - see Printing options for more details.
- Experiment titles are shown on the index page of report sets - for more details see the 'Report sets' section of the Introduction to results, reports & data topic.

Limits

You can enter anything you like up to a maximum of 60 characters.

Experiment notes

In brief

You can enter any notes that relate to the experiment as a whole, in the *Experiment notes* field.

Details

Although you can enter anything you like here, it is usually best to record notes which refer to *specific animals* or *specific tests*, on the relevant Animal details or Test details reports.

Experiment notes are not actually used anywhere else in ANY-maze, although they do appear on the index page of report sets - for more details see the 'Report sets' section of the Introduction to results, reports & data topic.

Formatting

When the experiment notes field is active (i.e. the cursor is in that field) then the *Experiment notes format* section of the ribbon bar will be enabled. This contains a number of options, which apply to any text that is currently selected in the notes field. (If no text is currently selected, then the formatting you select will apply to new text when you start typing).

Reset formatting	Resets the formatting of the selected text to the default formatting (i.e. removes any colour, bold, italic etc.)
Text colour	Opens a dialog box allowing you to select a colour for the selected text.
Bold	Makes the selected text **bold.**
Italic	Makes the selected text *italic.*
Underline	Makes the selected text underlined.
Increase text size	Increases the size of the selected text by one point size.
Decrease text size	Decreases the size of the selected text by one point size.

Limits

There are no limits to what you can enter.

ANY-maze help topic H0670

Entering and editing the Operon experiment settings

If when adding an Ugo Basile Operon to your protocol you opted to allow ANY-maze to add the relevant and irrelevant dimension options, then the Experiment page will contain some extra Operon-specific options. These allow you to do the following:

Choose whether you want to perform a 2D or 3D experiment

The difference between a 2D and 3D experiment simply defines whether the animal will use two or three dimensions (light, texture and odour) throughout their test. The following options are available:

2D experiment using two dimensions	This requires you to select the two dimensions that you wish to use. When you have done this, each animal will be randomized to start with one of these dimensions as relevant, and the other as irrelevant. This option is useful if, for example, you do not have the Olfactory add-on and must only work with texture and light. The extra-dimensional shift (EDS) stages of your experiment will simply switch the relevant and irrelevant dimensions.
2D experiment using all three dimensions	This will use all three dimensions of texture, odour and light, but each animal will only use two of them throughout the experiment. The extra-dimensional shift (EDS) stages of your experiment will simply switch the relevant and irrelevant dimensions.
3D experiment	This will use all three dimensions, and the animals will be randomized to start with one of the three dimensions as relevant. The extra-dimensional shift (EDS) stages at the end of the experiment will use the *third* dimension (i.e. the one not yet used) as the new relevant dimension, and will make the previously-relevant dimension into the irrelevant dimension.

Select which dimensions to use in your tests

For a 2D experiment, you simply need to select which two dimensions to use in the experiment.

If you are using the Operon without *the Olfactory add-on, then you will need to select* Light and Texture *from the list - you will obviously not be able to perform any experiments involving odour.*

Select how you would like to randomize the animals

When you select an experiment that uses all three dimensions (a 3D experiment, or a 2D experiment that will use three dimensions), you will be given a choice of how to randomize your animals between these dimensions. You can opt for ANY-maze to randomize these completely for both relevant and irrelevant dimensions, or you can specify which dimension is 'irrelevant' for each choice of relevant dimension.

Limitations

1. Because ANY-maze needs to randomize the animals you add between treatment groups and dimensions, you can't add any animals or treatments to your experiment until you've told ANY-maze how to do this randomization. ANY-maze will prevent you adding animals or treatments until you've completed these entries in the Operon-specific controls.
2. When you add some animals to your experiment, they will be randomised between the treatment groups and dimensions in the way specified by the setting you make in the Operon-specific controls. After this **you can no longer alter the Operon experiment settings** (otherwise you could have some animals randomised between one set of dimensions and others randomised between a different set) and the Operon-specific controls are therefore shown as disabled.
3. When animals are added to your experiment, ANY-maze will randomize the animals between the relevant dimensions, and also between treatment groups. This randomization will attempt to balance the groups as equally as possible. If you then add more animals later, these animals will be balanced *amongst themselves* - i.e. the existing animals will be assumed to be already randomized. This means that if you edit the dimensions or treatments that ANY-maze has already entered, you could end up with animals that are not equally balanced between the treatment groups and dimensions.

Setting up treatment groups and animals

Introduction

In most behavioural experiments the animals being tested are organised into different groups, usually based on the treatments they'll receive. As this type of arrangement is so common, ANY-maze uses it as the *default* basis for all experiments.

- Entering treatment groups
- Treatment group coding
- Assigning animals to the treatment groups
- Unassigned animals
- Editing and deleting treatment groups

However, animals don't *have* to be organised into treatment groups:

- Choosing not to use treatment groups
- Adding animals to an experiment which doesn't use treatment groups

You also have the option of importing animals (and tests) from some other system, for example Excel. This can be very helpful if you already have this information in another program, as it avoids having to re-enter everything in ANY-maze. In this case you can still choose whether or not to place animals into treatment groups and you also have the option to import data and then add to it; for example, you could import a list of animals and then place them into treatment groups once they're in ANY-maze.

- Importing animals
- Importing tests

See also:

- Understanding the relationship between the Treatment groups spreadsheet and the Animals spreadsheet

The Treatment groups spreadsheet

Introduction

The Treatment groups spreadsheet is used to set up the treatment groups for an experiment. To access the spreadsheet you should switch to the Experiment page and select the *View treatments* button - the Treatment groups spreadsheet will then be shown on the right-hand side of the page - see figure 1.

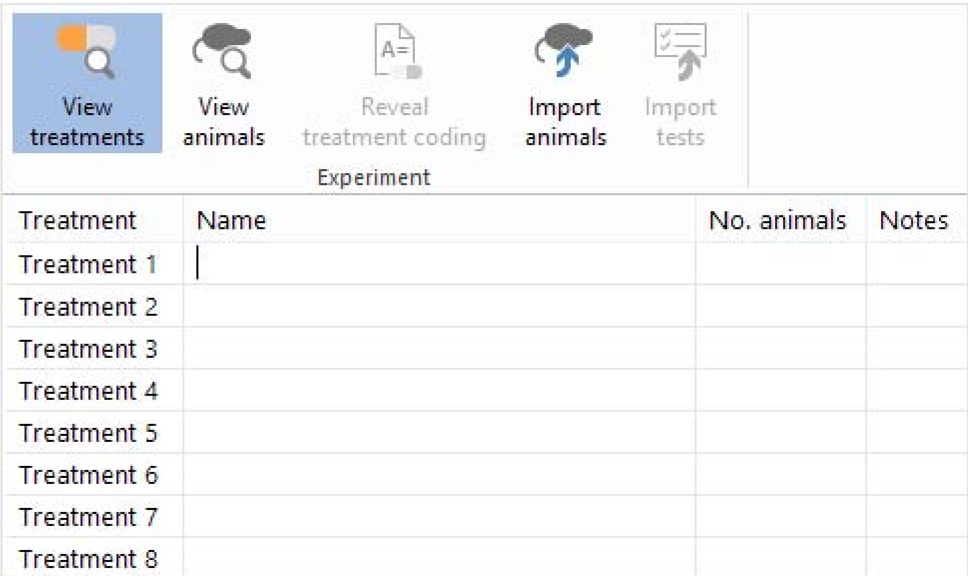


Figure 1. The Treatment groups spreadsheet, shown when the View treatments *button is selected.*

Entering treatment groups is very simple: for each group you just need to type in its name and the number of animals it contains. You can enter up to 20 different groups and each one can contain up to 999 animals.

What you need to do

Each of the steps below is a link, which you can click for more information.

1. Enter a treatment group name
2. Enter the number of animals in the group

3. If necessary, enter the treatment dose
4. If necessary, enter the treatment concentration

The Notes column isn't editable; ANY-maze will use this to give you relevant information about the treatment, for example the number of deleted or retired animals which were allocated to that treatment.

Altering this information

For full details about editing or deleting treatment groups refer to the Editing and deleting treatment groups and animals topic.

See also:

- Treatment group coding
- Assigning animals to the treatment groups
- Unassigned animals
- Editing and deleting treatment groups
- The Animals spreadsheet
- Understanding the relationship between the Treatment groups spreadsheet and the Animals spreadsheet

Treatment group name

In brief

You can enter anything you like as the name of a treatment group although all the names must be unique - ANY-maze will tell you if they're not.

Details

If you don't actually want to group the animals in your experiment, then you can switch off treatment groups and simply enter ungrouped animals instead.

If your groups aren't related to different treatments, perhaps they relate to the age of the animals, then you may wish to redefine the term 'treatment' - in this case to make it *Age group*. Thus you would enter the various age groups here, rather than treatment groups.

Note that you can use fields to record more 'grouping' details about animals. For example, you could create a field called 'Gender' and then group animals into males and females. This *grouping* is most relevant when performing statistical analysis of your results when you can use such groups as independent variables. Thus, for example, you could perform a 2 way ANOVA for treatment and gender.

Limits

You can enter anything you like as a treatment group name up to a maximum of 60 characters.

Number of animals in a treatment group

In brief

You should enter the number of animals you want in each treatment group in the column to the right of the treatment group name.

Details

You can, in fact, create a group with no animals in it at all (by leaving the number field blank), but this is only really useful if you intend to manually assign animals to their treatment groups.

Limits

You can enter any number up to 999.

Treatment dose

In brief

If the experiment's protocol specifies that ANY-maze should calculate the treatment dose for the animals, then the Treatment groups spreadsheet will include a *Dose* column. For each treatment group you should enter the dose in milligrams per kilogram (of animal weight).

Details

It's quite likely that you will base your treatment group names on the dose, for example, 'Drug X 1.0mg/kg' or 'Drug X 5.0mg/kg'. You should, however, still enter the dose in the dose column - i.e. 1.0 and 5.0 in this example.

Some treatments, such as a control, won't have a dose; in this case you should enter a dose of 0.

Limits

You can enter any number from 0 to 99,999.999

See also:

- Treatment concentration

Treatment concentration

In brief

An experiment's protocol can specify that ANY-maze should calculate the treatment dose for the animals in the experiment. This yields the mass of the treatment to be administered to each animal. However, in many cases you will administer the treatment as a solution and you can therefore setup the protocol so that ANY-maze will calculate the volume of the treatment that should be administered (such that the animal will receive the correct treatment mass). To work out the volume, ANY-maze needs to know the concentration of the treatment solution.

If the protocol is set up to calculate treatment volumes, then the Treatment groups spreadsheet will include a *Concentration* column - how you should use this column is described in the details section below

Details

The protocol actually allows you to specify the concentration in three ways:

- *The volume administered should be approximately the same for all treatment groups*
- *The concentration will be different for each treatment group*
- *The concentration will be the same for all treatment groups*

The volume administered should be approximately the same for all treatment groups

In this case ANY-maze will *calculate* the concentration for the different treatment groups and will report it in the concentration column. You should use this information to prepare appropriate solutions for each treatment.

The concentration will be different for each treatment group

In this case you should enter the treatment's concentration, in milligrams per litre, for each treatment group. In the case of a control group you should enter a concentration of 0.

The concentration will be the same for all treatment groups

In this case there won't actually be a concentration column in the Treatment groups spreadsheet, as the concentration will already have been specified in the protocol.

Limits

When specifying a concentration you can enter any number from 0 to 99,999.99

See also

- Treatment dose

The Animals spreadsheet

Tip - if you right click on the spreadsheet a menu will open which includes an option to specify whether pressing the 'Return' key on the keyboard should move to the next cell down or the next cell across in the spreadsheet.

Introduction

The Animals spreadsheet shows information about all the animals in your experiment in one place. It provides an efficient location to enter data about the animals and is particularly useful if you are manually assigning animals to their treatment groups.

To access the Animals spreadsheet you should switch to the Experiment page and select the *View animals* button, the spreadsheet will then be shown on the right-hand side of the page.

- Entering animal treatments
- Specifying whether the animals are lighter or darker than the apparatus
- Entering animal 'field' data
- Specifying the position of movable zones
- Setting animals' status
- Adding more animals
- Deleting animals

Entering animal treatments

To specify the animals' treatments you simply have to type in the relevant treatment code against each animal. If you want to see which treatment corresponds to which code, you can select the *Reveal treatment coding* button in the ribbon bar, although of course, you should carefully consider the repercussions of doing this if you are running your experiment blind.

You can edit, or delete, an animal's treatment codes at any time. If you delete a code then the relevant animal will simply be recorded as 'unassigned' - i.e. not part of any treatment group.

Specifying whether the animals are lighter or darker than the apparatus

One of the options in the Protocol specifies whether the animals are lighter or darker than the background of the apparatus. Normally you will be able to select one of the options and ANY-maze will apply it to **all** the animals in your experiment.

However, you might be working with different strains of animal and thus some may be lighter and some darker than the apparatus. In this situation you can specify this in the protocol, and ANY-maze will then include a column in the Animals spreadsheet for you to specify, for each animal, whether it is lighter or darker than the background. To specify the setting for an animal just type either 'L' or 'D' in the appropriate cell - ANY-maze will complete the entry with the word 'Lighter' or 'Darker'.

If this column doesn't appear in your spreadsheet, it simply means that the protocol specifies that all animals are either lighter or darker than the apparatus and therefore the system doesn't need to ask you about each animal individually.

Entering animal 'field' data

If the experiment protocol includes any fields for recording information about animals, the Animals spreadsheet will include one column for each of them.

For numeric fields, such as an animal's weight, you should enter the relevant value - the units will be shown in parentheses in the column's title.

For choice fields, such as an animal's sex, you should enter one of the field's choices. In fact ANY-maze will auto-complete entries as you type. For example, if the choices are 'Male' and 'Female' then as soon as you type 'M' the system will complete the entry with the word 'Male' because what you're typing doesn't match any other entry. This usually means that you only need to enter the first letter or two for choice field entries.

For text fields you can enter anything you like up to a limit of 80 characters. Remember that you can't use text fields in any analysis so whenever possible using numeric or choice fields is preferable.

Specifying the position of movable zones

If an experiment's protocol includes any movable zones (i.e. zones which can adopt different positions in different tests) **and** these zones are defined as only changing position *between* animals, then the spreadsheet will include a column for each of the zones.

For example, in a water-maze you might have an island which the animals have to learn how to find. To control against possible effects of island position, you decide to locate the island in different positions for different animals - of course, for an individual animal it will remain in the same place otherwise he'll never learn where it is! So, in this case, you might specify that for Animal 1 the island is in the North West, for Animal 2 it's in the South West, and so on.

You should enter the name of the zone position for each animal in the relevant zone columns. In the above example, you'd enter 'North West' for Animal 1 in the 'Island position' column. These columns will auto-complete so you probably won't need to type in the whole of the position's name.

If you leave a cell blank then the relevant zone's position will simply be *undefined* for the animal, you can always come back later and specify what it is.

Setting animals' status

The Animals spreadsheet always includes a column for the animals' status. Animals can have one of the following statuses which you can set by simply typing the status into this column:

Normal	The animal will be tested in all appropriate tests and its results will be included in all appropriate analysis.
Retired	The animal won't have any further tests performed. Any tests performed so far will be included in all appropriate analysis unless the analysis is specifically set to 'Exclude retired animals'.
Automatically retired	The animal has been retired from the experiment by the system (see Retired status). This occurs when an animal fails to satisfy some type of Stage end rule. You cannot alter this status manually - the only way to 'un-retire' an animal with this status is to alter the rule which caused it to be retired.
Deleted	The animal won't have any further tests performed. Any tests performed so far will be entirely ignored in all analysis.

Although you can change an animal's status here, it's usually easier to do this on the Animal details report. The only exception to this is when an animal which has received no tests is set to *Deleted* status - in this case the animal won't have an '*Animal details report*' and therefore the only way you can 'undelete' it is to change its status here.

The status column features auto-complete, so in fact you generally only need to type in the first letter of the status - the rest will be filled in automatically.

See also:

- The Treatment groups spreadsheet
- Understanding the relationship between the Treatment groups spreadsheet and the Animals spreadsheet

Treatment coding

Details

Although each treatment group you enter is given a name, ANY-maze will also give it a single letter code.

The reason ANY-maze uses these codes is to allow you to run experiments 'blind'. When performing a blind experiment ANY-maze will randomly code the treatments so the experimenter won't actually know which treatment is which. Furthermore, until the experiment is complete, the system will only refer to an individual animal's treatment by its code, again so the experimenter won't know which treatment group the animal belongs to.

Of course, someone will probably need to know which group is which, perhaps so they can label treatments with their codes. To see the codes for the treatments you simply need to select the *Reveal treatment coding* button. When viewing the Treatment groups spreadsheet, this will cause an extra column to be added to the spreadsheet showing each treatment's code; to hide the codes you just need to click the same button again. When viewing the Animals spreadsheet, this will cause both the treatment code and the treatment **name** to be shown for each animal.

See also:

- Assigning animals to the treatment groups
- Unassigned animals

Assigning animals to the treatment groups

Details

When you enter treatment groups the system will create records for the individual animals. In the example shown in figure 1 it will, unsurprisingly, create 24 such records.

Treatment	Name	No. animals	Notes
Treatment 1	Saline	6	
Treatment 2	Compound X 1.0 mg/kg	6	
Treatment 3	Compound X 5.0 mg/kg	6	
Treatment 4	Compound X 10.0 mg/kg	6	

Figure 1. An experiment which consists of four treatment groups of 6 animals each.

But what treatment should the first animal receive? It might seem obvious to say it should receive treatment 1, but the second animal, treatment 1 too? As ANY-maze tests animals in order (i.e. animal 1 is tested first, then animal 2, and so on) *assigning* animals to treatment groups in this way would mean that you would test all the animals for treatment 1 and then all the animals for treatment 2 - probably not what you'd want to do.

As you may know already, the way ANY-maze assigns animals to treatment groups is actually defined in the protocol. You'll find full details in the Choosing how to assign animals to the treatment groups topic, but in essence you can either choose to have ANY-maze assign the animals to treatments using one of a number of standard methods or you can tell the system that you'll perform the assignment yourself.

So, what actually happens here is that ANY-maze uses the details you've entered to create one record for each treatment group and one record for each animal - it then looks at the protocol and, unless you've chosen manual assignment, assigns the animals to the groups in the appropriate way. For example, one of the standard methods of assigning animals is called *cyclic*. Using this method the first animal is assign to the first treatment group, the 2nd animal to the 2nd group, the 3rd animal to the 3rd group and so on until all the groups include one animal, it then cycles round and the next animal is assigned to the first group, the next one to the 2nd group and so on.

Manually assigning animals to treatment groups

In fact, most people prefer to use manual assignment. In this case *you* need to specify which treatment each animal will receive. The easiest way to do this is by selecting the *View animals* button, which will show the Animals spreadsheet, instead of the Treatments groups spreadsheet - see figure 2.

Animal	Status	Treatment	Island	Sex
1	Normal	A	NW	Male
2	Normal	C	SW	Female
3	Normal	B	SE	Male
4	Normal	D	NE	Female
5	Normal	A	NW	Female
6	Normal	D	SW	Male

Figure 2. The easiest way to enter animal treatments when manually assigning animals to treatment groups in using the Animals spreadsheet shown here. You simply need to enter the appropriate treatment code against each animal.

In the Animals spreadsheet you can quickly and easily enter the appropriate treatment code against each animal. By the way, if you're wondering why you'd be entering treatment *codes* (rather than treatment names) and what these codes actually are - you'll find the answers here.

See also:

- Treatment coding
- Unassigned animals

Unassigned animals

Details

When you choose to manually assign animals to treatment groups (see Assigning animals to the treatment groups) ANY-maze will just create the requisite group and animal records without specifying which animal belongs to which group. This means the system will know that the experiment includes, for example, 4 treatment groups and 24 animals but, as the animals haven't been assigned to the groups, it won't be able to say which animal belongs to which group - so the animals, in this case, are said to be *unassigned*.

The fact that animals are unassigned is almost irrelevant to ANY-maze - you can still test them and you can still access their results - I've only mentioned it because on the *treatments pane* ANY-maze will include a special, group called *Unassigned*, for such animals - see figure 1.

Treatment	Name	No. animals	Notes
Unassigned		12	
Treatment 1	Drug	0	
Treatment 2	Saline	0	
Treatment 3			
Treatment 4			
Treatment 5			

Figure 1. While an experiment includes unassigned animals ANY-maze includes a special group for them on the treatments pane.

Of course, as soon as you've manually assigned all the animals to treatments, the *Unassigned* group will disappear from the treatments list.

See also:

- Treatment coding
- Assigning animals to the treatment groups

Choosing not to use treatment groups

Details

Although in most experiments you will probably choose to divide your animals in different groups, this might not always be the case, and in these circumstances it makes sense to turn off treatment groups altogether.

If you want to work in this way, you simply need to switch to the Protocol page, select the Treatment groups element and **deselect** the option *Use treatment groups*. Doing this will remove the *View treatments* option from the Experiment page - causing the right-hand side of the page to always show the Animals spreadsheet.

See also:

- Adding animals to an experiment which doesn't use treatment groups

Editing and deleting treatment groups and animals

Overview

At any time in an experiment you can come back to the treatments pane and alter any of the details you've entered. There's only one restriction - you can't delete animals which have already been tested.

You'll find full details in the following topics:

- Changing a treatment group's name
- Adding more animals to a treatment group
- Reducing the number of animals in a treatment group
- Adding a new treatment group
- Removing a treatment group altogether

Changing a treatment group's name

In brief

You can change a treatment group's name by simply typing in something different.

Details

You shouldn't remove the name altogether as this will remove the group - see Removing a treatment group altogether.

Limits

You can enter any name, up to a maximum of 60 characters, although you can't change the name so it's the same as another group's - if you try to do this ANY-maze will tell you.

Adding more animals to a treatment group

In brief

You can increase the number of animals in a treatment group at any time by simply typing in a larger number.

Details

If you add more animals during an experiment (i.e. after you've performed at least one test but before you've completed all the tests) and the experiment includes more than one stage, then ANY-maze will ask you where in the test schedule you want the new animals to be included - the options available are described fully in the Adding animals topic.

Limits

Each treatment group is limited to 999 animals.

Reducing the number of animals in a treatment group

In brief

If you want to reduce the number if animals in a group then you can simply alter the group size shown on the treatments pane.

Details

ANY-maze won't remove animals which have actually been tested. For example, if you have an experiment in which one of the groups contains 10 animals and so far you've tested 7 of them then trying to reduce the group's N to 6 won't work - ANY-maze will tell you that it can't be reduced below 7.

Of course, if you really do want to reduce the N to 6 then there's still a way to do it. First, you'd reduce the N to 7 as described above (which would simply remove the *untested* animals) and then you'd have to change the status of one of the tested animals to Deleted.

The point here is that ANY-maze will never delete the results of a test - this is a basic rule of the system. Therefore, when you try to delete untested animals it lets you but when you try to delete tested animals it doesn't. Of course, you might wonder what changing the status of an animal to *Deleted* does if it doesn't *delete* the animal. The answer is it *marks* it as deleted, this means that ANY-maze will simply ignore the animal, **but** you can always come back later and change the status back to *Normal* which will 'undelete' the animal and its results will reappear.

The reason ANY-maze works this way is to ensure that no matter what happens you can't lose test results.

Adding a new treatment group

In brief

To add a new group, simply enter the new group's name and number of animals.

Details

As occurs when you add more animals to an existing group, ANY-maze may ask you where in the test schedule you want the animals to be included - this is described fully in the Adding animals topic.

The *Notes* column isn't editable; ANY-maze will use this to give you relevant information about the treatment, for example the number of deleted or retired animals which were allocated to that treatment.

Limits

You can enter up to 20 groups in an experiment.

Removing a treatment group altogether

In brief

If you want to remove a treatment group entirely then you should delete its name.

Details

As occurs when you simply reduce the number of animals in a group, ANY-maze will only actually remove animals which have not been tested - thus to remove an entire group in this way it's necessary that none of the animals in the group have been tested yet. Of course, this is always the case before you start an experiment and this is when you'll find this facility most useful.

Note: Reducing the number of animals in the group to zero will almost work (the animals will be removed) but the group itself will still exist.

See also:

- Reducing the number of animals in a treatment group

Add animals

Overview

When you add new animals to an experiment which is in progress, ANY-maze may need to ask you where in the *Test schedule* you want the animals to be included.

Details

This is best explained by an example. Imagine you're performing a water-maze experiment in which you first train the animals to find an island and then treat them and test their ability to remember where the island is. Thus your experiment has two stages, let's call them *Training* and *Retest*.

Now imagine you've performed the *Training* and you've just started the *Retest* stage, at this point you add some more animals to the experiment.

What should ANY-maze do? Immediately go back to the *Training* stage so these new animals can get trained, or finish the *Retest* stage of the trained animals and only then go back to the *Training* stage to train the new animals? This is the question ANY-maze is asking you.

In fact what ANY-maze actually asks is whether you want the new animals to be inserted at the start or end of the test schedule. If you consider the above example, then it's probably clear that adding the new animals at the start of the schedule will mean that they'll be trained straight away, while adding them to the end means that the existing animals will finish their *Retests* before then new animals get trained - figures 2, 3 and 4 demonstrate this.

Test	Animal	Stage	Trial	Apparatus	Status
1	1	Training	1	New apparatus	Completed
2	2	Training	1	New apparatus	Completed
3	3	Training	1	New apparatus	Completed
4	4	Training	1	New apparatus	Completed
5	1	Training	2	New apparatus	Completed
6	2	Training	2	New apparatus	Completed
7	3	Training	2	New apparatus	Completed
8	4	Training	2	New apparatus	Completed
•	1	Retest	1	New apparatus	Ready
•	2	Retest	1	New apparatus	
•	3	Retest	1	New apparatus	
•	4	Retest	1	New apparatus	

Figure 1. A very simple water-maze experiment - the animals have completed their training and are about to start the Retest stage.

Test	Animal	Stage	Trial	Apparatus	Status
1	1	Training	1	New apparatus	Completed
2	2	Training	1	New apparatus	Completed
3	3	Training	1	New apparatus	Completed
4	4	Training	1	New apparatus	Completed
5	1	Training	2	New apparatus	Completed
6	2	Training	2	New apparatus	Completed
7	3	Training	2	New apparatus	Completed
8	4	Training	2	New apparatus	Completed
•	1	Retest	1	New apparatus	Ready
•	2	Retest	1	New apparatus	
•	3	Retest	1	New apparatus	
•	4	Retest	1	New apparatus	
•	5	Training	1	New apparatus	
•	6	Training	1	New apparatus	
•	5	Training	2	New apparatus	
•	6	Training	2	New apparatus	
•	5	Retest	1	New apparatus	
•	6	Retest	1	New apparatus	

New animals added at the end of the Test schedule. Existing animals will finish their tests first.

Figure 2. Two new animals have been added to the experiment. In this case they've been added at the end of the test schedule.

Test	Animal	Stage	Trial	Apparatus	Status
1	1	Training	1	New apparatus	Completed
2	2	Training	1	New apparatus	Completed
3	3	Training	1	New apparatus	Completed
4	4	Training	1	New apparatus	Completed
5	1	Training	2	New apparatus	Completed
6	2	Training	2	New apparatus	Completed
7	3	Training	2	New apparatus	Completed
8	4	Training	2	New apparatus	Completed
•	5	Training	1	New apparatus	Ready
•	6	Training	1	New apparatus	
•	5	Training	2	New apparatus	
•	6	Training	2	New apparatus	
•	1	Retest	1	New apparatus	
•	2	Retest	1	New apparatus	
•	3	Retest	1	New apparatus	
•	4	Retest	1	New apparatus	
•	5	Retest	1	New apparatus	
•	6	Retest	1	New apparatus	

New animals added at the start of the Test schedule. Existing animals have to wait for them to catch up.

Figure 3. Two new animals have been added to the experiment. In this case they've been added at the 'start' of the test schedule.

Adding animals to an experiment

In brief

Any time you are viewing the Animals spreadsheet on the Experiment page, you can select the *Add animals* button. This causes the *Add animals* window to open (see figure 1) where you can simply enter the number of animals you want to add.

When using Treatment groups, it is usually best to add animals to an experiment by increasing the number of animals in one or more of the treatment groups.

Figure 1. The Add animals window.

Details

Here you can add up to 999 animals to your experiment. The animals will then appear in the Animals spreadsheet, where you will be able to enter additional information about them, should you wish to.

Deleting animals from an experiment

In brief

Any time you are viewing the Animals spreadsheet on the Experiment page, you can select the *Delete animals* button. This causes the *Delete animals* window to open (see figure 1) where you can enter the animal numbers of the animals you want to delete.

When using Treatment groups, it is usually best to delete animals from an experiment by reducing the number of animals in one or more of the treatment groups.

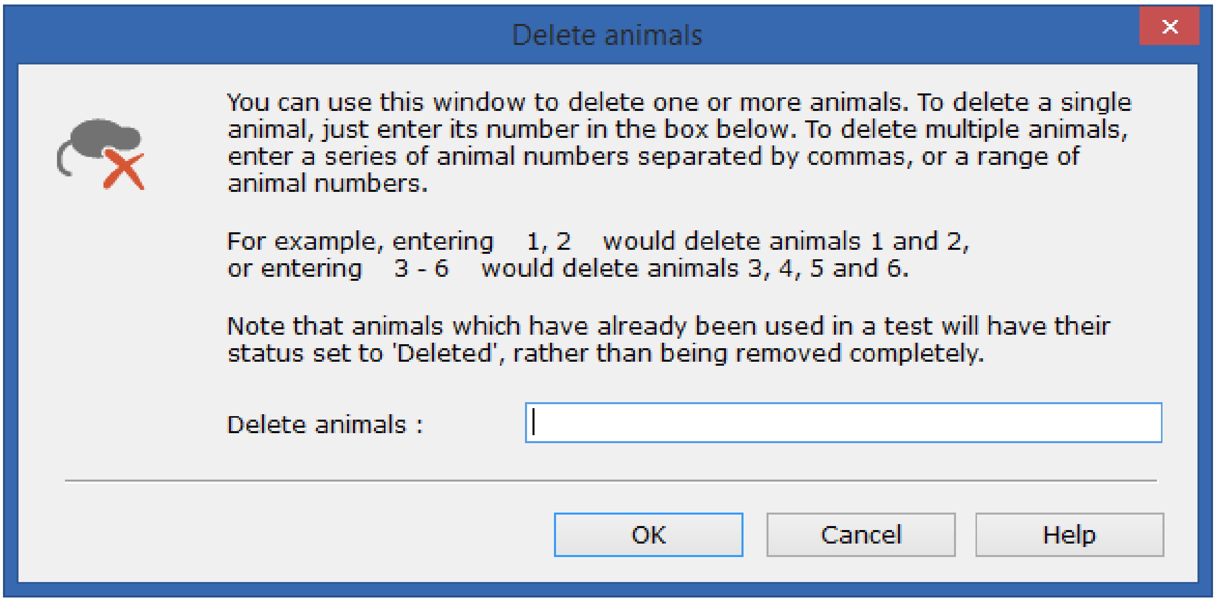


Figure 1. The Delete animals window.

Details

Here you can delete any of the animals in your experiment. If the animals have not yet been used in any tests, they will be deleted from the Animals spreadsheet. If they have been used in any tests, then their status will be changed to *Deleted*.

To delete one or more animals, simply enter their animal numbers, separated by commas (for example, '3, 4, 5'. If the protocol has been set up to use Animal IDs, you can enter the Animal IDs, separated by commas.

You can also delete a range of animals by entering the start and end animal **numbers** as a range, i.e. '3 - 6' will delete animals 3, 4, 5 and 6. (You can't use Animal IDs to enter a range, as they may not be sequential).

Importing lists of animals or tests

Overview

Although you will usually just type in details of your experiment's animals and tests, there may be circumstances when this information is already recorded in some other system and therefore it would make more sense to import the data then re-enter it all into ANY-maze. This option is provided through ANY-maze's import options.

- Importing animals
- Importing tests

Importing animals

Introduction

Normally you add animals to ANY-maze simply by entering a *Treatment group* and specifying the group's *N*. For example, entering groups of *Saline* and *Drug* each with an N of *6* would add twelve animals to the experiment. However, depending on how you have set the treatment group assignment options in the protocol, you may still have to specify exactly which of the two treatments each animal is to receive. Also, if your protocol includes any animal specific fields, such as sex or weight, then you may need to enter the field data for each animal, and if the protocol includes any zones whose positions vary between the animals you'll need to enter their positions too, and and, finally, if you have chosen to use your own animals' IDs then you will need to enter this data as well.

Of course, using the Animals spreadsheet this is all quite easy to do, but even so, if all the information is already stored in some other system, then it would make more sense if you could simply *import* it rather than having to type it all in again. This is possible by using the *Animal import* option, which reads data about the animals in an experiment from a *Comma-Separated Values* (CSV) file.

Creating a list of animals to import

ANY-maze uses a standard file format called *Comma-Separated Values,* usually abbreviated to CSV, for animal import. A CSV file is simply a text file with data on different rows and with individual values separated by commas - for example:

```
Animal,Treatment,Sex,Weight,Platform
123,Saline,Male,200,NW
456,Drug,Female,220,SW
```

Microsoft Excel includes an option to save files in CSV format, and this often the easiest way to create the files (in this case the commas separate the columns of data that form the spreadsheet).

The actual data in an import file consists of a *Header row* and any number of *Data rows*. The Header row is the first non-blank row in the file (all blank rows, anywhere in the file are simply ignored). The header row defines the data that will be found in the subsequent Data rows. Valid entries in the header row are:

Animal Animal can only be included if the protocol has been set up to use Animal IDs. If this is the case then the entries in the data rows specify the ID for each animal and can be any text, up to a maximum of 80 characters. If the value in a data row specifies an ID for an animal that already exists in the experiment then ANY-maze will either report an error or will simply ignore the entire data row, depending on the active import policies.

Treatment Specifies the name of the treatment group the animal should be added to. If the treatment group does not exist then a new group with the specified name will be created.

Field titles Any animal fields defined in the protocol can be included in the import. In the above example, 'Sex' and 'Weight' are animal fields.

For a field defined in the protocol as a *choice* field the data for each animal must be one of the choices specified for the field. For a field defined as a *numeric* field, the data must be a numeric value. For a field defined as a *text* field, the data can be anything, up to a maximum of 80 characters.

Zone positions Any zones defined in the protocol as having a position that varies between animals can have their positions included in the import. In the above example 'Platform' is such a zone, with its position being specified as 'NW' for animal 123 and 'SW' for animal 456. The name of a position must exactly match one of the zone's positions as defined in the protocol.

If the header row includes an invalid entry then ANY-maze will either report an error or it will simply ignore that 'column', depending on the active import policies.

Examples of some import files

A very simple import file might look like this:

```
Treatment
Saline
Saline
Drug
Drug
```

Importing this file into an empty experiment would add two treatments to the experiment, one called *Saline* and one called *Drug.* It would also add four animals to the experiment, animals 1 and 2 would be placed in the *Saline* treatment group, while animals 3 and 4 would be placed in the *Drug* treatment group.

A slightly more complex file might look like this:

```
Animal,Treatment
C1A2,Saline
C1A3,Saline
```

```
C1A4,Drug
C2A1,Drug
```

Importing this file would have a similar effect to the previous example, except that the animals would be given IDs of *C1A2, C1A3* etc. Note that this would only work if the experiment's protocol was set use your own animal IDs.

Note that there is no need to enter an animal number *and* an Animal ID - in fact, this is illegal, so a file that looks like the following would **not** be imported:

```
Animal,Animal ID,Treatment
1,C1A2,Saline
2,C1A3,Saline
3,C1A4,Drug
4,C2A1,Drug
```

Performing an import

To actually import an animal list, you simply need to switch to the Experiment page and click the *Import animals* button in the ribbon bar. This will cause a window to open where you can select the CSV file that contains the animal list. ANY-maze will then read the file and attempt the import - this is a two stage process:

During the first part of the import process (called pre-processing), ANY-maze reads the entire file and checks that it all makes sense. For example, if the header row contains an entry for 'Sex' then the system will check that the protocol does indeed include a field called 'Sex'. Any errors detected are reported and the import does not occur.

If the check is successful, then the system will proceed to the second stage and will actually import the data and create the treatment groups and animals. At the end of the process a report is displayed showing exactly what the system did - see figure 1 below:

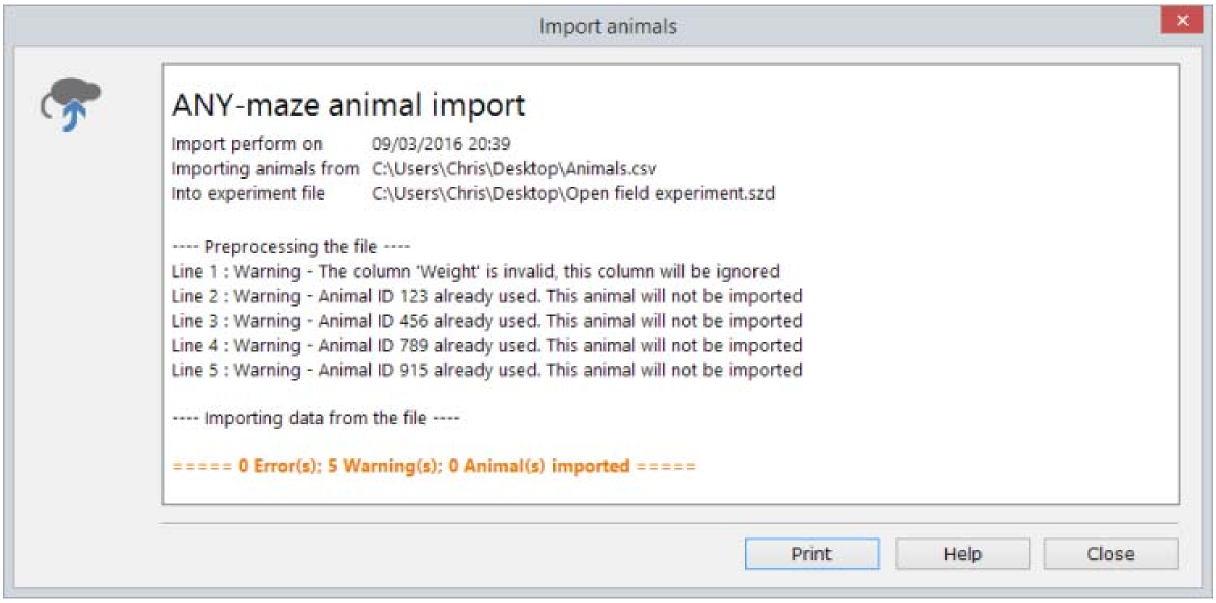


Figure 1. An example of an 'Import animals' report. In this case the CSV file included a column for 'Weight', but there was no 'Weight' field in the protocol. Import policies specified that invalid columns should be ignored, so the import was performed but the animal weights were not imported.

In most cases, you will probably import an animal list before you start any tests, but there are no *rules* about when you import animals. For example, you could import one file, perhaps including 30 animals, perform some of their tests and then import a second file, that might add another 20 animals to the experiment.

See also:

- Importing tests
- Import policies

Importing tests

Introduction

When the test schedule for an experiment is very complicated, it's often necessary to use ANY-maze's manual scheduling feature. This allows you to enter tests in any order, making the test schedule completely flexible.

Of course, if you know your schedule is complicated (and that you will therefore need to use manual scheduling), it implies that you have already planned your experiment, in which case you may well have created a test schedule in some other system, for example, in Microsoft Excel. In these circumstances it would make sense to simply import the schedule from that other system, rather than having to retype all the tests into ANY-maze. This is what the test import feature does.

Creating a list of tests to import

ANY-maze uses a standard file format called *Comma-Separated Values,* usually abbreviated to CSV, for test import. A CSV file is simply a text file with data on different rows and with individual values separated by commas - for example:

```
Animal,Apparatus,Novel side
C1A4,Box 1,Left
C1A2,Box 2,Right
```

Microsoft Excel includes an option to save files in CSV format, and this often the easiest way to create the files (in this case the commas separate the columns of data that form the spreadsheet).

The actual data in an import file consists of a *Header row* and any number of *Data rows*. The header row is the first non-blank row in the file (all blank rows, anywhere in the file are simply ignored). The header row defines the data that will be found in the subsequent data rows.

Each data row adds a test on a specific animal to the experiment, with the tests being scheduled in the order they appear in the file. For example, the file:

```
Animal
1
2
2
1
```

will create a schedule in which animal 1 will be tested, then animal 2, then animal 2 again and then animal 1 again. Note that imported tests still have to agree with the number of trials specified in the experiment's protocol, so in this case the protocol would need to include at least two trials otherwise the import would fail.

Valid entries in the file are:

Animal
This is the animal number or ID of the animal to be tested. If the protocol uses Animal IDs then you should specify either an animal's ID or an ANY-maze number prefixed by a # sign. For example #4. Otherwise, you should simply specify the animal's number (as in the example above).

Apparatus
This is the name of the apparatus that the test should be performed on. This is only necessary if the protocol includes more than one piece of apparatus. If the protocol does include multiple apparatus, but the import file does not have an apparatus 'column', then the animals will be assigned to apparatus using the rules specified in the stages element of the protocol.

Field titles
The names of any test fields defined in the protocol can be included in the import file.

For a field defined in the protocol as a *choice* field the data for each test must be one of the choices specified for the field. For a field defined as a *numeric* field, the data must be a numeric value. For a field defined as a *text* field, the data can be anything, up to a maximum of 80 characters.

Zone positions
Any zones defined in the protocol as having a position that varies within animals can have their positions included in the import. In the first example above, 'Novel side' is such a zone, with its position being specified as 'Left' for the test on animal C1A4 and 'Right' for the test on animal C1A2. The name of a position must exactly match one of the zone's positions as defined in the protocol.

Performing an import

To actually import a tests list, you simply need to switch to the Experiment page and click the *Import tests* button in the ribbon bar (if the *Import tests* button is disabled it means that you have not chosen to use manual scheduling in the protocol).

Selecting this button will cause a window to open where you can select the CSV file that contains the tests list. ANY-maze will then read the file and attempt the import - this is a two stage process:

During the first part of the import process (called pre-processing), ANY-maze reads the entire file and checks that it all makes sense. For example, it checks that all the animals specified in the import file actually exist in the experiment. Note that the processing of this check is affected by the current import policies.

If the check is successful, then the system will proceed to actually import the data and add the tests to the experiment's test schedule. At the end of the process a report is displayed showing exactly what the system did - see figure 1 below:

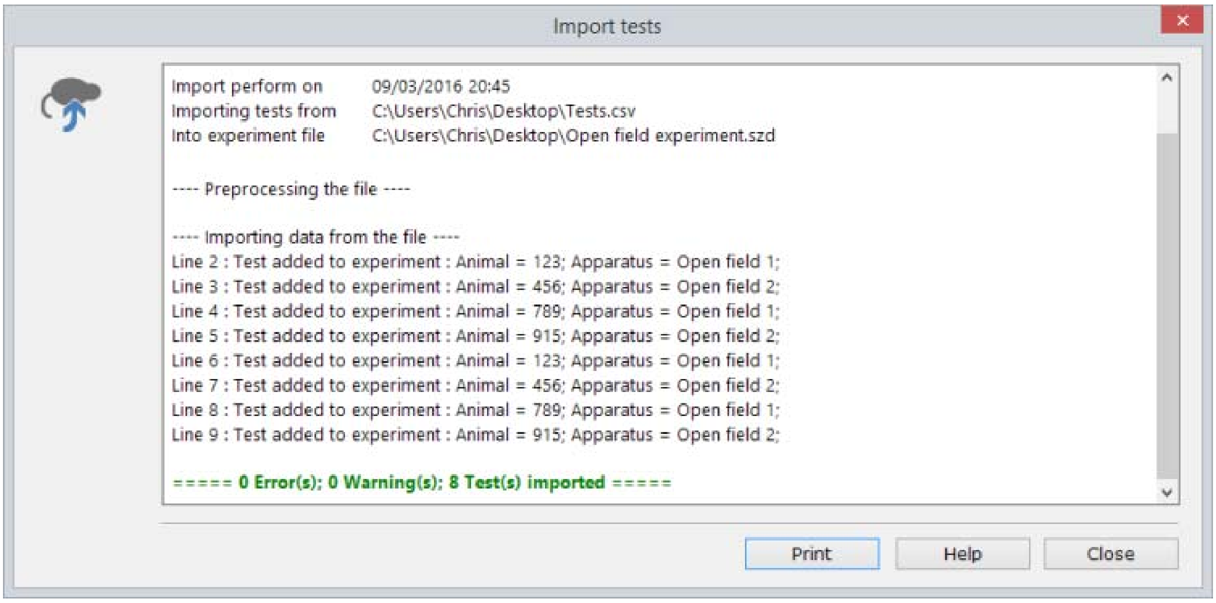


Figure 1. An example of an 'Import tests' report.

See also:

- Importing animals
- Import policies

Browse for import file path

Introduction

ANY-maze includes the ability to import lists of animals and tests from CSV files. This feature is described fully in the Importing lists of animals or tests topic.

The first stage when importing is to choose the file to import and this is the purpose of this window.

Details

You should use this window to navigate to a CSV (or text) file that contains the list of animals or tests that you wish to import. This file can be anywhere on your computer or network and can be called anything you like.

Details of the format of the animal and test import files can be found in the following two topics:

- Importing animals
- Importing tests

Understanding the relationship between the Treatment groups spreadsheet and the Animals spreadsheet

Details

The right-hand side of the Experiment page can show either the Treatment groups spreadsheet or the Animals spreadsheet. It's helpful to understand that these two spreadsheets actually show the same information, just from two different points of view.

As the name implies, the Treatment groups spreadsheet shows the different treatment groups together with the number of animals in each one. For example, the spreadsheet might show two groups - 'Saline', containing 6 animals, and 'Drug', containing another 6 animals.

On the other hand, the Animals spreadsheet shows all the animals in the experiment, together with additional information about each animal, including (if relevant) its treatment. So in the example from the previous paragraph, there would be 12 rows in the spreadsheet, one for each animal; 6 of the animals would be shown having a treatment of 'Saline' and 6 having a treatment of 'Drug'.

Any changes you make in one spreadsheet will necessarily alter the other, because they show the same data. Thus, in our example, if you changed the treatment of an animal in the Animals spreadsheet from being 'Saline' to being 'Drug', and you then switched to view the Treatment groups spreadsheet, it would now show 5 animals in the 'Saline' group and 7 in the 'Drug' group.

See also:

- The Treatment groups spreadsheet
- The Animals spreadsheet

The Tests page

Introduction

The Tests page is the heart of ANY-maze - it's here that you actually perform tests. The page is divided into two panes: the Experiment management pane, shown on the left, is used to access reports which help you manage the overall experiment and view results for individual animals and tests; while the Apparatus pane, shown on the right, is used to actually run the tests in your apparatus.

You can drag the divider between the Experiment management and Apparatus panes. If you double click the divider then the Experiment management pane will be sized to show the full width of the currently displayed Experiment management report.

- The experiment management reports
- Running tests
- Adjusting apparatus layout
- Recording and tracking with videos

The experiment management reports

You can drag the divider between the Experiment management and Apparatus panes. If you double click the divider then the Experiment management pane will be sized to show the full width of the currently displayed Experiment management report.

Introduction

On the right-hand side of the *Test page* is the *Experiment management pane*, which shows reports and tables which can be used to manage the current experiment.

- The Test schedule report
- The Animal details report
- The Test details report
- The Test track plot report
- The Test heat map report
- The Test data report
- The Test video report
- The Test charts report
- The Test technical details report

The Test schedule report

Introduction

The Test schedule report is perhaps the single most important report in ANY-maze. It not only helps you to see where you've got to in an experiment, but it also provides access to the individual Animal and Test details reports.

The Test schedule report is the *home* page of the experiment management pane (shown on the left-side of the Tests page) and therefore when you first switch to the Tests page it will be the report that's shown. You can use links in the Test schedule report to access other reports, but at any time you can get back to it quickly by simply clicking the *Test schedule* button in the ribbon bar

- Using the Test schedule report to manage an experiment
- Choosing what's shown in the Test schedule report
- Weighing animals on the Test schedule report
- Animal treatment dosage shown on the Test schedule report
- Accessing details about individual animals and tests

Using the Test schedule report to manage an experiment

The Test schedule report shows a list of the tests which will be performed in an experiment in the order in which they'll be performed. Essentially, ANY-maze will just work through this list prompting you to perform each test in turn.

The order of tests is defined in the protocol but essentially ANY-maze will works through each *Stage,* in an experiment in the order in which they appear in the protocol, and within each stage it'll work through the animals in numerical order. If the animals can be tested more than once in the stage (i.e. have multiple trials) then the protocol will define the trial order.

For full details about test order refer to Specifying the order in which animals will be tested.

Other useful topics related to managing experiments:

- Resolving common problems when testing
- Skipping a test
- Skipping an entire stage for an animal
- Undoing a performed test
- Cancelling a performed test
- Reperforming a performed test

- Adding additional scoring to a performed test
- Ending an animal's tests in a stage manually
- Retiring an animal from an experiment
- Deleting an animal from an experiment

Choosing what's shown in the Test schedule report

The information that's shown in the Test schedule report is actually defined in the Protocol. This is very helpful as it allows you to just include information that's relevant to your experiment. For example, if the experiment doesn't include multiple trials then including *Trial number* in the report would be unnecessary, but if it does then knowing the trial's number would be extremely useful.

To quickly change the data shown, select the *Select columns* button in the ribbon bar

A word of advice - don't select too many columns as you'll have to scroll from side to side to see them which can become quite frustrating.

As well as choosing what columns are shown you can also use the three check boxes shown in the ribbon bar to choose what tests are included:

Show all stages	If checked, the tests in all the stages of the experiment are listed, otherwise only the tests in the current stage are shown. This box is disabled when an experiment is complete as in this situation all stages are always shown.
Show completed tests	If checked, tests which have been performed are listed, otherwise they're not. This box is disabled when an experiment is complete as, in this situation, all tests are completed and they're all shown.
Show cancelled tests	Only enabled if *Show completed tests* is checked. If checked, any cancelled tests (or tests performed on deleted animals) are listed, otherwise they're not.

Accessing details about individual animals and tests

The Test schedule report acts as the access point for the individual Animal details and Test details reports. To access one of these reports simply click the appropriate animal or test number on the Test schedule report. Alternatively you can open the Testing status menu and select the appropriate option - see figure 10.

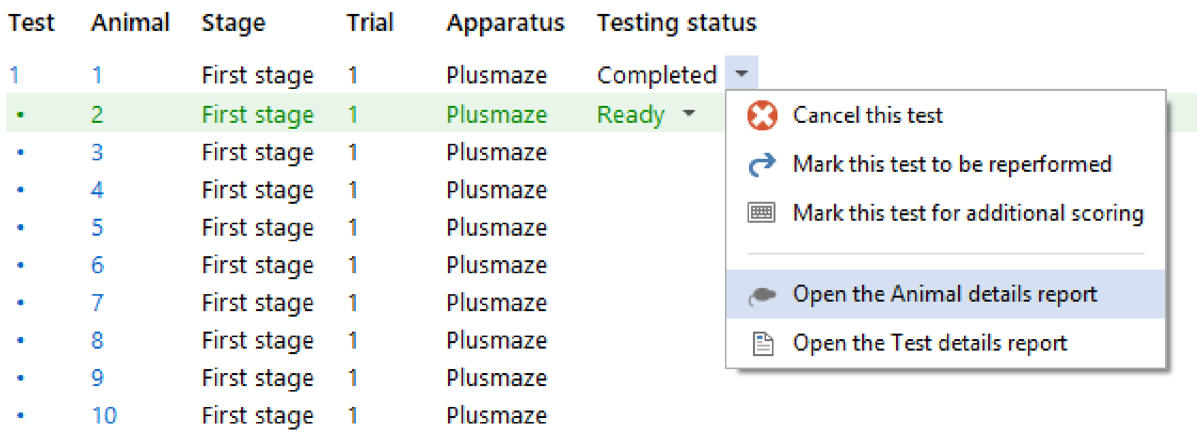


Figure 10. To access an Animal details report you can simply open the relevant Testing status menu and select the appropriate option.

After following a link you can get back to the Test schedule report either by clicking the *Test schedule* button or by working back through the reports you've viewed using the *Previous report* button; both these buttons are in the ribbon bar.

See also:

- Printing reports
- Saving reports
- Copying reports
- Sending reports by e-mail

Weighing animals on the Test schedule report

The Weighing status column

If the experiment's protocol specifies that you will weigh animals before their tests, then you will find that Test schedule report includes a *Weighing status* column - see figure 1.

Exactly what the *Weighing status* column will show depends on whether you are manually weighing animals or using the ANY-maze Animal scales.

Test	Animal	Stage	Trial	Weighing status	Testing status
1	1	First stage	1	Weighed: 228.0g	Completed
•	2	First stage	1	Enter the weight (g)	
•	3	First stage	1		
•	4	First stage	1		

Figure 1. In this experiment, animals are set to be weighed manually before their first trial.

Manually weighing animals

If you are manually weighing animals, then the *Weighing status* column will prompted you to enter the animal's weight - as in figure 1. You should:

1. Weigh the animal
2. Type its weight into the field
3. Press return or enter

The field will change to show the weight you just entered and, if relevant, ANY-maze will prompt you to weigh the next animal.

If after weighing an animal, you want to weigh it again (perhaps you typed in the wrong value), then you simply need to open the *Weighing status menu,* by clicking the little down arrow next to the weight, and select *Reweigh the animal* from the options that appears - see figure 2.

Test	Animal	Stage	Trial	Weighing status	Testing status
1	1	First stage	1	Weighed: 228.0g	Completed
•	2	First stage	1	Weighed: 220.0g	Ready
•	3	First stage	1	Enter the weight (	
•	4	First stage	1		

Reweigh the animal

Figure 2. The animal from figure 1 has been weighed and its weight (220g) has been entered. If you want to reweigh the animal you can open the Weighing status menu and select the option to 'Reweigh the animal'.

Weighing animals using the ANY-maze Animal scales

If you are using the ANY-maze Animal scales to weigh your animals then the Weighing status column will prompted you to weigh each animal - see figure 3.

Test	Animal	Stage	Trial	Weighing status	Testing status
1	1	First stage	1	Weighed: 228.0g	Completed
•	2	First stage	1	Weigh the animal	Waiting for the animal to be weighed
•	3	First stage	1		
•	4	First stage	1		

Figure 3. Animal 2 is ready to be tested, but you must weigh the animal first.

To weigh an animal, simply place the animal in the basket and place the basket on the scales - the animal will be weighed automatically. ANY-maze weighs animals for a few seconds and uses the average weight registered during the period, this avoids problems that would otherwise arise if the animal moves while being weighed. During the weighing process the Weighing status counts down the seconds until the weighing cycle will complete - see figure 4.

Test	Animal	Stage	Trial	Weighing status	Testing status
1	1	First stage	1	Weighed: 228.0g	Completed
•	2	First stage	1	Weighing the animal: 3s	Waiting for the animal to be weighed
•	3	First stage	1		
•	4	First stage	1		

Figure 4. While weighing ANY-maze counts down the seconds until the weigh cycle will complete.

When the weighing cycle ends, the Weighing status prompts you to remove the animal from the scales - see figure 5. Removing the animal confirms the animals weight and stores it in ANY-maze. If after weighing an animal, you want to weigh it again (perhaps you weighed the wrong animal!), you simply need to open the Weighing status menu, by clicking the little down arrow next to the weight, and select *Reweigh the animal* from the menu that appears - see figure 2, above.

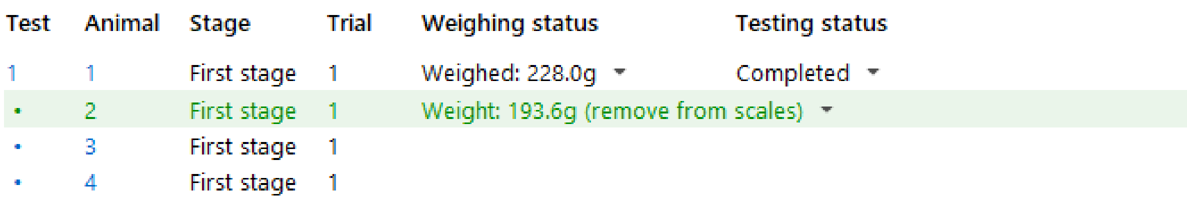


Figure 5. When the weigh cycle completes, you will be prompted to remove the animal (and basket) from the scales - doing so causes ANY-maze to store the animal's weight.

Skipping weighing of an animal

ANY-maze will work through the animals in the experiment, weighing each one in turn. If you don't want to weigh a particular animal, you should open the Weighing status menu, by clicking the little down arrow next to the weight, and select *Skip weighing this animal* from the menu that appears. Note the if a test is waiting for the animal to be weighed, then skipping weighing may or may not mean that you can then start the test - this will depend on whether the protocol is set to prevent a test from starting if the animal has not been weighed. However, if the protocol does prevent the test from being started, you can still skip the test and move on to the next animal.

If you have skipped weighing of an animal then opening it's weighing status menu will show an option to *Unskip weighing the animal* - see figure 6.

Test	Animal	Stage	Trial	Weighing status	Testing status
1	1	First stage	1	Weighed: 228.0g	Completed
•	2	First stage	1	Weighed: 193.6g	Ready
•	3	First stage	1	Weighing skipped	
•	4	First stage	1	Weigh the animal	
•	5	First stage	1		

Unskip weighing the animal

Figure 6. You can use the Weighing status menu to both 'Skip' and 'Unskip' (shown here) the weighing of an animal.

Weighing all the animals before starting a stage

One of the options in the protocol for when to weigh animals is to 'Weigh all the animals before starting the stage'. If you select this then the Test schedule report will include one row per animal (before stage's 'test' rows) for you to weigh the animals. In other words, you will first weigh all the animals and then test them - see figure 7.

Test	Animal	Stage	Trial	Apparatus	Weighing status	Testing status
	1	Training	Weigh	Water-maze 1	Enter the animal weight (g)	
	2	Training	Weigh	Water-maze 1		
	3	Training	Weigh	Water-maze 1		
	4	Training	Weigh	Water-maze 1		
	5	Training	Weigh	Water-maze 1		
	6	Training	Weigh	Water-maze 1		
•	1	Training	1	Water-maze 1	-	Waiting for the animal to be weighed
•	2	Training	1	Water-maze 1	-	
•	3	Training	1	Water-maze 1	-	
•	4	Training	1	Water-maze 1	-	
•	5	Training	1	Water-maze 1	-	
•	6	Training	1	Water-maze 1	-	

Figure 7. In this experiment, the animals are set to be weighed BEFORE any of them are tested in their training stage - this causes special 'weigh' rows (highlighted in the figure) to be included in the Test schedule report.

See also:

- Setting up animal weighing in the protocol
- The ANY-maze Animal scales

Animal treatment dosage shown on the Test schedule report

The Dose column

If the experiment's protocol specifies that ANY-maze should calculate the animal's treatment doses then the Test schedule report will include a column for *Dose* - see figure 8.

Test	Animal	Stage	Trial	Apparatus	Weighing status	Dose	Treatment	Testing status
•	2	Training	1	Water-maze 1	Weighed: 200.0g	0.200mg	B	Ready
•	4	Training	1	Water-maze 1	Weighed: 220.0g	1.100mg	A	
•	5	Training	1	Water-maze 1	Weighed: 210.0g	0.210mg	B	
•	6	Training	1	Water-maze 1	Weighed: 200.0g	0.200mg	B	
•	10	Training	1	Water-maze 1	Weighed: 220.0g	1.100mg	A	
•	13	Training	1	Water-maze 1	Weighed: 225.0g	1.125mg	A	
•	14	Training	1	Water-maze 1	Weighed: 215.0g	0.215mg	B	
•	15	Training	1	Water-maze 1	Weighed: 200.0g	1.000mg	A	
•	16	Training	1	Water-maze 1	Weighed: 230.0g	0.230mg	B	
•	17	Training	1	Water-maze 1	Weighed: 210.0g	1.050mg	A	

Figure 8. In this experiment the protocol specifies that ANY-maze should calculate the treatment dosage for each animal - this is reported in the Dose column of the Test schedule report.

For ANY-maze to calculate the dose for an animal, the animal must first be weighed, so until you weigh an animal the dose column will be blank. Once the animal has been weighed, the dose column will report the dose that the animal should be given - this is based on the dose information entered for the treatment groups, and the animal's weight. For example, if the treatment specifies that the animal should be given a dosage of 1.0mg/kg, and the animal weighs 200g, then the dose will be shown as 0.2mg (see animal 1, in figure 8).

Dose as a volume

One of the options when setting up treatment doses in the protocol, is for ANY-maze to calculate the dose as a volume (usually the volume to inject) rather than as a mass. In this case the *Dose* column of the Test schedule report will show the volume to administer (in millilitres), rather than the mass - see figure 9. This is calculated based on the information specified in the protocol, the dose and concentration specified when the treatment was entered, and the animal's weigh.

Test	Animal	Stage	Trial	Apparatus	Weighing status	Dose	Treatment	Testing status
•	2	Training	1	Water-maze 1	Weighed: 200.0g	4.762ml	B	Ready
•	4	Training	1	Water-maze 1	Weighed: 220.0g	5.238ml	A	
•	5	Training	1	Water-maze 1	Weighed: 210.0g	5.000ml	B	
•	6	Training	1	Water-maze 1	Weighed: 200.0g	4.762ml	B	
•	10	Training	1	Water-maze 1	Weighed: 220.0g	5.238ml	A	
•	13	Training	1	Water-maze 1	Weighed: 225.0g	5.357ml	A	
•	14	Training	1	Water-maze 1	Weighed: 215.0g	5.119ml	B	
•	15	Training	1	Water-maze 1	Weighed: 200.0g	4.762ml	A	
•	16	Training	1	Water-maze 1	Weighed: 230.0g	5.476ml	B	
•	17	Training	1	Water-maze 1	Weighed: 210.0g	5.000ml	A	

Figure 9. In this experiment the protocol specifies that ANY-maze should calculate the treatment dosage for each animal and report it as the volume to administer. In this example, the option to 'administer approximately the same volume for all animals' was chosen, which is why the dose is similar for the different treatment groups.

If an animal's dose is shown as *Treatment: No dose* it means that no dose information has been entered in the Treatment groups spreadsheet for the relevant animal's treatment group.

See also:

- Setting up calculation of Treatment doses in the protocol
- Specifying the dose for treatment groups
- Specifying the treatment concentration for treatment groups

The Animal details report

Introduction

The animal details report brings together all the information ANY-maze holds about an individual animal. Here you can enter details about the animal, alter its status and view its test schedule and results.

To access this report you should click the relevant animal's number on the Test schedule report.

- Entering or editing the animal's treatment
- Altering an animal's status to deleted or retired
- Specifying whether the animal is lighter or darker than the apparatus background
- Recording data for 'animal fields'
- Recording the positions of movable zones
- Recording notes about an animal
- Viewing the animal's test schedule
- Ending an animal's trials in a particular stage
- Viewing an animal's results

Entering or editing the animal's treatment

At the very top of the animal details report is a drop list titled *Treatment*. Here you can select the animal's treatment or specify that the treatment is *Undefined* - i.e. the animal's not part of any treatment group. If you are performing a blind experiment then the drop list will show treatment codes, otherwise it will show the treatment names. If you want to find out how the treatments have been coded then you should switch to the Experiment page and select the *Reveal treatment coding* button in the ribbon bar.

Altering an animal's treatment group can be very useful if you mis-dose an animal - you simply need to change the animal's treatment. Of course you may also choose to change a different animal to receive this animal's treatment so as to maintain your group Ns.

*If you want to enter the treatments for lots of animals then it's easier to use the **Animals spreadsheet** rather than accessing each animal's details report separately.*

Altering an animal's status to deleted or retired

All animals have a status which is either Normal, Retired or Deleted. You can alter the animal's status simply by choosing the appropriate option from the *Status* drop list at the top of this report. The actual status's are described in detail in:

- Retiring an animal from an experiment
- Deleting an animal from an experiment

An animal with *Normal* status is neither *Retired* nor *Deleted*.

Specifying whether the animal is lighter or darker than the apparatus background

One of the tracking options in the Protocol specifies whether the animals are lighter or darker than the background of the apparatus. Normally you will be able to select one of these options and ANY-maze will apply it to all the animals in your experiment. However, in some cases you might plan to work with animals of different strains and some may be lighter and some darker than the background. In this case you will find that the animal details report includes an *Animal colour* drop list where you can select the appropriate setting for each animal individually.

If you want to specify this setting for lots of animals then it's easier to use the Animals spreadsheet, on the Experiment page, rather than accessing each animal's details report separately.

Recording data for 'animal fields'

If the experiment's protocol includes any fields defined for animals, such as weight, sex, age etc., then they will appear on this report and you can simply enter the appropriate information. Remember that you can use numeric fields (like weight) as dependent variables in analysis and choice fields (like sex) as independent variables.

If you want to enter data for lots of animals then it's easier to use the Animals spreadsheet, on the Experiment page, rather than accessing each animal's details report separately.

Recording the positions of movable zones

If the protocol includes any movable zones whose positions change between, but not within animals, then they will be listed in this report and you will be able to select the relevant position of each zone for this animal.

For example, in a water-maze you might have an island that's positioned in the North West for some animals and the South East for others (note that the position remains the same *within* an animal's trials). This zone would be shown on this page as a drop list and you'd be able to choose either 'North West', 'South East' or 'Undefined' as its position.

If you want to enter the position of a movable zone for lots of animals then it's easier to use the Animals spreadsheet, on the Experiment page, rather than accessing each animal's details report separately.

Recording notes about an animal

You can record notes that relate to an animal in the area immediately below the *Animal Notes* title on the Animal details report. You can enter anything you like. It's usually a good idea to record something if you retire or delete an animal explaining the reason why.

The first 80 characters that you type here will be included in the Data page spreadsheet (assuming the 'Animal notes' column is actually shown). For this reason it's a good idea to be fairly succinct, although there is, in fact, no limit to how much you can enter.

When the animal notes field is active (i.e. the cursor is in the field), the *Animal notes format* section of the ribbon bar will be enabled. This contains a number of options, which apply to any text that is currently selected in the notes field. (If no text is currently selected, then the formatting you select will apply to new text when you start typing).

Reset formatting	Resets the formatting of the selected text to the default formatting (i.e. removes any colour, bold, italic etc.)
Text colour	Opens a dialog box allowing you to select a colour for the selected text.
Bold	Makes the selected text **bold.**
Italic	Makes the selected text *italic.*
Underline	Makes the selected text underlined.
Increase text size	Increases the size of the selected text by one point size.
Decrease text size	Decreases the size of the selected text by one point size.

Viewing the animal's test schedule

This report includes a *mini* test schedule for the individual animal. This makes it easy to review all the tests which an animal has received, which can be very helpful in a multi-stage/multi-trial experiment.

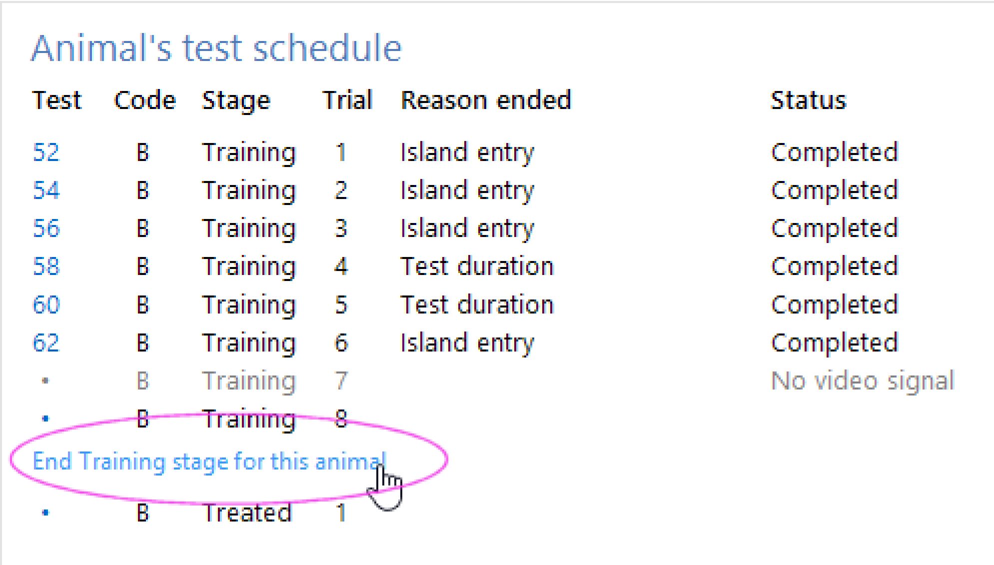
The data shown here is the same as that shown on the main Test schedule report (with the exception that animal number column isn't included) and, as on the main report, the Test numbers are links which will take you to the individual Test details reports.

One difference between the main Test schedule report and this one, is that here the reason why each stage ended is also shown. Also you can use the animal's test schedule to manually end the animal's tests in a stage or to have the animal

- skip a stage entirely.

Ending an animal's trials in a particular stage

If a stage includes multiple trials, then the protocol may specify that the user can manually end the trials for individual animals - see Creating stage end rules. If this is the case then the animal's test schedule will include a link titled *End xxx stage for this animal*, clicking the link will end the stage with the reason *User ended the stage*.


Animal's test schedule

Test	Code	Stage	Trial	Reason ended	Status
52	B	Training	1	Island entry	Completed
54	B	Training	2	Island entry	Completed
56	B	Training	3	Island entry	Completed
58	B	Training	4	Test duration	Completed
60	B	Training	5	Test duration	Completed
62	B	Training	6	Island entry	Completed
•	B	Training	7		No video signal
•	B	Training	8		
End Training stage for this animal					
•	B	Treated	1		

Figure 1. Ending a stage manually for an animal. The highlighted link is only included if the protocol specifies that the user can manually end the trials for individual animals.

If you end a stage in this way then the animal's test schedule will change to include a link titled *Restart this stage for this animal* and clicking it will restart the stage. Note that if you restart an earlier stage for an animal (i.e. a stage prior to the current stage of the experiment) then all the current testing will be 'blocked' until the animal catches up - i.e. completes the earlier stage.

Viewing an animal's results

At the bottom of the report is a section which shows the results for all the animal's tests. The exact measures which are shown are defined in the protocol and you can easily alter them by clicking the link shown just below the report section's title.

Note that this report is not intended to be the main location for accessing an experiment's results (this is better done on the Results page or Data page) - rather the results are shown here to provide an easy way to view all the data for an individual animal in one place.

See also:

- Printing reports

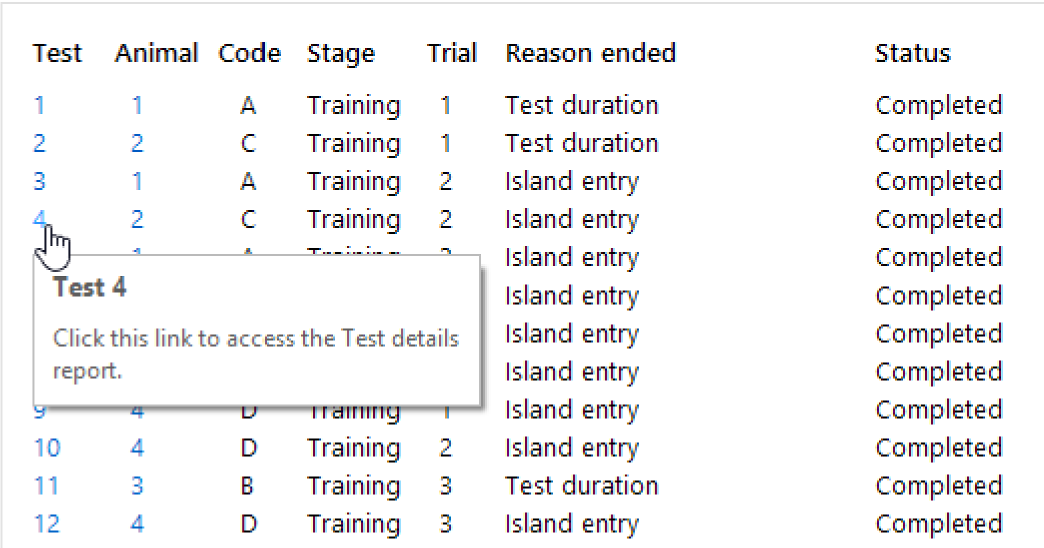
- Saving reports
- Copying reports
- Sending reports by e-mail

The Test details report

Introduction

The Test details report, together with its sub reports, bring together all the information that ANY-maze holds about an individual test.

To access this report you should click the relevant test's number on the Test schedule report.



Test	Animal	Code	Stage	Trial	Reason ended	Status
1	1	A	Training	1	Test duration	Completed
2	2	C	Training	1	Test duration	Completed
3	1	A	Training	2	Island entry	Completed
4	2	C	Training	2	Island entry	Completed
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Island entry	Completed
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Island entry	Completed
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Island entry	Completed
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Island entry	Completed
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Island entry	Completed
10	4	D	Training	2	Island entry	Completed
11	3	B	Training	3	Test duration	Completed
12	4	D	Training	3	Island entry	Completed

Figure 1. To access a Test details report you simply have to click the test's number in the Test schedule report - the numbers are links.

Note: Before a test is performed it doesn't actually have a *test number* instead the test is represented by a *marker* on the Test schedule report - however, clicking this marker will still access the Test details report, just like clicking its number would.

- Changing the animal that was tested in the test
- Changing the animal that will be tested in a manually scheduled test
- Skipping an unperformed test
- Deleting an unperformed test
- Cancelling a performed test
- Reperforming a performed test

- Reviewing a performed test so as to add additional scoring to it
- Recording details about the animal and/or test
- Specifying the position of movable zones
- Recording notes about a test
- Viewing a test's results
- Accessing related reports

Changing the animal that was tested in the test

At the top of the Test details report ANY-maze shows the number of the animal that will be / was tested in the test. If the test has been performed then you can alter this number, which can be very useful if you accidentally tested the wrong animal.

If you do change the animal number then ANY-maze will automatically reschedule the tests for affected animals. For example, if you're performing an experiment in which each animal receives 4 training trials and after training animal 1 in its first trial you accidentally train it again instead of training animal 2 then, when you alter the animal number for the test to '1', ANY-maze will recognise that this was the second trial for animal 1 and will therefore reduce the number of trials for this animal (otherwise it would have 5 trials) and increase the number for animal 2 (otherwise it would only have 3 trials).

Changing the animal that will be tested in a manually scheduled test

When manually scheduling tests, the test details report of an *unperformed* test will include a field showing the animal number of the animal the test is to be performed on. You can edit this, to change the animal to test.

Skipping an unperformed test

When you are viewing the Test details report of a test that hasn't been performed yet, the ribbon bar will include a *Skip test* button. Clicking this button will cause the test to be skipped by the ANY-maze scheduler. When a test has been skipped an 'un-skip' button is shown instead, clicking it will cause the test to be placed back into the test schedule, which will often mean it becomes the next test to perform.

You find more information about skipping tests in Skipping a test.

Deleting an unperformed test

If you are using manual scheduling, then, when you view the Test details report for an *unperformed* test, the ribbon bar will include a *Delete test* button; clicking this button will cause the test to be deleted.

The action of deleting the test necessarily means that the Test details report can no longer be displayed (because the test whose details it's showing has been deleted), therefore, after deleting a test, the Test schedule report will automatically be displayed.

Note that you can't delete unperformed tests when you are using automatic test scheduling because if you did, the test scheduler would immediately create a new test to replace it. This is because the scheduler will always ensure that each animal has as many trials as are defined in the stage elements of the protocol. However, in this case you can skip the unperformed test instead.

Cancelling a performed test

When viewing a test that has been performed, the ribbon bar includes a *Cancel test* button. Clicking this button will cause the test to be cancelled. This means that ANY-maze will ignore the test completely, although it won't actually throw it away - thus you can un-cancel the test if you later change your mind (to do this click the *Un-cancel test* button in the ribbon bar).

It's important to understand that cancelling a test will cause the scheduler to introduce a new test to replace it. This is because the cancelled test will be completely ignored so, if an animal is supposed to have, say, 2 trials in a stage and you cancel one of them, the scheduler will think that a new trial is required for the animal. If what you actually want to do is stop testing an animal then you should consider retiring or deleting it instead, or perhaps, ending the current stage for the animal.

There's more about cancelling tests in the Cancelling a performed test topic.

Reperforming a performed test

Reperforming a test does what the name implies, it causes a performed test to be reperformed, but without altering the test's position within the animal's trials. For example, if you record a video of a test then you might decide at some point in the future that you want to *retrack* the test using the video.

In this case cancelling the 'original' test will not be a good idea because, as mentioned above, the scheduler will *replace* the cancelled test with a new one. This new test will be added to the end of the animal's tests in the stage and this will probably mean that the test's trial number will change. For example, imagine you have performed four trials on animal 1 and you now want to retrack trial 1. If you cancel this test then the replacement test will be put in at the *end* of the trials and will therefore be trial 4, not 1. So, in these circumstances, cancelling a test is a bad idea - what you should do is *reperform* the test.

To reperform a test click the *Mark test to be reperformed* button in the ribbon bar. Reperforming a test simply causes ANY-maze to rerun the test, but the test's position within the animal's trials remains the same.

For full details about reperforming tests refer to the Reperforming a test topic.

Reviewing a performed test so as to add additional scoring to it

In some tests you may want to score a wide range of different behaviours using keys. While ANY-maze is able to do this (you can define up to 26 'key' behaviours), *you* will probably find it very hard to score more than a few behaviours simultaneously.

To overcome this, ANY-maze allows you to review a video of a test and add additional scoring to the test results. For example, imagine you want to track animals in your apparatus and, at the same time, score 8 different behaviours. To do this, you could run the tests 'live' and have ANY-maze track the animals and simultaneously record a video of the test. While ANY-maze is doing this you could score 2 of the behaviours you're interested in, using keys. Then, when the test is finished, you could review the video and score a further 3 behaviours and then review it again and score the last 3 behaviours - so you'd end up having scored them all. There's no rule about how many times you can review the test in this way, so you could review 8 times and score one behaviour each time, if you wanted to.

This ability to add scoring to a performed test doesn't alter the way you look at the results - as it will appear as if all the behaviours were scored at the same time - it just makes it easier for you to actually perform the scoring.

Full details about how to add additional scoring to a test can be found in the Adding additional scoring to a performed test topic.

Recording details about the animal and/or test

If the protocol includes any fields then they will be listed on this report. Fields which are being recorded for the animal will be shown with the prefix '*Animal* - ' while those being recorded for tests will be shown with the prefix '*Test* - '. Note that 'numeric' fields will show their units (if defined in the protocol) in parenthesis after the field name.

The data you can enter for a field will depend on how the field's defined in the protocol, but essentially you can enter up to 80 characters of text for any *text* fields, a valid whole number for a *numeric* field and any valid choice (as defined in the protocol) for a *choice* field - choice fields will auto-complete

You don't have to make any entries for any of the fields if you don't want to.

Specifying the position of movable zones

If the protocol includes any movable zones whose position can change *within individual animals,* (i.e. the zone won't necessarily be in the same position for all of an animal's trials) then the zone (or zones) will be shown in the report and you'll be able to enter or edit their positions.

To enter a position you simply need to pick it from a drop list. If you want to you can set a zone's position to *Undefined,* which will just mean that the zone will have no position in the test and therefore won't influence the test results.

You can alter a position at any time (including after a test has been performed) but changes you make *while a test is actually running* won't affect any procedures (for the current test) which rely on the zone.

Recording notes about a test

You can record anything you like about a test in the notes section. If you cancel or reperform a test (see above) then it's usually a good idea to make some notes explaining why. Beyond this you can record any observations you like. There's no limit to the notes you can make, but if you include the notes in the Data page spreadsheet only the first 80 characters will actually be shown.

When the test notes field is active (i.e. the cursor is in the field), the *Test notes format* section of the ribbon bar will be enabled. This contains a number of options, which apply to any text that is currently selected in the notes field. (If no text is currently selected, then the formatting you select will apply to new text when you start typing).

Reset formatting	Resets the formatting of the selected text to the default formatting (i.e. removes any colour, bold, italic etc.)
Text colour	Opens a dialog box allowing you to select a colour for the selected text.
Bold	Makes the selected text **bold.**
Italic	Makes the selected text *italic.*
Underline	Makes the selected text underlined.
Increase text size	Increases the size of the selected text by one point size.
Decrease text size	Decreases the size of the selected text by one point size.

Viewing a test's results

The results of performed tests are included in this report. The list of measures shown is defined as part of the protocol and you can quickly change them by clicking the link shown just below the section's title.

Note that this report is not intended to be the main location for accessing an experiment's results (this is better done on the Results page or Data page) - rather the results are shown here to provide an easy way to view all the data for an individual test in one place.

Accessing related reports

Selecting the *Related reports* button in the ribbon bar opens a menu of related reports for the current test:

The Animal details report	This will open the report for the animal the test was performed on.
The Test track plot report	This report shows the track that the animal followed around the

	apparatus during the test.
The Test heat map report	This report shows a graphical representation of how long the animal spent in different parts of the apparatus during the test.
The Test data report	This report shows a spreadsheet detailing the events, in chronological order, that occurred during the test. You can choose exactly which events should be included as part of the protocol. For example, you can include a columns for animal's x and y coordinates, thus giving you a list of all the animal's position throughout the test.
The Test video	This will only be included if a video of the test was actually recorded by the system. If present this option will open the Test video report where you can watch the video.
The Test charts report	This will only be included if the protocol includes one or more charts. Selecting this option will display all the defined charts for the test.
The Test technical details report	This report shows various technical aspects of the test such as the frame processing statistics.

See also:

- Printing reports
- Saving reports
- Copying reports
- Sending reports by e-mail

The Test track plot report

Introduction

The track plot report shows the *track* that an animal took through the apparatus during a test.

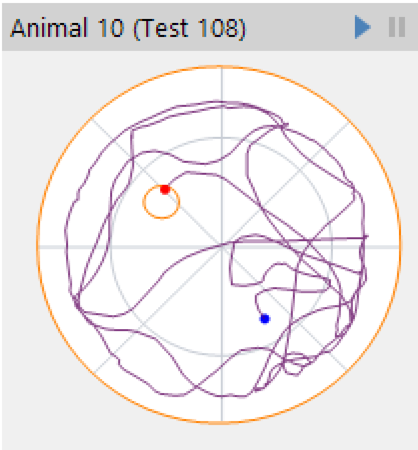


Figure 1. An example of a track plot from a water-maze experiment.

To access the report you should first access the relevant Test details report, then click the *Related reports* button in the ribbon bar and select the *Test's track plot of the centre point* or *Test's track plot of the head* from the menu which appears.

- 'Playing' a track plot
- Copying or saving a plot
- Viewing the test's results

'Playing' a track plot

The title bar of a track plot includes two buttons which can be used to 'play' the track:

Starts playing the plot, clicking the button repeatedly will increase the playback speed from x 1 to x 2 to x 4 and finally to x 8 before cycling back to x 1 again. While playing the playback position and time will be shown.

Pauses the playback, to restart click the Play button. If you click the Pause button a second time then playback will be cancelled.

While a plot is actually being played a simple *playback position bar* is displayed which show where in the test the playback has got to. You can click in the bar to jump to a particular position or drag the position indicator to 'fast-forward' or 'rewind' to an arbitrary place.

If an experiment's protocol includes a hidden zone then ANY-maze obviously won't be able to show the track while the animal is in it. For this reason when the playback reaches a point at which the animal become hidden, the word 'HIDDEN' will be displayed in the centre of the plot until such time as playback reaches a point where the animal became visible again.

If in an experiment you used keys to score behaviours which ANY-maze can't detect automatically (such as grooming or rearing) then the name of the key will be shown in the track plot next to the animal marker whenever the key was held down during the test.

Copying or saving a plot

If you want to copy or save *just* the plot, and not the entire report, right click over the plot itself and select either *Copy plot* or *Save plot...* from the menu which appears. This menu also includes a *Save plot as a movie* option; selecting it will save the plot playback as a movie, which can then be included in a PowerPoint presentation, or played in any standard media player.

Viewing the test's results

The results of the test are also included in this report. Exactly which measures are shown is defined in the protocol and you can easily alter them by clicking the link shown just below the Results section title.

See also:

- Printing reports
- Saving reports
- Copying reports
- Sending reports by e-mail

The Test heat map report

Introduction

The heat map report shows a graphical representation of how long the animal spent in different parts of the apparatus. To access the report you should first access the relevant Test details report and then click the *Related reports* button in the ribbon bar and select the *Test's heat map of the centre point* or *Test's heat map of the head* from the menu which appears.

In version 4 of ANY-maze heat maps were called occupancy plots.

- What a heat map shows
- The heat map scale
- Comparing heat maps
- Copying or saving a heat map
- Viewing the test's results

What a heat map shows

A heat map shows a graphical representation of the amount of time the animal spent in different parts of the apparatus. The map uses a range of colours to indicate time spent in an area, with blue as the shortest time and red as the longest - see figure 1.

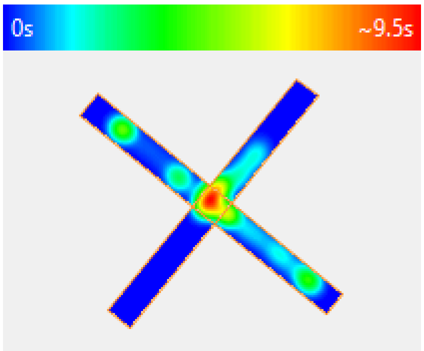


Figure 1. A heat map uses a range of colours to show how long the animal spent in different parts of the apparatus. The blue end of this range indicates less time, the red end indicates more time.

At first glance, it might seem that there's little real difference between what a heat map shows and what a track plot shows, but consider the following example: You have two tests; in the first test the animal moved slowly from the left side of the apparatus to the right; while in the second test the animal sat on the left side for 90% of the test and then ran quickly to the right. The track plot of these two tests would be identical - a track from the left to the right of the apparatus - but the heat maps would be very different. The first one would show a 'track' of more or less the same colour (probably green) across the apparatus, while the second test would show a red area on the left (where the animal sat for a long time) and then a pale blue track across the apparatus.

Another situation in which heat maps are very useful is in long-term tests, or tests where the animal moves around the apparatus a lot. In these cases, the track plot will typically become 'saturated' - like the left-hand image in figure 2. The heat map, however, never saturates and allows you to see where the animal spent most of the time in the test - as in the right-hand image in figure 2.

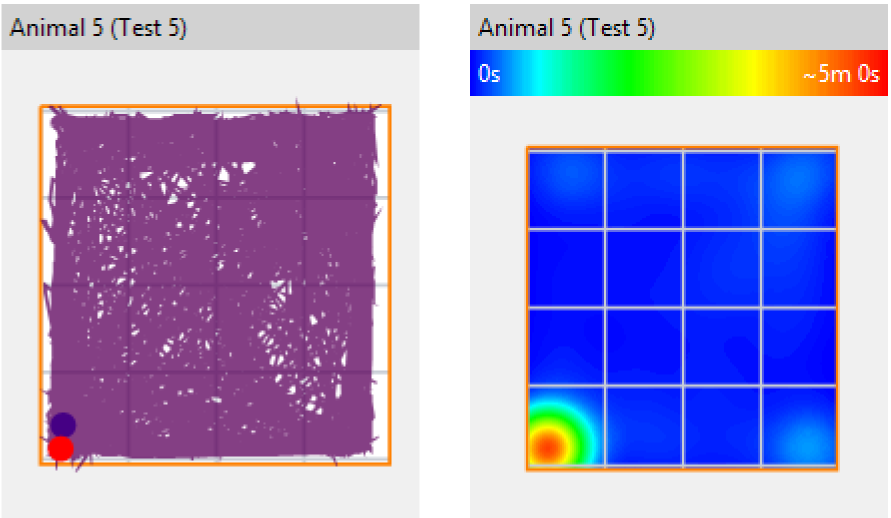


Figure 2. A track plot and heat map of the same test. In long tests track plots can become 'saturated', as occurred here, but a heat map never saturates, making it easy to see where the animal spent time during the test.

The heat map scale

At the top of a heat map is a scale, which shows the amount of time represented by the red area in the map. The scale is linear, so you can get an idea about the time represented by any colour simply by seeing how far along the scale it comes, for example yellow is about three quarters of the way along the scale bar, so if red represents 8 minutes, yellow would represent around 6 minutes.

The value shown in the scale for the red colour is *approximate* (this is why it has a ~ character before it). The reason ANY-maze can't give an exact value relates to the way that heat maps are generated, but the approximation is quite accurate and certainly serves to give an indication of how long the animal spent in different parts of the apparatus. If you want to know a precise value for how long the animal spent in a specific location, you should create a zone covering that location and then look at the result for the time the animal spent in the zone - this will be entirely accurate (see figure 3). By the way, it's worth remembering that you can add zones to your protocol at any time, including *after* tests are performed.

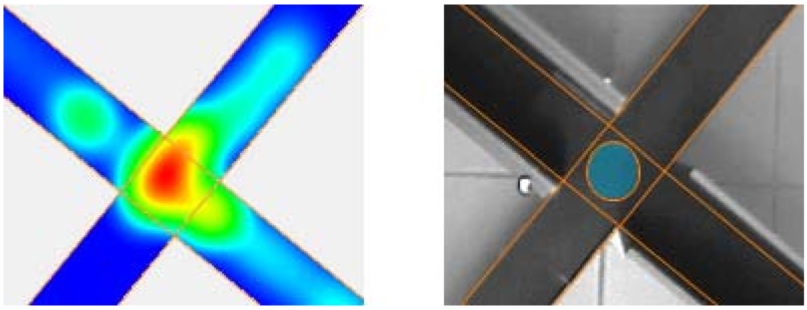


Figure 3. The left-hand image in this figure shows the same heat map as figure 1. According to the scale (see figure 1) the animal spent around 9.5s in the red area. The zone in the right-hand image covers a similar part of the maze to the heat map's red area; the time the animal spent in this zone is reported by ANY-maze as 9.4s.

Comparing heat maps

It might seem logical to assume that a heat map will show the point at which the animal spent the longest time in a test in red and the areas in which it spent no time in blue, but this has the disadvantage that plots of two tests can't be compared.

Consider the following situation: You perform two 5 minute tests; In the first test the animal spent 30 seconds in the corner of the apparatus and wandered around throughout the apparatus for the rest of the test. In the second test the animal spent 4 minutes in the corner of the apparatus and then walked across the apparatus and spent the last 30 seconds in the opposite corner.

If the plots were generated such that the point at which the animal spent the longest time *in each test* (the so called 'local maximum') was set as the 'red' point, then in the first test, this would be the place where the animal spent 30 seconds, while in the second test it would be the place where the animal spent 4 minutes. Comparing these plots would then give the erroneous impression that the animals spent the same amount of time in this position in the two tests, as they'd both be shown in red.

To address this, ANY-maze calculates the maximum across all tests in the experiment (both for the position of the head and the centre) and uses *this* maximum as the 'red' value in all heat maps (the so called 'global maximum'). Thus, in the example, the global maximum would be the point at which the animal spent 4 minutes (in the second test). This would mean that the points at which the animals spent 30 seconds in the two tests (i.e. where the animal was at the start of the first test and where it was at the end of the second one) would both be plotted in the same colour, making comparisons valid.

One repercussion of this is that the global maximum may change as you perform more tests, and this can cause the heat map of an already-performed test to change as well. Of course the plot itself doesn't change, but the colours used to represent the times do. For this reason it's best to only *compare* heat maps when you've completed all the tests in an experiment.

Copying or saving a heat map

If you want to copy or save *just* the heat map, and not the entire report then right click over the heat map itself and select either *Copy plot* or *Save plot...* from the menu which appears.

Viewing the test's results

The results of the test are also included in this report. Exactly which measures are shown is defined in the protocol and you can easily alter them by clicking the link shown just below the Results section title.

See also:

- Printing reports
- Saving reports
- Copying reports
- Sending reports by e-mail

The Test data report

Introduction

The Test data report shows a chronological list of events that occurred during a test in a spreadsheet format - see figure 1 for an example.

Time	Centre position X	Centre position Y	Speed	In NW	In SW	In NE	In SE	In Island
0:00:00.000	#N/A	#N/A	#N/A	0	1	0	0	0
0:00:00.500	225	356	#N/A	0	1	0	0	0
0:00:00.840	224	359	#N/A	0	1	0	0	0
0:00:01.570	227	374	0.068m/s	0	1	0	0	0
0:00:01.840	231	374	0.068m/s	0	1	0	0	0
0:00:02.010	238	378	0.068m/s	0	1	0	0	0
0:00:02.260	250	383	0.141m/s	0	0	0	1	0
0:00:02.590	269	381	0.233m/s	0	0	0	1	0
0:00:02.760	279	378	0.233m/s	0	0	0	1	0
0:00:02.920	291	375	0.270m/s	0	0	0	1	0
0:00:03.260	315	365	0.309m/s	0	0	0	1	0
0:00:03.430	328	358	0.333m/s	0	0	0	1	0
0:00:03.590	342	348	0.430m/s	0	0	0	1	0
0:00:03.760	356	337	0.404m/s	0	0	0	1	0
0:00:03.930	366	325	0.357m/s	0	0	0	1	0
0:00:04.100	376	313	0.357m/s	0	0	0	1	0

Figure 1. An example of the Test data report, showing the animal's position, speed and whether or not it was in certain zones.

Specifying the events shown in the report

The actual events listed in the test data report are defined as part of the protocol - full details are given in the Test data report settings topic.

Details

Time	This column shows the time at which the event occurred. Event times are recorded in ANY-maze to the nearest millisecond. You can choose exactly how the times are reported using the option on the Test data report settings page of the protocol. The options include: the time since the test start in hours, minutes and seconds; the time since the test start simply as a number of seconds (this is useful if you want to export the data); the real-time (i.e. the time shown on your wristwatch) at which the event occurred.
Event columns	The event columns shown depend on what you selected in the Test data report settings page of the protocol, but they are reasonably intuitive. Note that a 1 is used to mean a state is ON and 0 to mean it is OFF. So, for example, a column that shows whether the animal is in a certain zone will show a 1 when the animal is in the zone and a 0 when it isn't. In some circumstances a column may show #N/A as its value. This means that the value is *Not applicable*. This is shown, for example, in the x,y coordinate columns at time zero as ANY-maze doesn't know where the animal is.

Each row of the report shows details for the time shown in the *Time* column, so in the example shown above, at time 2.76 seconds the animal was at position 279,378 moving at a speed of 0.233ms and was inside the SE zone.

Transferring data from the report to other programs

The easiest way to transfer data to other programs is simply to copy the spreadsheet (or a sub-section of it) to the Windows Clipboard and then paste the data into a program such as Excel, or SPSS. You can also save data in a variety of different formats.

For full details refer to the Copying and saving data topic. Although this describes the options for copying and saving data on the Data page the spreadsheet used in this report is the same and the options available are identical.

See also:

- The Test data report settings

The Test video report

Introduction

ANY-maze includes the ability to automatically record videos of the tests performed in an experiment. When a test video has been recorded in this way, the Test details report's *Related reports* menu will include an entry for the *Test video report* where you can watch the video. (For details about how to automatically record test videos see What to record while testing).

Playing a video

When you open this report the video will start playing automatically. However, you can control the video using the buttons shown just below the video window:

Play	Restarts the video when it's paused or stopped.
Pause	Pauses the playing video.
Rewind	Rewinds the video to the start and stops playback.
Jump to label	Opens a menu listing any labels included in the video. Clicking a label will jump to that point in the recording.

Using the *Search bar* to move through a video recording

When playing a video, ANY-maze shows playback progress using a bar at the bottom of the video window. Using this bar you can scroll backwards and forwards through the video to any point.

In fact, to be strictly correct, you won't be able to scroll to *any point* because the search bar has a limited resolution based on its width, so if the bar is 1000 pixels wide and you are watching a video that lasts 4 hours, you will be able to jump to any point with, approximately, a 15 second resolution.

As an alternative to scrolling the search bar, you can simply change the playback time which is shown on the left end of the bar. To change the time, point the mouse at either the hours, minutes or seconds and roll the mouse wheel. Each 'step' of the wheel will alter the value being pointed at by 1. So, for example, to move forward 10 minutes in a video simply point at the minutes part of the time and roll the mouse wheel forward 10 steps. In fact, if you hold down the SHIFT key while rolling the wheel then the value will move forward 5 units, making it easier to move quickly.

This technique can also be used to move to an exact point in the video. For example, if you want to move to the point 2:14:00 (2 hours, 14 minutes), then you could point at the hours and roll the mouse wheel 2 steps, then point at the minutes and roll it 14 steps (as you roll the numbers change dynamically, so you don't have to count the steps).

See also:

- What to record while testing

The Test charts report

Introduction

The test chart report shows the charts for the selected test. The charts themselves are defined in the protocol.

- Altering the period of the test that the x-axis shows
- Scrolling the chart
- Copying or saving the charts
- Printing the charts
- Measuring values from the charts

Altering the period of the test that the x-axis shows

Usually the charts displayed in the test charts report show the entire duration of the test across their x-axis (although this can be altered in the protocol). This is useful as it allows you to quickly gain an idea for how the value being plotted changed across the duration of the test.

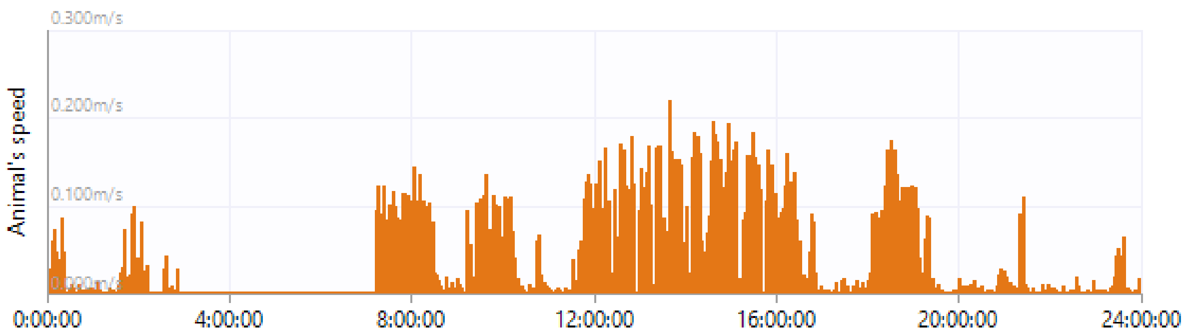


Figure 1. An example of a chart showing the animal's speed across the duration of a 24 hour test.

However, you will often want to 'zoom in' on a certain area of the chart and look at it in more detail. For example, in a 6 hour test you might see that the value being plotted changed a lot at around 2 hours and so you'd like to look at exactly what happened at this time. Zooming-in like this is easy to do, you can simply place the mouse cursor **over the x-axis** at the time of interest and then roll the mouse wheel - this will 'zoom in' on the time that the mouse is pointing at. You can also zoom back out in the same way, just roll the mouse wheel in the opposite direction.

In fact there are other ways that you can change the x-axis width and these are described in detail in the Working with charts topic.

Scrolling a chart

If you alter the width of a chart's x-axis (see previous section) then the chart will usually show just a part of the test's duration, so some parts of the trace will necessarily be off the chart to the left or to the right. To address this you can simply scroll the chart from side to side. To scroll a chart you should use the mouse to drag **the ex-axis** from side to side. Again, there are other ways to scroll the chart and these are described in detail in the Working with charts topic.

Copying or saving the charts

You can copy or save the *entire* Test chart report, simply by selecting the *Copy* or *Save* button in the ribbon bar. You can also copy or save individual charts by right clicking on the specific chart and choosing the appropriate option from the menu which appears.

Printing the charts

To print the Test chart report you should simply click the *Print* in the _ ribbonbar_.

Measuring values from the charts

It's possible to measure values directly from charts - perhaps to determine the height of a peak, or the time it took for a certain change to occur. Full details about how to do this are included in the Working with charts topic.

See also:

- Working with charts
- An introduction to charts
- Setting the default x-axis width for the Test charts report
- Printing, copying and saving charts
- Printing reports
- Saving reports
- Copying reports
- Sending reports by e-mail

ANY-maze help topic H0708

Working with charts

Contents

Altering the period the x-axis shows

Charts always start off displaying a default period on the x-axis, exactly what this is will depend on how the chart was setup in the protocol, but typically it will be the entire duration of the test. This is useful as it allows you to quickly gain an idea for how the value being plotted changed across the duration of the test, but you will often want to 'zoom in' on a certain area of the chart and look at it in more detail and there are various ways to do this:

- You can simply place the mouse cursor over the x-axis at the time of interest and then roll the mouse wheel - this will 'zoom in' on the time that the mouse is pointing at. You can also zoom back out in the same way, just roll the mouse wheel in the opposite direction.
- If you right-click over the chart a menu will open - one of the options on this menu is *Set x-axis period*, selecting this opens a sub menu from which you can select a standard period, such as 1 minute, 5 minutes, 1 hour, etc. The chart will update such that the time at the x-axis origin remains the same with the x-axis showing the specified period from that point on.
- The other way to alter the x-axis width is to use the mouse to select an area of the chart (by clicking the left button and dragging). When you release the mouse the entire chart width will be used to show just the area you selected - try it, it's more intuitive than it sounds! One word of warning though, this will alter the y-axis too.

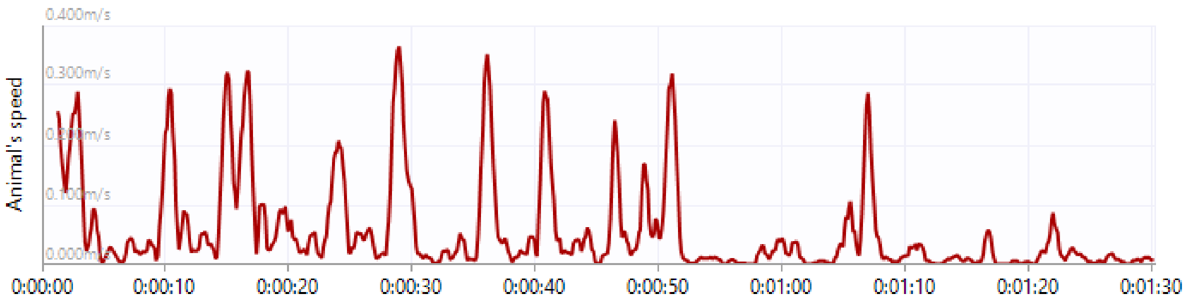


Figure 1. An example of a chart showing the animal's speed across the duration of a one and a half minute test.

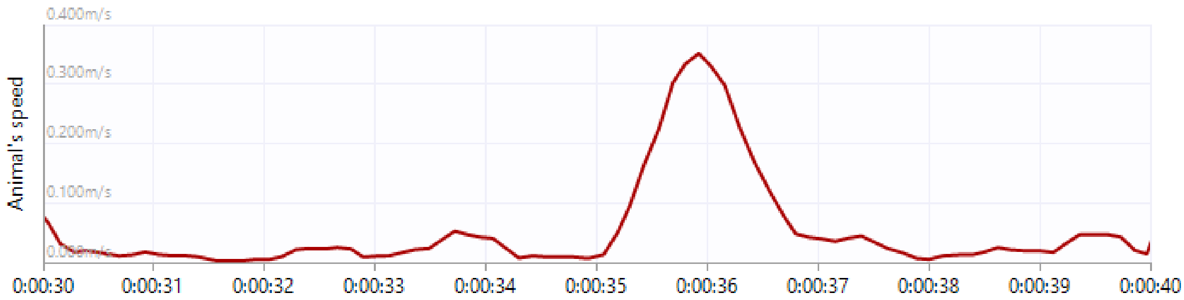


Figure 2. The same chart, but zoomed to show just the period from 30 seconds to 40 seconds.

Scrolling a chart from side to side

If you alter the width of a chart's x-axis (see previous section) then the chart will usually show just a part of the test's duration, so some parts of the trace will necessarily be off the chart to the left or to the right. To address this you can scroll the chart from side to side. To do this just move the mouse so it is over the x-axis and then drag the chart from side to side.

Altering the y-axis range

Charts usually automatically set their y-axis range so it runs from 0 to a value a little larger than the maximum value plotted. However, this might mean that you can't see details in some parts of the trace. For example, if you are looking at a chart which has small variations followed by a large peak, then the y-axis range will be large enough to show the peak, but that means you won't be able to see the detail in the small variations. To address this you can adjust the y-axis so that it covers a smaller range of values - the top of the peak will then be off the top of the chart, but you'll be able to see the detail of the variations.

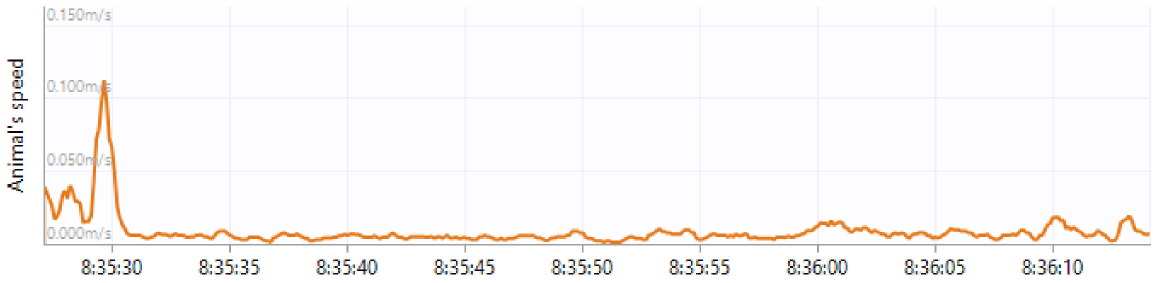


Figure 3. A chart with a large peak on the left. This has caused the y-axis scale to have a maximum of 0.15, making it hard to see any detail in the rest of the trace.

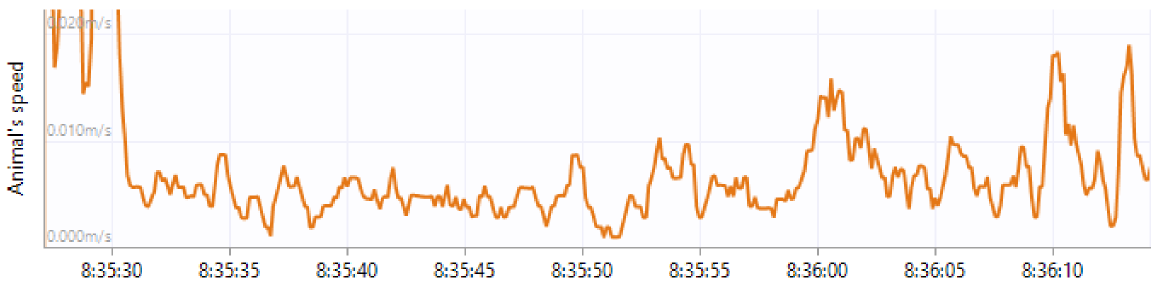


Figure 4. The same chart, but with the y-axis range reduced to have a maximum of 0.02 - the peak is off the top of the chart, but the detail in the rest of the chart can now be seen.

To change the range in this way, you need to move the mouse over the y-axis and roll the mouse wheel. Rolling it one way will 'zoom in', rolling it the other way will 'zoom-out'. As an alternative, you can use the mouse to select an area of the chart (by clicking the left button and dragging), when you release the mouse the x-axis will show just the area you selected. One word of warning though, this will alter the x-axis too.

To restore the y-axis to its default range, right-click on the chart and select *Y-axis: auto-scale* from the menu which appears.

Scrolling a chart from up and down

If you change the range of the y-axis (see previous section), the trace will probably have some parts which are 'off the top' (or 'off the bottom') of the chart. You can scroll the chart to make these parts of the trace visible, simply by dragging the y-axis up and down.

Zooming in on a part of a trace

To quickly zoom in on a specific part of the trace you can simply select it with the mouse. The x and y axes will then be scaled so that just the area of the trace that you selected is shown.

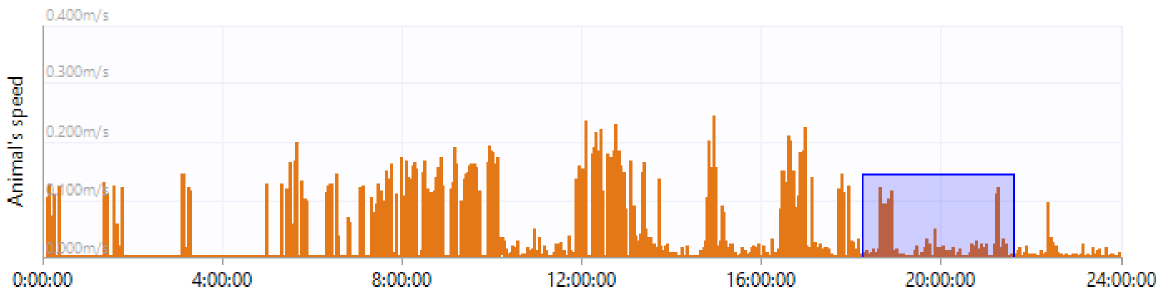


Figure 5. To zoom you just need to select a part of the trace with the mouse - this is the blue area shown here.

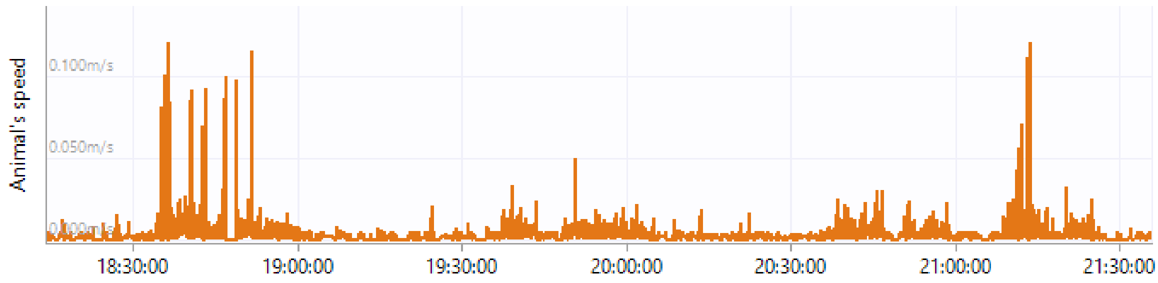


Figure 6. The same chart after the zoom. The chart now shows the just the area that was selected.

Returning to the previous view of a chart

After changing the range of either axis, scrolling or zooming-in you'll often want to return to the view of the chart you had before, fortunately this is easy to do - just right-click and select *Previous view* from the menu which appears.

Measuring values from a chart

In some circumstances you may want to measure a value from a chart. For example, you might be looking at a chart which shows to periods of 'activity' (like the chart in figure 7) and you may wish to know how long the gap between these two periods was.

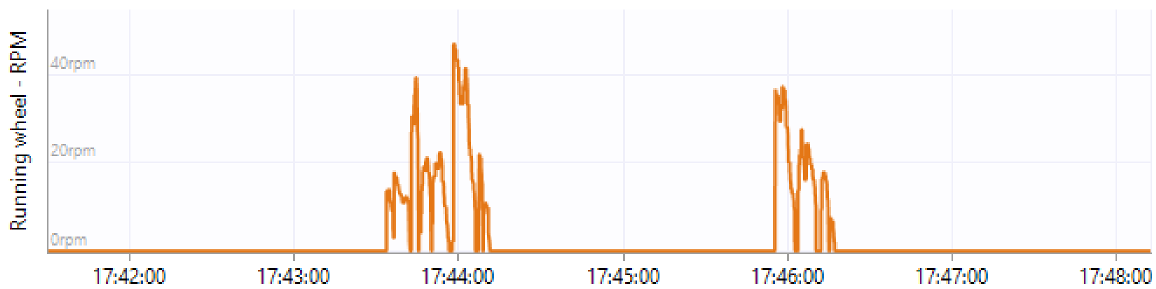


Figure 7. A chart showing two periods of 'activity'.

To find this out you can switch the chart into measurement mode by right-clicking on the chart and then selecting *Measurement mode* from the menu which appears. When in measurement mode you can use the mouse to select areas of the chart and view their details. For example, in figure 8, I've selected the area between the two periods of activity and I can see that this area covers the period from 17:44:12 (that's 17 hours, 44 minutes, 12 seconds) to 17:45:55, which is a period of 1 minute 43s.

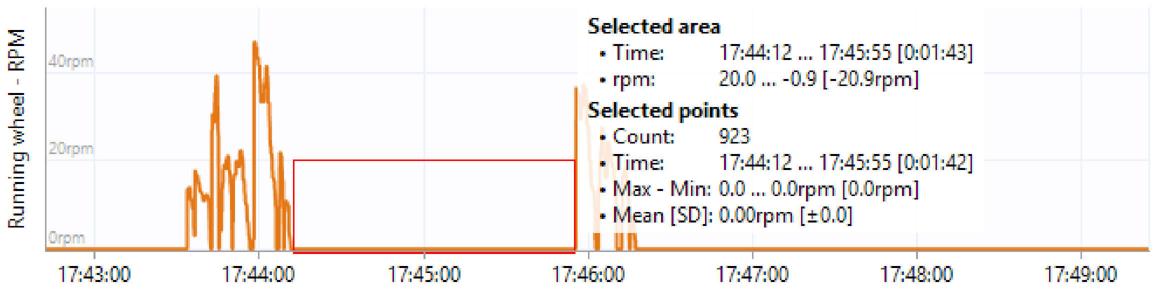


Figure 8. Using the chart 'Measurement mode' to measure the time between the periods of activity. The measurement data for the area selected shows that this covers a period of 1 minute 43 seconds.

The measure information shown for a chart describes two things: the *area* that was selected in the chart and the *chart points* which fall within that area. The difference here is that the area describes the area of the chart surface that was selected - this need not even have any of the trace in it, but it can still be reported as being a certain time in width and a certain value (in whatever the y-axis units are) in height. The points on the other hand, describe the actual data from the trace. Sometimes it's useful to see these points, which you can do by right-clicking over the chart and selecting *Show points* from the menu which appears (note that you must be in measurement mode for this option to appear on the menu).

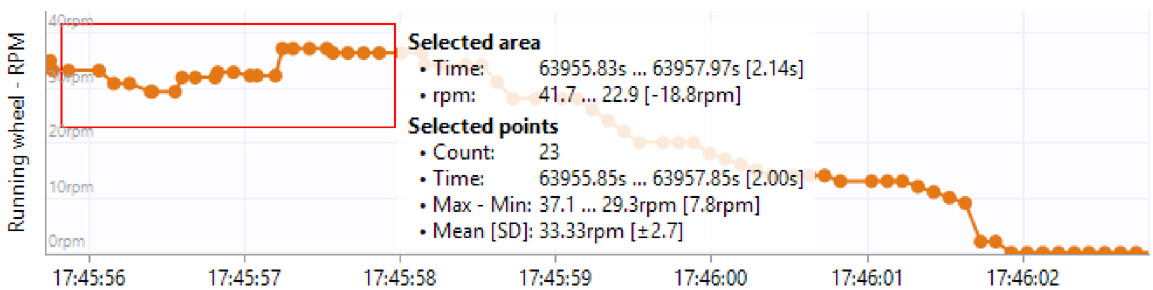


Figure 9. A chart in which the points are being shown. 'Measurement mode' shows us that the points within the selected area (the red box) cover a period of 2 seconds and have a range of 37.1 to 29.3 and an average of 33.33.

If you right click on the measurement data you'll find that the menu includes an option to *Copy measurements* which copies the measurements to the clipboard.

Copying or saving a chart

You can copy or save all the charts shown in a report by right-clicking on any of the charts and selecting *Copy all charts* or *Save all charts* from the menu which appears. If you want to copy of save a specific chart then you should right-click on that specific chart and then select *Copy this chart* or *Save this chart* from the menu which appears.

Charts can be saved in a variety of formats, which are described in detail in the Saving charts topic.

Printing a chart

Charts can't be printed on their own, but they can be printed as part of a report, such as the Tests chart report. In this case just selecting the *Print...* button in the ribbon bar will print the report.

See also:

- Saving charts

The Test technical details report

Introduction

The Test technical details report lists a variety of technical information about a particular test and the tracking that ANY-maze performed in it.

Details

Test date, time & user	Shows the date and time at which the test STARTED and the user who was logged on to ANY-maze during the test.
Apparatus details	Shows the name of the apparatus used in the test, the name of the video source which provides images for the apparatus, the name of the interface, camera or video file which supplied the images, as well as details of the physical digitiser board or camera that the interface represents.
Image processing details	Reports the number of frames that ANY-maze processed during the test - this depends on the frame rate of the device and the speed of your computer. For example, with multiple cameras operating at a high frame rate (30 frames per second) a slow computer might 'drop frames' and may only process perhaps 15 frames per second. Also shows the number of frames successfully tracked - this is the number of processed frames in which ANY-maze successfully identified the animal. In images with lots of reflections and or noise this value may be substantially less than the number of frames processed.
Recording details	Reports the period for which ANY-maze recorded results (i.e. the test duration) and the number of frames that were actually recorded for analysis in the test's results. This number may be less than the number of frames successfully tracked as frames are recorded at a maximum frequency specified in the protocol's tracking options. For example, if ANY-maze tracks in 15 frames per second but the recording frequency is set to just 8 frames per second then only half the frames in which the animal is tracked will actually be recorded.

See also:

- Printing reports
- Saving reports

- Copying reports
- Sending reports by e-mail

Running tests

Introduction

Generally speaking, running tests in ANY-maze is quite straight-forward, however you will inevitably come across problems - the apparatus will be misaligned with the apparatus map, an animal might jump off a plusmaze, you'll start a test accidentally, etc.

In this section you'll find links to topics which not only explain how to perform tests but also explain how to resolve the problems that you might encounter.

General information about running tests

- Buttons and keys for controlling testing
- Testing do's and don'ts
- Resolving common problems when testing

Before starting a test

- What to do if the apparatus map isn't aligned with the video image
- Specifying what's displayed during a test
- Setting the positions of movable zones before a test
- Skipping a test
- Skipping an entire stage for an animal
- Adding a test to a manual schedule
- Deleting a test from a manual schedule

Running a test

- Starting tests
- Running tests which include an accustomisation period
- Controlling trials which are set to start automatically
- Pausing a running test
- Recording behaviours during a test using keys
- Restarting a test ANY-maze that has ended automatically
- Stopping a test manually

After testing

- Undoing a performed test
- Cancelling a performed test
- Reperforming a performed test
- Adding additional scoring to a performed test
- Ending an animal's tests in a stage manually
- Retiring an animal from an experiment
- Deleting an animal from an experiment

General information about running tests

Introduction

This section of the help provides general guidance on running tests, together with details of the function of each of keys and buttons used to control tests.

- Buttons and keys for controlling testing
- Testing do's and don'ts
- Resolving common problems when testing

Buttons and keys for controlling testing

Introduction

In general tests are controlled using the buttons shown just above the video picture of the apparatus. However, in some situations you'll probably find it's more convenient to use a test control key or a test control switch.

- Controlling tests using the toolbar buttons
- Controlling tests using a key or switch

Controlling tests using the toolbar buttons

The actions of most of the toolbar buttons depend on the current test *status*, for example, the stop button has a different action if you click it when *Waiting to start...* than it does if you click it when *Testing...*

Below is a full list of all the buttons and their actions for each status. Although this list may appear rather complicated, button's actions are usually quite intuitive in context:

Add a test

*Note: This button is only displayed when **manually scheduling tests.***

Ready...	Opens the Add test window where you can add a test for an animal by entering its animal number. In fact you can enter multiple animal numbers, so as to add more than one test at a time.
All other statuses	*No action*

Start test

Ready...	When using auto-start, sets the status to *Waiting to start...*, when the user leaves the camera's field of view the test will start. When not using auto-start, starts the test. See Starting tests
Waiting to start...	Starts the test. Useful in the unlikely event that auto-start fails to start the test.
Accustomising	*No action*
Testing	*No action*
Paused	*No action*

Waiting to end test...	Causes the test to continue as if the event which could have caused the test end had never occurred.
Set zone's position	Same as *Ready...* status action.

Pause test	
Ready...	*No action*
Waiting to start...	*No action*
Accustomising	*No action*
Testing	Pauses the test. The test clock is stopped and the animal's position stops being recorded, however the system will continue to track the animal so as not to lose it.
Paused	Restarts the test.
Waiting to end test...	*No action*
Set zone's position	*No action*

End test	
Ready...	*No action*
Waiting to start...	Returns to *Ready...* status.
Accustomising	Cancels the accustomisation period and returns to the *Ready...* status.
Testing	Asks you whether you want to end the test and save its results or end the test and set it back to the beginning as if it had never been started. While the message is displayed tracking continues as normal. You can also just cancel the message and continue the test. Note that this message can be suppressed using an option in the protocol.
Paused	Same as *Testing* status action.
Waiting to end test...	Confirms the end of the test.
Set zone's position	*No action*

Undo last test	
Ready...	Asks you to confirm that you want to undo the last test performed - if confirmed the test is set back to *unperformed* and the test results are **irrevocably** lost.

All other statuses No action

Set zone positions

Note: This button is only displayed when the protocol includes one or more movable zone.

Ready... Cycles through the movable zones defined in the protocol prompting you to specify the position of each one.

All other statuses No action

Take background image

Note: This button is only displayed when the protocol specifies that the animal will be in the apparatus before the start of the test.

Ready... If the protocol specifies that the animal will be in the apparatus before the start of the test, you should use this button to take a picture of the apparatus *before* you put the animal in. Thus ANY-maze will have a picture of the empty apparatus which it can use to detect the animal's location when the test does start. Note that you can click the button as often as you like, each time you click it a new background image will be captured, replacing any picture you've already taken.

All other statuses No action

Change the video file being played

Ready... Causes the Choose video window to open where you can choose a video file to play. Clicking the button a second time will restore the default video source (as specified in the protocol). For more details refer to Tracking in a video of a test.

All other statuses No action

Adjust the apparatus map

Ready... Switches to Map adjustment mode. In this mode you can adjust the apparatus map so it correctly aligns with the apparatus in the picture.

All other statuses No action

Pan the video image

Ready...	Switches to *Image panning mode*. In this mode you can adjust the part of the image which is visible in the live video window - perhaps shifting it so that the apparatus becomes fully visible. Note that this button is disabled if the entire video image is already visible - which will, in fact, normally be the case.
All other statuses	*No action*

Record a movie for a presentation

All statuses	Starts recording a standard format video file of the apparatus, showing **exactly** what you are seeing on the screen - i.e. the video with the tracking overlaid. For more details refer to: Recording movies of tests for inclusion in a presentation.

Show keys

Note: This button is only shown if the protocol includes at least one key.

All statuses	Opens a small window which lists any keys defined in the protocol, together with their keystrokes. This can be used to check which key stroke is being used to record which behaviour.

Open test debug output window

Note: This button is only shown if the protocol includes at least one procedure or has any I/O.

All statuses	Opens the test debug output window. This window is fully described in the Debugging procedures topic.

Clear test debug output window

Note: This button is only shown if the protocol includes at least one procedure or has any I/O.

All statuses	Clears the contents of the debug output window. This button is only enabled when the debug output window is open.

For details of the buttons on the right of the toolbar refer to Recording videos of tests.

Controlling tests using a key or switch

When defining a piece of apparatus in the protocol you can optionally specify one of the computer's 'F' keys (F1 - F12) as the apparatus's test control key and/or an input switch as the test control switch.

The exact action of a test control key (or switch) depends on the apparatus's status as follows:

Ready...	When using auto-start, sets the status to *Waiting to start...*, when the user leaves the camera's field of view the test will start. When not using auto-start, starts the test. See Starting tests
Waiting to start...	Starts the test. Useful in the unlikely event that auto-start fails to start the test.
Accustomising	Ends the accustomisation period prematurely - i.e. as if the full period had passed. What happens next depends on how the accustomisation period has been defined, either the test will start or the status will change to *Waiting to start...*
Testing	Acts like the *Stop* button - i.e.: Asks you whether you want to end the test and save its results or end the test and set it back to the beginning as if it had never been started. While the message is displayed tracking continues as normal. You can also just cancel the message and continue the test. Note that this message can be suppressed using an option in the protocol. Also note that you will only be able to end a test using a test control *switch* if the apparatus element in the protocol has been set to allow this.
Paused	Restarts the paused test
Waiting to end test...	Acts like the *Start* button - i.e.: Causes the test to continue as if the event which could have caused the test to end had never occurred.
Set zone's position	Same as *Ready...* status action.

Testing do's and don'ts

Introduction

To ensure ANY-maze testing works smoothly there are a few things it's good to be aware of. In fact most of them are either obvious or don't really require any change from what you would probably do anyhow, but it's still good to know what they are.

- A general point - ANY-maze is always watching
- When you start ANY-maze (or when you switch the cameras on) try to make sure no one is in the camera's view
- Before a test starts there should be a period of at least 4 seconds during which the apparatus is "empty" and no one is in the camera's view
- Don't leave things behind in the camera's view when you start a test
- Once a test has started you CAN walk around in the area outside the apparatus
- If your apparatus has a transparent lid or cover of some kind then keep it closed ALL THE TIME except, obviously, when you're putting an animal in or taking an animal out of the apparatus.

First a general point - ANY-maze is always watching!

You might think that ANY-maze is only analysing images while a test is running but this isn't actually the case. It's always analysing the video pictures it receives and it uses them to better understand your apparatus. This fact is partly responsible for some of the other recommendations given below.

When you start ANY-maze (or when you switch the cameras on) try to make sure no one is in the camera's view

This is not critical and usually *just happens anyway* because to start ANY-maze you'll need to be at the computer and, ideally, the computer won't be in the camera's view.

If there is someone visible when you start ANY-maze then, when the person LEAVES the camera's view, a 'person' icon will probably be shown in the top-left corner of the video picture (usually this icon is displayed when someone ENTERS the camera's view) - just click the icon and you'll fix the problem.

Before a test starts there should be a period of at least 4 seconds during which the apparatus is 'empty' and no one is in the camera's view

In order for ANY-maze to quickly identify the animal at the start of a test it should have had a chance to see what the empty apparatus looks like before the test starts (this is one of those things it's analysing when tests aren't running). Here 'empty' means not just that there's no animal in the apparatus but also that you are not anywhere in the video picture either.

In fact this is another thing which usually *just happens anyway* because you obviously have to get the animal before you start a test and this usually takes at least 4 seconds and while you're doing it you probably won't be visible to the camera.

Don't leave things behind in the camera's view when you start a test

Generally, at the start of a test you will walk into the camera's view, place an animal on the apparatus and then walk away again - as ANY-maze sees you do this it will start the test.

However, if you leave something behind in the camera's view (such as the animal's box) then ANY-maze will probably think there's still someone in the picture and it won't start the test.

Note: If you don't intend to use the auto-start feature then you don't need to worry about this.

Once a test has started you CAN walk around in the area outside the apparatus (although there are usually good scientific reasons for avoiding this)

If you reappear in the camera's view AFTER a test has started you won't interfere with the tracking. However, you should be careful not to come between the camera and the apparatus as this will almost certainly have an adverse effect.

If your apparatus has a transparent lid or cover of some kind then keep it closed ALL THE TIME except, obviously, when you're putting an animal in or taking an animal out of the apparatus

This is rarely an issue as most apparatus doesn't have a lid or cover; but if it does, and the camera has too look through it to see the animal, then it's important that it's always looking through it even when tests aren't running - again this is related to the fact that ANY-maze is always analysing the apparatus and it needs to be looking at the apparatus as it will appear while tests are running, i.e. with any lid or cover closed.

Before starting a test

Introduction

This section of the help provides guidance on how to perform common tasks before running a test. It also describes common issues which can arise at this point and describes how to cope with them:

- What to do if the apparatus map isn't aligned with the video image
- Specifying what's displayed during a test
- Setting the positions of movable zones before a test
- Skipping a test
- Skipping an entire stage for an animal
- Adding a test to a manual schedule
- Deleting a test from a manual schedule

Adjusting the apparatus map

Introduction

Clearly it's important that the apparatus map aligns correctly with the apparatus shown in the live video picture, as ANY-maze uses the map to determine where in the apparatus the animal is.

Nevertheless, it's easy to *lose* this alignment by moving the apparatus, perhaps when cleaning the room, or by adjusting the camera's position, focus or zoom - see figure 1.

Fortunately, resolving this issue is extremely simple. You don't need to drag the apparatus around the floor - instead, you can drag the apparatus map around the screen.

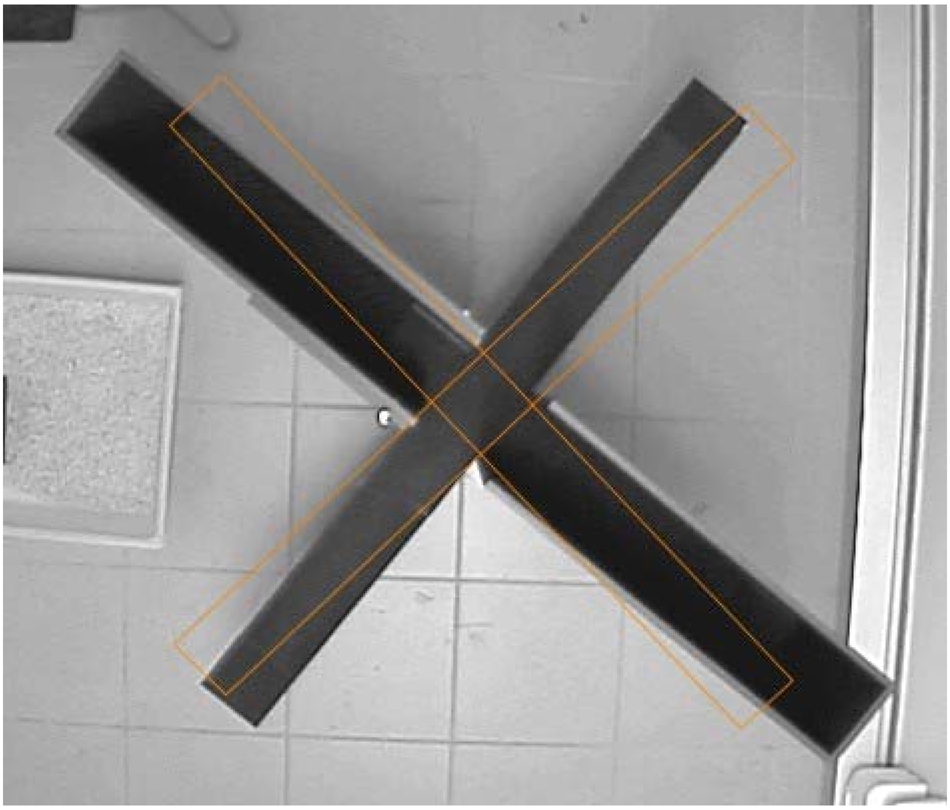


Figure 1. The plusmaze in the video picture doesn't align correctly with the apparatus map. The map needs to be moved a little to the left and rotated slightly.

Details

To adjust the apparatus map you simply need to click the button displayed above the apparatus video picture on the *Tests* page. This may cause a message to be displayed advising you that you're about to enter *Map adjustment mode*, just click OK.

In map adjustment mode you can move the map (i.e. drag it), rotate the map or even resize it. All these operations are performed using the mouse. Basically ANY-maze divides the video picture into nine sections (see figure 2) and the operation that the mouse will perform depends on the section it's in - to make this clearer the mouse pointer changes to indicate the operation.

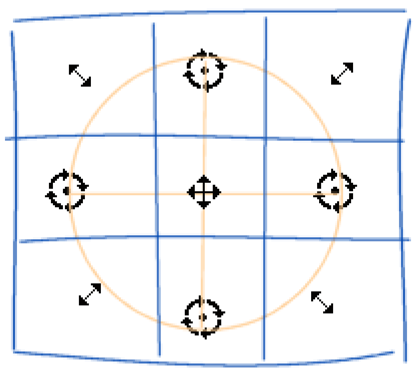


Figure 2. The type of adjustment performed depends on where in the video picture you position the mouse pointer.

Mouse pointer styles and the adjustments they perform:

Holding the left mouse button down and moving the mouse drags the map.

Holding the left mouse button down and moving the mouse rotates the map around its centre point.

Holding the left mouse button down and moving the mouse resizes the map. This can be useful if the camera has been moved closer or further from the apparatus, if you've adjusted the lens focus or if you've adjusted the zoom on a vari-focal lens.

One point to be aware of is that the map is constrained by the edges of the video image (well, to be strictly accurate it has to stay inside a margin of 4 pixels) so you won't be able to perform any adjustments which would take it outside this limit.

Exiting map adjustment mode

To exit map adjustment mode simply click the button again.

Resetting adjustments

You can always undo the adjustments that are applied to the apparatus map by entering map adjustment mode, right-clicking on the video picture and selecting "Reset adjustments" from the menu which appears.

How ANY-maze uses the adjusted map

The way ANY-maze applies map adjustments is probably not quite as you'd expect. Although the adjusted map is shown (and used) while tracking, what ANY-maze actually does at the end of the test is to 'un-adjust' the animal's track before saving it - essentially this means that the recorded track has the inverse of the apparatus adjustments applied to it which maps it to the 'standard', unadjusted apparatus map, i.e. the map you drew as part of the Protocol.

The reason ANY-maze works this way is so that all recorded tracks are relative to the same apparatus map. This is required by the ANY-maze results analysis system and also makes it much simpler for you to alter the map at any point in the experiment, perhaps to add a new zone.

Anyhow, this doesn't actually make much difference except when you look at a test's track plot - there you'll find that the plot is drawn relative to the *unadjusted map*.

Specifying what's displayed during a test

Introduction

While a test is running ANY-maze can:

- Indicate the animal's centre point
- Indicate the position of the animal's head and/or tail
- Highlight the part of the image it considers to be the animal
- Indicate the zone(s) which it considers the animal to be in
- Indicate which, if any, 'behaviours keys' are pressed
- Indicate whether it considers the animal to be immobile
- Indicate whether it considers the animal to be freezing

With a few exceptions you can choose any combination of these indicators or, if you like, none at all.

Indicating the animal's centre point

ANY-maze will indicate the centre of the animal using a small orange dot. This dot will only actually be shown if you requested it in the What to display while testing element of the protocol.

If the centre point isn't displayed during a test then you can have it shown by right clicking the video picture and selecting *Show animal marker* from the menu which appears. Note, however, that this will only change the **current** test - when the next test starts the dot won't appear. To change it so the centre point's always displayed you should alter the options in the What to display while testing element of the protocol.

Indicating the position of the animal's head and/or tail

If you have chosen to track the animal's head and tail, ANY-maze will indicate the position of the animal's head using a small green dot and the position of the tail using a small yellow dot. In fact, these dots will only actually be shown if you requested them in the What to display while testing element of the protocol.

If the head or tail point isn't displayed during a test then you can have it shown by right clicking the video picture and selecting *Show head marker* or *Show tail marker* from the menu which appears. Note, however, that this will only change the **current** test - when the next test starts the dot won't appear. To change it so the head and/or tail point is always displayed you should alter the options in the What to display while testing element of the protocol.

When tracking the head, you also have the option to show lines which join the animals head and tail to the centre point and/or to show the animal's orientation - see figure 1. Again, these options can be set using the What to display while testing element of the protocol.

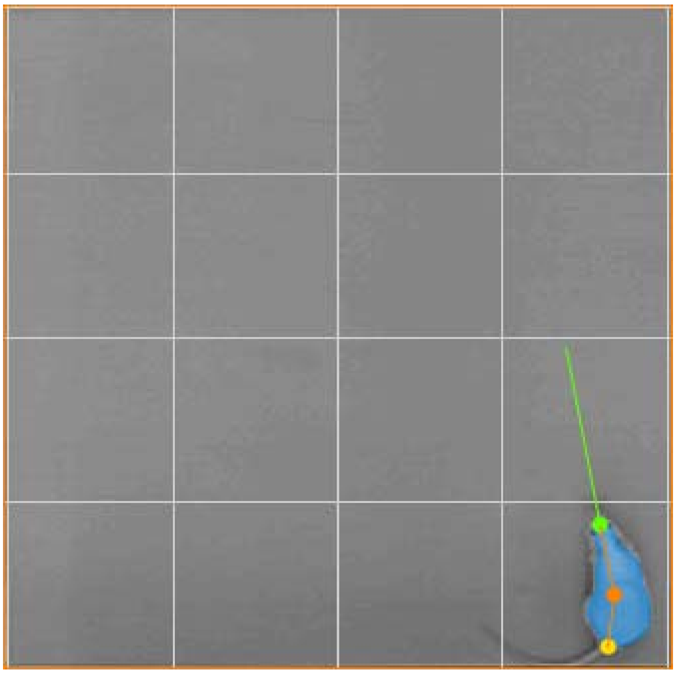


Figure 1. In this test all possible animal markers are shown: The centre, head and tail points; lines which join the head and tail to the centre; and the animal's orientation, which is indicated by the line extending forward from the animal's head.

Highlighting the part of the image ANY-maze considers to be the animal

ANY-maze can indicate the entire area of the animal by shading it in blue (see figure 1, above). This is another option which is specified in the What to display while testing element of the protocol.

You can show the animal area in a test in which it's not displayed by right clicking the video picture and selecting *Show animal area* from the menu which appears. This will only effect the **current** test, to permanently change the setting you should alter the What to display while testing element of the protocol.

Indicate the zone(s) which ANY-maze considers the animal to be in

ANY-maze can indicate which zone (or zones) it considers the animal is in by highlighting them in green and/or display the zones' names next to the position of the animal. This can be very useful if you want to check that zone entries are being scored correctly, especially when they're based on the entire area of the animal.

To specify that a zone is highlighted you should check the box labelled *Highlight this zone during tracking*... which is shown on the Zone settings page of the protocol.

As you can specify zone highlighting on a zone by zone basis, you can choose to switch it on for some zones and off for others. This can be very helpful if two zones share certain parts of the apparatus, or if one zone is entirely enclosed in another one - in these cases highlighting all zones can become quite confusing.

There's no method to temporarily switch highlighting on (because it has to be switched on for each zone separately) but you can temporarily switch it *off* in a test by right clicking the video picture and removing the check mark from the *Show zone highlighting* option on the menu which appears. If you want to permanently switch highlighting off you should alter the definition of the appropriate zones in the protocol.

Indicating which, if any, 'behaviour keys' are pressed

If an experiment's protocol includes any keys that are being used to score behaviours that ANY-maze can't detect automatically (such as grooming or rearing), then you can choose to have the behaviour's names displayed next to the animal while the relevant key is pressed (i.e. while the behaviour is occurring). This acts as a nice confirmation that the system is registering the behaviour and can also help you to remember that you're holding the key down!

You specify whether the names of active behaviours are shown using the What to display while testing element of the protocol. Unlike for the previous indicators there is no method to temporarily alter this setting.

Indicating whether ANY-maze considers the animal to be immobile

As part of the tracking options you can specify whether ANY-maze should detect periods when it considers the animal to be immobile. If you choose to do this then it's a good idea to also specify that it should indicate immobility, which it does by displaying the text "IMMOBILE" next to the animal marker.

Indicating whether ANY-maze considers the animal to be freezing

As part of the tracking options you can specify whether ANY-maze should detect periods when it considers the animal to be freezing. If you choose to do this then it's a good idea to also specify that it should indicate freezing, which it does by displaying the text "FREEZING" next to the animal marker.

Setting the positions of movable zones before a test

Introduction

If an experiment's protocol includes some movable zones (i.e. zones whose positions can be different in different tests) then you may find that you need to specify the position of the zone(s) before you start each test. However, this will only be the case if the protocol specifies that the zone's positions can vary *within* the animals (i.e. the position won't necessarily be the same for all the tests for a particular animal) and the protocol doesn't explicitly specify what the positions of the zone should be in each trial.

Details

Specifying the position of a zone

ANY-maze will automatically ask you to specify the position of any movable zones which can vary *within* an animals' tests and which don't have their positions defined in the protocol. It will do this before the start of **every** test. Specifically, the apparatus will show a status message of the form :
Animal 1, Training stage trial 1: Please click on the xxx's position

In the apparatus image the possible positions which the relevant zone can adopt will be shaded in purple and you simply need to click on whichever position the zone is in for this test - when you click a position it will change to be shaded in blue. You can change the position you've selected by simply clicking a different one.

Once you've specified the position of the zone you should click the button. If the protocol includes another zone whose position needs to be specified then you'll be asked to specify its position, otherwise the test will be set to the '*Ready*' state and you can start it in the normal way.

ANY-maze remembers the positions of movable zones between tests

ANY-maze will remember the position of all movable zones from one test to the next. This means that when you start the next test the position of any movable zone will already be shown where it was in the previous test, if it's not changed then you don't need to do anything.

Starting tests directly

You don't actually need to use the button to get to the 'Ready' state in order to start testing as ANY-maze will still respond to clicks on the *Start test* button when the status is asking you to specify the position of movable zones. This is useful when the position of a movable zone hasn't changed (see previous item) as it means that you can start the test immediately, without having to click the button first.

It's not strictly necessary to specify the position of movable zones before performing a test

In fact, you don't *have* to specify the position of any zones before you perform a test - you can always set them afterwards using the Test details report - however there is one very important point to consider: If a movable zone is used by any procedures and its position hasn't been specified, then the procedure probably won't work in the way you expect. For example, if the *island zone* in a water-maze is movable and you've created a procedure to detect when the animal enters the zone and to then end the test - if the island's position isn't specified, clearly the procedure will never end the test. For this reason it's usually best to specify the positions before running the tests.

Weighing the animal before a test

Introduction

All protocols include an option to record the weight of your animals. If you choose to do this, then you can either weigh the animals and type in their weights manually, or you can weigh them using the ANY-maze Animal scales, in which case ANY-maze will automatically record their weights. You can also choose whether to record an animal's weight before every one of its tests (assuming it will be tested multiple times) or just before the first test it has in the experiment. These options are set up using the Animal weights element of the protocol.

Depending on the options you choose, you may need to weigh/record animal weights before some, or all, of the tests you perform in an experiment:

- Recording animal weights manually
- Weighing animals using the ANY-maze Animal scales
- Weighing animals when you have multiple apparatus
- Weighing animals before every test
- Weighing an animal again
- Sorting out problems when weighing animals

Recording animal weights manually

If you chose to record animal weights, whether manually or using the ANY-maze Animal scales, then ANY-maze will automatically include a field on the Test details report for the animal weight.

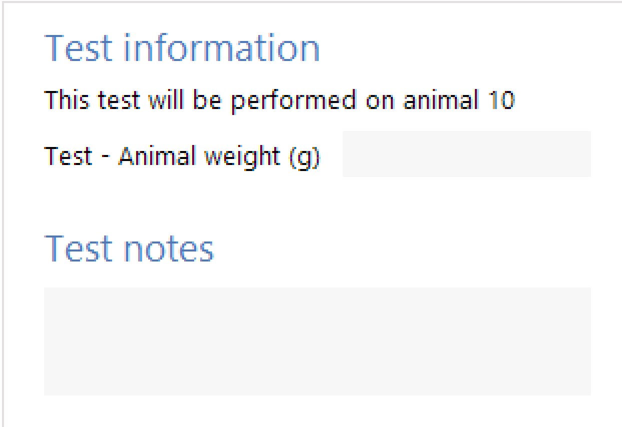
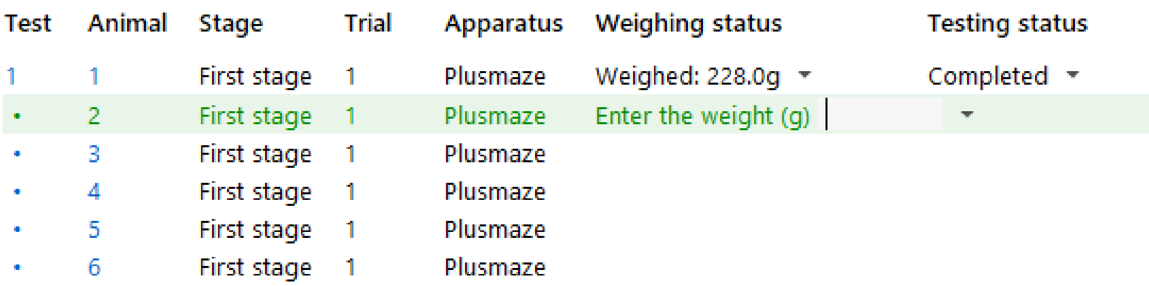


Figure 1. The animal weight field on the Test details report.

When recording the weight manually, all you have to do is type the weight into this field. In fact, if in the protocol you selected the option to Prevent tests from starting until the animal has been weighed then before a test you will see the following status:



Test	Animal	Stage	Trial	Apparatus	Weighing status	Testing status
1	1	First stage	1	Plusmaze	Weighed: 228.0g	Completed
•	2	First stage	1	Plusmaze	Enter the weight (g)	
•	3	First stage	1	Plusmaze		
•	4	First stage	1	Plusmaze		
•	5	First stage	1	Plusmaze		
•	6	First stage	1	Plusmaze		

Figure 2. Status shown when the animal weight must be entered manually before a test can be performed.

Clicking the status message will open the Test details report ready for you to type in the animal's weight.

Note that you'll only see this status if the weight has not already been entered. Also, you'll see it only before the animal's *first* test, if the protocol is set to Manually enter animal weights for each animal (as opposed to entering them for each *test*).

Weighing animals using the ANY-maze Animal scales

Using the ANY-maze Animal scales to weigh your animals has the advantage that ANY-maze will automatically store the animal weight for you, so you just need to weigh the animal and then move straight on to testing it.

If you have just one piece of apparatus and you have selected the option in the protocol to Prevent tests from starting until the animal has been weighed then before a test you will see the following status:

Test	Animal	Stage	Trial	Apparatus	Weighing status	Testing status
1	1	First stage	1	Plusmaze	Weighed: 228.0g	Completed
•	2	First stage	1	Plusmaze	Weigh the animal	Waiting for the animal to be weighed
•	3	First stage	1	Plusmaze		
•	4	First stage	1	Plusmaze		
•	5	First stage	1	Plusmaze		
•	6	First stage	1	Plusmaze		

Figure 3. Status shown when the animal must be weighed before a test can be performed.

You just need to put the animal into the weighing basket (which should already be on the scales) and ANY-maze will weigh it and then store its weight. After the animal has been weighed, removing it from the basket causes the apparatus to move to the *Ready...* state - ready to begin the test

Figure 4. Status shown after an animal has been weighed.

If you haven't selected the option to Prevent tests from starting until the animal has been weighed then ANY-maze won't automatically expect you to weigh the animal before the test, rather it will go straight to the *Ready...* status, ready to begin the test. However, you can still weigh the animal, you just need to click the *Weigh animal* button.

Things to be aware of when using the ANY-maze Animal scales

1. When you first power-up the scales, they will automatically 'zero'. It's best to zero WITHOUT the animal basket on the scales, so if the basket was on the scales at power up, you should remove it and then press the scales' 'Zero' key. You just need to do this one before each session.

2. With the scales correctly zeroed (see previous paragraph) you need to 'Tare' them. To do this, place the animal basket on the scales (with its lid, if you use one) and then press the 'Tare' key - the display will then read 0.0g with the word NET shown in the bottom left. At this point, you're ready to weigh animals.
3. In between tests, you may find that the scales need to be tared again. For example, if you weigh an animal and then remove it from the animal basket, the animal may leave some urine or faeces behind. This will probably only weigh a gram or two, but to weigh the next animal accurately, you will either need to clean out the basket, or tare the scales. ANY-maze will attempt to detect this situation, and instead of asking you to weigh the animal, will suggest that the scales probably need tareing. Simply ensure that the basket is empty (and, if you're using a lid on your basket, that the lid is **on** the basket) and press the 'Tare' button on the scales.

   If you tare the scales, and then later clean the basket, the weight of the urine or faeces will disappear and the scales will register a small negative weight. Again, ANY-maze will detect this and warn you before asking you to weigh the animal.

   Note that there are some circumstances under which ANY-maze is unable to determine accurately whether the scales really *do* need to be tared. In these cases, you can simply ignore this message and continue the weighing process. For example, if you are using a lid that is a simlar weight to your animals, then removing the lid and putting the animal in the basket will cause this message to appear. In this case, just carry on and put the lid on the basket, and ANY-maze will continue the measurement process and weight the animal accurately.

   This process does require that the basket is left on (or is returned to) the scales between animals. If, for example, you weigh an animal, then remove the basket from the scales, remove the animal from the basket (with it leaving some faeces behind) and you then place another animal in the basket and put the basket back on the scales, then the weight of the faeces left behind by the first animal *will* be added to the weight of the second animal - this is because the scales never had a chance to tare with just the basket on the scales.

Weighing animals when you have multiple apparatus

When you have multiple apparatus and they all need to weigh an animal, then the status of each one will look like this:

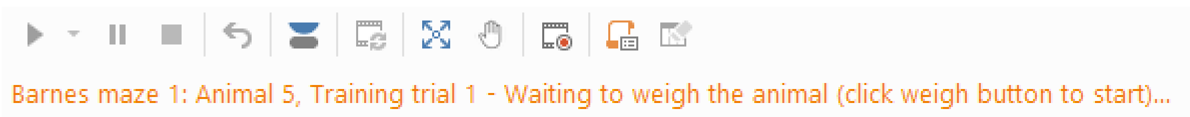


Figure 5. Status shown when testing in multiple apparatus and the animal must be weighed before a test can be performed.

You just need to click the *Weigh animal* button in the apparatus whose animal you intend to weigh and then proceed as described in the previous section.

If you change your mind about which apparatus you want to weigh the animal for, then just click the *Weigh animal* button in a different apparatus.

In experiments in which the animals will be tested in multiple trials, you may want to record their weight before every trial, or you may want to just record their weight once. You can specify which option you want to use, in the Animal weights element of the protocol.

If you choose to just weigh the animals once and you include the option to Prevent tests from starting until the animal has been weighed then ANY-maze will only expect you to weigh the animal before the *first* test (note that the weight will be registered for the *animal* not for the test). In fact, you can weigh the animal at the start of *any* test (not just the first one) by selecting the *Weigh animal* button, but if you do this then the weight measured will replace any weight previously registered for the animal.

Weighing an animal again

You can weigh an animal at any time when the *Weigh animal* button is enabled, just click the button to begin the weighing process. The weight you measure will replace any weight already recorded for the same test or animal.

Sorting out problems when weighing animals

- I see a message saying the scales are not connected
- I see a message saying the scales are not ready
- I weighed the wrong animal
- I need to weigh the animal again
- I forgot to Zero the scales at the start of the session
- I forgot to Tare the scales after I put the basket on
- I started recording the animal weight just once in my experiment, but now I want to record it for every test
- I started recording the animal weight for every test, but now I want to record it just once for the whole experiment
- I can't run a test because ANY-maze wants me to weigh the animal but my scales are unavailable
- I weighed some of my animals manually, and some using the ANY-maze scales, how do I now enter the manually weighed values
- I have another problem not listed here

I see a message saying the scales are not connected

Check that the scales' USB cable is connected to your PC and also to the scales (it's a good idea to unplug it from the scales and plug it in again, so as to be sure it's plugged in correctly).

Also check that the scales are connected to DC power, that their DC power block is plugged into an outlet and that the outlet is switched on. (You can use the scales with batteries - if you are doing this, then there's no need to connect the DC power supply.)

I see a message saying the scales are not ready

There are a number of reasons why the scales may not be ready:

- When the scales first switch on they can take up to 10 seconds to initialise. During this period ANY-maze will report them as being *Not ready*.
- If you remove the animal basket from the scales they'll show a large negative weight and ANY-maze will report them as being *Not ready*. Just place the animal basket back on the scales. Note that if you put the animal in the basket and then put the basket on the scales, they will immediately go from 'Not ready' to weighing the animal.
- The scales may have some sort of error. In this case the I/O icon in the status bar will be shown with a warning symbol over it (see figure 6) and clicking the icon will display information about the error and how to fix it.

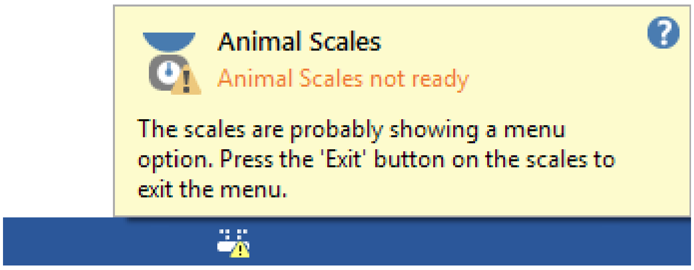


Figure 6. Example of the information shown when the scales have an error. Note the warning symbol on the I/O icon in the status bar.

I weighed the wrong animal

If you have weighed the wrong animal but you haven't yet started its test, then click the  *Weigh animal* button and then weigh the correct animal. The new weight will replace the old one.

If you have realised that you weighed the wrong animal but you have already performed the test, weigh the animal manually and then navigate to the relevant Test details report and overwrite the animal weight shown with the value you just weighed.

I need to weigh the animal again

If you realise that you made some sort of mistake when you weighed the animal (for example, you left something in the animal basket or you weighed the animal without the lid on the basket) then, provided the animal's test hasn't been started, you can weigh it again simply by clicking the *Weigh animal* button.

If the test has been performed, then you should weigh the animal manually and then navigate to the relevant Test details report and overwrite the animal weight shown with the value you just weighed.

I forgot to Zero the scales at the start of the session

This doesn't matter, provided that the scales have been showing 0.0g whenever you are ready to weigh an animal (i.e. when the empty basket is on the scales). However, you can zero the scales at any time by:

- Removing the basket from the scales
- Pressing the Zero key on the scales
- Placing the basket back on the scales
- Pressing the Tare key on the scales

If the scales were not showing 0.0g when the empty basket was on them, then you will need to reweigh any animals you've weighed in this session.

I forgot to Tare the scales after I put the basket on

This doesn't matter, provided that the scales have been showing 0.0g whenever you are ready to weigh an animal (i.e. when the empty basket is on the scales). However, you can tare the scales at any time by:

- Placing the basket on the scales
- Pressing the Tare key on the scales

If you find the scales won't tare then follow the instructions to Zero them instead.

If the scales were not showing 0.0g when the empty basket was on them, then you will need to reweigh any animals you've weighed in this session.

I started recording the animal weight just once in my experiment, but now I want to record it for every test

You should change the option in the Animal weights element of the protocol to Weigh animals using the ANY-maze scales before every test, any weights you recorded already will be retained and ANY-maze will record weights for any new tests you perform. However, you'll need to manually enter the animal weight on the Test details report for any tests which have been performed and for which no weight was recorded (although you may not know these weights, in which case you can just leave them blank).

I started recording the animal weight for every test, but now I want to record it just once for the whole experiment

You should change the option in the Animal weights element of the protocol to Weigh animals using the ANY-maze scales just once - before their first test. Any weights you have recorded for tests other than the first test will be discarded.

I can't run a test because ANY-maze wants me to weigh the animal but my scales are unavailable

If your scales aren't available, but the protocol is set to Prevent tests from starting until the animal has been weighed, then you won't be able to perform any tests. To resolve this just unselect the Prevent tests from starting until the animal has been weighed, option in the protocol.

I weighed some of my animals manually, and some using the ANY-maze scales, how do I now enter the manually weighed values?

You can enter (or edit) an animal's weight at any time. Just navigate to the relevant Test details report and enter the animal's weight into the relevant field.

I have another problem not listed here

If you have any other problem, just contact ANY-maze technical support (techsupport@anymaze.com) and we'll be happy to help.

Skipping a test

Introduction

In some circumstances you may want to skip the next test in the schedule and come back to it later - for example, the animal may not be ready for the test yet.

Details

How to skip a test:

There are two ways you can skip a test:

- Open the Testing status menu on the Test schedule report (by clicking the little down arrow next to the testing status) and then select *Skip test* from the options which appear.
- Alternatively:
  1. Click the "Test number" on the Test schedule report. (In fact this isn't a number but a marker as tests aren't actually numbered until they've been performed).
  2. The Test details report will open. Click the *Skip test* button in the ribbon bar.
  3. Click the *Back* button in the ribbon bar to return to the *Test schedule report.*
  4. The test will be marked as *Skipped,* and ANY-maze will step over it in the schedule.

Unskipping a test

When you want to perform a skipped test, you can either select *Unskip test* from the Testing status menu, or you can return to the Test schedule report and click the *Un-skip test* button in the ribbon bar. In either case the test will be reinserted in to the schedule and will usually be performed immediately.

Skipping all the tests for one animal

Note that skipping a test only affects THAT test, not all the tests for an animal. This means that skipping one trial for an animal won't stop any subsequent trials from being scheduled. If what you want to do is to temporarily remove an ANIMAL from the experiment then it's better to retire the animal from the experiment and then un-retire it when you want to start testing it.

Skipped tests prevent an experiment from progressing from one stage to the next

If an experiment includes multiple stages then ANY-maze won't progress from one stage to the next until all the tests in a stage have been performed. Thus if you reach the end of a stage and you still have some *skipped* tests in the stage then the stage will be 'blocked'. If you try to start the next test for the animal (i.e. a test in the stage after the one containing the skipped tests) then ANY-maze will warn you that you need to unskip the skipped tests and then perform them, or if you don't want to perform the tests then it will offer to delete them for you.

Removing a test altogether

If what you want to do is never perform a particular test then skipping it isn't the correct solution (skipping means 'I don't want to perform this test now'). Instead you should adopt one of the following measures:

1. If you don't want to perform any more tests on a specific animal then you should either Retire the animal or Delete it from the experiment altogether - which of these you choose will depend on whether you want to use the results of any tests the animal's already had.
2. If you're running a multi-stage experiment and you don't want to perform any more tests on this animal *in the current stage* then you should end the stage for the animal - the animal will continue to be tested in subsequent stages.
3. If you're running a multi-stage experiment and you don't want to perform *any* of the tests in a specific stage then you should skip the entire stage.

If the test has already been performed then you should Cancel the test as skipping only applies to tests which haven't been performed yet.

Skipping an entire stage for an animal

Introduction

In some circumstances you might not want an animal to have any trials in a stage at all. This can be done simply by skipping the stage.

Details

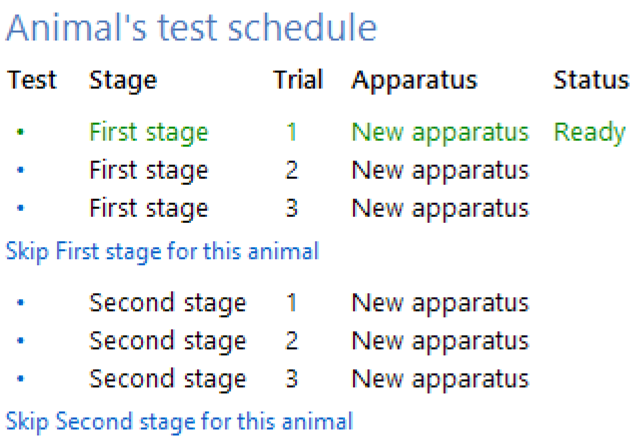
In order to skip a stage, you need to ensure that you've checked the box labelled *Allow the user to specify when an animal should stop having trials in this stage, or whether an animal should skip the stage entirely* which is shown on the Stage end rules settings page in the protocol.

How to skip a stage:

1. Click the *Animal number* on the Test schedule report.
2. The Animal details report will open. Half way down this report is a section which shows the individual animal's *Test schedule.* At the end of the relevant stage you will see a link labelled *Skip the xxx stage for this animal* - see figure 1. Click the link. (NOTE: The link won't appear if the animal has already started the stage - i.e. it has started any trial in the stage.)
3. All the trials for the animal in this stage will be removed and the link will change to read *Unskip the xxx stage for this animal.* Clicking this link will reinstate the stage's trials.

When manually scheduling tests, skipping a stage will cause any tests in the stage to be moved to the next unskipped stage. If there are no further stages then the tests will be deleted.

After a stage subsequent to the skipped is started (i.e. the animal starts a test in the subsequent stage) then the skipped stage can no longer be unskipped.



Animal's test schedule

Test	Stage	Trial	Apparatus	Status
•	First stage	1	New apparatus	Ready
•	First stage	2	New apparatus	
•	First stage	3	New apparatus	

Skip First stage for this animal

Test	Stage	Trial	Apparatus	Status
•	Second stage	1	New apparatus	
•	Second stage	2	New apparatus	
•	Second stage	3	New apparatus	

Skip Second stage for this animal

Figure 1. If specified in the protocol, an animal's test schedule (which is part of the Animal details report) can include a link which will skip the animal's trials in a particular stage.

Adding a test when using a manual schedule

This topic describes how to add tests to a manual schedule.

Introduction

If you are manually scheduling tests in an experiment then you need to add each test to the schedule before it can be performed.

Details

How to add a test:

1. Click the + button shown just above the video picture of the apparatus in which you want to perform the test (note that this button is only shown when manually scheduling tests).
2. The Add test window will open.
3. Enter the *animal number* (or animal ID, if you are using your own IDs) of the animal you intend to test and click OK.
4. The test will appear in the Test schedule report and, assuming no other test is already scheduled to be performed in the apparatus, the apparatus will be ready to perform the test on the animal you specified.

In fact you can enter more than one animal number into the Add test window, allowing you to add a number of tests all at once. Simply separate the numbers with commas.

See also:

- Deleting a test from a manual schedule
- The Add test window

The Add tests window

This topic describes how to add tests to a manual schedule.

Introduction

If an experiment's protocol is set to have test's scheduled manually then you will need to add each test to the experiment before it can be performed. To do this, click the button shown just above the video image of the apparatus in which you want to perform the test, this will cause the *Add Test* window to open. (Note that this button is only shown when manually scheduling tests.)

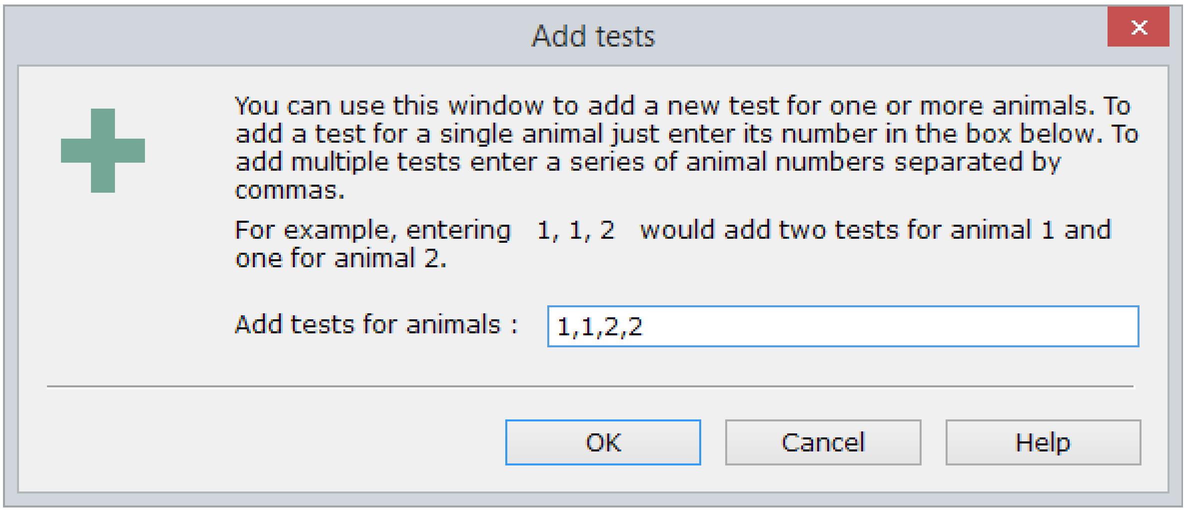


Figure 1. The Add tests window.

Specifying the animals to tests

To add tests you simply have to specify the animal numbers of the animals you'd want to test, *in the order you want to test them*. For example, you could enter 10, 1, 3 in which case ANY-maze will expect you to perform a test on animal 10 then one on animal 1 and then one on animal 3.

If your protocol includes an animal ID field, then you should enter the IDs of the animals you wish to test, rather than their ANY-maze animal numbers.

If you repeat an animal's number, for example 10, 10, 10 then you're simply specifying that you will test the animal repeatedly, in this case in three trials.

As you enter the animals to test, the Test schedule report will update to show the tests you've scheduled, so, for example, entering 2, 2, 4, 4, 1, 1 would create a schedule like this:

Test	Animal	Stage	Trial	Apparatus	Status
•	2	First stage	1	New apparatus	Ready
•	2	First stage	2	New apparatus	
•	4	First stage	1	New apparatus	
•	4	First stage	2	New apparatus	
•	1	First stage	1	New apparatus	
•	1	First stage	2	New apparatus	

Figure 2. The Test schedule report after entering 6 tests.

Of course, if you want to, you can simply add each test individually just before you perform it - this can be a good idea, as it keeps things very simple.

Removing a test you've entered

It's easy to remove a test you've added to the experiment. To do this click the *place-holder* (i.e. the little dot) in the test number column in the Test schedule report (see figure 2) - this will open the Test details report. Then click the *Delete test* button in the ribbon bar. Further details can be found here.

Editing the animal to test

You can edit the animal a test will be performed on (or the animal it was performed on), by changing the test's animal number in the Test details report.

Deleting a test when using a manual schedule

This topic describes how to delete tests from a manual schedule.

Introduction

If you are manually scheduling tests in an experiment then you can delete any unperformed tests you have previously added.

Details

How to delete a test:

1. Click the 'Test number' on the Test schedule report. (In fact this isn't a number but a marker, as tests aren't actually numbered until they've been performed).
2. The Test details report will open.
3. Click the *Delete test* button in the ribbon bar.
4. When you click the button the test will be deleted, which necessarily means that the Test details report you are seeing will no longer be valid (as the test it refers to has just been deleted). Therefore ANY-maze will automatically show the Test Schedule report instead.

See also:

- Adding a test to a manual schedule
- The Test schedule report
- The Test details report

Running a test

Introduction

This section of the help provides guidance on how to actually perform a test, this includes how to start a test, how to end it and how to perform some common tasks while the test is running:

- Starting tests
- Running tests which include an accustomisation period
- Controlling trials which are set to start automatically
- Pausing a running test
- Recording behaviours during a test using keys
- Restarting a test ANY-maze that has ended automatically
- Stopping a test manually

Starting tests

Introduction

How you start a test in ANY-maze depends on whether the experiment's protocol has been set to use ANY-maze's auto start feature.

As the name implies, auto-start causes a test to automatically start without user intervention. This works by tracking *you*, the experimenter, and starting the test as you walk away from the apparatus. This has the advantage that you don't need to rush back to the computer to press a key.

Of course there maybe occasions when auto-start isn't necessary, perhaps you'll be tracking animals for 4 hours and a few seconds lost at the start is really not important.

Details

Preparing for a test

Naturally, you may need to perform various tasks in order to prepare for a test - for example you might need to clean the apparatus, close off a maze arm or perhaps reposition an island. In any event, provided the apparatus's status is *Ready...* ANY-maze will just ignore you while you perform these tasks.

That said, you may wish to refer to the following topics which relate to the types of changes you might want to make in ANY-maze before starting a test:

- Setting the positions of movable zones before a test
- Adjusting the apparatus map so it aligns with the video picture

Once the apparatus is ready for a test then, strictly speaking, you should ensure that there's a period of at least 4 seconds during which no one appears in the camera's view. However, this almost always happens without you having to think about it because you will necessarily have to go and get the animal and this rarely takes less than 4 seconds.

Making sure the user icon isn't shown

This issue is only relevant if you're using the auto-start feature.

When you're ready to start the test the user icon **should not** be shown in the top-left of the video picture. This icon indicates that ANY-maze thinks there's a user in the image and clearly if there isn't then it must be confused. To correct the problem just click the icon and it'll go away - effectively you're telling ANY-maze that there isn't a user in the picture.

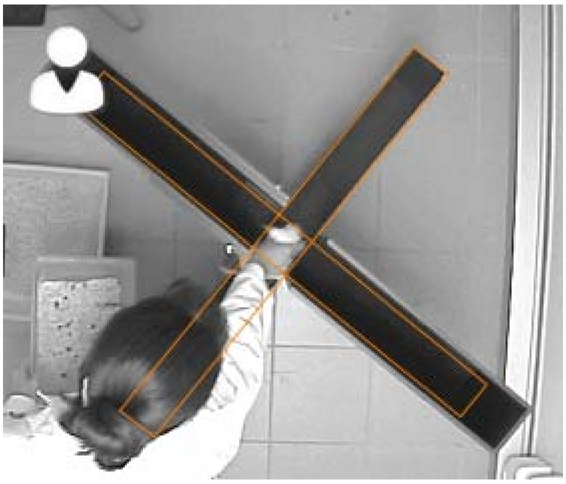


Figure 1. An example of the 'user icon'. ANY-maze uses this icon to indicate when it thinks there someone in the video picture.

If you're wondering how this situation could arise then consider the following: Imagine that yesterday you left a plastic cage on the floor next to your plusmaze. Today, you start ANY-maze and get ready to perform the first test. While preparing the apparatus you remove the cage. Now, ANY-maze will see that the cage is missing and it'll think it must mean something is covering it - something like a person - so it'll think there's someone in the image. What you have to do is tell it that the image it's now seeing is actually 'empty' - which is what clicking the icon does.

Starting a test using auto start

1. Ensure the test status is *Ready...*
2. WITHOUT putting the animal in the apparatus, click the button or press the test control key (or close the test control switch) - the test status will change to *Waiting to start...*
3. Walk up to the apparatus and put the animal in
4. Quickly walk away from the apparatus - as ANY-maze sees you leave the camera's view it will automatically start the test
5. In the unlikely event that the test doesn't start automatically you should click the button or press the test control key (or close the test control switch.)

Once a test has started you can re-enter the camera's view, this will cause the user icon to appear in the top-left corner of the video picture, as ANY-maze 'sees' you but this won't actually have any effect. What you should not do is place any part of yourself between the camera and the apparatus - **this will affect the tracking.**

Starting a test manually

1. Ensure the test status is *Ready...*
2. Walk up to the apparatus and put the animal in
3. Quickly return to the computer and click the button, press the test control key or close the test control switch - the test will start.

The ability to use a test control switch can be very useful when you need to start tests manually, as you could connect a switch to a long cable and actually attach it to your apparatus - this way you wouldn't need to rush back to the computer to start a test.

When using manual start there's no technical reason why you have to leave the camera's view during the test - provided you don't appear between the camera and the apparatus ANY-maze will just ignore you, even if you move about. Nevertheless there are often good scientific reasons why you'd want to be distant from the apparatus during a test.

Starting tests in all the apparatus at the same time

If your protocol defines multiple apparatus, you can start tests in all of them simultaneously by simply click the *Start all* button in the ribbon bar. (Note that this button is not shown unless the protocol include two or more apparatus).

Doing this has the same effect as clicking the individual *Start test* buttons. For example, if you are using auto-start, this will set all the apparatus to *Waiting to start...* status and you'll then work your way through them placing the animals in each one. On the other hand, if you are not using auto start, then clicking the *Start all* button will cause tests to immediately start in all the apparatus, which can be very helpful if you want all the tests to be synchronised.

Running tests which include an accustomisation period

Introduction

In some experiments you might place the animal in the apparatus for a while, before you actually want to start the test (or at least before you want to start tracking). For example, in the water-maze you might place the animal on the island for 20 seconds before its first training trial so it knows that there's actually an island there. In ANY-maze this is called an *accustomisation period*.

You might think that including an accustomisation period before a test would make no difference to ANY-maze as you don't actually want to track the animal; however, going directly from accustomisation to tracking can confuse the tracking system at the start of the test and may cause it to take longer to 'lock-on' to the animal. For this reason you need to tell ANY-maze if your experiment will include an accustomisation period which you do as part of the protocol - see Including an accustomisation period in a stage for details. In fact, there's a side benefit - ANY-maze will time the accustomisation period for you.

Details

Starting an accustomisation period

Starting an accustomisation period is just like starting a test - if you're using auto-start then:

1. Ensure the test status is *Ready...*
2. Click the ▶ button or press the test control key (or close the test control switch) - the test status will change to *Waiting to start...*
3. Place the animal in the apparatus
4. Quickly walk away - the accustomisation period will start automatically
5. In the unlikely event that accustomisation period doesn't start automatically, click the ▶ button or press the test control key (or close the test control switch)

If you're not using auto-start:

1. Ensure the test status is *Ready...*
2. Place the animal in the apparatus
3. Quickly return to the computer and click the ▶ button or press the test control key (or close the test control switch) - the accustomisation period will start

Cancelling an accustomisation period or ending it prematurely

If you want to cancel an accustomisation period (and set the test back to the *Ready...* status) simply click the button.

If you want to end an accustomisation period prematurely (i.e. before the full time specified in the protocol) then press the test control key or close the test control switch - the effect will be as if the accustomisation period had ended.

What happens when an accustomisation period ends

There are two possibilities for what happens when an accustomisation period ends, either the test will start immediately (without any intervention from you), or the test will go to the *Waiting to start...* status and you'll need to explicitly start the test. Which of these occurs is defined in the protocol - see Including an accustomisation period in a stage for details.

Clearly, in the first case (where the test automatically starts) there's nothing for you to do, but in the second case you need to start the test yourself. In fact the most common reason for specifying that you need to start the test is because you want to move the animal in the apparatus before starting to track it. For example, if you've placed the animal on an island in the water-maze during the accustomisation period, then you will, of course, want to move it off the island before you actually start testing it!

Starting a test following an accustomisation period

As you'd probably expect, starting the test following accustomisation is much like starting it in normal circumstances - if you're using auto-start:

1. You DON'T need to press a key because the test will already be *Waiting to start...*
2. If necessary, move the animal in the apparatus
3. Quickly walk away - the accustomisation period will start automatically
4. In the unlikely event that accustomisation period doesn't start automatically, click the button or press the test control key (or close the test control switch)
5. If you don't need to move the animal then just click the button or press the test control key (or close the test control switch)

If you're not using auto-start:

1. If necessary, move the animal in the apparatus
2. Quickly return to the computer and click the button or press the test control key (or close the test control switch) - the accustomisation period will start

Controlling trials which are set to start automatically

Introduction

Using the Automatic repetition of trials options for a stage, you can specify that trials should start either a certain period after the previous trial or at a certain time of day.

- Making trials start at their programmed time
- Preventing trials from starting at their programmed time
- Pausing the automatic start of a trial

Making trials start at their programmed time

To make a trial start automatically at its programmed time you don't have to do anything. ANY-maze will simply start the trial for you.

When a trial is set to start automatically ANY-maze will show the time when the trial will begin, for example "The test will start at 9:00am". When the programmed start time is less than one hour away, the display will change to show a countdown to the test start, for example "The test will start in 23:47".

Note, ANY-maze will never automatically start a test whose programmed start time is in the past, for example, if the test should start 10 minutes after the start of the previous trial, but the previous trial lasts 20 minutes, then the next trial WILL NOT start automatically - you will need to start it manually by clicking the *Start test* button.

Preventing trials from starting at their programmed time

To prevent tests from starting at their programmed time you just need to click the little arrow next to the *Start test* button and un-check the option *Automatically start tests at protocol specified times* in the menu that appears.

To switch automatic starting back on, simply check the box.

Pausing the automatic start of a trial

Imagine a test is about to start at its programmed time but you want to pause the countdown for a while - perhaps because you want to wait for noise from another room to cease.

To achieve this, you could switch off the automatic start (see the previous section) and then switch it back on later, but if the test start time passes while the automatic start is off then the test won't start when you switch it back on (because ANY-maze never automatically starts a test whose programmed start time is in the past). Instead you can simply pause the countdown by clicking the *Pause test* button and then unpause it a while later by clicking the same button again. While paused the countdown stops and when unpaused it *continues from where it left off*, this does mean that the actual start time will change.

Pausing a running test

Introduction

During a test you may encounter circumstances where you would like to pause the test clock, perhaps adjust something in the apparatus, and then restart the test. This is easily achieved using the pause button.

Details

To pause a test simply click the button. The test clock will be paused and ANY-maze will stop recording tracking results - although it will continue to track the animal.

While the test is paused you can enter the image without causing any problems. To restart the test simply click the button again.

A pause in a test necessarily breaks the animal's track because ANY-maze stops recording the track while paused. This is reflected in the track plot report where a symbol is shown at the point where the animal was when the test was paused and another symbol is shown at the point where the animal was when the test was restarted.

Pausing all running tests

If you are running tests in multiple apparatus simultaneously, you can pause all the tests at the same time by clicking the *Pause all* button.

You can also resume all paused tests by clicking the *Resume all* button.

Recording behaviours during a test using keys

Introduction

As you may know, you can define keys in the protocol which you can use to record behaviours which ANY-maze can't score automatically such as *grooming* and *rearing*.

Details

Each behaviour you want to score is assigned a key (as part of the protocol) and while the animal is exhibiting the behaviour you simply need to press the key and hold it down until the behaviour stops. You can hold down more than one key if more than one behaviour is being exhibited at the same time.

If the protocol includes any keys then the toolbar shown above the video picture will include a *Show keys* button. Clicking this button will open a little window listing the defined keys and what behaviours they are being used to score - see figure 1.

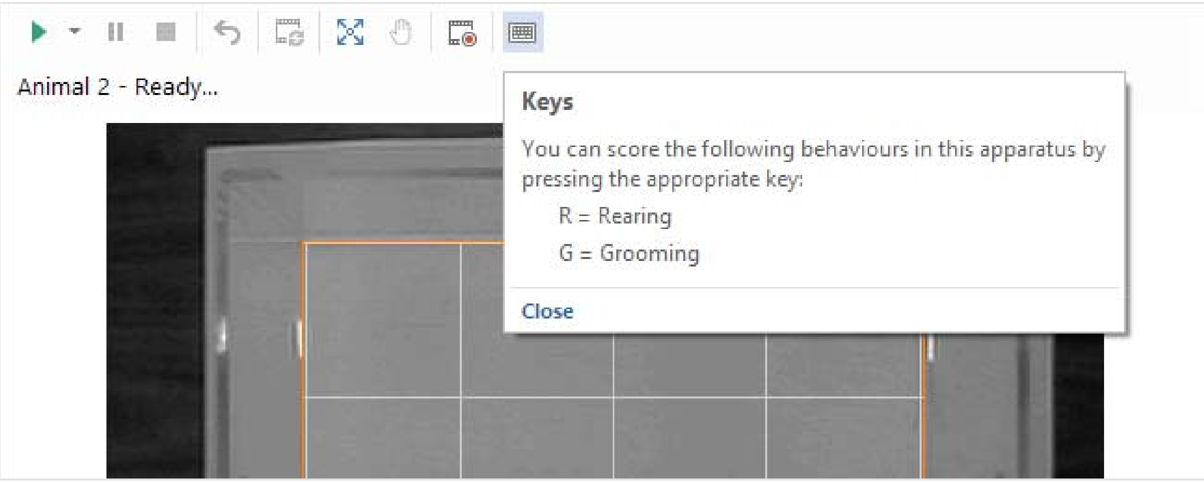


Figure 1. You can use the Show keys *button to find out which keys are being used to score behaviours.*

If you intend to record more than just a few behaviours using keys then you may find that it very hard to do as you'll have to be looking out for all the behaviours simultaneously and press the right key at the right time.

To overcome this, ANY-maze allows you to repeatedly review a video of a test and score just some of the behaviours in each review - thus you can easily score 10 different behaviours by simply scoring 2 at a time and reviewing the video 5 times.

Clearly, for this to be possible you must record a video of the test, but you can set up the protocol so that this is done automatically.

For full details on how to actually add additional scoring to a test see this topic.

Restarting a test ANY-maze that has ended automatically

Introduction

As you may know, you can use procedures to determine when some situation occurs in a test and to take some action because of it; and one of the available actions is to end that test. For example, you might create an procedure that will end a water-maze test if the animal finds the 'Island' zone.

In some situations a test might be ended by a procedure but *you* may want to continue the test. This might be the case, for example, if the animal climbed onto the water-maze island and then immediately jumped off.

Details

When you create a procedure that can end a test, you can actually choose whether the user (i.e. you) will be allow to force the test to continue or not. If the action you include in the procedure allows for test continuation, then, when the action is triggered, ANY-maze won't immediately end the test, instead it will change the test status to *Waiting for test end*. While this is the status tracking doesn't stop and you have 10 seconds during which you can click the button (or press the test control key or close the test control switch) to force the test to continue. If you do this then the fact that the test 'ended' is forgotten and the test continues without any interruption in the tracking data.

On the other hand, if you really do want the test to end then you can either:

- Wait for the 10 second period to expire
- Click the button
- Walk into the camera's view. This is the easiest option as you'll assumedly want to remove the animal from the apparatus.

Whichever method you use, ANY-maze will stop tracking and will only record the data up to the point where the action was *triggered* - i.e. the tracking data recorded since then is simply thrown away.

Stopping a test manually

Introduction

A test will automatically end when the duration specified in the test's stage has elapsed. Alternatively, a test may also be ended by a procedure, as is typically the case in the water-maze, when the animal finds the platform.

However, there may be circumstances when you want to end a test yourself which you can do simply by clicking on the *End test* button.

Details

As well as being able to end a test by clicking the *End test* button, you can also press the press the test control key or close the test control switch (although these options can be disabled in the protocol).

Whatever method you use to tell ANY-maze that you want to end the test, you will usually then see a message asking whether or not you want to save the results recorded so far. If you choose to save the results then the test will end and ANY-maze will move on to the next test in the schedule. If you don't save the results, then the test will simply be reset back to the start again as if you had never begun it. Alternatively, you can cancel the message and the test will simply continue.

In fact, this message can be suppressed by an option in the protocol, which when selected, causes the results to always be saved.

Ending all the running tests

If you are running tests in more than one piece of apparatus simultaneously, you can end all the tests by clicking the *Stop all* button. In this case a message will still be displayed (see previous paragraph) and your response will be applied to all the running tests.

After testing

Introduction

This section of the help provides guidance on some common tasks performed after a test has been completed:

- Undoing a performed test
- Cancelling a performed test
- Reperforming a performed test
- Adding additional scoring to a performed test
- Ending an animal's tests in a stage manually
- Retiring an animal from an experiment
- Deleting an animal from an experiment

Undoing a performed test

Introduction

In some situations you might want to 'undo' a test you've just performed - for example, you might have started the test accidentally.

Details

How to Undo a test

1. To undo the last test you performed in a specific piece of apparatus click the button shown above the apparatus image on the Tests page.
2. A message will be displayed asking you to confirm the action. BE CAREFUL - undoing a test will irrevocably lose any data recorded for the test.
3. Clicking *Yes* will throw away the test's results and set the test back to being 'unperformed'.

Tests are 'undone' in the order they were performed

Undoing a test 'undoes' the last test performed on the relevant apparatus. Using 'Undo' again will undo the test before last, and so on back through the tests you've performed. You can only undo test you've performed since you opened the experiment file. For example, if you performed 3 tests yesterday and 2 today then you can undo the 2 tests from today but not those you performed yesterday.

If you want to 'undo' a test that's not the previous test you performed (or it's a test you performed in a previous session) then you should cancel the test instead.

Don't undo tests you performed on the wrong animal

If you accidentally test the wrong animal, for example ANY-maze is expecting you to test animal 4 and you actually test animal 5, then you **shouldn't** undo the test. Instead you should use the Test details report to change the number of the animal the test was performed on. ANY-maze will then automatically adjust the test schedule and animal 4 (in the example) will be scheduled for the next test.

Sometimes it's better to 'Cancel' a test than 'Undo' it

When you undo a test you lose all reference to the fact that the test was performed at all. If for example, you accidentally ran a test without putting an animal in the apparatus then this would be exactly what you'd want, but if you undo a test which was actually performed on an animal then it might be better to Cancel the test as this will mean that ANY-maze still records the fact that the test was performed on the animal, although it will ignore it in ALL other respects.

When 'undo' is unnecessary

If you do accidentally start a test then you don't necessarily need to 'undo' the test. Instead you can stop the test, by clicking the button and then simply answer *No* when asked if you would like to *save the results recorded so far*. If you do answer *Yes* (perhaps without meaning to), then subsequently 'undoing' the test will have the same effect as answering *No* would have had.

Cancelling a performed test

Introduction

In some situations you may want to cancel a test you've performed. For example, you might have started the test by accident (although then undoing the test may be a better option) or you might have tested the animal too many times.

Details

How to cancel a test:

There are two ways to cancel a performed test:

- Open the Testing status menu on the Test schedule report (by clicking the little down arrow next to the testing status) and then select *Cancel test* from the options which appear.
- Alternatively:
  1. Click the *Test number* on the Test schedule report.
  2. The Test details report will open
  3. Click the *Cancel test* button in the ribbon bar.
  4. Click the *Back* button shown in the ribbon bar to return to the *Test schedule report* .
  5. The test will be marked as *Cancelled.* Note: The test will disappear from the Test schedule report unless you have the checked the option to *Include cancelled tests* shown in the ribbon bar.

Cancelled tests are rescheduled

If you cancel a test, what you're doing is telling ANY-maze to ignore the test and act as if it hadn't been performed. This means that the test scheduler will see that the animal is missing a trial (i.e. the one you just cancelled) and will automatically add a new test into the schedule to compensate. For example, if you're performing a stage in which each animal is to be tested 3 times and you've tested animal 1 in all 3 trials and you then cancel its trial 1, then a new test will be added as it will only have had 2 trials. Note that in this situation what was trial 2 would become trial 1, what was trial 3 would become trial 2 and the new test would be trial 3. This is because trials are numbered in the order they're actually performed - clearly then by telling ANY-maze that trial 1 should be ignored it will act as if the 2nd trial the animal had was it's first. (If this *isn't* what you want, then you may wish to consider reperforming the test instead of cancelling it.)

This rescheduling of tests can seem a little odd but it makes sense if you consider what you're telling the system - in the above example you told it, '*each animal is to receive 3 trials*' if you then remove one of those trials it will put a new one in to satisfy the requirement you gave it.

Of course, you might want to cancel a test because it should never have been performed. For example, imagine that in the above example animals should be tested AT MOST 3 times or until you tell ANY-maze they've had enough trials. So you tested the animal in 3 trials but now you realise that the third trial was unnecessary and you'd like to cancel it but when you do so it gets rescheduled. Well actually that's correct, you're now in the same situation as you would have been in if you'd not performed the test yet (trials 1 and 2 performed and trial 3 ready) - now you simply need to tell ANY-maze to end the stage's trials for the animal.

Cancelled tests are NOT permanently removed

When you mark a test as cancelled it isn't actually thrown away, rather its *marked* as cancelled. ANY-maze ignores all tests which are marked this way so the effect is much the same as removing the test EXCEPT for two things:

1. You can still see that the test was actually performed because it'll still be listed in the Animal details report - this may be important, after all the animal did actually spend some time in the apparatus even if you chose to cancel the test afterwards.
2. You can un-cancel a test by simply clicking the *Un-cancel this test* button shown in the ribbon bar when the Test details report of a cancelled test is displayed.

Cancelling all of an animal's tests won't remove the animal

If you want to remove an animal from an experiment it might seem logical to just cancel all of its tests, but this won't work because ANY-maze will simply reschedule the tests as you cancel them (see above). Instead, to remove an animal you should either retire or delete it.

Reperforming a test

Introduction

In some circumstance you may wish to *reperform* a test. For example, you might realise that adjusting some settings, such as brightness or contrast, will improve the tracking, or you might want the tracking to continue for longer than it did when you first performed the test.

The key point here is that you are still saying that the animal had this test and that the test was still its Xth trial in the relevant stage - just that you want to reperform the tracking of that particular test. Contrast this with cancelling a test, where you're saying that the test should be removed altogether (when you do that the trials 'above' the cancelled test all move down and, if appropriate, a new test is added in *at the end of the stage* to replace it).

Note that when you reperform a test the new results (of the reperformed test) replace those of the original test, if what you want to do is review the test and add *additional* scoring to it, then see this topic.

Details

How to reperform a test:

There are two ways to reperform a test:

- Open the Testing status menu on the Test schedule report (by clicking the little down arrow next to the testing status) and then select *Reperform test* from the options which appear.
- Alternatively:
  1. Click the *Test number* on the Test schedule report.
  2. The Test details report will open.
  3. Click the *Mark test to be reperformed* button in the ribbon bar.
  4. Click the *Back* button shown in the report's title bar to return to the *Test schedule report.*
  5. The test will be scheduled as the next test to perform (unless there are other tests to be reperformed before it).

Changing your mind

Having specified that you want to reperform a test you can still change your mind. Either select *Un-mark test to be reperformed* from the Testing status menu, or return to the *Test details report* and select the *Un-mark test to be reperformed* button in the ribbon bar.

Note that once you actually *have* reperformed the test then you can't do this anymore.

The results of a reperformed test replace the original results

When you reperform a test the results of the *reperformed test* entirely replace the results of the original test - there is then no way to recover the original results should you change your mind. Note that this is only the case *after* the test has actually been reperformed - until then you can simply cancel the request to reperform the test, as described in the section above.

Until reperformed a test continues to have its original results

When you specify that a test is to be reperformed (by clicking the *Reperform this test* link - see the first section above), ANY-maze simply notes your request and schedules the reperformance of the test as the next thing to do. However, until the test is actually reperformed the results of the original test are still valid and will be shown in all relevant reports.

Adding additional scoring to a performed test

Introduction

Although ANY-maze allows you to set up as many as 46 keys to score additional behaviours that the system can't assess automatically, this is of little use if you are going to try to measure the behaviours during the test as you're unlikely to be able to detect and score more than 3 or 4 behaviours simultaneously.

To address this (and let you score as many behaviours as you want), ANY-maze lets you review a video of a test and add additional scoring to it (using keys) as if that scoring had been performed while the test was being run. What's more, you can do this any number of times, so if you wanted to, you could review a test 10 times and score two behaviours on each pass.

This ability to add additional scoring to a test has no side-effects on the test results, where it will appear that all the scoring was performed at the same time.

Prerequisites for adding additional scoring to a test

In order to add additional scoring to a test, the test must have either been tracked from a video or a video must have been recorded of the test. Clearly without a video, the system can't replay the test for you to review it.

The easiest way to ensure that you have a video of test, is to simply specify that ANY-maze should automatically record a video of all the tests in your protocol.

How to add additional scoring to a test

To add additional scoring to a test you should follow these steps. First either:

- Open the Testing status menu on the Test schedule report (by clicking the little down arrow next to the testing status) and then select *Mark this test for additional scoring* from the options which appear.
- Alternatively:
    1. In the Test schedule report click the test number of the test you want to add additional scoring to. This will cause the Test details report to open.
    2. Click the *Mark test for additional scoring* button in the ribbon bar.
    3. ANY-maze will check that it has a video of the test and, assuming it has, it will mark the test as *Waiting to have additional scoring added to it* - you'll see this at the top of the report.

In either case, the test will immediately be scheduled as the next test to run in the apparatus the original test was performed in. Of course, you won't actually be rerunning the test in your physical apparatus, but ANY-maze will need to use the apparatus's *video window* (on the Tests page) to play the video of the test. You will see that the Test schedule report shows the test as the next thing to run and the apparatus will have a status of *Ready to add additional scoring...*

If at this point, if you change your mind, you can either select *Un-mark this test for additional scoring* from the Testing status menu, or you can go back to the Test details report and click the *Un-mark test for additional scoring* button in the ribbon bar.

Assuming you wish to proceed, then you just need to click the *Start test* button for the apparatus:

- ANY-maze will automatically rewind the video to just before the test start and will then start the test at exactly the same moment as it started when it was originally performed.
- You can then use the keys on the keyboard to score any of the behaviours you defined in the keys element of the protocol.
- ANY-maze will automatically end the test at the same moment that it ended when it was originally performed, but you can end it beforehand if you wish, by clicking the *Stop test* button. If you do this a message will be displayed asking whether you want to save the results for the scoring you have just performed or not. If you answer *No,* only the additional scoring for the review you're ending will be lost, all the previous results for the test will be retained.

There is no limit to how many times you can add additional scoring to a test, you just need to go back to step 2, above and click the *Mark test for additional scoring* button again.

Choosing when to start a test

Ordinarily, when adding additional scoring to a test, ANY-maze will start the test automatically at the same time as it started when originally performed. If you wish you can override this default behaviour and start the test yourself. To do this, when you reach step 6, in the above instructions, you should click the little down arrow next to the *Start test* button, and un-check the option to *Automatically start tests when adding additional scoring* in the menu that appears.

You will then be able to start the test at any point in the video by clicking the start test button.

When using this option it is important to understand that ANY-maze times all events in a test in relation to the test start and that by using this option you can alter the test start in relation to the video, thus altering the times at which scoring appears to occur. This doesn't affect existing results, but will affect the results scored in the manually started test.

What happens if key strokes overlap?

A possible problem that can occur when adding additional scoring to a test, is that two keystrokes may *overlap*. For example, imagine you pressed key A at time 10s and released it at time 15s. Then you scored the test again and pressed key A at time 12s and released it at time 17s - what should ANY-maze score, one key press or two, and how long should it say the key was pressed for?

This issue is addressed by the options available in the Additional scoring - overlapping keystrokes sub-element in the Analysis element of the protocol, and you'll find a full description of the options here.

Ending an animal's trials in a stage manually

Introduction

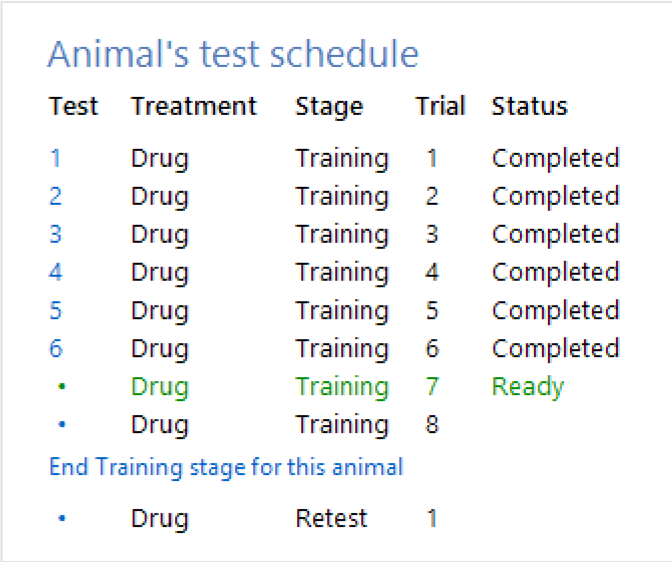
In a stage which includes multiple trials you may want to stop testing a specific animal before it has had the maximum number of trials. For example, in a training stage, you may wish to stop training the animals *either* after a maximum of 10 trials or when they achieve some type of training goal.

Details

In order to end the trials for an animal yourself, you need to ensure that you've checked the box labelled *Allow the user to specify when an animal should stop having trials in this stage, or whether an animal should skip the stage entirely* which is shown on the Stage end rules settings page in the protocol.

How to end the trials in a stage:

1. Click the *Animal number* on the Test schedule report.
2. The Animal details report will open. Half way down this report is a section which shows the individual animal's *Test schedule*. At the end of the relevant stage you will see a link labelled *End the xxx stage for this animal* - see figure 1. Click the link. (NOTE: The link won't appear if the animal has already ended the stage - perhaps because it's had the maximum number of trials.)
3. All future trials for the animal will be removed from the schedule and the report will show the text *Xxx stage ended for the following reason: User ended the stage.*



Animal's test schedule

Test	Treatment	Stage	Trial	Status
1	Drug	Training	1	Completed
2	Drug	Training	2	Completed
3	Drug	Training	3	Completed
4	Drug	Training	4	Completed
5	Drug	Training	5	Completed
6	Drug	Training	6	Completed
•	Drug	Training	7	Ready
•	Drug	Training	8	

End Training stage for this animal

•	Drug	Retest	1	

Figure 1. If specified in the protocol, an animal's test schedule (which is part of the Animal details report) can include a link which will end the animal's trials in a particular stage.

Restarting a stage you ended manually

To restart a stage for an animal is almost identical to ending it (see above), only you'll click a link labelled *Restart this stage for this animal.* ANY-maze will automatically reschedule all outstanding trials for the animal.

Ending stages automatically

Although you can always end stages manually (provided you specified this option in the protocol - see above), there may be situations when ANY-maze could automatically detect your stage end criteria for you.

For example, in the water-maze you might want to train animals to find the island with 30 seconds in two consecutive trials. As ANY-maze obviously knows when an animal finds the island it could determine when an animal achieves this goal - this would save you from having to check for this following each trial of each animal.

To do this you can use Stage end rules, which are part of the stages element of the protocol. Of course, even if you're using stage end rules you can still end a stage manually if you need to.

Retiring an animal from an experiment

Introduction

When running an experiment which includes multiple trials (and/or stages) you may find yourself in a situation where you've performed some tests on an animal but you don't want to perform any more. This might be the case if the animal has become ill.

Of course, you could choose to delete the animal from the experiment but this would cause the results for the performed tests to be excluded from the analysis and you might not want this to occur (particularly if you only have a few animals, or the experiment is being run over a long period of time).

The solution in these cases is to *retire* the animal. As the name implies, the animal takes no further part in the experiment but results of tests it has already had are retained and are included in the experiment's results (although you can exclude them if you want to).

You should note that retiring an animal removes it from ALL future trials and stages of the experiment - if what you want to do is to stop testing an animal in a particular stage but continue testing it in subsequent stages then you should end the stage for the animal, rather than retiring it.

Details

How to retire an animal

1. Click the *Animal number* on the Test schedule report.
2. The Animal details report will open. Towards the top of this report is a drop list titled *Status*.
3. Open the drop list and change the animal's status to *Retired*. It's also a good idea to record why you retired the animal in the notes area.
4. The animal will be retired from the experiment and won't take part in any future tests.

How to un-retire an animal

If you want to un-retire an animal (i.e. have it participate in tests again) then you should simply follow the steps above and change the animal's status to *Normal*.

Note that when an animal 'comes out of retirement' it's possible that it will be behind the other animals in the experiment, for example, they may be in a future stage while the newly un-retired animal still has trials in an earlier stage to complete. The effect of this will be to 'block' the tests of the other animals until the un-retired animal has caught up.

A retired animal can disappear completely from the Test schedule report

If an animal has had at least one test and you then retire it will still have its performed tests shown on the Test schedule report (at least it will have, if the *Include completed tests* box is checked), however, if the animal has never had any tests performed and you retire it then it'll disappear from the Test schedule report completely.

This may not seem too important, but it will cause a problem if you want to *un-retire* the animal because to do this you need to access the Animal's details report which you'd normally do by clicking the animal number on the Test schedule report. The solution to this problem is to switch to the Experiment page and there select the *View animals* button in the ribbon bar. This will open the Animals spreadsheet where you can edit the status of any animal in the experiment.

Deleting an animal from an experiment

Introduction

If you want to remove an animal from an experiment then the easiest thing to do is to *delete* it. In fact, deleting an animal is a bit of a misnomer as animals aren't actually deleted; they're *marked* as deleted - this has the advantage that you can undelete them later if you want to.

To be strictly accurate, you *can* completely remove an animal from an experiment but only if it's had no tests performed on it. In this case you can use the Experiment page to delete animals (see Deleting animals from an expeirment), or reduce the N of the animal's treatment group - this is described fully in the Reducing the number of animals in a treatment group topic.

Details

How to delete an animal

1. Click the *Animal number* on the Test schedule report.
2. The Animal details report will open. Towards the top of this report is a drop list titled *Status*.
3. Open the drop list and change the animal's status to *Deleted*. It's also a good idea to record why you deleted the animal in the notes area.
4. The animal will be deleted from the experiment - it won't take part in any future tests and those tests it's already had will be excluded from the experiment's results.

Note that the animal still actually exists - this has the advantage that you can still access the Animal's details report where you can see what tests were performed on the animal and see the notes explaining why it was deleted. Of course it also means you can un-delete the animal.

How to un-delete an animal

If you want to un-delete an animal (i.e. include it's performed tests in the experiment and have it participate in tests again) then you should simply follow the above steps and set the animal's status back to *Normal*.

Note that when an animal is un-deleted it's possible that it will be behind the other animals in the experiment, for example, they may be in a future stage while the newly un-deleted animal still has trials in an earlier stage to complete. The effect of this will be to 'block' the tests of the other animals until the un-deleted animal has caught up.

A deleted animal can disappear completely from the Test schedule report

If an animal has had at least one test and you then delete it will still have it's one performed test shown on the Test schedule report (at least it will have, if the *Include cancelled tests* box is checked), however, if the animal has never had any tests performed and you delete it then it'll disappear from the Test schedule report completely.

This may not seem too important, but it will cause a problem if you want to *un-delete* the animal because to do this you need to access the Animal's details report which you'd normally do by clicking the animal number on the Test schedule report. The solution to this problem is to switch to the Experiment page and there select the *View animals* button in the ribbon bar. This will open the Animals spreadsheet where you can edit the status of any animal in the experiment.

ANY-maze help topic H0740

Adjusting the apparatus layout

Introduction

The right-hand side of the Tests page displays the video pictures of the apparatus that tests are being performed in. If you just have a single piece of apparatus then the video picture will usually fit comfortably into this area, but if you have a couple of apparatus, they will probably be a bit squashed; and if you have more than that, the video pictures probably won't be large enough to show the entire apparatus.

In fact, the tracking is unaffected by the fact that *you* can't see the entire apparatus (and ANY-maze will also automatically pan the video picture during a test so the animal is always shown), but nevertheless, not being able to see the entire apparatus can be awkward and can make it harder to score behaviours using keys.

There are, however, various ways you can address these issues: you can scale the video pictures, so they'll always fit inside the space available; you can reorganise the way the video pictures are arranged, for example, putting them all into a single row, and then simply scroll the row side to side to see the different apparatus; or you can show little thumbnails of all the apparatus and then choose those you want to view by selecting the thumbnails. And you can combine these features in any way that suits you.

Important: None of the options described here affects the tracking, they just alter how the apparatus video pictures are *displayed* and not how the animals are actually tracked.

- Scaling the apparatus images so they fit the space available
- Arranging the apparatus into rows or columns
- Using thumbnails to select apparatus to view

Scaling the apparatus images so they fit the space available

Usually, when the video picture of some apparatus won't fit within the space available on the tests page, ANY-maze simply crops the picture - as in figure 1, below.

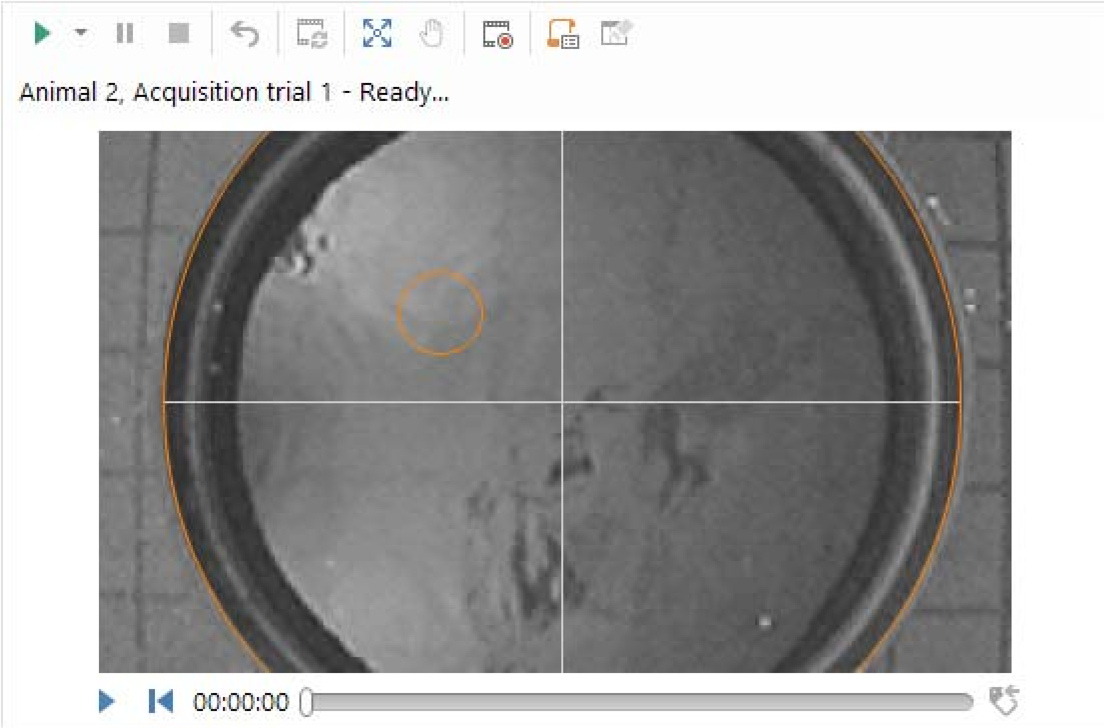


Figure 1. The video picture of the apparatus does not fit in the space available, so ANY-maze has simply cropped it.

Although it looks as if the animal will be invisible it if swims to the top or bottom of the maze, what actually happens is that ANY-maze will pan the image so that the animal is always visible. Nevertheless, it would still better if you could see the entire apparatus and this is what the *Scale to fit* option does - see figure 2.

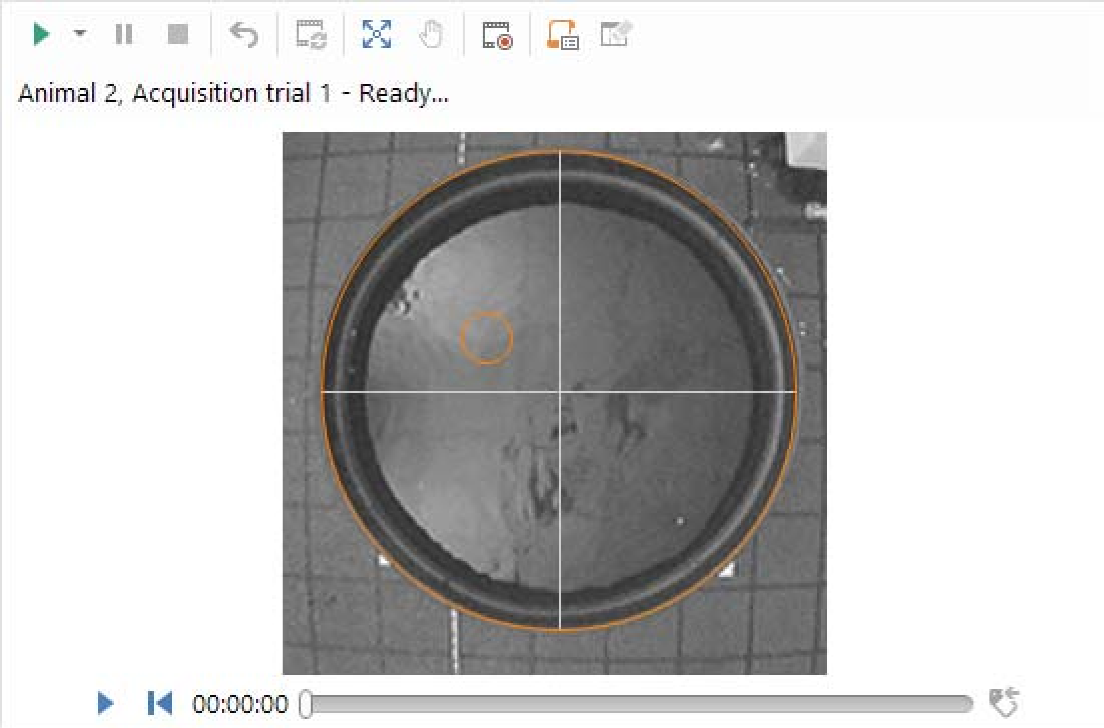


Figure 2. Here the video picture has been resized to exactly fit the space available - what ANY-maze calls 'Scale to fit'.

To scale the view in this way you just need to select the *Scale to fit* button in the ribbon bar. This will cause *all* the apparatus video pictures on the Tests page to be scaled.

Arranging the apparatus into rows or columns

When your protocol includes multiple apparatus, ANY-maze will, by default, *Tile* the video pictures within the overall space available - see figure 3.

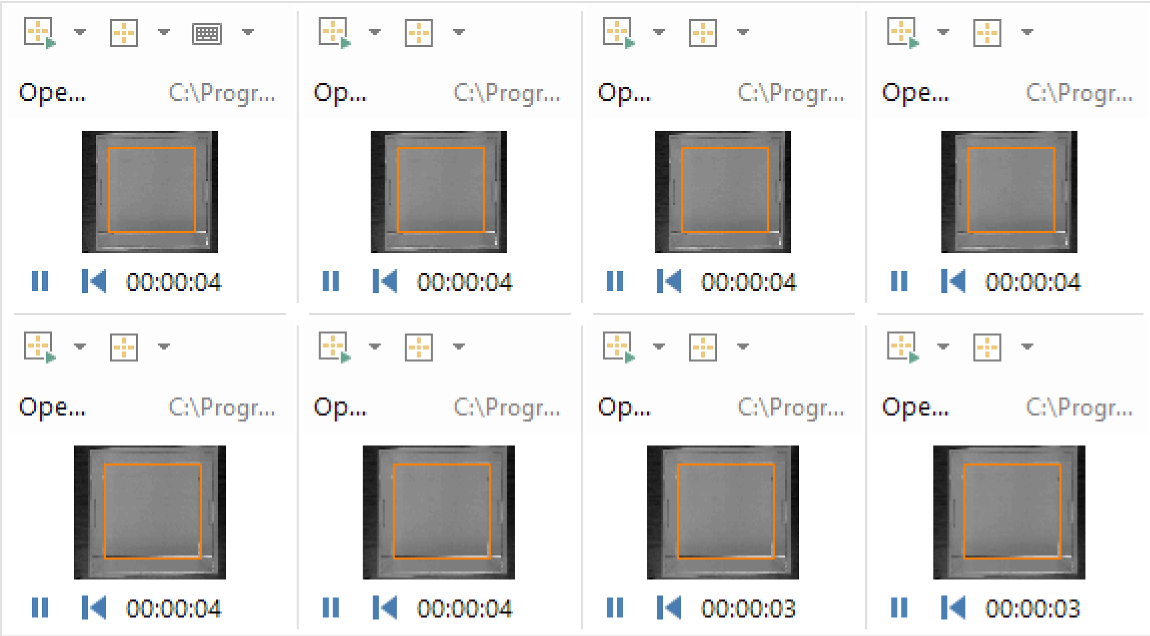


Figure 3. Eight open fields laid out using the default 'tiled' option. Here, the pictures have been scaled to fit, but they're almost too small to be usable.

When you just have a two or three apparatus, tiling in this way is usually fine, especially if combined with the Scale to fit option described above. However, with larger numbers of apparatus, and/or with a small screen, this can become unsatisfactory, ss is the case in figure 3.

To address this you can choose to reorganise way the video pictures are laid out, by placing them into a fixed number of rows or columns. In this case the video pictures will be as tall as the rows (or as wide as the columns) and you will probably need to scroll the rows side to side (or up and down, for columns) in order to see all the apparatus - see figure 4 and 5.

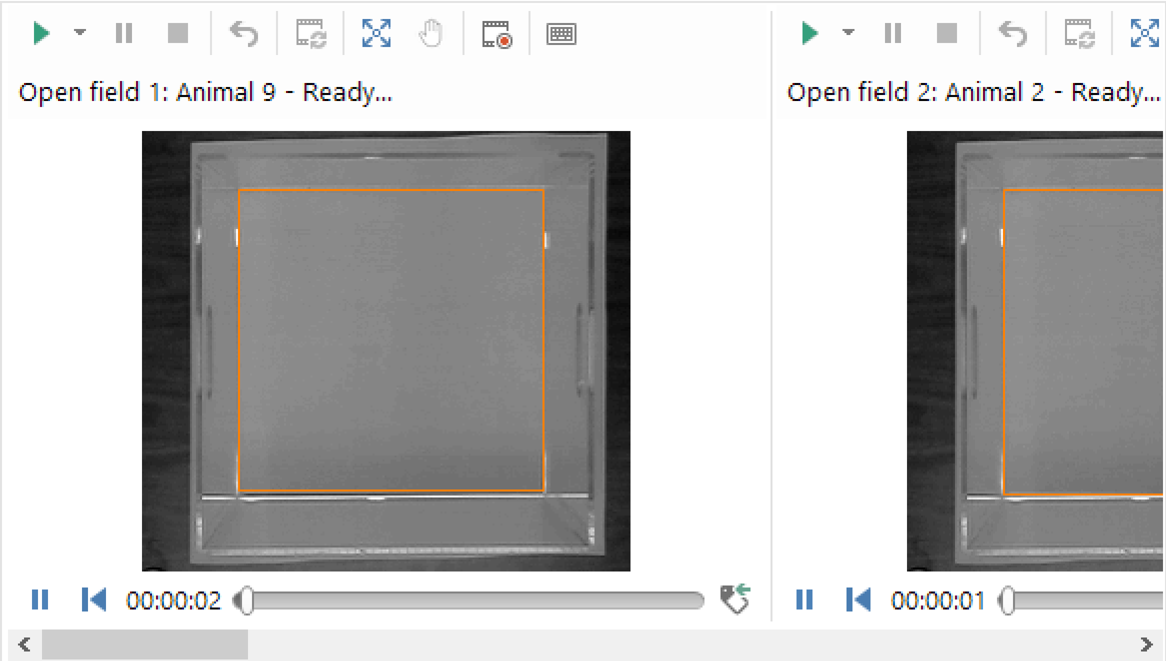


Figure 4. Here the same eight open fields shown in figure 3, have been laid out in a single row. Note the scroll bar at the bottom - to see the other apparatus you would need to scroll side-to-side.

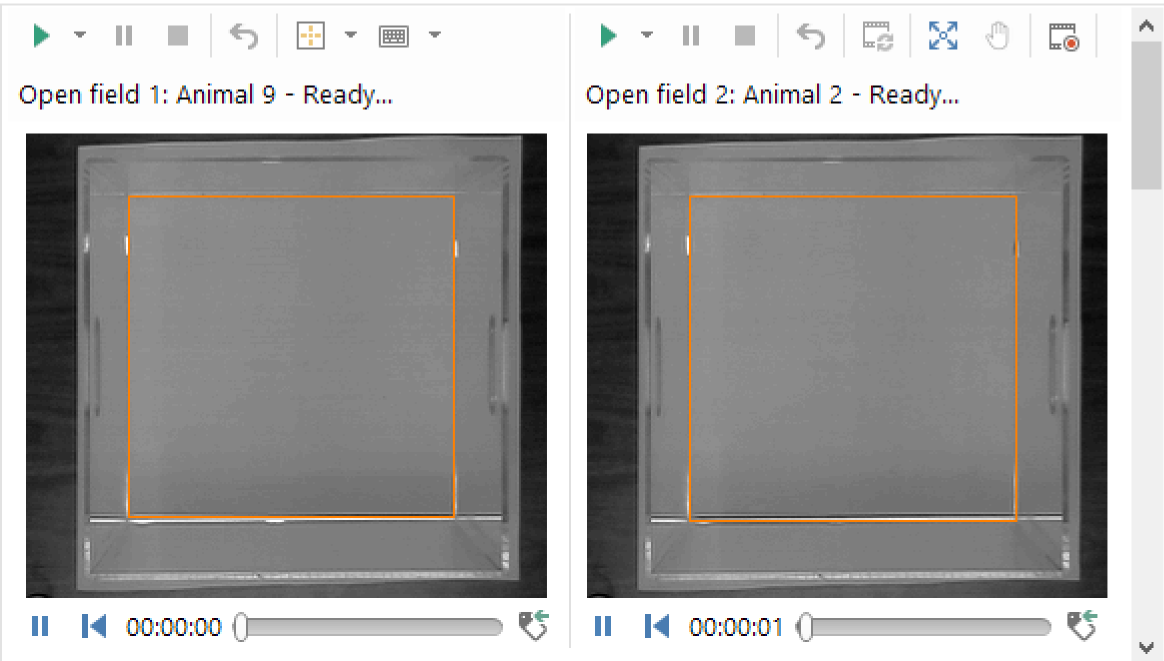


Figure 5. Here the same apparatus have been laid out in two columns. Note that

this time there is a scroll bar at the side, so to see the other apparatus you would need to scroll up-and-down.

Clearly the advantage here is that the apparatus video picture is a good size, but the disadvantage is that you can't see them all, so you have to scroll. By the way, whether or not *you* can see the apparatus, ANY-maze will still track in it.

To arrange the apparatus in rows or columns you simply need to select *Rows* or *Columns* button in the ribbon bar and then choose the appropriate number of rows or columns from the menu which appears. To return to the default *Tiled* view, you should select the *Tile* button. Remember you can combine rows and columns with the scale to fit option, if you want to.

Using thumbnails to select apparatus to view

When working with large numbers of apparatus you may find it useful to include *thumbnails* on the Tests page - see figure 6.

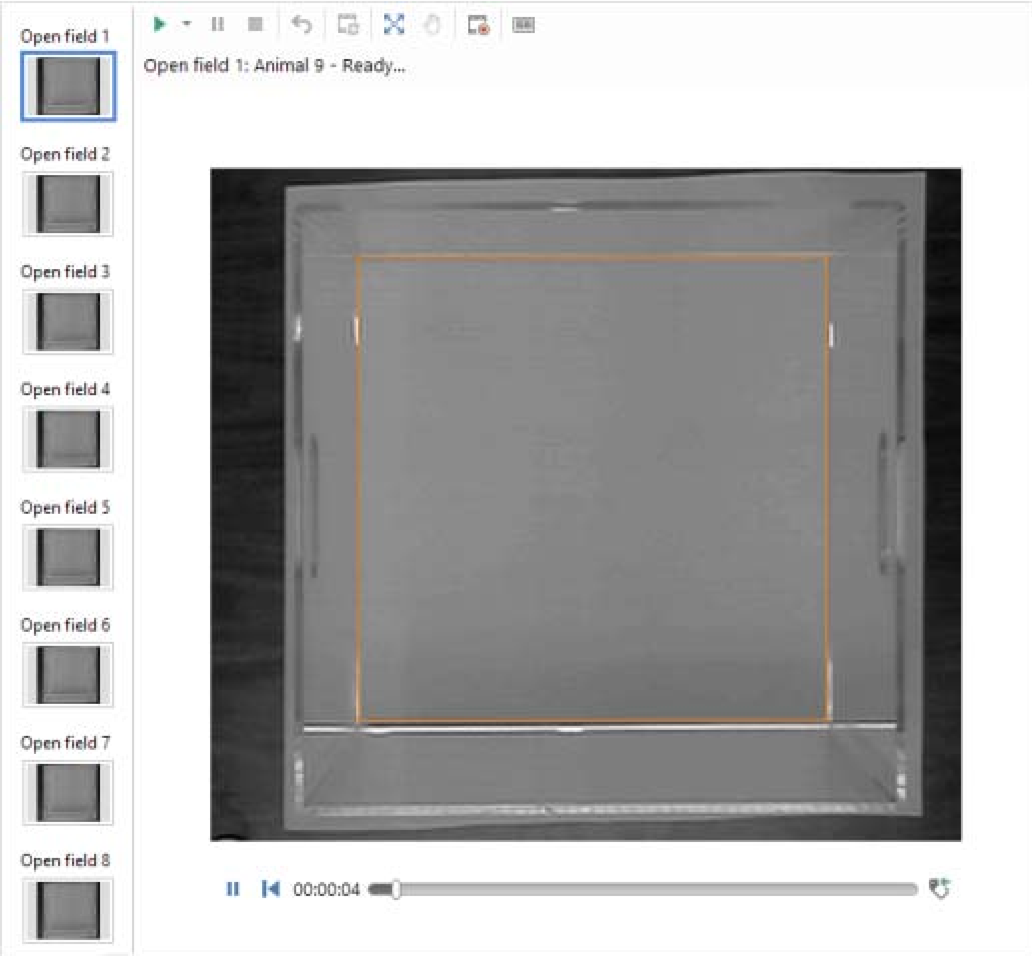


Figure 6. Here the same eight apparatus as in figures 3, 4 and 5 are shown as thumbnails. In this example just the apparatus 'Open field 1' has been selected to be displayed.

There are two options for thumbnails: single-select and multi-select. Single-select always shows just one piece of apparatus, and you select which one it is by clicking the relevant thumbnail image - this is the option being used in figure 6. Multi-select allows you to show more than one apparatus at the same time, with the apparatus being laid out using tiles, rows or columns, as described above. In either case you can use the scale to fit option too.

To show thumbnails, you should click the *Thumbnails* button in the ribbon bar and then choose either *Single-select* or *Multi-select* from the menu which appears. To remove the thumbnails select the *No thumbnails* option from the same menu.

Remember, whether or not a piece of apparatus is currently displayed, makes no difference to the tracking - so in figure 6, all eight apparatus can be tracking even though we're currently only looking at one of them.

ANY-maze help topic H0741

Recording and tracking with videos

The videos described here are not suitable for playback in standard media players or inclusion in PowerPoint or other presentation software. If you would like to record such a video (which in ANY-maze is called a 'movie') you should refer to the Recording movies of tests for inclusion in a presentation topic.

Introduction

ANY-maze includes a built-in video recorder which can record videos of tests. Not only can you then review the video, perhaps to check on specific behaviours, but you can also track in a video in just the same way as you can in the live picture from a camera.

Using unlicensed copies of ANY-maze to record tests

Tracking in videos offers a way to extend your ANY-maze system without having to purchase multiple licences. This is possible because you can record videos using unlicensed copies of the program.

For example, if you have a water-maze, a plusmaze and an open-field you could set-up three ANY-maze systems, one for each piece of apparatus. You could then install unlicensed copies of ANY-maze on two of these systems and a licensed copy on the other one (it's OK this is quite legal, we even encourage you to do it!). Now you could run experiments simultaneously in all your apparatus. However, in the unlicensed copies of ANY-maze, you would just video the tests, so you could track in them later using the licensed copy.

Recording videos using other software

ANY-maze can track in videos recorded in AVI or Windows Media Video file formats. Therefore you can record videos of tests using any software that can record in one of these formats and then track in the video using ANY-maze.

Clearly, however, the details of how to record videos using such software is outside the scope of this help.

- Recording videos of tests in ANY-maze
- Tracking in a video of a test

See also:

- Recording movies of tests for inclusion in a presentation

Recording videos of tests

Introduction

There are two distinct ways of recording videos of tests - you can record a video manually or you can set up the protocol so that ANY-maze will automatically record a video of all the tests in an experiment.

Automatically recording tests

If you know in advance that you will want to record videos of the tests in an experiment then you should set up the protocol so that the videos are recorded automatically - see What to record while testing. Automatically recording tests in this way has a number of advantages:

- You don't need to remember to start recording - it occurs automatically when the test starts.
- You can use procedures to actually control the video recorder during the test - for example you could pause the recorder while the animal's asleep.
- ANY-maze will include a link to the test's video in the list of *Related reports* for the test - thus when you are viewing any test specific report, the video will just be a click away.
- ANY-maze will file the videos automatically for you, so you won't have to give them file names and choose where to save them.

You'll find more information about automatic recording of tests in the What to record while testing topic.

Manually recording tests

In some cases you may wish to record a video manually - this is most useful when you don't actually want to *track* the test at the time, instead you want to record a video and then perform the tracking later.

There are two places in ANY-maze where you can record a video, on the Tests page and on the Video page. Neither of these is any better than the other, the only difference is that you can record on the Video page without having to create an experiment first - so if you *just* want to record a video this the easiest place to do it. On the other hand, if you're already running an experiment and you decide that you'd like to record a video then you'll already be on the Tests page and recording it there will be more logical.

Whether you're recording a video on the Tests or the Video page the method used is the same - you use the VCR buttons shown at the top of the page:

Record	Opens a window where you can specify the file to record to (see below) and then starts recording.
Pause	Pauses a recording.
Stop	Ends a recording.
Cancel	Cancels a recording - any video that's been recorded will be thrown away.
Add label	Opens the Add label window where you can enter some text which will be used to label the video at the moment you clicked the button. During playback you will be able to select the label and jump directly to that point in the video.

If you are recording on the Video page and you then switch to another page (such as the Protocol or Tests) the ***video recording will stop****. This doesn't happen if you record a video on the Tests page. In general this is less of a problem than it sounds because when recording on the Video page you usually won't have an experiment open - in which case all the other pages will be disabled, so you won't actually be able to switch to them.*

Specifying the file to record to

When you manually record a video you will need to specify the name of the file you want the video to be recorded to. You do this using the *Video recorder window*

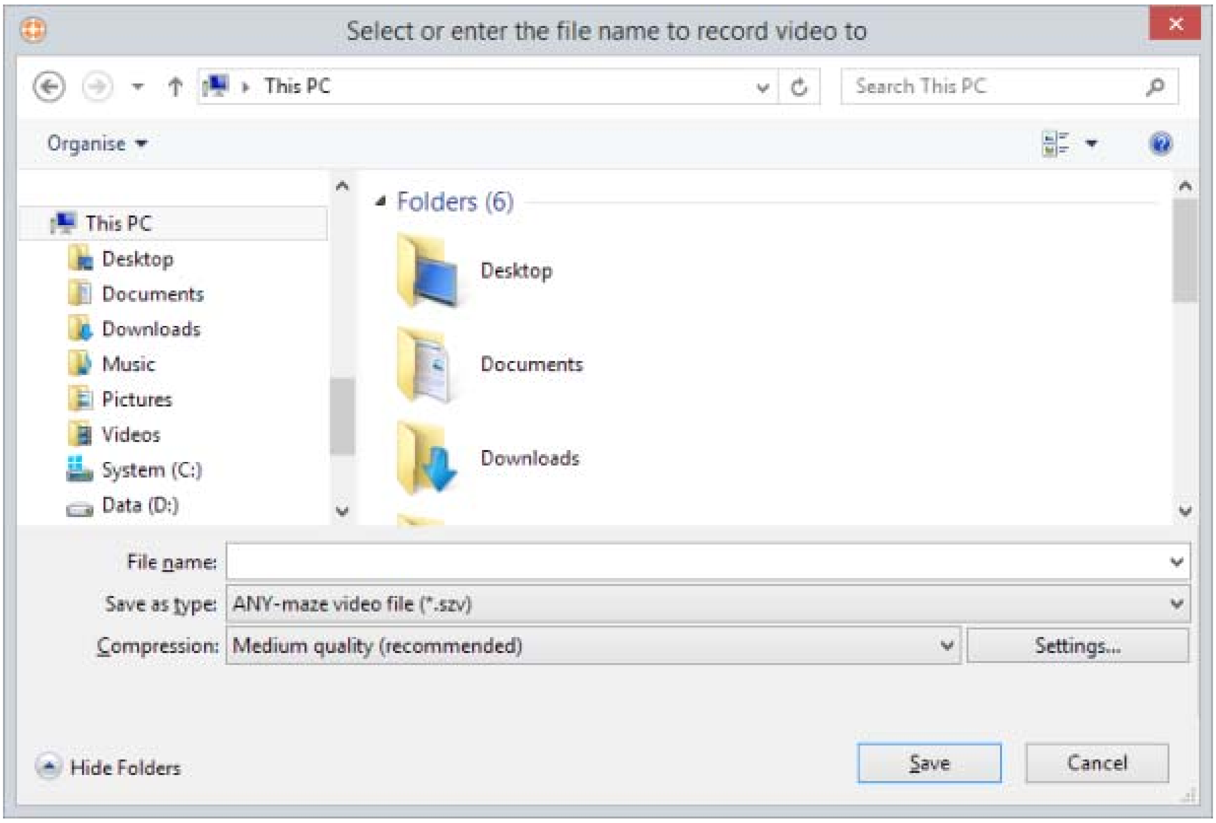


Figure 1. The Video recorder window.

Files list

The files list shows the folders and files of the selected format (see *Type of file* below) stored in the current location. You can double click a folder to open it and you can click a file to transfer its name to the File name field.

File name

Use this field to enter a name for the video you're going to record. You can enter up 255 characters but your entry mustn't contain any of the following characters: \ / : * ? " < > |.

Type of file

There's only one choice here *ANY-maze video file* so you can ignore this list.

Compression

You can choose one of three standard compression schemes for the video: *Best quality, Medium quality* or *Low quality* or a scheme you've saved previously using the options on the ANY-maze video recorder settings window.

If you're not sure which scheme to use I suggest you select *Medium quality* which is fine for most situations.

Settings

Clicking this button will open the *ANY-maze video recorder settings* window where you can fine tune the video compression settings ANY-maze will use to record the video. See the ANY-maze video recorder settings window topic for more details.

Appending recordings to existing video files

If you choose to record to a file that already exists, the system will ask you whether you want to append the new recording to the end of the existing file, or replace the file with the new video. By appending recordings you can easily build up a single file which contains a series of different tests. By the way, the system will automatically include a label in the video to mark each point where you appended a new recording.

The ANY-maze video recorder settings window

Overview

A typical video camera supplies around 30 images per second, therefore to create a video these images simply need to be recorded to a file and then later played back, again at a speed of 30 per second.

The problem with this is size. A typical image might measure say 400 x 400 pixels and if each pixel requires 1 byte of memory a 5 minute video will require: 400 x 400 x 30 x 60 x 5 = 1,440,000,000 bytes or a about 1.3GB for just 5 minutes of video! Obviously, the solution to this is to compress the video pictures.

In fact in ANY-maze two techniques are employed, firstly the number of frames per second can be reduced, for example a video playing at 15 frames per second still appears fine, and secondly the images themselves are compressed.

There are many video compression techniques employed nowadays but, from ANY-maze's point of view, they suffer from one of two problems, either the compression requires too much computation (which might make it impossible to track and record a video at the same time in multiple apparatus) and/or the compression loses too much information from the picture thereby reducing the quality and thus making it impossible to track using the resulting video file. For these reasons ANY-maze uses its own video compression technique which can be controlled using the options on the *Video recorder settings* window.

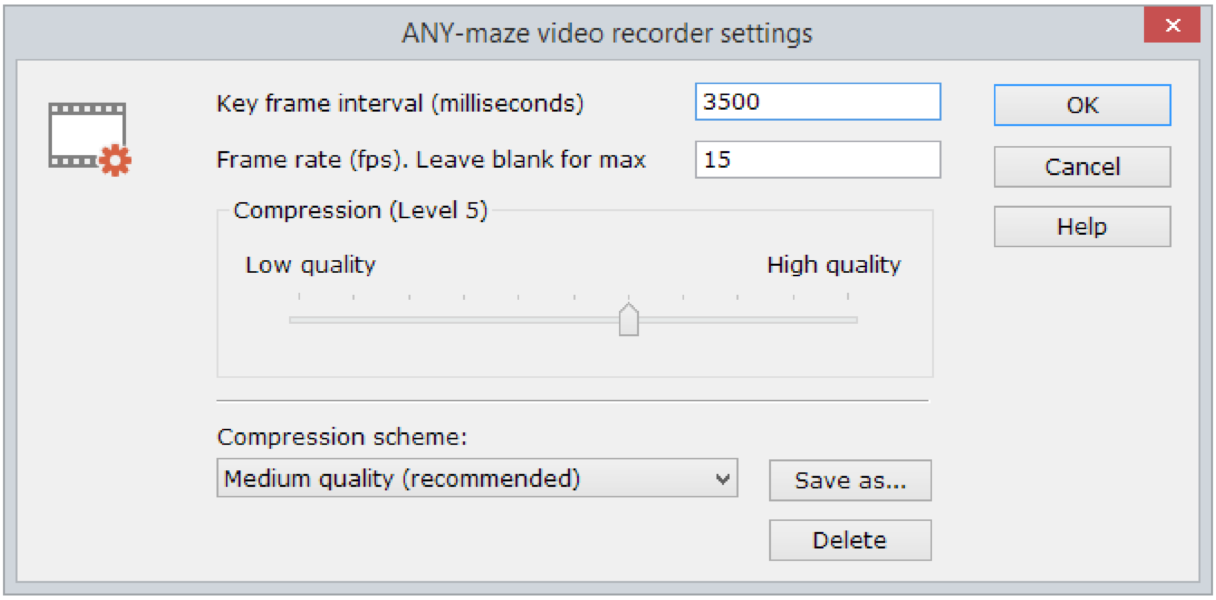


Figure 1. The ANY-maze video recorder settings window.

Details

Key frame interval	Allows you to specify how often a *key frame* should be saved. To understand this setting you need to know what a key frame actually is. One way in which ANY-maze video compression works is to compare an image to the previous image and save just the differences. However, this has a disadvantage in that each image depends on its predecessor so if, for example, you wanted to jump to the image at time 60 seconds in a video the system would have to get and process all the images up to this time. This would clearly be absurd if you wanted to jump to time 10 hours! To overcome this the system saves occasional *key frames* which don't rely on any previous image, thus to jump to an image the system finds the immediately previous key frame and then decodes just the images from there on. As you might expect a key frame is larger than a normal frame and so it's not a good idea to save them too frequently, on the other hand if you don't save them often enough, it can make jumping to different parts of a video slower. So, the key frame interval, sets how often key frames are stored - settings of around 4000 milliseconds are usually fine but you could use larger values if you want to improve compression.
Frame rate	Specifies how many frames should be recorded per second. Smaller values create smaller files but values below 10 tend to create video that looks jerky during playback. If you want to record every frame from the camera (whatever its frame rate is) then simply leave this field blank.
Compression quality	One of the compression techniques used in ANY-maze involves throwing away some of the information in the video pictures. Generally speaking quite a lot of information can be removed from a video image without the subjective image quality changing very much. This option is used to set just how much information is discarded - the lower the quality the better the compression will be, but the more information will be lost. You will probably want to experiment with different settings depending on what you actually intend to use the recorded videos for.

Saving your settings as a scheme

If you make adjustments to the settings in this window you can optionally choose to save your new settings as a 'compression scheme'. The scheme will then appear in the list at the bottom of the

Record video window, making it easy for you to reuse it.

Saving a scheme

To save your settings as a compression scheme simply click the *Save as...* button. The Save video compression scheme window will then open where you can give your new scheme a name.

Editing and deleting scheme

If you want to edit a scheme you've already saved then you should first select it in the *Compression scheme* drop-down list and then make your adjustments. Next, click the *Save as...* button and resave the scheme using the same name as it has already - the new settings will then overwrite the old ones.

To delete a scheme you should select it in the *Compression scheme* drop down list and then click the *Delete* button.

You can't edit or delete the standard: Best quality, Medium quality and Low quality schemes.

Adding a label to a video

Overview

If you click the  button while recording a video the *Add label window* will open. This allows you to add a text label to the recording which will be used to mark the video at *the time you **clicked** the button*. During playback of the video you can then jump directly to any such labelled point by simply selecting the label from a menu.

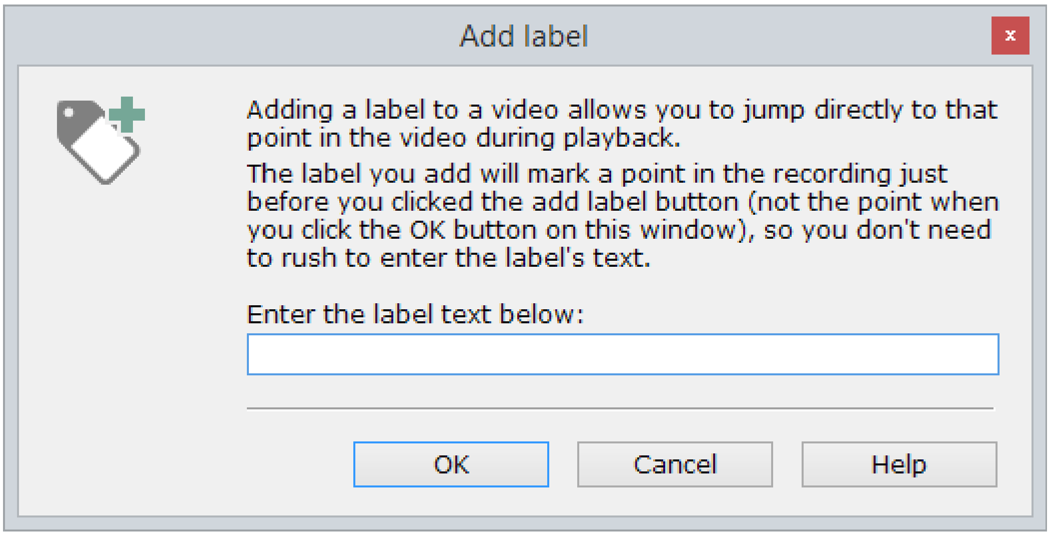


Figure 1. The add label window.

Details

You can enter anything you like as a label, up to a maximum of 79 characters.

Tracking in a video of a test

Introduction

Sometime it's convenient to record a video of a test, or tests, and then actually perform the tracking in the recorded video. This can be particularly useful if you only have one ANY-maze licence but you want to perform two experiments at the same time because you can use any number of unlicensed (i.e. free) copies of ANY-maze to record videos and then later perform the tracking using the licensed copy.

- Recording the video
- Designing a protocol for tracking in a video file
- Tracking in the video
- Controlling video playback
- Using the Search bar to move through a video recording

Recording the video

Full details about how to record a video within ANY-maze are given in the Recording videos of tests topic.

Alternatively you can record a video using any other video recording software that can save the video as either an AVI or Windows Media video file.

When recording a video which you later plan to track, there is a very important consideration that you must take into account: ANY-maze **must** see the empty apparatus before the test begins. When performing tests 'live' this isn't usually an issue, because ANY-maze is always analysing the image from the camera and so it will see the empty apparatus during the period before you put the animal in. However, when recording a video of a test you might start the video after you put the animal in the apparatus, in which case ANY-maze will never see the apparatus and tracking will therefore be compromised. There are a couple of ways you can address this requirement:

- Record a separate video of every test but make sure you start the video recording at least 5 seconds before you put the animal in the apparatus. This is usually quite easy to achieve, just start the video and *then* go and get the animal from its cage, as that usually takes more than 5 seconds.
- Alternatively you could just record a single video of the entire experiment. This way you will record all the period between tests and you'll necessarily always record the empty apparatus in the period after you finish cleaning the apparatus between tests. This is a simple method although you may want to add a label to the video before each test, identifying the animal being tested, as this will allow you to quickly locate specific tests

in the overall video file.

Designing a protocol for tracking in a video file

There's nothing you *have* to do to make a protocol suitable for tracking in a video file but you may want to give some thought to what you'll use as the video source.

Normally the video source you choose in a protocol show the image from the camera which is viewing your apparatus. But what if you plan to record videos on two *other* computers and just use this computer to track in the videos? In this case you won't be able to choose the camera that can see the apparatus because it's not even connected to this computer.

The answer in this situation would be to record a short video of the apparatus and then set the video source to use this video as the source of video images. This will mean that whenever you use this protocol it will, by default, show the same video file - however, all you'll then need to do is open the correct video file when you want to actually perform an experiment - how you do this is explained in the next section.

Of course, it might be that the camera *is* connected to this computer and therefore you won't have to use a video file as the video source - you can just use the camera in the normal way.

Tracking in the video

To track in a video recording you simply need to play the video instead of viewing the live image from your camera. To do this should click the button shown in the apparatus pane toolbar, which will cause the Choose video window to open where you can choose the video file you want to play.

Once the video is playing you can track in it in just the same way as you would if the image was coming from a camera - although, of course, you can do such things as pause playback and jump to different parts of the video, which wouldn't normally be able to do with a live image - see Controlling video playback below.

Changing the video file

If you want to change the video you're playing then you should click the button again, this will restore the live video image, and then click it a second time, which will display the Choose video window where you can choose a different video file.

Aligning the apparatus

When tracking in a video it's very common for the apparatus map to not align with the apparatus in the video picture. You can easily fix this by adjusting the apparatus map.

Offsetting the video image

When you play a video the size of the picture shown will NOT necessarily be the size of the picture recorded in the video - it will be the size of the picture supplied by the apparatus's video source as defined in the protocol.

This might seem strange but imagine what would happen if you had a big video source with a correspondingly big apparatus map and you then chose to play a small video - the apparatus map would extend beyond the edges of the picture.

In fact, when the picture in the video file is smaller than the video source ANY-maze just adds a black border around the picture to make it bigger. On the other hand, when the picture in the video file is larger than the video source ANY-maze clips some of the picture so it fits.

When a picture is clipped in this way you can use the button to offset the video picture making parts that are off-the-edges come into view. This can be very helpful if part of the apparatus has been clipped.

Controlling video playback

When playing a video you can use the buttons below the video picture to control playback:

	Play	Restarts the video when it's paused or stopped.
	Pause	Pauses the playing video.
	Rewind	Rewinds the video to the start and stops playback.
	Jump to label	Opens a menu listing any labels included in the video. Clicking a label will jump to that point in the recording.

Using the Search bar to move through a video recording

When playing a video, ANY-maze shows playback progress using a bar at the bottom of the video window. Using this bar you can scroll backwards and forwards through the video to any point.

In fact, to be strictly correct, you won't be able to scroll to *any point* because the search bar has a limited resolution based on its width, so if the bar is 1000 pixels wide and you are watching a video that lasts 4 hours, you will be able to jump to any point with, approximately, a 15 second resolution.

As an alternative to scrolling the search bar, you can simply change the playback time which is shown on the left end of the bar. To change the time, point the mouse at either the hours, minutes or seconds and roll the mouse wheel. Each 'step' of the wheel will alter the value being pointed at by 1. So, for example, to move forward 10 minutes in a video simply point at the minutes part of the time and roll the mouse wheel forward 10 steps. In fact, if you hold down the SHIFT key while rolling the wheel then the value will move forward 5 units, making it easier to move quickly.

This technique can also be used to move to an exact point in the video. For example, if you want to move to the point 2:14:00 (2 hours, 14 minutes), then you could point at the hours and roll the mouse wheel 2 steps, then point at the minutes and roll it 14 steps (as you roll the numbers change dynamically, so you don't have to count the steps).

Open ANY-maze video file

Overview

To open a video file in ANY-maze you should either:

- Click the  button in the toolbar just above the video picture on the Tests page.
- Select *ANY-maze video file* as a video source's *Source of video images.*
- Click the *Browse for video file* button on the Video page.

In all cases the *Choose video to play* window will be displayed.

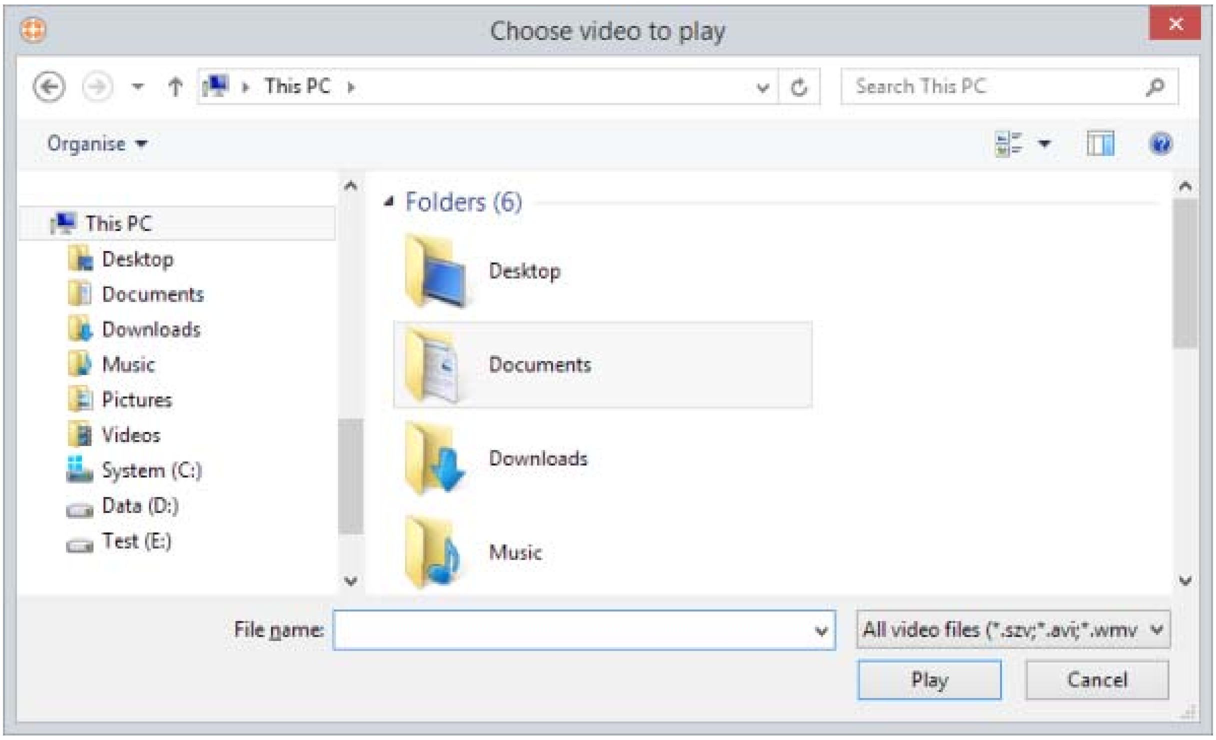


Figure 1. The Choose video to play window. Here you can browse your computer and/or network to find the video file you want to play.

Details

You will probably be familiar with this window already as it's based on the standard 'open file' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by *dragging* the bottom right corner with the mouse.

Files list The files list shows the folders and experiment files stored in the current location. You can click a file to select it, in which case clicking the *OK* button will open the file, or you can simply double click a file to open it without having to click OK. You can also open a folder, so you can see what's inside it, by double clicking the folder in this list.

File name You can use this field to enter the name of the file to open, however, it's usually much easier just to click a file in the *Files list*.

Type of file You can use this list to choose what types of files are shown in the *Files list*.

See also:

- Tracking in a video of a test
- Video source : Selecting the source of video images
- Playing videos files in the Video page

Recording movies of tests for inclusion in a presentation

The option described here records a movie of a test in a format suitable for playback in PowerPoint (or any standard media player). These movies are not suitable for subsequent tracking. If you would like to record a video of a test so you can track in it later then you should refer to the Recording and tracking with videos topic.

Overview

ANY-maze uses the name *movie* to refer to recordings of tests that are designed to be played back in PowerPoint or any standard media player.

An important point about movies is that they record *exactly* what you see on the screen during a test, this includes the animal marker, animal and zone shading, etc. Also note that, although I talk about recording *a test*, the recording actually just starts when *you* start it and ends when *you* end it - so you could choose to record an entire experiment; just part of a test - anything at all.

- Recording a movie
- Supported formats for recording movies
- Playing movies and including them in PowerPoint presentations

Recording a movie

To record a movie you should click the button that's shown just above the video picture on the Tests page. This will cause the *Record video as a movie...* window to open:

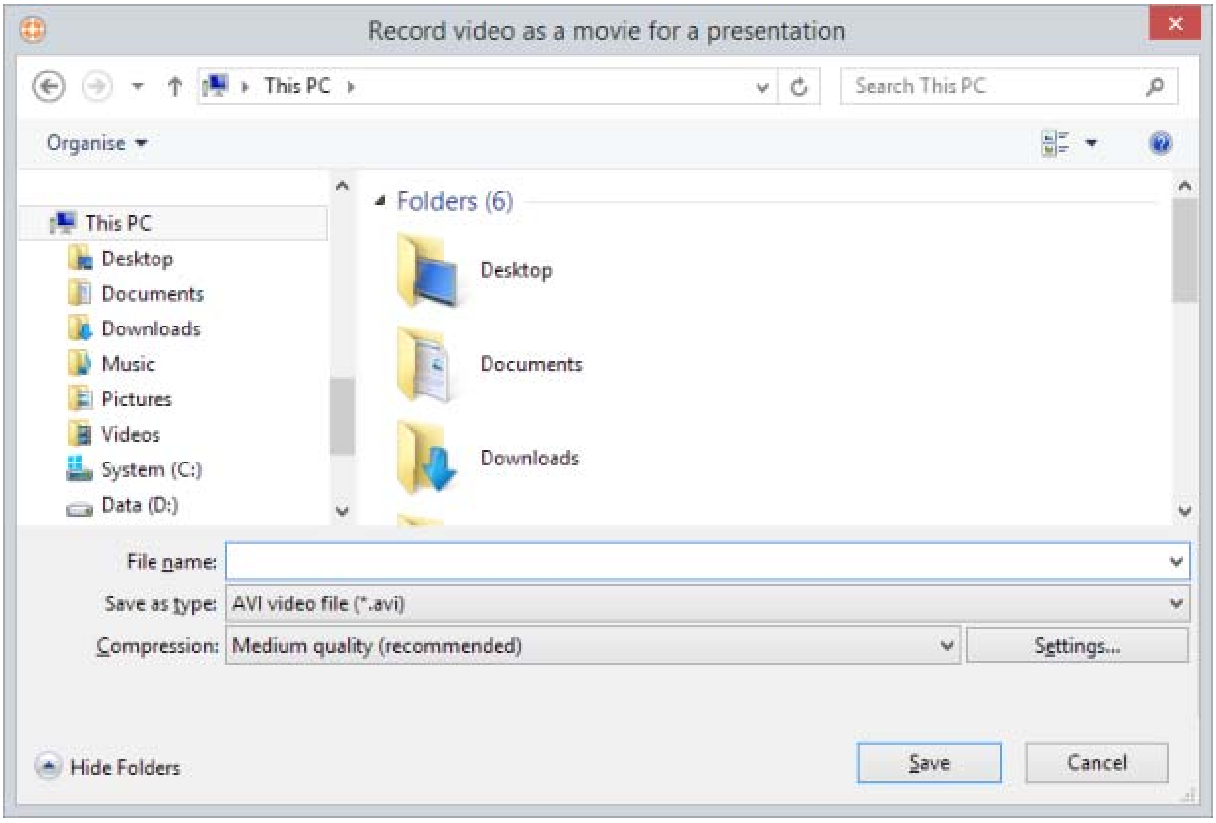


Figure 1. The Record movie window.

Files list The files list shows the folders and files of the selected format (see *Type of file* below) stored in the current location. You can double click a folder to open it and you can click a file to transfer its name to the File name field.

File name Use this field to enter a name for the video you're going to record. You can enter up 255 characters but your entry mustn't contain any of the following characters: \ / : * ? " < > |.

Type of file Select the format of the moviue you want to record. (Under Windows XP and Vista, you'll only have one choice - *AVI video file* - so you can ignore this list). For more information about the format used to record the movie see Supported formats for recording movies below.

Compression You can choose one of three standard compression schemes for the video - *Best quality, Medium quality* and *Low quality*. If you're recording to AVI format under Windows XP or Vista, you can choose any Compression scheme you've saved yourself using the options in the AVI compression window. If you're not sure which option to choose, I suggest you use *Medium quality,* which is fine for most situations.

Settings — Only available for AVI format. Clicking this button will open the *AVI compression window* where you can fine tune the compression settings for the movie you're going to record - see the AVI compression window topic for more details.

Once you have specified the name of the file you want to record to, you should click the OK button and recording will begin. While ANY-maze is recording a movie, a small flashing red dot will be displayed in the bottom right corner of the video image on the screen (however, this is won't appear in the recorded movie).

To end a recording, simply click the  button again.

Supported formats for recording movies

All movies recorded in ANY-maze will be in a standard format, which you can play in PowerPoint, Windows Media Player and a host of other programs. Under Windows 7 and above, these videos can be recorded to MP4 or WMV format; under Windows XP and Vista, they will be recorded to AVI format.

Although the files will be recorded in a standard format, this doesn't define the *compression technique* used - this will depend on the compression scheme you choose in the *Record Movie Window* and the video codecs you have installed on your computer. There's more about this in the AVI compression window topic, but, if you don't want to get too involved in the details then I suggest you simply use one of the standard compression schemes - *Best quality, Medium quality* or *Low quality*. These will choose a compression method based on the quality you're indicating and the codecs you have installed.

Under Windows XP, you can install the Microsoft MPEG-4 codec from within ANY-maze, and the program will then use it by default. For Windows Vista onwards, you can install one of the following encoders to improve the quality of AVI movies:

- Microsoft Media Video 9 VCM codec
- Xvid codec

These codecs should work on all operating systems from Windows XP onwards, although they don't necessarily work well on all computers. Try each of them to see which works on your computer.

Playing movies and including them in PowerPoint presentations

Playing a movie

If you want to take a quick look at a movie you've just recorded, then the easiest way is to use the ANY-maze Video page. Just switch to the Video page and, select the *Browse for a video file* option. Note, however, that although the movie will have been recorded in colour, ANY-maze only plays movies in black and white - for this reason you may prefer to open your movie in Windows Media Player.

Including a movie in a PowerPoint presentation

To include a movie in a PowerPoint presentation you should first select the 'Insert' tab in PowerPoint, and then select the 'Video > Video on my PC' option in the ribbon bar. You will then be able to choose the movie using a standard *File open* window. The movie will then be added to your current slide. You can adjust how the movie is played-back, by clicking on it (so it is selected) and then using the options on the 'Format' and 'Playback' tabs (which are only displayed when a movie's selected).

The above instructions are for Microsoft PowerPoint 2013, your version of PowerPoint may differ in some details.

See also:

- AVI compression window

AVI video compression schemes

Recording videos to AVI format is only supported under Windows XP and Vista. Under Windows 7 and above, ANY-maze will record to MP4 or WMV format; the compression options for these video formats are much simpler, and so this window is not available.

Introduction

ANY-maze can record movies of tests in a standard format called AVI. These movies can then be incorporated into PowerPoint presentations or played in standard media players.

When a movie is recorded in an AVI file, it's usually compressed so the file isn't unusably big. This compression is performed by something called a video codec. If you wish to, you can choose which video codec to use to compress your movie and also adjust such things as the frame rate and the level of compression applied.

Having found some settings that generate movie files that satisfy your needs you can optionally choose to save them as a *Compression scheme* making it easy to reuse them in the future.

All these options are available in *AVI compression window,* which can be accessed by clicking the *Settings* button on the Record movie window.

- Choosing the codec to use
- Making adjustments to the recorded movie
- Saving your settings as a scheme

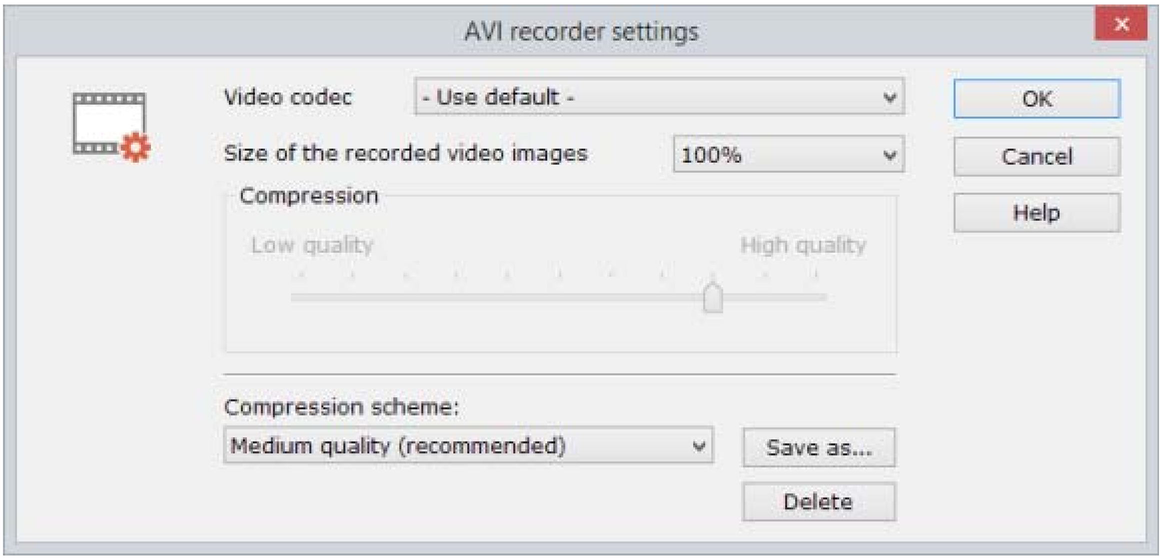


Figure 1. The AVI compression window.

Choosing the codec to use

The *Video Codec* drop-down list includes an entry for every video codec you have installed on your computer. Note, however, that ANY-maze may not be able to use all the listed codecs to record movies; for example, any codec that ends with the letters DMO is for a different video file format to the one used by ANY-maze and won't work. (If you try to use an unsupported codec ANY-maze will display a message when recording starts telling you this).

If you're not sure which codec is best to use them choose the option *Use default*, this will cause ANY-maze to choose the most suitable codec from those you have installed.

One reason you may wish to alter the codec you use to record a movie, is that different codecs place greater or lesser demands on your processor. For example, using the Microsoft Video 1 codec is less computationally intensive than using an MPEG-4 codec; on the other hand, the movie files recorded by MS Video 1 are much larger. Therefore, if you're using an older computer you may wish to use a less demanding codec, as this will have less impact of the tracking.

Modern installations of Windows include relatively few codecs that ANY-maze can use, however, you'll find various "codec packs" available for download on the Internet - a Google search for "codec pack" usually turns up quite a few, although we make no claims about their quality or suitability of use in ANY-maze.

Some other good choices for use in ANY-maze are:

- Microsoft Media Video 9 VCM codec
- Xvid codec

These codecs should work on all operating systems from Windows XP onwards, although they don't *necessarily work well on all computers. Try each of them to see which works on your computer.*

Making adjustments to the recorded movie

Altering the image size

By default the movie you record will be the same size as the image shown on the screen. However, you can alter the size by choosing an option from the *Size of recorded video images* list - but be warned, *increasing* the size will place much greater demands on your computer.

Altering the compression level

Some codecs can have the amount of compression they apply to the images adjusted. This can help to further reduce the size of the recorded files, but at the expense of quality. For codecs that support this option, you can adjust the quality using the *Compression* slider.

Saving your settings as a scheme

If you make adjustments to the settings in this window you can optionally choose to save your new settings as a *compression scheme*. The scheme will then appear in the list at the bottom of the Record movie window, making it easy for you to reuse it.

Saving a scheme

To save your settings as a compression scheme simply click the *Save as...* button. The Save video compression scheme window will then open where you can give your new scheme a name.

Editing and deleting scheme

If you want to edit a scheme you've already saved then you should first select it in the *Compression scheme* drop-down list and then make your adjustments. Next, click the *Save as...* button and resave the scheme using the same name as it has already - the new settings will then overwrite the old ones.

To delete a scheme you should select it in the *Compression scheme* drop down list and then click the *Delete* button.

You can't edit or delete the standard: Best quality, Medium quality and Low quality schemes.

Save video compression scheme

Details

Use this window to enter a name for your compression scheme. You can enter anything you like, up to a maximum of 64 characters.

The Results page

Introduction

As the name implies, the Results page in ANY-maze is designed for you to view an experiment's results.

The first thing to understand is that the Results page is designed to work with collated (i.e. grouped) results. If you want to look at the results for an individual test or an individual animal, then the Test details and Animal details reports are better places to go. Also, if you want to access the *raw* results, to transfer them to another program for example, then the Data page spreadsheet will be more suitable. But for looking at results for different treatment groups, age groups, sexes etc. the Results page is the right place.

Using the Results page you can view an experiment's results as text reports, graphs, track plots or heat maps. You can also perform quite sophisticated statistical analysis - for a general introduction see An introduction to viewing results.

You'll find full details about creating and working with results reports in the following topics:

- Viewing results in text reports
- Viewing results as graphs
- Performing statistical analysis
- Viewing results as track plots or heat maps
- Working with the results report
- Viewing reports that are defined in the Protocol
- Publishing or e-mailing report sets

Viewing results in text reports

For general information about the Results page, refer to the **Results page** *topic.*

Introduction

Text reports are the basic report style in ANY-maze. They show the results collated by groups using a simple tabular format. For each group the N, mean and raw data are shown - you can also optionally include the standard error of the mean and/or standard deviation.

Test duration (s)

Treatment	*N*	*Mean*	*SD* (note 1)	*Data* (notes 2, 3)
Saline	6	9.38	±6.03	19.0, 8.5, 11.3, 3.7, 11.4, 2.4
Compound X 1.0 mg/kg	5	18.72	±11.53	3.1, 30.7, 24.3, 10.3, 25.2
Compound X 5.0 mg/kg	5	33.46	±16.19	43.5, 35.7, 51.9, 26.2, 10.0
Compound X 10.0 mg/kg	5	83.76	±31.34	118.5, 110.3, 85.9, 53.1, 51.0

Notes:
1. SD = Standard deviation.
2. The data analysed has been limited in the following way: Trial = Treated trial.
3. Animals which are marked as retired have been excluded from this report.

Figure 1. An example of a simple text report.

To create a text report you should:

1. Switch to the Results page.
2. Select the *Text* button in the *Report style* section of the ribbon bar.
3. If the report settings are not already displayed, click the *Edit report settings* button in the *Actions* section of the ribbon bar.
4. On the *Text report settings* page you can:
   - Select the results to include in the report
   - Specify how the data should be grouped
   - Choose what data should be shown
   - Limit the data included in the report
5. Once you've specified the settings you want to use, you should click the *View the*

report button in the *Actions* section of the ribbon bar.

Selecting the results to include in a text report

In brief

You should use the list of measures shown at the top of the *Text report settings page* to choose the results that you want to include in the report. The measures are shown in groups, and clicking a group's title opens it so you can select the measures inside. You can choose as many measures as you like - each one will be analysed separately in the report.

Details

A text report will show analysis of each of the measures you select, in the order in which they appear in the measures list.

To select a measure in the list, first click the title of its group, for example *Apparatus measures,* and then check the box next to the measure's name, for example *Test duration* - see figure 1.

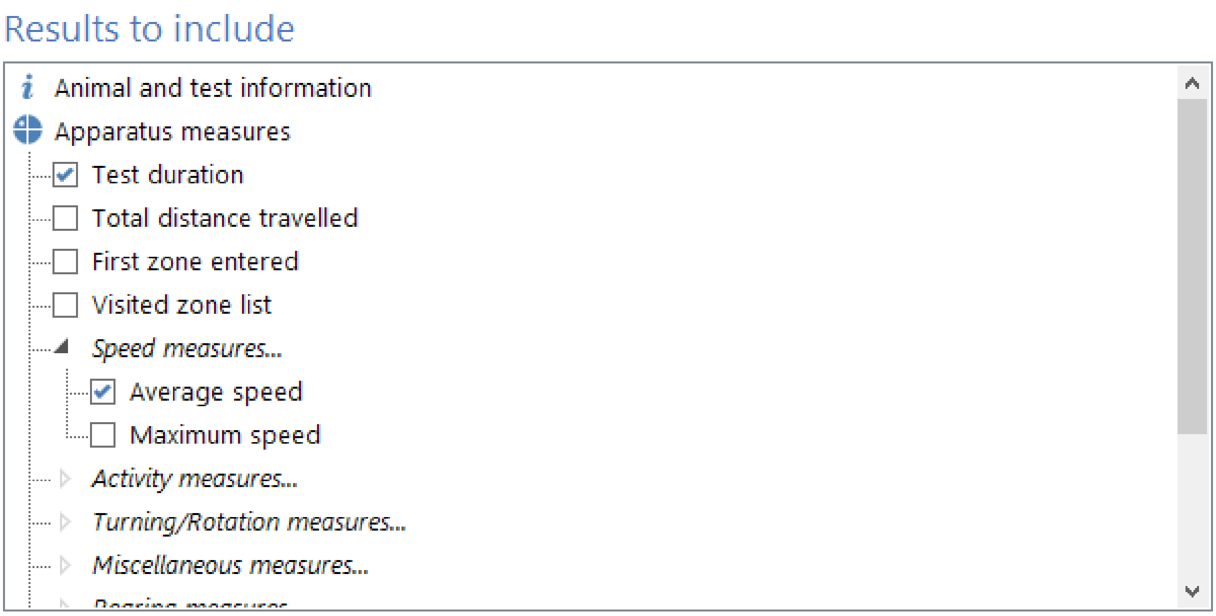


Figure 1. Measures are organised into groups - clicking a title opens the relevant group.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of measures.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures

* Key measures
* On/off input measures
* Signal measures
* Sensor measures
* Rotary encoder measures
* Movement detector measures
* Output switch measures
* Syringe pump measures
* Laser controller measures
* OPAD measures
* Procedure measures
* Event measures
* Virtual switch measures

Specifying how the data should be grouped in a text report

In brief

A text report shows the results of measures for groups of tests. For example, a text report which is grouped by *Treatment* would show the results for each treatment group in the experiment - see figure 1.

Test duration (s)

Treatment	*N*	*Mean*	*SD* (note 1)	*Data* (notes 2, 3)
Saline	6	9.38	±6.03	19.0, 8.5, 11.3, 3.7, 11.4, 2.4
Compound X 1.0 mg/kg	5	18.72	±11.53	3.1, 30.7, 24.3, 10.3, 25.2
Compound X 5.0 mg/kg	5	33.46	±16.19	43.5, 35.7, 51.9, 26.2, 10.0
Compound X 10.0 mg/kg	5	83.76	±31.34	118.5, 110.3, 85.9, 53.1, 51.0

Notes:
1. SD = Standard deviation.
2. The data analysed has been limited in the following way: Trial = Treated trial.
3. Animals which are marked as retired have been excluded from this report.

Figure 1. A text section grouped by treatment.

You can select up to three *nested* groups, where the nesting causes groups to be further subdivided - see figure 2.

Test duration (s)

Group (note 1)	*N*	*Mean*	*SD* (note 2)	*Data* (notes 3, 4)
Saline				
Male	3	13.90	±4.42	19.0, 11.3, 11.4
Female	3	4.87	±3.21	8.5, 3.7, 2.4
Compound X 1.0 mg/kg				
Male	3	12.57	±10.78	3.1, 24.3, 10.3
Female	2	27.95	±3.89	30.7, 25.2
Compound X 5.0 mg/kg				
Male	2	22.85	±18.17	35.7, 10.0
Female	3	40.53	±13.10	43.5, 51.9, 26.2
Compound X 10.0 mg/kg				
Male	2	80.65	±41.93	110.3, 51.0
Female	3	85.83	±32.70	118.5, 85.9, 53.1

Notes:
1. The data have been grouped by Treatment, then by Sex.
2. SD = Standard deviation.
3. The data analysed has been limited in the following way: Trial = Treated trial.
4. Animals which are marked as retired have been excluded from this report.

Figure 2. A text section grouped by treatment and then by sex.

Details

You must select at least one group. If what you really want is just the raw data without any grouping or analysis, then you should probably use the spreadsheet shown on the Data page.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Choosing what data should be shown in a text report

In brief

Text sections always show the groups' names, Ns, means and raw data. You can also optionally choose to show the groups' standard deviations and/or standard errors.

If you wish to, you can also specify that the raw data shown should be labelled with animal and trial numbers - see figure 1. In fact this is extremely useful because the animal and trial number labels are *links* to the relevant Animal details and Test details reports, respectively.

35.7 (8,1)

Figure 1. Raw data can be labelled with animal and trial numbers. In this case the value is for animal 8, trial 1.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes setting all these options to un-checked.

Limiting the data included in a text report

In brief

By default, a text report will include all the tests performed in the experiment. However, there may be occasions when you're not interested in results of particular tests and you'd like them to be excluded from the report.

For example, in a water-maze experiment you might train the animals to find an island, then treat them and then test them again to see whether the treatment has affected their ability to find the island. In this experiment you would probably be most interested in the results of the post-treatment tests and perhaps the last training tests (which you could use for comparison). However, the earlier training tests would be of little actual interest.

To actually choose which tests are included, you should use the list shown at the bottom of the *Text report settings* page - see figure 1.

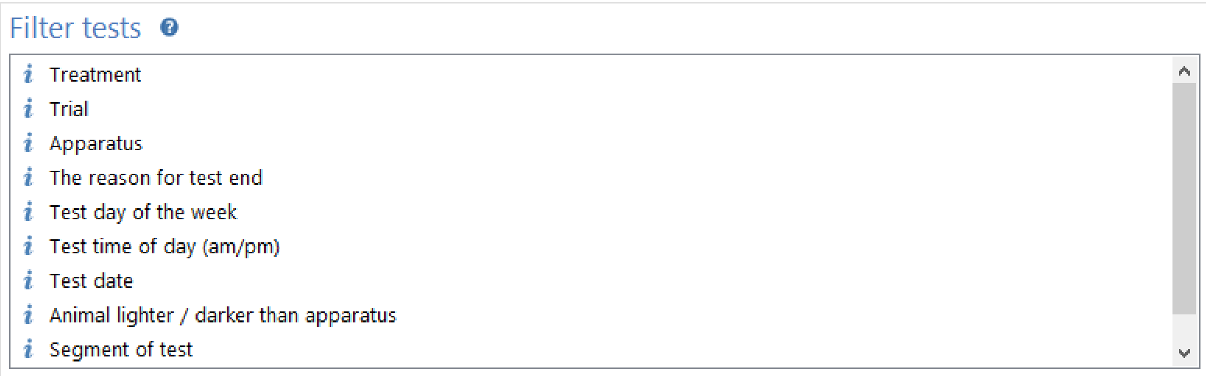


Figure 1. The list of measures used to limit the tests included in a text report.

Details

The list includes various measures - clicking a measure's name expands the list to show the individual values of the measure. For example, one of the measures listed is *Treatment* - clicking it will show a list of the experiment's treatments - see figure 2.

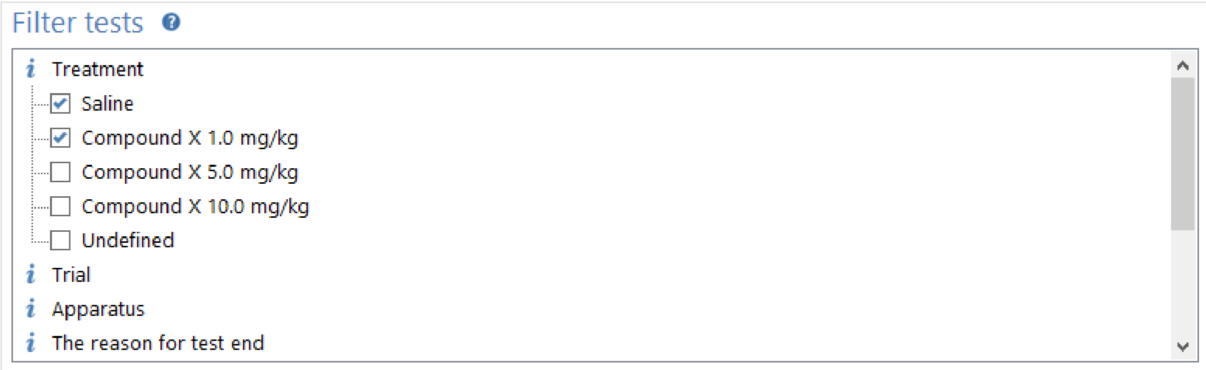


Figure 2. In this case the report will just include those tests performed on animals which received the treatments 'Saline' or 'Compound X 1.0 mg/kg'.

To limit the tests you should select items you want <u>included</u> - for example, in figure 2 the report will include all the tests performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg'.

You can choose to limit the tests based on more than one measure - for example you could also open the *Trial* measure and select just the *Re-test* trial. In this case only the tests which match the selected treatments AND the selected trials would be included. So, in our example, just the *Re-test* trials performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg' would be included in the report.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Note that if you don't select anything at all in this list, then *all* tests will be included.

Limiting analysis to certain 'segments' of tests

One of the measures you can use to limit a report is *Segment of test*; however this doesn't work quite the same as other limiting measures. The important point to understand is that whenever you use the *Segment of test* measure, ANY-maze divides tests into segments and analyses them rather than analysing entire tests.

For example, if you are using 30 second segments and you limit a report to include the segments 0-30 and 30-60, then the report will show results for <u>each segment</u> and not for each test. This means

that, in this example, you'd be given TWO results for each test, one for the period 0-30 and one for the period 30-60 and NOT, as you might expect, one result for the first 60 seconds.

The critical point is this: whenever you use the *Segment of test* measure, segments, rather than tests, become the entity that ANY-maze analyses.

The segment length is set using the *Analysis across time* options on the Protocol page, and you can use the *Set segment length* button in the ribbon bar to take you directly from the Results page to the relevant part of the protocol.

Selecting trials

Limiting the tests to just those which match certain trials can be a little less obvious than it appears. Going back to the example from the start of this topic, you'll recall that we were interested in a post-treatment trial and the **last** training trial. However, for some animals the last training trial might have been trial 3, for others trial 4 etc. depending on how quickly they learnt the task. For this reason, when listing the *Trials* in a stage, ANY-maze includes not just the individual trials (trial 1, trial 2, etc.) but also *Last trial* which will match the last trial for each animal, irrespective of which trial it happened to be. (For obvious reasons the *Last trial* isn't listed for stages which consist of just a single trial).

Excluding retired animals

As you may know, ANY-maze allows you to mark an animal as Retired. This means that the results of any tests the animal has already had are maintained, but the animal doesn't receive any further tests.

Retiring animals is common in experiments which include some type of training - animals which fail to achieve the training goal are retired.

Ordinarily, tests performed on retired animals are still included in reports, but you can choose to exclude them by checking the box shown at the bottom of the settings page titled *Don't include results for retired animals in the report.*

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of filters.

Viewing results as graphs

For general information about the Results page, refer to the Results page *topic.*

Introduction

As you would expect, graph reports show an experiment's results as a line, column or scatter graph. You can include multiple series on a single graph making it easy, for example, to see how a treatment affected results in different stages of an experiment.

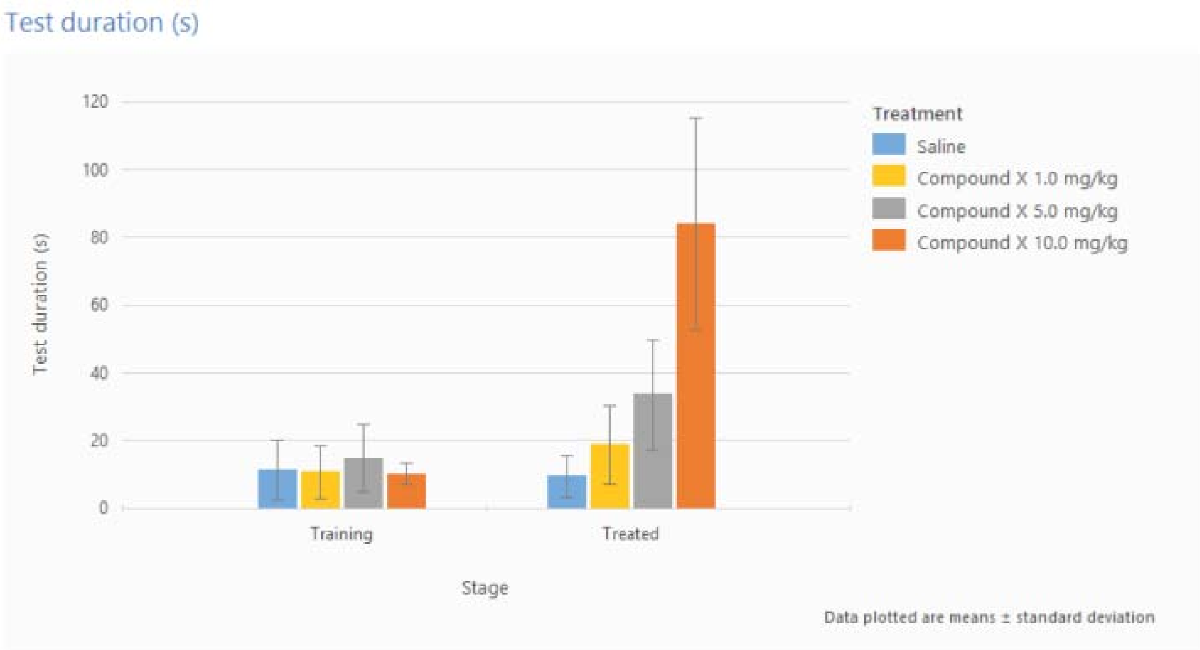


Figure 1. An example of a graph report.

To create a graph report you should:

1. Switch to the Results page.
2. Select the *Graph* button in the *Report style* section of the ribbon bar.

3. If the report settings are not already displayed, click the *Edit report settings* button in the *Actions* section of the ribbon bar.
4. On the *Graph report settings* page you can:
   - Select the measures you want to analyse
   - Specify how you want the graph organised
   - Specify the type of graph you want
   - Specify what the graph should show
   - Limit the data included in the graph
5. Once you've specified the settings you want to use, you should click the *View the report* button in the *Actions* section of the ribbon bar.

See also:

- Column graph format
- Line graph format
- Scatter graph format
- Graph printing format

Selecting the measures to analyse in a graph report

In brief

You should use the list of measures shown at the top of the *Graph report settings* page to choose those that you want to include in the report. The measures are shown in groups, and you should click a group's title to open it so you can select the measures inside. You can choose as many measures as you like - each one will be analysed separately in the report, i.e. it will appear as a separate graph (or graphs).

Details

A graph report will show a graph (or perhaps graphs) of each of the measures you select, in the order in which they appear in the measures list.

To select a measure in the list, first click the title of its group, for example *Apparatus measures,* and then check the box next to the measure's name, for example *Test duration* - see figure 1.

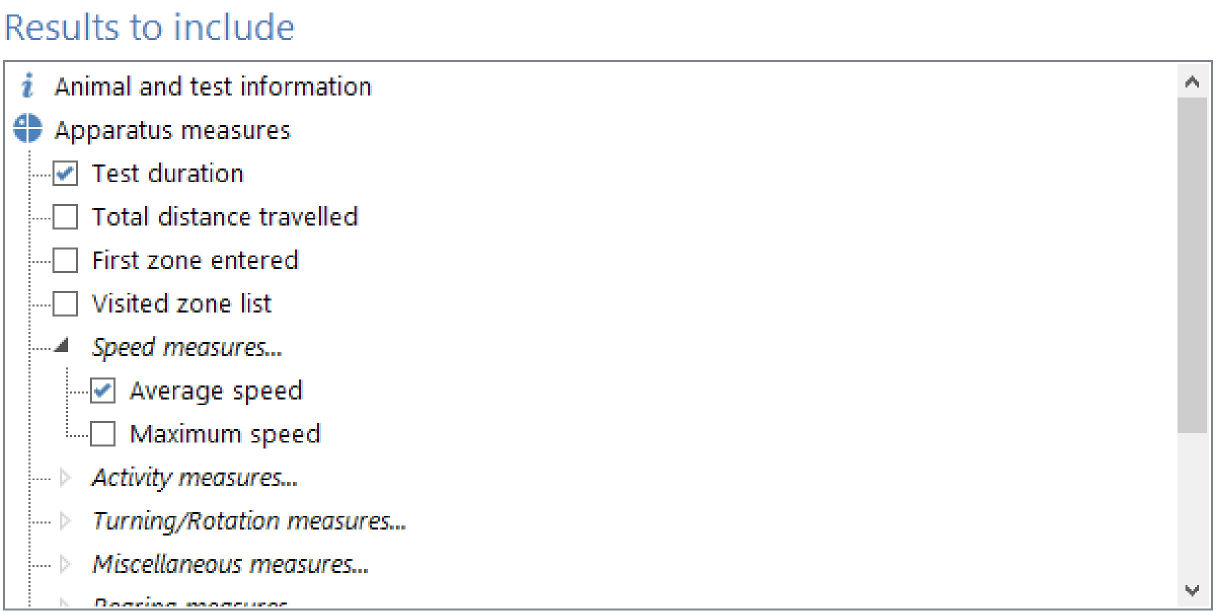


Figure 1. Measures are organised into groups - clicking a title opens the relevant group.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure — Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.

Select all — There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.

Deselect all — There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.

Expand all — There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.

Collapse all — There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of measures.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures

- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Specifying how graphs in a graph report should be organised

In brief

You should use the drop-down list labelled *On the x-axis show* to select what will be shown on the x-axis of your graph. For example, choosing *Treatment* will cause one column (or point if you use a line graph) to be shown for each of the treatment groups in your experiment.

You can optionally choose to break data into different series and/or across different graphs by making appropriate selections in the other two drop-down lists.

Details

Graphs plot the means of measures for particular groups. At its simplest, a graph will just show the different groups on the x-axis with the measure value on the y-axis - see figure 1.

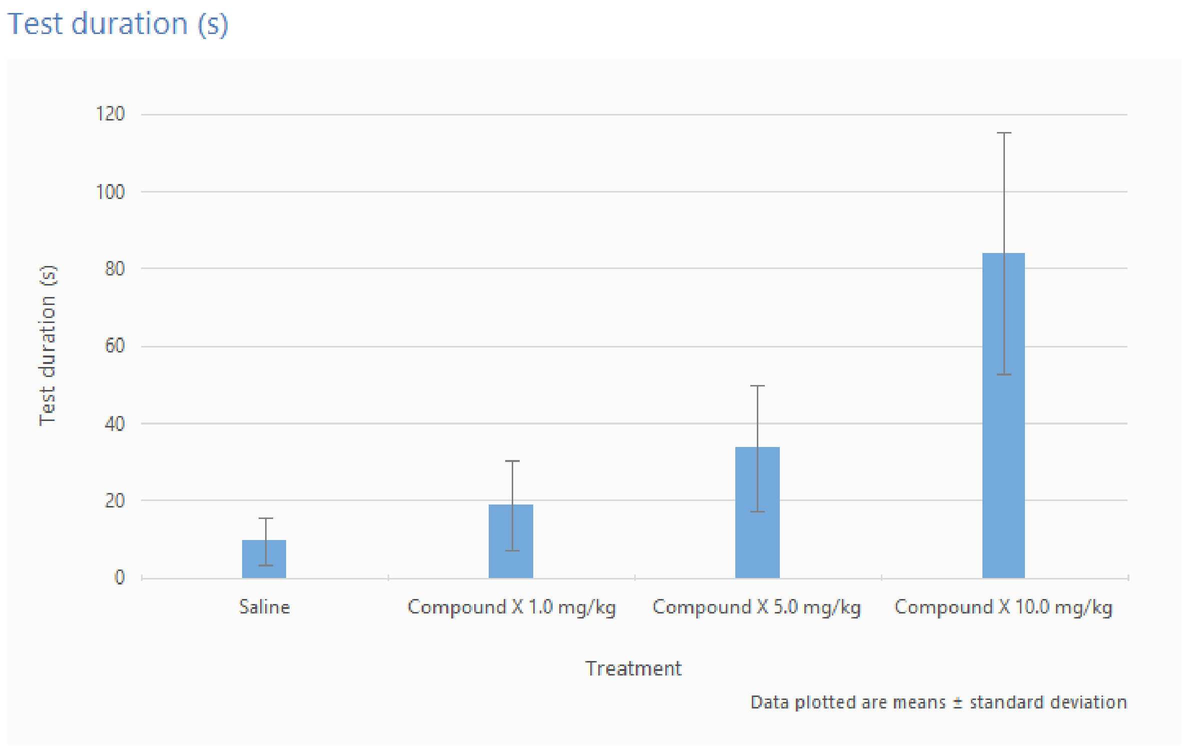


Figure 1. A simple graph showing the mean Test duration for different treatment groups.

However, you might want to split your data into different series too. For example, if your protocol includes multiple stages then you might want to show the stages on the x-axis and show different series for each of the treatment groups - see figure 2.

Test duration (s)

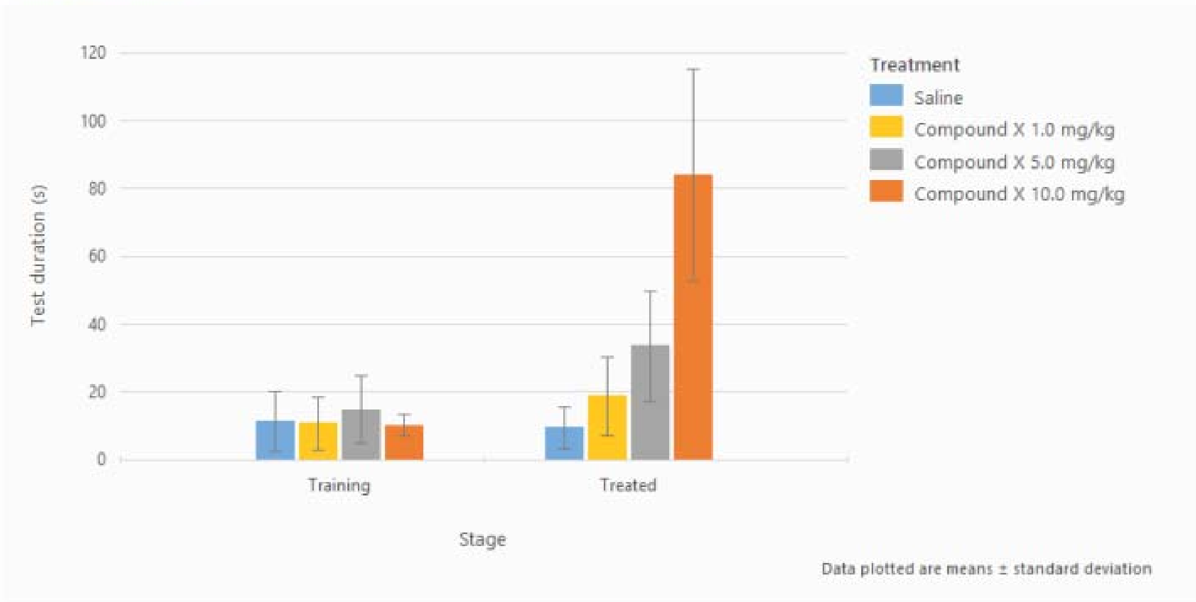


Figure 2. A graph showing the mean Test duration in different stages for different treatment groups.

Alternatively, in this situation, you might prefer to use different graphs, rather than different series on a single graph.

In fact, you can show different graphs *and* include different series on each of them. For example you might want different graphs for each stage in an experiment, with each graph having a different series for each treatment with the x-axis of each graph showing the different trials in the stage.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Specifying the type of graphs to show in a graph report

In brief

ANY-maze can create column, line or scatter graphs. The system will automatically select column graphs if the data on the x-axis is discrete, or line graphs if it's continuous. However, you can change this if you like using the options on the *Graph report settings* page.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Specifying exactly what the graph should show

In brief

ANY-maze plots mean values on line and column graphs and individual data points on scatter graphs, but you also can choose to include such things as error bars on column graphs or group medians on scatter plots.

Details

On column and line graphs, ANY-maze plots mean values and if you would like it to, it can also include error bars of either the standard deviation (SD) or standard error of the mean (SEM). The error bars can either be positive only, or positive and negative.

On scatter plots, the system shows the individual data points and can optionally include a line indicating the groups' means or medians.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Limiting the data included in a graph report

In brief

By default, a graph report will include all the tests performed in the experiment. However, there may be occasions when you're not interested in results of particular tests and you'd like them to be excluded from the report.

For example, in a water-maze experiment you might train the animals to find an island, then treat them and then test them again to see whether the treatment has affected their ability to find the island. In this experiment you would probably be most interested in the results of the post-treatment tests and perhaps the last training tests (which you could use for comparison). However, the earlier training tests would be of little actual interest.

To actually choose which tests are included you should use the list shown at the bottom of the *Graph report settings* page - see figure 1.

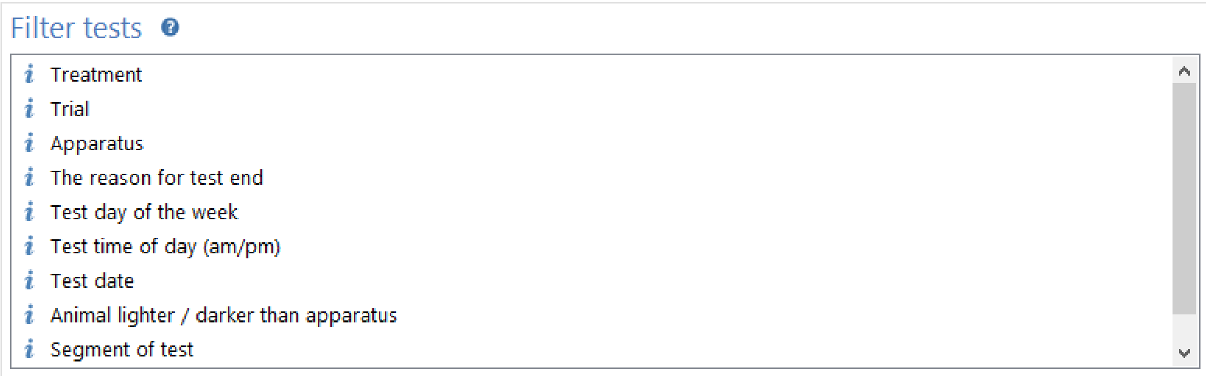


Figure 1. The list of measures used to limit the tests included in a graph report.

Details

The list includes various measures - clicking a measure's name expands the list to show the individual values of the measure. For example, one of the measures listed is *Treatment* - clicking it will show a list of the experiment's treatments - see figure 2.

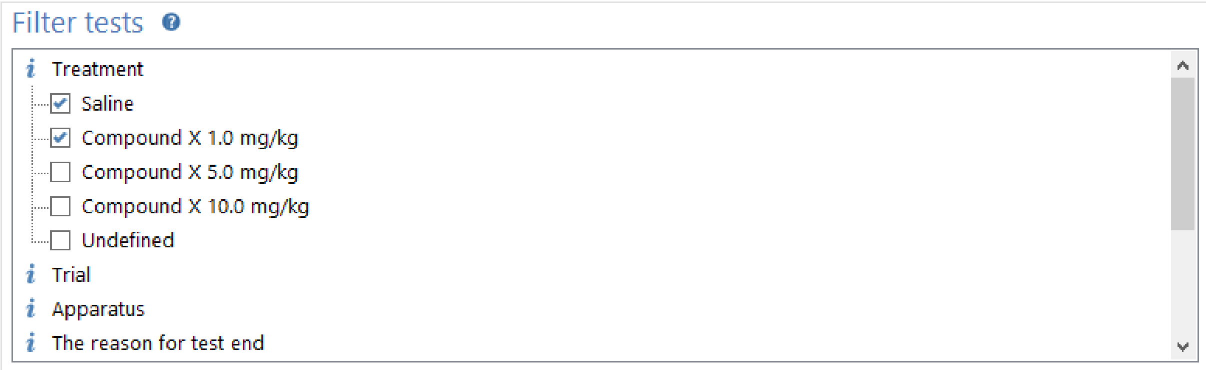


Figure 2. In this case the report will just include those tests performed on animals which received the treatments 'Saline' or 'Compound X 1.0 mg/kg'.

To limit the tests you should select items you want included - for example, in figure 2 the report will include all the tests performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg'.

You can choose to limit the tests based on more than one measure - for example you could also open the *Trial* measure and select just the *Re-test* trial. In this case only the tests which match the selected treatments AND the selected trials would be included. So, in our example, just the *Re-test* trials performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg' would be included in the report.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Note that if you don't select anything at all in this list, then *all* tests will be included.

Limiting analysis to certain 'segments' of tests

One of the measures you can use to limit a report is *Segment of test*; however this doesn't work quite the same as other limiting measures. The important point to understand is that whenever you use the *Segment of test* measure, ANY-maze divides tests into segments and analyses them rather than analysing entire tests.

For example, if you are using 30 second segments and you limit a report to include the segments 0-30 and 30-60, then the report will show results for each segment and not for each test. This means

that, in this example, you'd be given TWO results for each test, one for the period 0-30 and one for the period 30-60 and NOT, as you might expect, one result for the first 60 seconds.

The critical point is this: whenever you use the *Segment of test* measure, segments, rather than tests, become the entity that ANY-maze analyses.

The segment length is set using the *Analysis across time* options on the Protocol page, and you can use the *Set segment length* button in the ribbon bar to take you directly from the Results page to the relevant part of the protocol.

Selecting trials

Limiting the tests to just those which match certain trials can be a little less obvious than it appears. Going back to the example from the start of this topic, you'll recall that we were interested in a post-treatment trial and the **last** training trial. However, for some animals the last training trial might have been trial 3, for others trial 4 etc. depending on how quickly they learnt the task. For this reason, when listing the *Trials* in a stage, ANY-maze includes not just the individual trials (trial 1, trial 2, etc.) but also *Last trial* which will match the last trial for each animal, irrespective of which trial it happened to be. (For obvious reasons the *Last trial* isn't listed for stages which consist of just a single trial).

Excluding retired animals

As you may know, ANY-maze allows you to mark an animal as Retired. This means that the results of any tests the animal has already had are maintained, but the animal doesn't receive any further tests.

Retiring animals is common in experiments which include some type of training - animals which fail to achieve the training goal are retired.

Ordinarily, tests performed on retired animals are still included in reports, but you can choose to exclude them by checking the box shown at the bottom of the settings page titled *Don't include results for retired animals in the report.*

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of filters.

Column graph format

Introduction

If you right click on a graph, a menu will appear. One of the options shown is *Format graph...*, and selecting it will open the *Graph format* window. This window is divided into four *pages*, of which the first is used to define the format of column graphs - see figure 1.

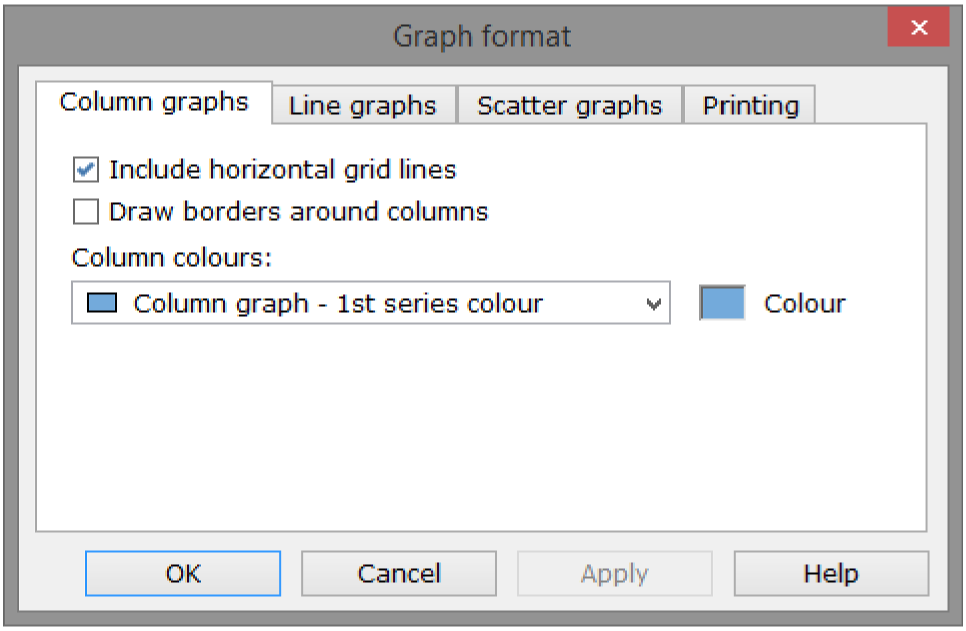


Figure 1. The Column graphs *page of the* Graph format *window.*

Details

Include horizontal grid lines	Checking this box will cause pale grid lines to be drawn across the graph for each value shown on the graph's y-axis.
Draw borders around columns	The columns of a graph are drawn in different colours, but you may want to draw borders around them to help differentiate them further.
Column colours	Column graphs can include up to 16 series where each one is drawn in a different colour. To change the colour of a series' column, just select it in the drop-down list and then click the 'colour box' shown

to the right of the list. The standard 'colour picker' window will open where you can choose the colour you want to use.

Note that you can also change the colours used in column graphs using the *Appearance* settings on the Options page.

Line graph format

Introduction

If you right click on a graph, a menu will appear. One of the options shown is *Format graph...*, and selecting it will open the *Graph format* window. This window is divided into four *pages*, of which the second is used to define the format of line graphs - see figure 1.

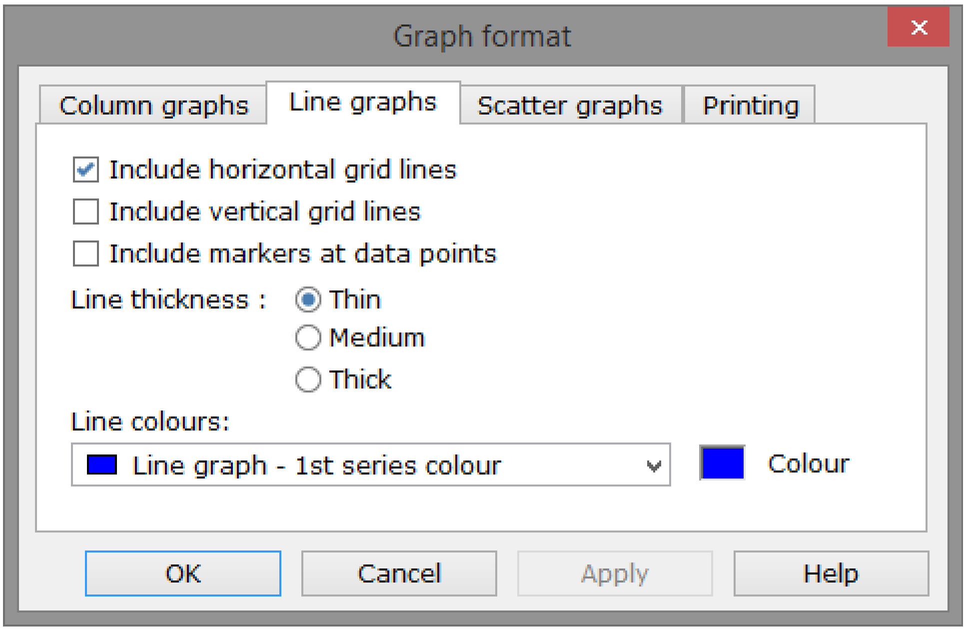


Figure 1. The Line graphs *page of the* Graph format *window.*

Details

Include horizontal grid lines	Checking this box will cause pale grid lines to be drawn across the graph for each value shown on the graph's y-axis.
Include vertical grid lines	Checking this box will cause pale grid lines to be drawn up the graph for each value shown on the graph's x-axis.
Include markers at data points	Checking this box will cause small markers to be drawn at the data points along the line graph. A different marker is used for each series shown.

Line thickness	Choose the thickness for all the lines on the graph. When showing more than one series, a thicker line usually helps to make the graph clearer.
Line colours	Line graphs can include up to 16 series where each one is drawn in a different colour. To change the colour of a series' line, just select it in the drop-down list and then click the 'colour box' shown to the right of the list. The standard 'colour picker' window will open where you can choose the colour you want to use. Note that you can also change the colours used in line graphs using the *Appearance* settings on the Options page.

Scatter graph format

Introduction

If you right click on a graph, a menu will appear. One of the options shown is *Format graph...*, and selecting it will open the *Graph format* window. This window is divided into four *pages* of which the third is used to define the format of scatter graphs - see figure 1.

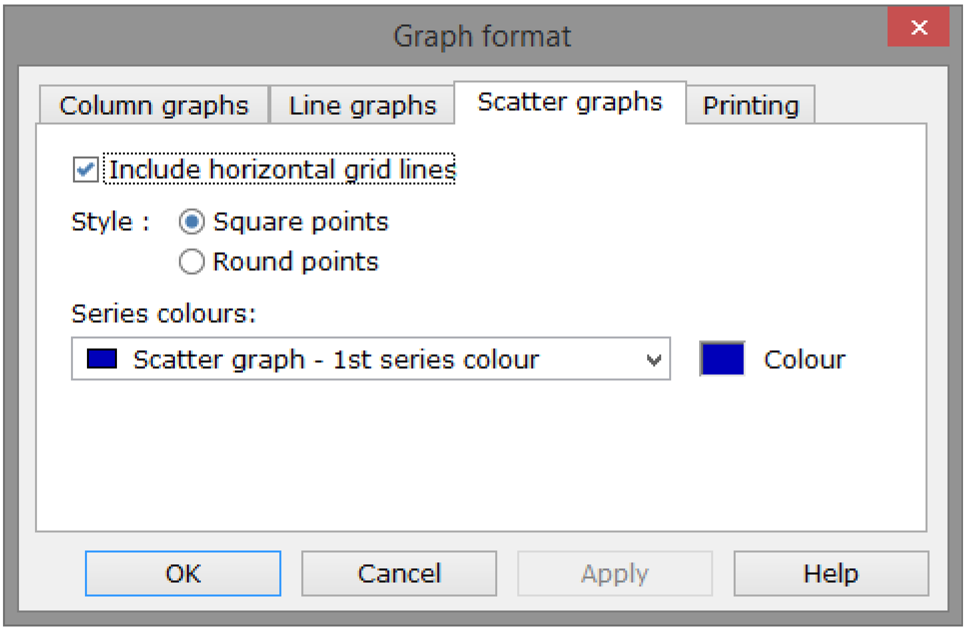


Figure 1. The Scatter graphs *page of the* Graph format *window.*

Details

Include horizontal grid lines	Checking this box will cause pale grid lines to be drawn across the graph for each value shown on the graph's y-axis.
Point style	Using this option, you can choose whether the points plotted on a scatter graph are square in shape or round.
Series colours	Scatter graphs can include up to 16 series where each one is drawn in a different colour. To change the colour of a series' points, just select it in the drop-down list and then click the 'colour box' shown to the right of the list. The standard 'colour picker' window will

open where you can choose the colour you want to use.

Note that you can also change the colours used in scatter graphs using the *Appearance* settings on the Options page.

Graph printing format

Introduction

If you right click on a graph, a menu will appear. One of the options shown is *Format graph...*, and selecting it will open the *Graph format* window. This window is divided into four *pages*, of which the last is used to specify options to use when printing graphs - see figure 1.

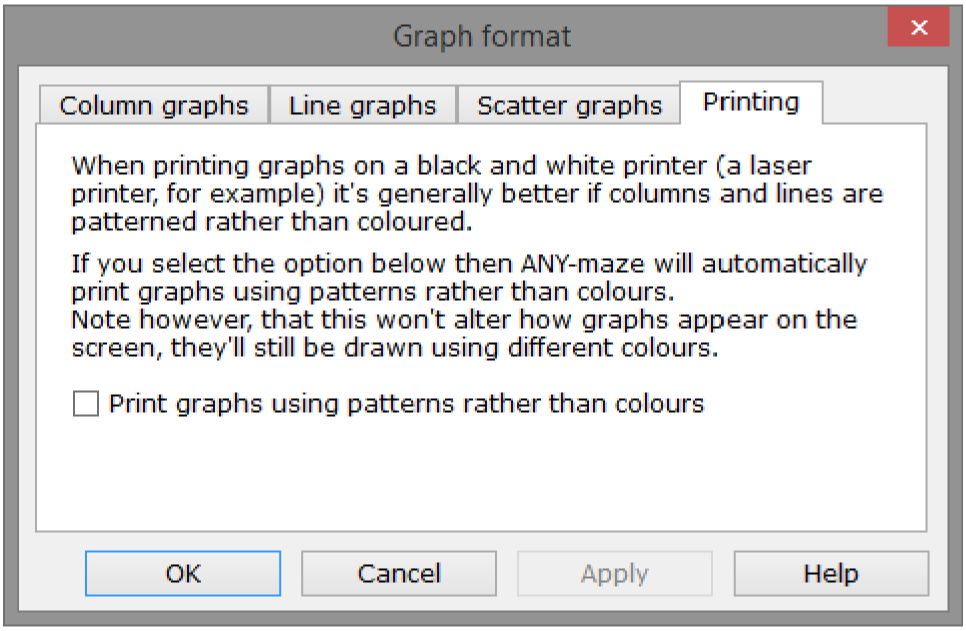


Figure 1. The Printing *page of the* Graph format *window.*

Details

In fact, there's only one 'Printing' option, which is to choose whether graphs should be printed using patterns or colours.

On the screen, ANY-maze will always display graphs using different colours for different series. However if you print such a graph on a black and white printer, the colours are converted to shades of grey, some of which may appear to be almost identical.

For this reason it's a good idea, when printing on a black and white printer, to print graphs using patterns for the different series and you can do this by checking the box shown here.

By the way, you don't have to use this option only for black and white printers; you can use it for colour printers too - this is a good idea if you expect you'll photocopy the printouts or if you simply want to save colour ink.

Note that this option can also be set using the *Printing* settings on the Options page.

This option ONLY affects printed graphs. On screen, ANY-maze always uses different colours for different series.

Performing statistical analysis

For general information about the Results page, refer to the Results page *topic.*

Introduction

A great feature of ANY-maze is its ability to perform statistical analysis of an experiment's results. While not as sophisticated as full statistical analysis programs such as SPSS or Systat, ANY-maze is able to analyse most results using its battery of built-in statistical tests.

Statistical tests are performed by simply creating an appropriate statistical analysis report. As you'll see, stats reports are much like text reports except that they include analysis of the groups shown.

Test duration (s)

Treatment	*N*	*Mean*	*SD* (note 1)	*Data* (notes 2, 3)
Saline	6	9.38	±6.03	19.0, 8.5, 11.3, 3.7, 11.4, 2.4
Compound X 1.0 mg/kg	5	18.72	±11.53	3.1, 30.7, 24.3, 10.3, 25.2
Compound X 5.0 mg/kg	5	33.46	±16.19	43.5, 35.7, 51.9, 26.2, 10.0
Compound X 10.0 mg/kg	5	83.76	±31.34	118.5, 110.3, 85.9, 53.1, 51.0

ANOVA (notes 4, 5) : $F_{(3,17)} = 17.0868$ $p = 0.000$

Student-Newman-Keuls test (notes 5, 6)

Saline vs. Compound X 1.0 mg/kg	$q_{(17,2)} = 1.1655$	$p = 0.421$
Saline vs. Compound X 5.0 mg/kg	$q_{(17,3)} = 3.0056$	$p = 0.114$
Saline vs. Compound X 10.0 mg/kg	$q_{(17,4)} = 9.2848$	$p = 0.000$
Compound X 1.0 mg/kg vs. Compound X 5.0 mg/kg	$q_{(17,2)} = 1.8401$	$p = 0.211$
Compound X 1.0 mg/kg vs. Compound X 10.0 mg/kg	$q_{(17,3)} = 8.1192$	$p = 0.000$
Compound X 5.0 mg/kg vs. Compound X 10.0 mg/kg	$q_{(17,2)} = 6.2792$	$p = 0.001$

Notes:
1. SD = Standard deviation.
2. The data analysed has been limited in the following way: Trial = Treated trial.
3. Animals which are marked as retired have been excluded from this report.
4. One-way ANOVA, fixed factor.
5. $p \leq 0.05$ is highlighted in red.
6. The error term (mean square error) = 334.800

Figure 1. An example of a statistical analysis report.

To create a stats report you should:

1. Switch to the Results page.
2. Select the *Statistical* button in the *Report style* section of the ribbon bar.
3. If the report settings are not already displayed, click the *Edit report settings* button in the *Actions* section of the ribbon bar.
4. On the *Statistical analysis settings* page you can:
   - Select the dependent variables you want to analyse
   - Select the independent variables you want to use
   - Specify the analysis you want to perform
   - Set optional information you want shown
   - Limit the data being analysed
5. Once you've specified the settings you want to use, you should click the *View the report* button in the *Actions* section of the ribbon bar.

Changing the level of significance (alpha) for statistical tests

Note that the level of significance (alpha) for statistical tests can be changed using the *Analysis options* on the Protocol page, and you can use the *Set alpha* button in the ribbon bar to take you directly from the Results page to the relevant part of the protocol.

Selecting the dependent variables to analyse in a stats report

In brief

You should use the list of measures shown at the top of the *Statistical analysis settings* page to choose the dependent variable, or variables, you want to analyse. The measures are shown in groups, and you should click a group's title to open it so you can select the measures inside. You can choose as many measures as you like - each one will be analysed separately in the report.

Details

A statistics report will analyse each of the measures you select. The results of the analysis will appear in the order in which they appear in the measures list.

To select a measure in the list, first click the title of its group, for example *Apparatus measures*, and then check the box next to the measure's name, for example *Test duration* - see figure 1.

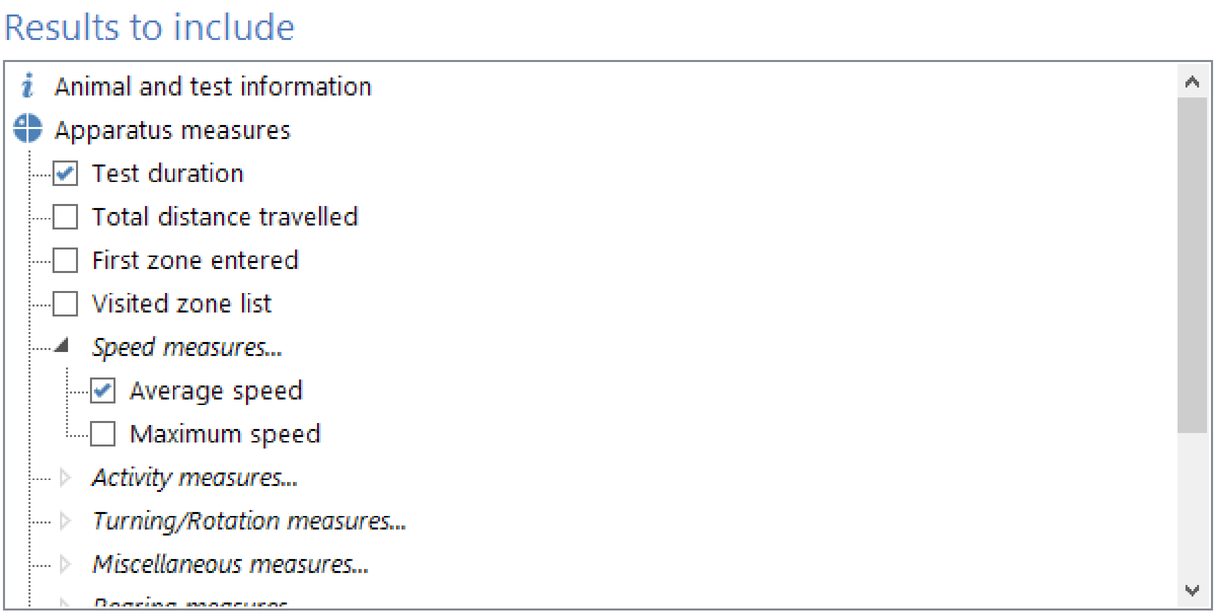


Figure 1. Measures are organised into groups - clicking a title opens the relevant group.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.
Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of dependent variables to analyse.

See also:

- Information measures
- Apparatus measures
- Zone measures

* Point measures
* Sequence measures
* Key measures
* On/off input measures
* Signal measures
* Sensor measures
* Rotary encoder measures
* Movement detector measures
* Output switch measures
* Syringe pump measures
* Laser controller measures
* OPAD measures
* Procedure measures
* Event measures
* Virtual switch measures

ANY-maze help topic H0764

Selecting the independent variables to use in a stats report

In brief

You should use the two drop-down lists shown on the statistical analysis settings page to select one or two independent variables that you want to use in the analysis. Only measures which can logically be used as independent variables are listed.

As you'd expect, if you choose two independent variables ANY-maze will perform an appropriate 2 way test.

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Specifying the analysis to perform in a stats report

In brief

When choosing the analysis to perform in a stats report, the first thing to decide is whether you want to analyse the dependent variables using parametric or non-parametric tests. ANY-maze will use your choice together with information it already knows about the dependent variables (for example, whether they represent nominal data) to choose an appropriate test. For full details of the tests used, refer to: Statistical tests included in ANY-maze.

If you wish to perform *post-hoc* analysis of any statistically significant results then, for parametric data, you can select a suitable test from the *Post-hoc test* drop-down list shown on the statistical analysis settings page. If you choose to use Dunnett's test (which is Dunnett's test for comparison to a control group) then you will also need to select the control group for the post-hoc comparisons. Note that for non-parametric data, ANY-maze will automatically select a post-hoc test for you - again you'll find details in Statistical tests included in ANY-maze.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Selecting the optional information to show in a stats report

In brief

Statistics reports always show groups' names, Ns, means and raw data as well as the results of the statistical analysis. If you wish to, you can also choose to include the groups' standard deviations and/or standard errors.

You can also specify that the raw data shown is labelled with animal and trial numbers - see figure 1. In fact this is extremely useful because the animal and trial number labels are *links* to the relevant Animal details and Test details reports, respectively.

35.7 (8,1)

Figure 1. Raw data can be labelled with animal and trial numbers. In this case the value is for animal 8, trial 1.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Limiting the data analysed in a stats report

In brief

By default, a stats report will analyse all the tests performed in the experiment. However, there may be occasions when you're not interested in results of particular tests and you'd like them to be excluded from the report.

For example, in a water-maze experiment you might train the animals to find an island, then treat them and then test them again to see whether the treatment has affected their ability to find the island. In this experiment you would probably be most interested in the results of the post-treatment tests and perhaps the last training tests (which you could use for comparison). However, the earlier training tests would be of little actual interest.

To actually choose which tests are included you should use the list shown at the bottom of the *Statistical report settings* page - see figure 1.

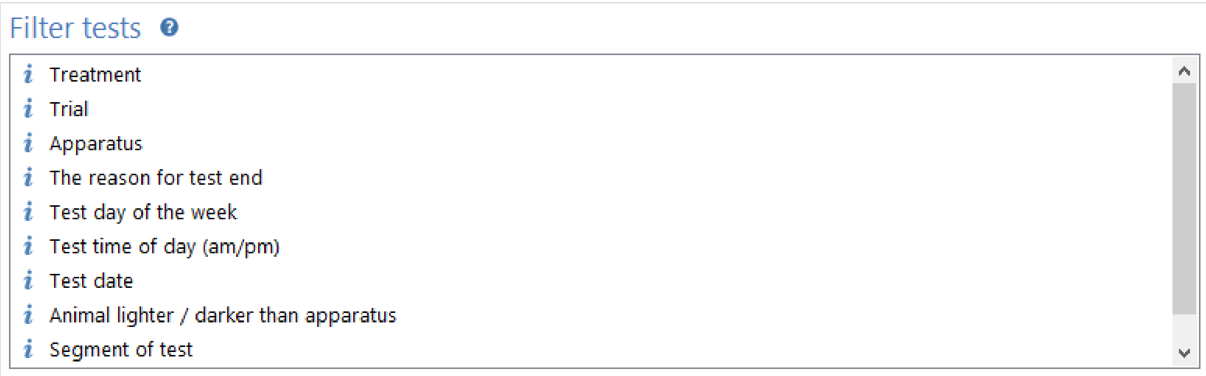


Figure 1. The list of measures used to limit the tests included in a stats report.

Details

The list includes various measures - clicking a measure's name expands the list to show the individual values of the measure. For example, one of the measures listed is *Treatment* - clicking it will show a list of the experiment's treatments - see figure 2.

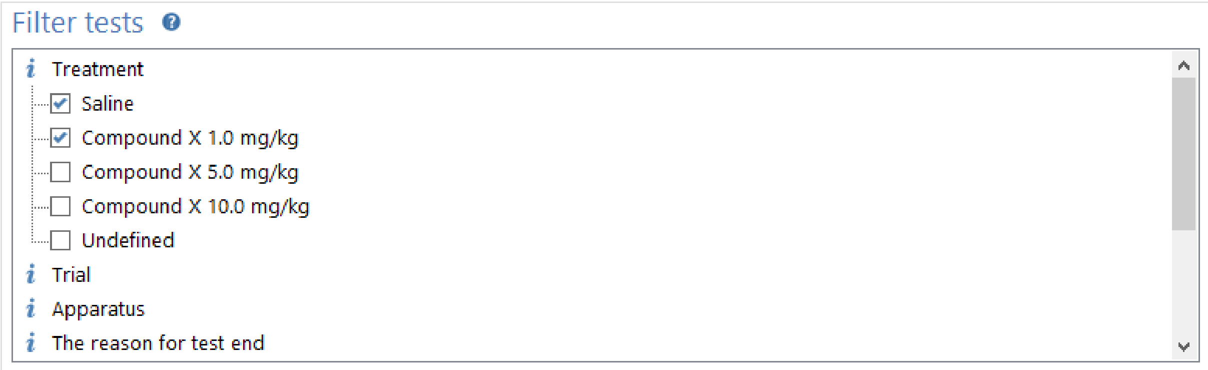


Figure 2. In this case the report will just include those tests performed on animals which received the treatments 'Saline' or 'Compound X 1.0 mg/kg'.

To limit the tests you should select items you want <u>included</u> - for example, in figure 2 the report will include all the tests performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg'.

You can choose to limit the tests based on more than one measure - for example you could also open the *Trial* measure and select just the *Re-test* trial. In this case only the tests which match the selected treatments AND the selected trials would be included. So, in our example, just the *Re-test* trials performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg' would be included in the report.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Note that if you don't select anything at all in this list, then *all* tests will be included.

Limiting analysis to certain 'segments' of tests

One of the measures you can use to limit a report is *Segment of test*; however this doesn't work quite the same as other limiting measures. The important point to understand is that whenever you use the *Segment of test* measure, ANY-maze divides tests into segments and analyses them rather than analysing entire tests.

For example, if you are using 30 second segments and you limit a report to include the segments 0-30 and 30-60, then the report will show results for <u>each segment</u> and not for each test. This means

that, in this example, you'd be given TWO results for each test, one for the period 0-30 and one for the period 30-60 and NOT, as you might expect, one result for the first 60 seconds.

The critical point is this: whenever you use the *Segment of test* measure, segments, rather than tests, become the entity that ANY-maze analyses.

The segment length is set using the *Analysis across time* options on the Protocol page, and you can use the *Set segment length* button in the ribbon bar to take you directly from the Results page to the relevant part of the protocol.

Selecting trials

Limiting the tests to just those which match certain trials can be a little less obvious than it appears. Going back to the example from the start of this topic, you'll recall that we were interested in a post-treatment trial and the **last** training trial. However, for some animals the last training trial might have been trial 3, for others trial 4 etc. depending on how quickly they learnt the task. For this reason, when listing the *Trials* in a stage, ANY-maze includes not just the individual trials (trial 1, trial 2, etc.) but also *Last trial* which will match the last trial for each animal, irrespective of which trial it happened to be. (For obvious reasons the *Last trial* isn't listed for stages which consist of just a single trial).

This issue is particularly important because 'Trials' will be a *repeated measure* independent variable and the statistical analysis requires that all animals have data for all *levels* of repeated measures. Thus if you were to analyse 'Last training trial' and 'Treated trial', the measure would have two levels and all animals would indeed have data for all levels. However, if you analysed **all** the 'Training trials' and the 'Treated trial' then the total number of levels might be say, 7 (6 training trials and 1 treated) but most animals would lack data for at least some levels (i.e. only animals which had 6 training trials would have data at all levels). In this case ANY-maze would remove the data for all the animals that don't have data at all levels (see Why tests are excluded from statistical analysis for more details about this) and the analysis would become almost useless.

Excluding retired animals

As you may know, ANY-maze allows you to mark an animal as Retired. This means that the results of any tests the animal has already had are maintained, but the animal doesn't receive any further tests.

Retiring animals is common in experiments which include some type of training - animals which fail to achieve the training goal are retired.

Ordinarily, tests performed on retired animals are still included in reports, but you can choose to exclude them by checking the box shown at the bottom of the settings page titled *Don't include results for retired animals in the report.*

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of filters.

Statistical tests included in ANY-maze

Introduction

ANY-maze includes a number of different statistical tests which it can use to analyse your data. As the system already knows certain things about the variables to analyse, such as whether they represent nominal data, whether there are repeated measures etc., it is able to choose an appropriate parametric or non-parametric test for itself - full details are given below:

Analysis of nominal data

ANY-maze can perform one-way (but not two-way) analysis of nominal data, such as the first zone entered in a test. To do this it uses the tests shown in table 1. If the result of this analysis is significant (by default, $p \leq 0.05$) and you've requested *post-hoc* analysis then the system will employ the tests shown in table 2.

Number of levels of independent variable	Number of levels of dependent variable	Independent samples?	Test
2	2	Yes	Fisher exact test
		No	Cochran Q*
	> 2	Yes	Contingency table
		No	Cochran Q
>2	2	Yes	Contingency table
		No	Cochran Q
	>2	Yes	Contingency table
		No	Not supported

Table 1. Tests used to perform one-way analysis of nominal data in ANY-maze.

**Strictly speaking this should be analysed using the McNemar test, but the Cochran Q test yields the same answer.*

<table>
<tr><th>Number of levels of dependent variable</th><th>Independent samples?</th><th>Ns are equal</th><th>Test</th></tr>
<tr><td rowspan="3">2</td><td>Yes</td><td></td><td>Not supported</td></tr>
<tr><td rowspan="2">No</td><td>Yes</td><td>Multi comparisons of data subjected to the Cochran Q test (Marascuilo & McSweeney)</td></tr>
<tr><td>No</td><td>Not supported</td></tr>
<tr><td rowspan="2">> 2</td><td>Yes</td><td></td><td>Not supported</td></tr>
<tr><td>No</td><td></td><td>Not supported</td></tr>
</table>

Table 2. Tests used to perform post-hoc analysis of nominal data in ANY-maze.

Analysis of continuous data

ANY-maze can perform one-way parametric or non-parametric analysis of continuous data using the tests shown in table 3. For two-way analysis the tests used are shown in table 4. If the result of this analysis is significant (by default, $p \leq 0.05$) and you've requested *post-hoc* analysis then the system will employ the tests shown in table 5.

<table>
<tr><th>Parametric data?</th><th>Number of levels of independent variables</th><th>Independent samples?</th><th>Test</th></tr>
<tr><td rowspan="4">Parametric</td><td rowspan="2">2</td><td>Yes</td><td>t-test</td></tr>
<tr><td>No</td><td>Paired sample t-test</td></tr>
<tr><td rowspan="2">>2</td><td>Yes</td><td>Between subjects ANOVA</td></tr>
<tr><td>No</td><td>Within subjects ANOVA</td></tr>
<tr><td rowspan="4">Non-parametric</td><td rowspan="2">2</td><td>Yes</td><td>Mann Whitney U</td></tr>
<tr><td>No</td><td>Wilcoxon</td></tr>
<tr><td rowspan="2">>2</td><td>Yes</td><td>Kruskal Wallis</td></tr>
<tr><td>No</td><td>Friedman</td></tr>
</table>

Table 3. One-way parametric and non-parametric tests used to analyse continuous data in ANY-maze.

<table>
<tr><th>Parametric</th><th>Independent samples?</th><th>Test</th></tr>
<tr><td rowspan="3">Parametric</td><td>Both IVs independent</td><td>2-way between subjects ANOVA</td></tr>
<tr><td>One IV independent</td><td>2-way between/within ANOVA</td></tr>
<tr><td>Neither IV independent</td><td>2-way within subjects ANOVA</td></tr>
<tr><td rowspan="2">Non-parametric</td><td>Both IVs independent</td><td>2-way Kruskal-Wallis</td></tr>
<tr><td>Either IV not independent</td><td>Not supported</td></tr>
</table>

Table 4. Two-way parametric and non-parametric tests used to analyse continuous data in ANY-maze.

<table>
<tr><th>Parametric</th><th>Comparison to control</th><th>Independent samples?</th><th>Ns are equal</th><th>Test</th></tr>
<tr><td rowspan="2">Parametric</td><td>Yes</td><td></td><td></td><td>Dunnett</td></tr>
<tr><td>No</td><td></td><td></td><td>Bonferroni,
Duncan,
Fisher LSD,
Scheffe,
Sidak,
SNK,
Tukey</td></tr>
<tr><td rowspan="8">Non-parametric</td><td rowspan="4">Yes</td><td rowspan="2">Yes</td><td>Yes</td><td>Not supported</td></tr>
<tr><td>No</td><td>Not supported</td></tr>
<tr><td rowspan="2">No</td><td>Yes</td><td>Not supported</td></tr>
<tr><td>No</td><td>Not supported</td></tr>
<tr><td rowspan="4">No</td><td rowspan="2">Yes</td><td>Yes</td><td>Non-parametric analog of the SNK test</td></tr>
<tr><td>No</td><td>Dunn</td></tr>
<tr><td rowspan="2">No</td><td>Yes</td><td>Non-parametric analog of the Tukey test for repeated measures</td></tr>
<tr><td>No</td><td>Not supported</td></tr>
</table>

Table 5. Post-hoc parametric and non-parametric tests used to analyse continuous data in ANY-maze.

Details as to whether a measure is nominal or continuous can be found under the measure's notes in the following topics:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Why tests are excluded from statistical analysis

Introduction

When performing analysis of nested data or repeated measures, ANY-maze will sometimes remove animals from the analysis.

In the case of nested data, all animals in the analysis must have the same number of replicates and if this isn't the case then ANY-maze will remove animals from the analysis to try to make it so.

In the case of repeated measures, ANY-maze requires that each animal included in the analysis has data at each of the levels of the repeated measure. Where this isn't the case, the system will exclude tests in an attempt to make it so.

Details

Nested ANOVA

In ANY-maze a nested ANOVA is used to analyse data in which each animal has more than one datum at the same level of a measure (and no data at the other levels). For example, imagine you have three treatment groups with 6 animals in each group and you tested the animals in 4 repeated trials. Now you want to look at the differences between the treatment groups (you're not interested in a possible effect of trial). So you will have 6 x 4 = 24 results for each treatment. But these 24 results are not from 24 animals, they are from 6 animals and each animal has 4 'replicates'. In this situation a nested ANOVA will be used.

Note that if the animals also have data at multiple levels of the measure then a *repeated measures* analysis will be used, rather than a nested ANOVA.

When performing a nested ANOVA, ANY-maze requires that all the animals have the same number of replicates. In the above example this was the case, but imagine that for some reason the 2nd animal in the first treatment group was only tested 3 times (perhaps it fell ill before it could be tested in its fourth trial). You would then have 4 replicates for all the animals except 1, which would have 3. To address this ANY-maze would remove the animal with 3 replicates from the analysis.

What ANY-maze actually does is to find the most common number of replicates (i.e. the number of replicates which the largest number of animals has) and it then removes all the animals with a different number of replicates. As an example, we might have animal A1 with 2 replicates (2R), animal A2 with 3R, A3 with 3R and A4 with 1R. In this case we have 1 animal with 1 replicate, 2 animals with 2 replicates and 1 animal with 3 replicates, so 2 replicates is the most common. Therefore animals A1 and A4 will be removed, leaving just animals A2 and A3 - we then have an equal number of replicates for all animals. Of course in a real experiment you are unlikely to have such 'uneven' data.

Repeated measures

So called *Repeated measures* analysis (also sometimes called *within subject* analysis) is used when each subject in an experiment (i.e. each animal) is tested repeatedly in different situations. The number of *levels* of the measure is simply the number of different situations. For example, imagine you tested some animals in a water-maze on three consecutive days - then 'Day' would be a repeated measure and the number of levels would be 3 (you tested them on three days).

In this type of analysis it's generally necessary to ensure you don't have any 'missing data', that's to say animals for whom you don't have a result for a particular situation. For example, in the above water-maze experiment you might have results for 'Distance travelled' of:

	Day 1	*Day 2*	*Day 3*
Animal 1	12.3	14.5	10.6
Animal 2	9.2	11.8	11.0
Animal 3	11.3	12.9	9.9
Animal 4	14.1	14.0	-

In this case animals 1, 2 and 3 have data for all the days but animal 4 is missing data for day 3.

Generally speaking there are two solutions to this problem:

1. Perform some type of statistical analysis that's 'unaffected' by the missing data. Strictly speaking this isn't possible as any analysis will be affected by the missing data, but there are at least some techniques (for example, pair-wise removal of missing data) which are still possible to calculate. However, it's important that you understand how the missing data might affect the results of the analysis (which can be non-trivial) and for this reason ANY-maze doesn't adopt this type of solution.
2. Remove the animal(s) with the missing data from the analysis. This is the so-called *case-wise* removal of missing data and it's the better solution as the resulting analysis is still accurate, although it does of course reduce the group Ns. Indeed, in extreme cases it can reduce the Ns for some (or all) groups to zero. Nevertheless, as your intention will assumedly be to always have the minimum amount of missing data this is the solution adopted by ANY-maze.

Specifically, what ANY-maze does is to check that all the animals have data for all the levels of any repeated measure in the analysis. If there's any missing data, then the animal is removed from the analysis and a note to this effect is added to the end of the stats report.

In fact in some situations ANY-maze may only remove some of the tests for an animal and leave others - this occurs when an animal has multiple tests within a particular level of an independent variable.

For example, imagine an experiment in which you tested four animals three times on days 1, 2 and 3 and four times on day 4. In this case if you analysed the data between the different days you would have something like this (where Ax = animal x and Tx = trial x).

Day 1	A1T1, A1T2, A1T3, A2T1, A2T2, A2T3, A3T1, A3T2, A3T3, A4T1, A4T2, A4T3
Day 2	A1T1, A1T2, A1T3, A2T1, A2T2, A2T3, A3T1, A3T2, A3T3, A4T1, A4T2, A4T3
Day 3	A1T1, A1T2, A1T3, A2T1, A2T2, A2T3, A3T1, A3T2, A3T3, A4T1, A4T2, A4T3
Day 4	A1T1, A1T2, A1T3, A1T4, A2T1, A2T2, A2T3, A2T4, A3T1, A3T2, A3T3, A3T4, A4T1, A4T2, A4T3, A4T4

So we have the same N for days 1, 2 and 3 but for Day 4 we have an extra trial for each animal (highlighted in red). In this case ANY-maze will 'match' the trials for the animals on the different days and then remove any un-matched trials. So, in the example it will remove the fourth trial for each animal on Day 4 because this trial has no matching trial on the other days.

See also:

- How ANY-maze processes missing data and empty cells in two-way ANOVAs

How ANY-maze processes missing data and empty cells in two-way ANOVAs

Introduction

Ideally, when performing a two-way ANOVA, each cell will contain the same number of observations, i.e. the 'N' for each level of the independent variables will be the same. However, departures from this ideal are common (perhaps because you had to remove an animal from the experiment) and unequal cell sizes, or entirely empty cells can often occur - see figures 1 and 2.

Group (note 1)	*N*	*Mean*	*Data*
Drug			
Male	3	0.3037	0.348, 0.269, 0.294
Female	4	0.2768	0.290, 0.273, 0.240, 0.304
Saline			
Male	4	0.3233	0.375, 0.296, 0.381, 0.241
Female	4	0.3055	0.222, 0.250, 0.417, 0.333

Figure 1. Example of an experiment in which there are unequal cell sizes.

Group (note 1)	*N*	*Mean*	*Data*
Drug			
Male	4	0.3037	0.304, 0.348, 0.269, 0.294
Female	0		
Saline			
Male	4	0.3233	0.375, 0.296, 0.381, 0.241
Female	4	0.3055	0.222, 0.250, 0.417, 0.333

Figure 2. Example of an experiment in which there is an empty cell.

How ANY-maze manages unequal cell sizes

ANY-maze uses the general linear model approach to manage unequal cell sizes. This approach, which uses the marginal sums of squares (also called the Type III or adjusted sums of squares), is commonly used by almost all statistics software.

How ANY-maze manages empty cells

Entirely empty cells are a more complicated problem. In cases where there is just one empty cell, or where the non-empty cells are 'connected' it is theoretically possible to perform a two-way ANOVA, but you must assume that there is no interaction. As this seems rather a gross assumption, ANY-maze doesn't take this approach - instead it reports that it can't perform the analysis and suggests that you perform separate one-way ANOVAs for the two independent variables. Although this is a more conservative approach, it doesn't require you to make any assumptions about the nature of the data or experimental design.

Viewing results as track plots or heat maps

For general information about the Results page, refer to the Results page *topic.*

Introduction

Track plot reports either show small plots of the tracks of the animals in their tests, or show the areas of the apparatus in which the animals spent the most time (heat maps, or *occupancy plots*). The plots can be shown in groups, which helps to highlight their differences - see figures 1 and 2.

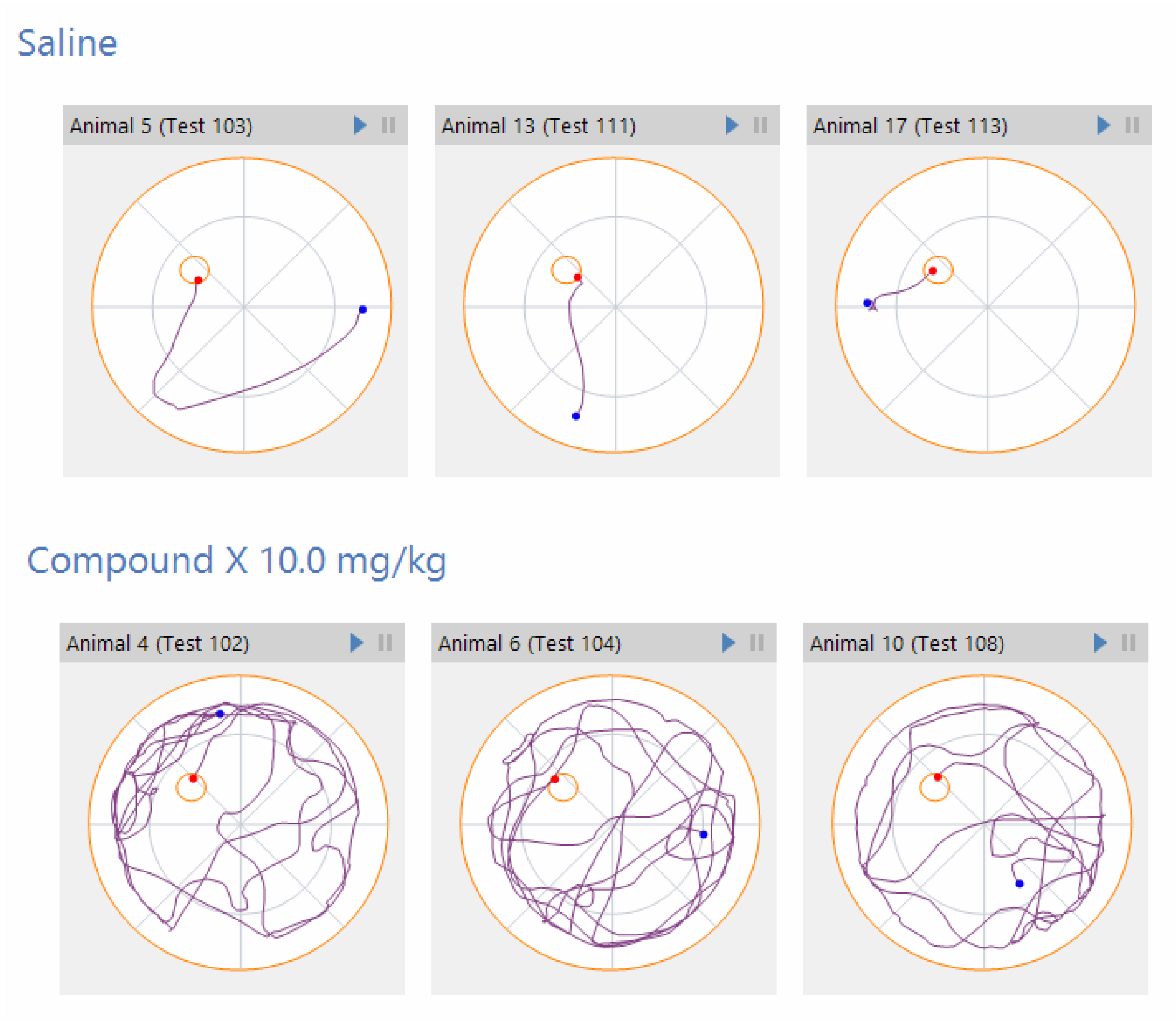


Figure 1. An example of a track plot report showing plots for two different groups of animals in a water-maze test.

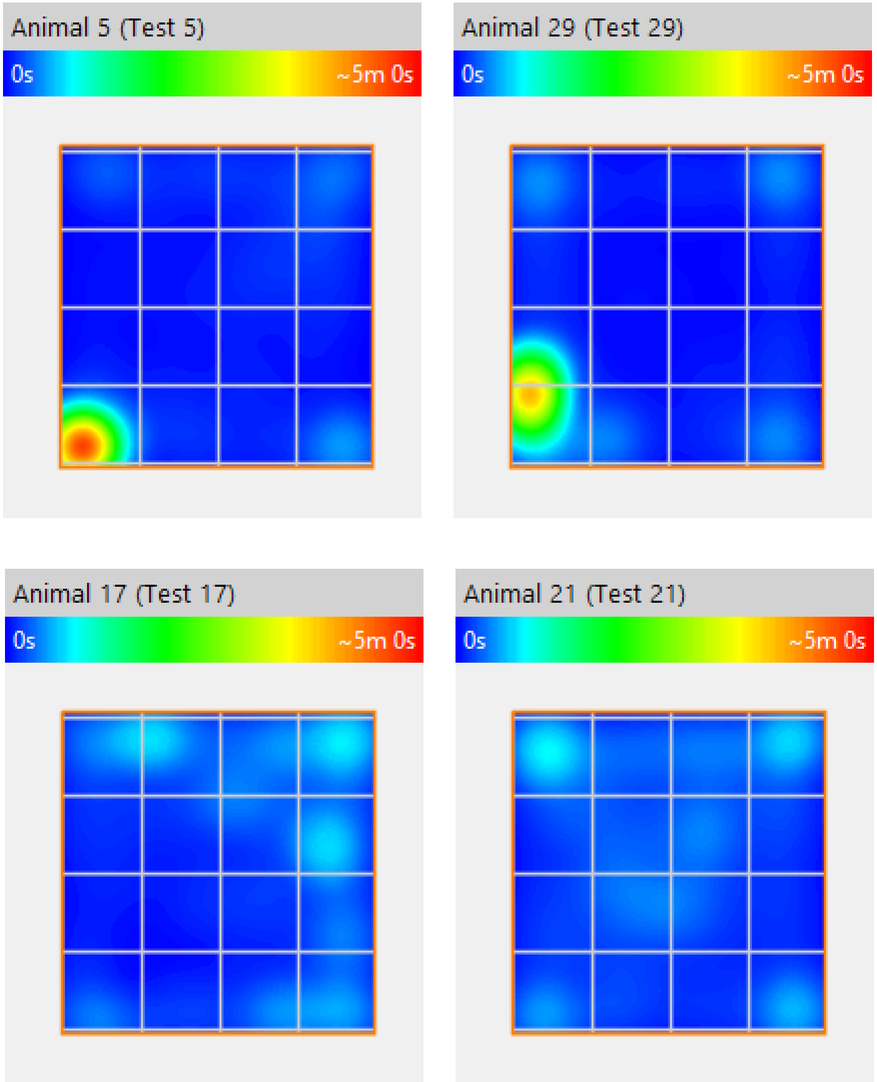


Figure 2. An example of a heat map report showing the results for two groups of animals in an open field test.

To create a track plot report you should:

1. Switch to the Results page.
2. Select the *Track plot* button in the *Report style* section of the ribbon bar.
3. If the report settings are not already displayed, click the *Edit report settings* button in the *Actions* section of the ribbon bar.
4. On the *Track plot report settings* page you can:

- Specify the type of report to create
- Specify how you want the track plots to be grouped
- Limit the tests included in the report

5. Once you've specified the settings you want to use, you should click the *View the report* button in the *Actions* section of the ribbon bar.

Specifying the type of track plot report to create

In brief

ANY-maze can show three types of plots - track plots, heat maps and grouped heat maps. For each type, the system can show the plots for the animal's centre point or, when head tracking is switched on, for the position of the animal's head.

Details

You should use the drop-down list labelled *What should the plots show* to specify the type of plot you want. The options are:

- Track of the animal's centre point
- Track of the animal's head
- Heat map of the animal's centre point
- Heat map of the animal's head
- Mean heat map of the group's centre point
- Mean heat map of the group's head

The 'track plot' reports simply show the track the animal followed during the test (there's one plot shown for each test). The individual heat maps are similar, except they show where the animal spent the most time during the test; but the *grouped* heat maps show one plot for each group, with all the test data amalgamated to create a single plot. These grouped plots are best for comparing the differences between groups.

What a heat map shows

A heat map shows a graphical representation of the amount of time the animal spent in different parts of the apparatus. The plot uses a range of colours to indicate time spent in an area, with blue as the shortest time and red as the longest - the colour range is shown in figure 1.

Figure 1. A heat map uses a range of colours to show how long the animal spent in different parts of the apparatus. The blue end of this range indicates less time, the red end indicates more time.

At first glance, it might seem that there's little real difference between what a heat map shows and what a track plot shows, but consider the following example. You have two tests; in the first test the animal moved slowly from the left side of the apparatus to the right; while in the second test the animal sat on the left side for 90% of the test and then ran quickly to the right. The track plot of these two tests would be identical - a track from the left to the right of the apparatus - but the heat maps would be very different. The first one would show a 'track' of more or less the same colour (probably green) across the apparatus, while the second test would show a red area on the left (where the animal sat for a long time) and then a pale blue track across the apparatus.

Another situation in which heat maps are very useful is in long-term tests, or tests where the animal moves around the apparatus a lot. In these cases, the track plot will typically become 'saturated' - like in figure 2.

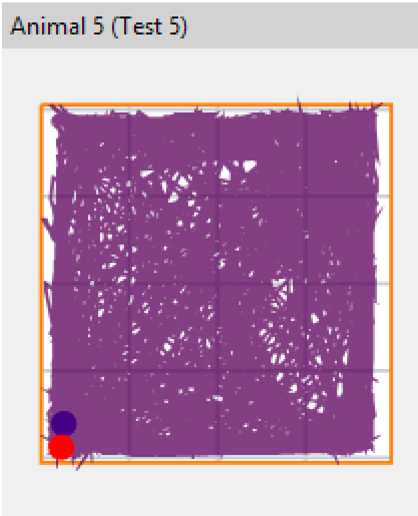


Figure 2. Example of a 'saturated' track plot. This is very likely to happen for longer tests, for example when you're tracking the animal in a home cage for 24 hours or more.

The heat map, however, never saturates and allows you to see where the animal spent most of the time in the test - see figure 3.

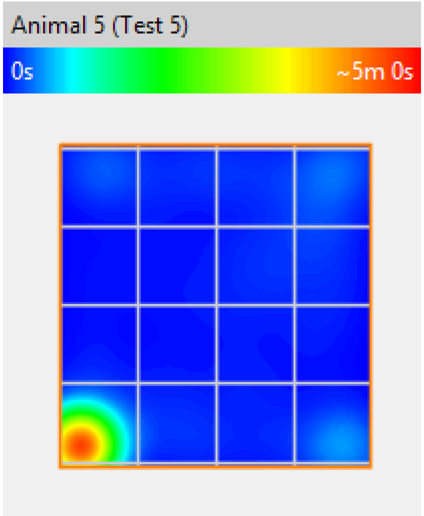


Figure 3. A heat map for the same test as is shown in figure 2.

Comparing heat maps

It might seem logical to assume that a heat map will show the point at which the animal spent the longest time in a test in red and the areas in which it spent no time in blue, but this has a disadvantage in that plots of two tests can't be compared.

Consider the following situation: You perform two 5 minute tests; In the first test the animal spent 30 seconds in the corner of the apparatus and wandered around throughout the apparatus for the rest of the test. In the second test the animal spent 4 minutes in the corner of the apparatus and then walked across the apparatus and spent the last 30 seconds in the opposite corner.

If the plots were generated such that the point at which the animal spent the longest time *in each test* (the so called 'local maximum') was set as the 'red' point, then in the first test, this would be the place where the animal spent 30 seconds, while in the second test it would be the place where the animal spent 4 minutes. Comparing these plots would then give the erroneous impression that the animals spent the same amount of time in this position in the two tests.

To address this, ANY-maze calculates the maximum across all tests in the experiment (both for the position of the head and the centre) and uses this maximum as the 'red' value in all heat maps (the so called 'global maximum'). Thus, in the example, the global maximum would be the point at which the animal spent 4 minutes (in the second test). This would mean that the points at which the animals spent 30 seconds in the two tests (i.e. where the animal was at the start of the first test and where it was at the end of the second one) would both be plotted in the same colour, making comparisons valid.

One repercussion of this is that the global maximum may change as you perform more tests, and this can cause the heat map of an already-performed test to change as well. Of course the plot itself doesn't change, but the colours used to represent the times do. For this reason it's best to only *compare* heat maps when you've completed all the tests in an experiment.

A final consideration regarding heat map comparisons relates to 'grouped' heat maps, which show the positions for groups of tests rather than an individual test. The important point is that group plots will have maxima which are greater than the global maximum of the individual plots. To address this, for these plots ANY-maze uses the maximum for all the plots included *in a single report*. This means that the plots in the report can be compared *to each other*, but it means that they can't be compared to individual test heat maps, or to other grouped plots.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Specifying how plots in a track plot report should be grouped

In brief

Track plot and heat map reports show a small plot of the animal's track or occupancy in a test. You can choose to group the plots by particular measures to make it easier to compare them.

Details

You should use the drop-down lists labelled *Group plots by* to specify how the track plots or heat maps will be grouped in the report.

You can group plots in up to three levels, for example by treatment, then by stage then by trial - you have to select a primary grouping measure, but the others are optional. Within the groups, ANY-maze will sort the plots by test number.

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults.

Limiting the tests included in a track plot report

In brief

By default, a track plot report will include all the tests performed in an experiment. However, you will often want to reduce this, as such a report will tend to be rather large. For example, you might want to look at plots for just a couple of treatments, or perhaps for just some specific trials.

To actually choose which tests are included you should use the list shown at the bottom of the *Track plot report settings* page - see figure 1.

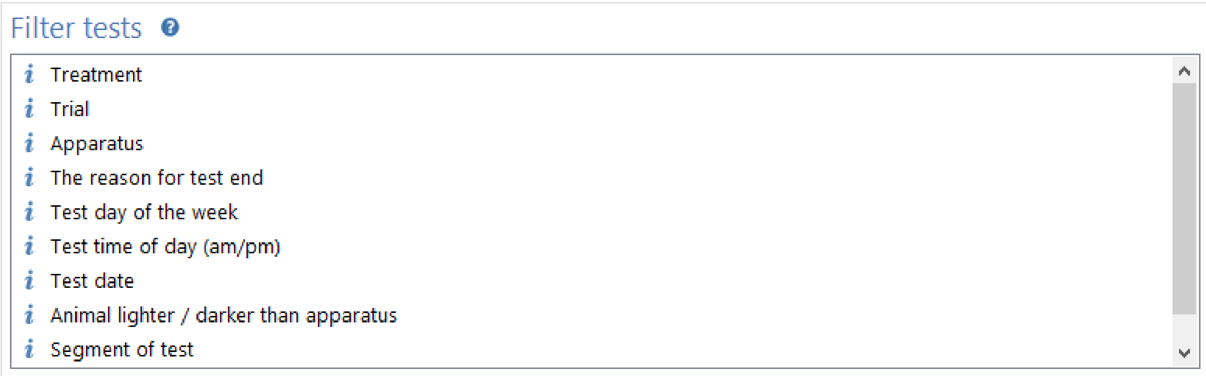


Figure 1. The list of measures used to limit the tests included in a track plot report.

Details

The list includes various measures - clicking a measure's name expands the list to show the individual values of the measure. For example, one of the measures listed is *Treatment* - clicking it will show a list of the experiment's treatments - see figure 2.

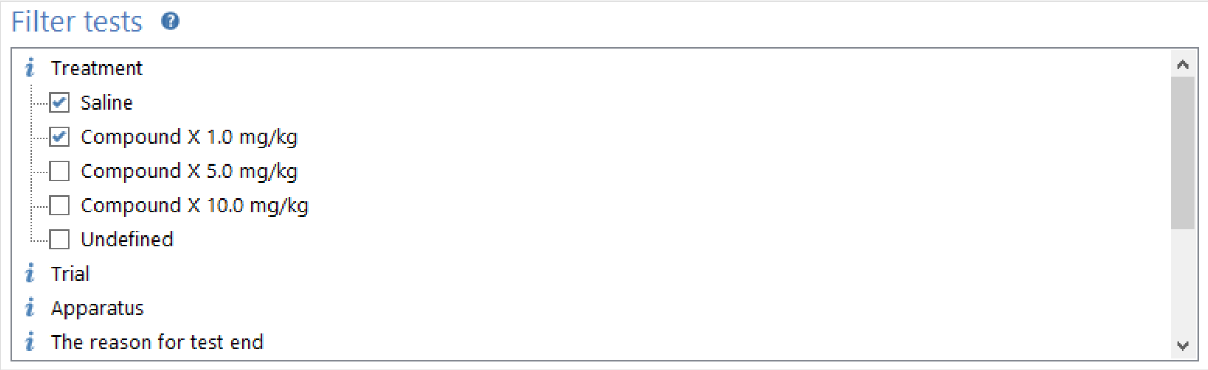


Figure 2. In this case the report will just include those tests performed on animals which received the treatments 'Saline' or 'Compound X 1.0 mg/kg'.

To limit the tests, you should select items you want included - for example, in figure 2 the report will include all the tests performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg'.

You can choose to limit the tests based on more than one measure - for example you could also open the *Trial* measure and select just the *Re-test* trial. In this case only the tests which match the selected treatments AND the selected trials will be included. So, in our example, just the *Re-test* trials performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg' would be included in the report.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Note that if you don't select anything at all in this list, then *all* tests will be included.

Selecting trials

Limiting the tests to just those which match certain trials can be a little less obvious than it appears - this is best explained by an example.

Imagine you have a water-maze experiment in which you train the animals to find an island, then treat them and then test them again to see whether the treatment has affected their ability to find the island. In this experiment you would probably be most interested in the results of the post-treatment tests and perhaps the last training tests (which you could use for comparison). However, the earlier training tests would be of little actual interest. So you would want to compare the *Last training trial* and *treated* trial. But what is the *Last training trial?* For some animals it might have been trial 3, for others trial 4 etc. depending on how quickly they learnt the task. For this reason, when listing the *Trials* in a stage, ANY-maze includes not just the individual trials (trial 1, trial 2, etc.) but also *Last trial* which will match the last trial for each animal, irrespective of which trial it happened to be. (For obvious reasons the *Last trial* isn't listed for stages which consist of just a single trial).

Excluding retired animals

As you may know, ANY-maze allows you to mark an animal as Retired. This means that the results of any tests the animal has already had are maintained, but the animal doesn't receive any further tests.

Retiring animals is common in experiments which include some type of training - animals which fail to achieve the training goal are retired.

Ordinarily, tests performed on retired animals are still included in reports, but you can choose to exclude them by checking the box shown at the bottom of the settings page titled *Don't include results for retired animals in the report.*

The *Clear report settings* button in the ribbon bar can be used to reset all the report settings back to their defaults, which includes removing all selections from the list of filters.

Heat map format

Introduction

If you right click on a heat map, a menu will appear. One of the options shown is *Format plot...*, and selecting it will open the *Heat map format* window, shown in figure 1.

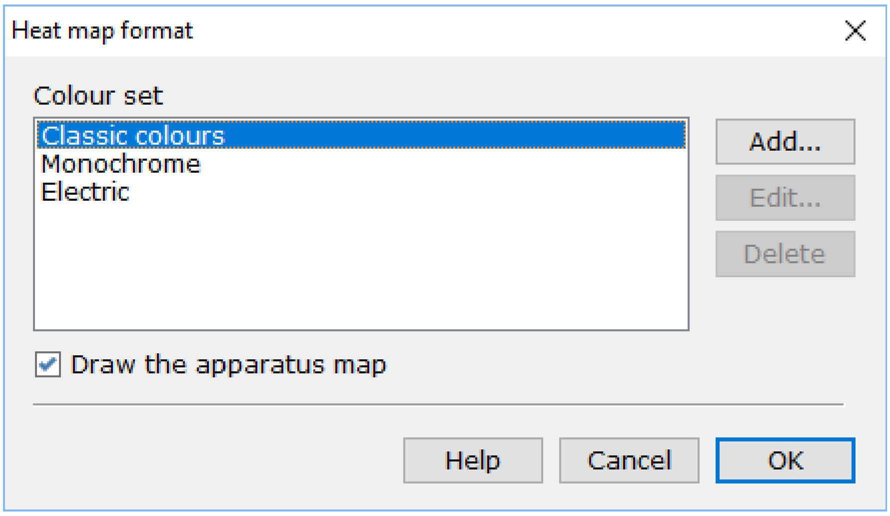


Figure 1. The Heat map format window.

Colour sets

You can change the colours used to display the heat map by selecting an alternative *Colour set*. By default there is only one colour set defined, the 'Classic colours', but you can define more using the *Add*, *Edit* and *Delete* buttons.

Add Adds a new colour set. The Heat map colour set window will open where you will be able to give the new colour set a name and specify the colours that it uses.

Edit Edits the colour set that is currently selected in the list. The Heat map colour set window will open, showing the current settings for the colour set, you can edit any of them. Note that you can't edit the 'Classic colours' colour set.

Delete Deletes the colour set that is currently selected in the list. Note that you can't delete the 'Classic colours' colour set.

Any colour sets that you create are stored on a per-user basis on your computer and *not* in the experiment file. This means that:

- You can easily apply a colour set to any of your experiments.
- Colour sets you define won't affect other users of ANY-maze.

Showing and hiding the apparatus map

By default heat maps have the apparatus map drawn over them, but if you wish, you can remove it simply by unchecking the option to *Draw the apparatus map*.

The Heat map colour set window

Introduction

The heat map colour set window, shown in figure 1, is used to define the colours in which a heat map is drawn. In essence you just need to specify two colours, the 'coldest' colour and the 'hottest' colour. However, there are a number of things you need to take into consideration when choosing the colours to use.

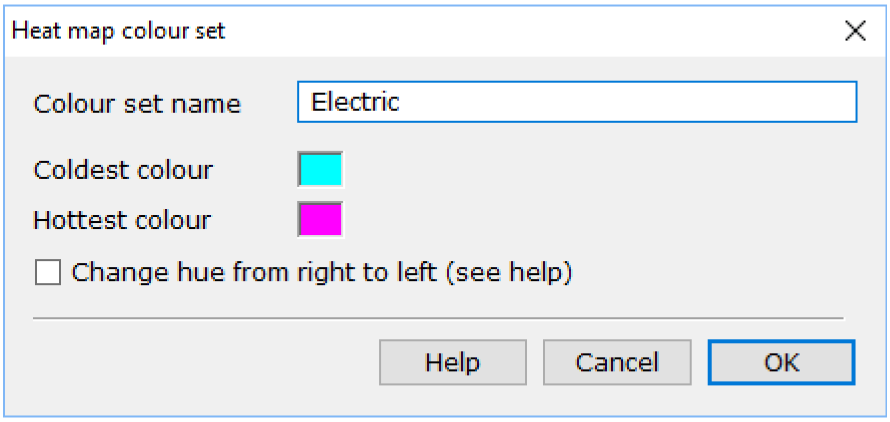


Figure 1. The Heat map colour set window.

Details

Colour set name	You must give a colour set a name, but it can be anything you like up to a maximum of 64 characters.
Coldest colour	Click the colour box to open the standard *Colour picker* window (shown in figure 2, below). Here you should select the colour to use for the coldest areas of the heat map - i.e. the parts where the animal spent no time. There are some things to consider when choosing a colour, which are explained below.
Hottest colour	Click the colour box to open the standard *Colour picker* window. Here you should select the colour to use for the hottest areas of the heat map - i.e. the parts where the animal spent the most time. There are some things to consider when choosing a colour, which are explained below.
Change hue...	This option affects how ANY-maze uses the colours you choose; its action is

explained below.

Choosing the colours, and how they're used

In essence, all you are doing when you define a colour set is to choose the two colours at either end of the heat map scale - ANY-maze then draws the heat map using the colours between them. However, this implies that there is some set sequence that can be used to move from one colour to another.

If, rather than choosing two colours, we were choosing two numbers, then it would be fairly obvious that if the 'coldest' number was 4 and the hottest number was 22, then we'd have a heat map scale that would go something like, 4, 5, 6 ... 20, 21, 22. So what we want to do is create a similar sequence for colours, starting from our coldest colour (say, dark orange) and going up to our hottest colour (say, light green) - but what colours come between them? Well, the rainbow provides a well-known ordered sequence for colours that goes red, orange, yellow, green, blue, indigo, violet. So we could say that in our example (moving from dark orange to light green) the colours would go: orange, yellow, green. And if we start with *dark* orange and end with *light* green, we could also say that we'll start with a dark colour and get progressively lighter. And in essence that's what ANY-maze does.

In the world of computers, colours are often defined as a mix of the primary colours Red, Green and Blue (usually referred to as RGB). But there's another, less well known way to define colours, which is based on their Hue, Saturation and Lightness (or HSL). Here, Hue is the colour itself (such as orange or green), Saturation is how much of the colour we have (so no saturation is grey and full saturation is very orange or very green) and Lightness (also sometimes called luminance) is how bright the colour is (so low lightness is a dark orange and high lightness is a pale orange). In fact, you'll see a colour's HSL values in the colour picker window which ANY-maze uses - see figure 2, below.

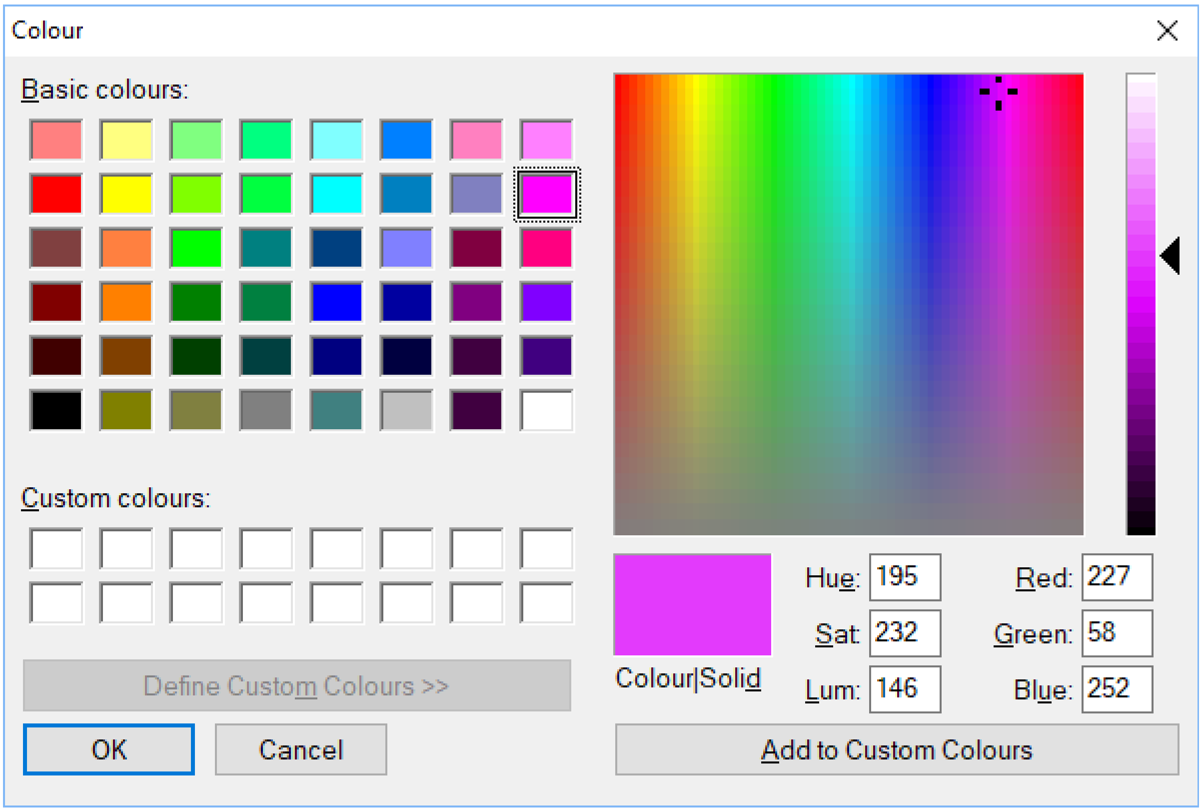


Figure 2. The colour picker window. The values for a colour's hue, saturation and lightness (or luminance) are shown just below the colour square.

When you use the colour picker, you are actually using HSL without really knowing it. The colourful square shows different Hues going from side to side, and different saturations going from bottom to top. And the rectangle to the right of the square has different lightness going from bottom to top.

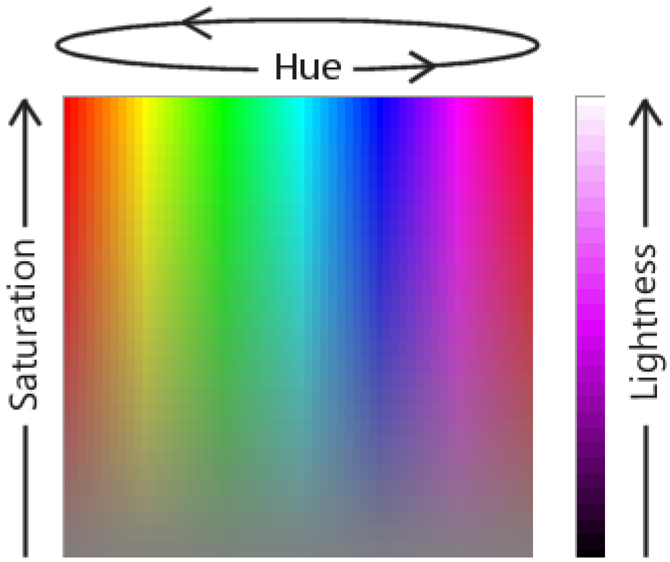


Figure 3. Using the colour picker window, you define a colour by setting its Hue, Saturation and Lightness.

Getting back to our heat map colours, we have a cold colour (dark orange in my example) and a hot colour (light green) and the challenge is to decide what colours come in between. You're probably starting to see the answer. If we break the colours down into their HSL values then we can move in sequence from the cold colour's hue to the hot colour's hue (effectively moving across the colourful square in the colour picker window). In the same way, we can also move from the cold colour's saturation value to the hot colour's saturation value, and we can do the same for lightness; so now we have a way to define what other colours come between the two colours you choose.

What ANY-maze actually does is to create 250 colours between the cold and hot colours you select. For each colour, it works out new hue, saturation and lightness values, being the values between the cold and the hot colours. For example, if the cold colour had a hue of 20 and the hot colour had a hue of 80 then the 125th colour (i.e. the one in the middle) would have a hue of 50 (half way between 20 and 80). Similarly if the cold colour had a lightness value of 80 and the hot colour had a lightness value of 100 then the 125th colour would have a lightness value of 90, and let's say that the cold and hot colours have the same saturation value of 200, so the 125th colour will also have saturation of 200. So the 125th colour would be H=50, S=200, L=90 which is a slightly muddy green!

Hopefully this all seems reasonably logical and you're now understanding how ANY-maze creates a sequence of colours for the heat map using the cold and hot colours that you choose. However, there are a couple of additional things you need to be aware of:

1. Although you can select any colours for hot and cold, it's usually best to choose

colours which have at least part of their HSL value the same. For example, you might fix Hue and Saturation and just choose two colours with different Lightness values - this might create a heat map where cold is, for example, a dark red and hot is a very pale red.

2. In fact, you'll usually want to change the hue as that's what makes the contrast in the heat map standout most clearly. The sequence of hue is not as obvious as saturation or lightness, as hue moves through the colours of the rainbow, whereas saturation and lightness simply start low and become greater. In fact, hue has another difference too: it cycles round, so the first hue value (red) is the same as the last one - think of the colours being wrapped around a cylinder; as you turn the cylinder, you get to see different colours, but there's no start or end - they just progress smoothly through all the colours. This means that you could, for example, have a start hue that is blue (which has a value of 160) an end hue that is yellow (which has a value of 40) and ANY-maze would work through the hues from 160 to 240 (the maximum value) and then restart at 0 and continue up to 40. So the colours would go from blue though purple to red then orange and finally yellow. *But,* you could go the other way through the hues (like turning our cylinder of colours in the opposite direction), in which case we'd move from blue to cyan to green to yellow. This is the option that's provided by the check box labelled *Change hue from right to left,* where the term 'right-to-left' comes from the sequence that the hues are shown in the colourful square in the colour picker window.

Working with the results report

For general information about the Results page, refer to the Results page *topic.*

Introduction

Full details about the various things you can do with ANY-maze results reports can be found in the following topics:

- Altering what's being analysed in the report
- Switching report styles
- Pinning a results report so you can come back to it later
- Printing, saving, copying and e-mailing the results report
- Changing the report's colours

ANY-maze help topic H0776

Altering what's being analysed in the report

In brief

To alter what's actually being analysed in a report, simply click the *Edit report settings* shown in the *Actions* section of the ribbon bar. This will take you to the 'settings' page for the current report style, where you can alter all aspects of the report.

Details

You'll find full details about the various settings for each of the report styles in the following topics:

- Text report settings
- Graph report settings
- Statistical analysis report settings
- Track plot report settings

Switching report styles

In brief

It's very easy to alter the style of a results report - just click the style you want in the *Report style* section of the ribbon bar.

Details

ANY-maze can show results in four different *styles*.

- Text report
- Graph report
- Statistical analysis report
- Track plot or heat map report

You can freely switch between styles so, for example, if you're viewing statistical analysis and you'd like to see a graph of the same data then you just need to switch to the *Graph* report style.

When you switch from one style to another, the settings used to create the report are transferred to the new style. However some settings used in the new style may not be defined for the style you *were* viewing - for example, when switching from *Statistical* style to *Graph* style, the graph type (line, column or scatter graph) won't be defined. In these cases ANY-maze will use the same settings as it used when it last created a report of that style (or default data if you've never created a report of the relevant style). So, in the example, if the last *graph* you'd looked at was a column graph then when you switch from *Statistical* style to *Graph* style, you'll be shown a column graph. Of course, you could then just click the *Edit report settings* button in the *Actions* section of the ribbon bar to alter the settings for the new style if you need to.

Pinning a results report

In brief

You can *Pin* a results report by clicking the *Pin this report* button in the *Stored reports* section of the ribbon bar. The report will then be added to the list of *Stored reports* in this drop-down list, and you'll be able to return to it at any time simply by selecting it from this list.

Details

The results page has been designed to be very interactive, making it easy for you to perform different analysis, view results in different formats etc. However, when you alter the report settings, perhaps to analyse a different measure, the previous report disappears and to get it back again you need to restore the report's settings - this makes it difficult to work with a number of reports simultaneously.

To overcome this problem ANY-maze allows you to '*pin*' a report. Doing this makes a copy of the report and adds it to a 'list of pinned reports'. You can then view the report at any time by simply selecting its name from the list. There's no limit to the number of reports you can *pin*, so if you want to work with, say, six different reports at the same time you can simply pin them all.

To pin a report	Click the *Pin this report* button in the *Stored reports* section of the ribbon bar. The *Pin report* window will open where you can give the pinned report a name - this is just so you can identify it, so you can use anything you like. When you click OK, the report will be added to the drop-down list of reports in the *Stored reports* section of the ribbon bar.
To view a pinned report	Simply select the name of the report in the drop-down list of reports in the *Stored reports* section of the ribbon bar.
To delete a pinned report	If you want to remove a report from the drop-down list of pinned reports, first select it from the list, so it opens, and then click the *Delete this report* button in the *Stored reports* section of the ribbon bar.
To rename a pinned report	If you want to rename a report that's already pinned, first click it, so it opens, and then click the *Rename this pinned report* button in the *Stored reports* section of the ribbon bar.

By the way, pinned reports are part of the experiment - they'll still be there if you close the experiment and reopen it again later.

ANY-maze help topic H0779

Name pinned report

Enter the name for this pinned report. You can enter anything you like (up to a maximum of 64 characters) but you should try to use a name which describes the report's contents without being too long-winded.

This name will be shown in the list of pinned reports shown in the ribbon bar on the Results page, and will be accompanied by an icon which will indicate the report's style (text, graph etc.), so you don't need to mention the style in the name.

By the way it's easy to rename a pinned report, so don't worry too much about getting this name exactly right the first time!

Printing, saving, copying and e-mailing the results report

In brief

Results reports can be printed, copied, etc. much like other documents in ANY-maze - i.e. using the buttons on the ribbon bar.

If you want to copy or save just a single graph or track plot, then you should right click on it and select the appropriate option (for example, Save graph...*) from the menu which appears.*

Details

Full details about these operations are to be found in the following topics:

- Printing reports
- Saving reports and data
- Copying reports and data
- Sending reports and data by e-mail

Changing the colours used in results reports

The font used for reports can't be changed; it follows the standard ANY-maze font. However you can change the size of this font using the Appearance settings on the Options page.

You can alter some of the colours used for graphs using the *Appearance* settings on the Options page.

Specifically, you can alter the colours of:

- Graph column colours
- Graph line colours
- Graph scatter point colours

Note that report background and axis colours are all part of the currently-selected ANY-maze theme and can only be changed by changing the theme.

Viewing reports that are defined in the Protocol

In brief

As you may know already, it's possible to define reports as part of a protocol. This has the advantage that reports are immediately available in any experiment that's based on the protocol, making it extremely easy to view your results.

Any such reports are listed as *Stored reports*, and are available from a drop-down list in the ribbon bar at the top of the Results page. To view a report, just select its name from this list.

Details

There are two benefits to defining results reports in the protocol:

- Firstly, you only have to 'design' the report once, and thereafter you can just reuse the same design again and again;
- Secondly, reports designed in the protocol can include a mix of the ANY-maze report styles - something that the reports created directly on the Results page can't do.

For example, you could easily create a report which shows statistical analysis of a measure, followed by a graph of the same data. For full details on defining reports in the protocol, refer to: Setting up a results report.

Of course, the disadvantage is that these types of reports are less flexible than reports you create directly on the Results page. In general, therefore, you'll probably want to utilise a mix of the two - perhaps including a 'standard' report in the protocol (which includes the results you know you'll always want to see) and then using the Results page to analyse individual experiments in detail.

Publishing or e-mailing report sets

In brief

Report sets, which are fully described in An introduction to results, reports & data, allow you to group a set of reports together and publish them to a folder in HTML format so other people can access them using a web browser such as Internet Explorer.

You can also package an entire report set into a single file and send it by e-mail, which is a great way to share results with a co-worker as they only need access to Internet Explorer and not the ANY-maze software.

Details

If the current experiment's protocol includes any report sets, then they will be listed in the drop-down list of reports in the *Stored reports* section of the ANY-maze ribbon bar.

To publish a report set

1. Select the report set's name from the drop-down list of reports in the *Stored reports* section of the ANY-maze ribbon bar.
2. Click the link *Publish the report set* on the page which is displayed.

The report set for this experiment will be published to the folder defined in the protocol, and a link to it will be added to the overall report set index.

Note that you can update a published report set by simply repeating the above steps. This is useful if you've changed something in the experiment, perhaps adding some more animals or altering the protocol.

To e-mail a report set

1. Select the report set's name from the drop-down list of reports in the *Stored reports* section of the ANY-maze ribbon bar.
2. Click the link *E-mail the report set* on the page which is displayed.
3. The *Send this document by e-mail* window will open. Enter your name and e-mail address, and the recipient's e-mail address.
4. Type in a message to accompany the report set file.
5. Click the OK button. The report set will be saved to a file*, the file will be attached to

the message and the message will be sent.

**Saving report sets to a file can take a long time, especially if the report set includes lots of graphs and/or track plots.*

The Data page

Introduction

While the Results page provides powerful facilities for viewing and analysing an experiment's results, the format used is not very satisfactory if you want to transfer data to another program. For this type of operation, a table of results is really more appropriate and this is what's provided by the Data page.

Essentially, the Data page shows a spreadsheet containing all the data in an experiment. In general, each row contains the data relating to an individual test, and there are different columns for each measure. In fact, although the spreadsheet can include all the data in the experiment, you will usually limit it to just show certain measures and possibly just certain tests.

You'll find full details about the Data page in the following topics:

- Definitions of the data shown
- Choosing what data is shown
- Showing repeated measures in columns
- Sorting the data
- Selecting cells, columns or rows
- Copying and saving data
- Notes on transferring data to Excel, Access, SPSS, SigmaStat, Systat and Statistica

Tips for working with the Data page

- You can resize the columns of the spreadsheet by dragging the column divider in the header area. Double clicking a divider will set a column's width to 'best fit'.
- The font face and size used in the spreadsheet are those used throughout ANY-maze; you can change the size using the Appearance settings on the Options page.
- If you include columns for either *Test notes* or *Animal notes*, only the first 80 characters of the notes will be displayed.
- Where a value is undefined, it will be shown as '#N/A' (Not Available) in the spreadsheet. When you copy or save the data, #N/A cells will just appear blank.

See also:

- Saving an experiment in XML format

ANY-maze help topic H0791

Definitions of the measures shown on the Data page

The columns of the Data page spreadsheet show the values of ANY-maze measures. Full details about individual measures and how they're calculated can be found in the following topics:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures
- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

Choosing what data is shown on the Data page

Overview

The Data page spreadsheet can include one column for every measure which ANY-maze can analyse. However, as this can quite literally include *hundreds* of measures, you will usually want to limit the columns to just those which are of specific interest to you.

As well as limiting the columns which are included on the spreadsheet, you can also select which rows (i.e. tests) are shown. This is generally less useful than limiting the columns, but in cases where an experiment includes multiple stages and/or trials you might, for example, want to exclude training tests.

Details

Follow these links for full details on selecting the data shown

- Selecting the measures (columns) shown
- Showing repeated measures in columns
- Limiting the tests (rows) shown
- Breaking tests into parts using the 'Segment of test' measure

Selecting the measures (columns) shown on the Data page

In brief

To alter the measures (i.e. columns) which are included in the spreadsheet:

1. Click the *Select data* button in the *Actions* section of the ribbon bar - this will cause the 'Data settings' page to be shown.
2. At the top of this page is a large list containing all the experiment's measures - you should check the measures you want included.

Details

You can use the measures list to choose which measures you would like to include on the Data page. As the list can potentially contain hundreds of measures, ANY-maze shows them in groups.

To open a group, simply click it with the mouse - the list will expand to show the group's measures. To close an open group, just click its title.

To actually choose an individual measure to show, click on the measure's name - a small tick will be shown in the box next to it. To exclude a measure, click it again - the tick will go away.

If you right click in the measures list, a menu containing some or all of the following options will appear:

Details about this measure	Only available if you click on a specific measure. Displays a comprehensive help topic about the measure, including information about how it is calculated.
Select all	There are potentially three 'Select all' options (exactly how many you'll see depends on what you click on in the measure list). One option selects all of the measures in the entire list, the others select all the measures in the current measure group or subgroup. Although the option to select all measures might appear to provide an easy way to view all the of analysis (and it is), you should be aware that it will usually create a very large report.
Deselect all	There are potentially three 'Deselect all' options (exactly how many you'll see depends on what you click on in the measure list). One deselects all the measures in the entire list - which is useful if you just want to reset the list and start over selecting the measures to include - the others deselect all the measures in the current measure group or subgroup.

Expand all	There are potentially two 'Expand all' options: One opens all of the groups so you can see all the measures that are available, the other opens all the subgroups in the current measure group.
Collapse all	There are potentially two 'Collapse all' options: One closes all of the groups making the list shorter and easier to work with, the other closes all the subgroups in the current measure group.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Once you've selected the measures you'd like included in the spreadsheet, click the *View spreadsheet* button in the *Actions* section of the ribbon bar. The spreadsheet will reappear, showing only the columns you selected.

Note that the *Test* column, which shows a test's number, is always included as the first column of the spreadsheet - you can't remove it.

See also:

- Showing repeated measures in columns
- Limiting the tests (rows) shown
- Breaking tests into parts using the 'Segment of test' measure

Showing repeated measures in columns

In brief

Usually ANY-maze will show one test (or segment of a test) in each row of the Data page spreadsheet. This creates a nice logical separation between the rows (tests) and the columns (measures), but for some statistical software, such as SPSS and Statistica, this arrangement causes problems when the data includes *repeated measures* (i.e. multiple trials, or multiple test segments), as these programs need each instance of a repeated measure to be shown in a different *column*.

To address this, ANY-maze includes specific options to organise the data in this way.

Details

An ANY-maze experiment will include *repeated measures* if the experiment consists of multiple stages and/or multiple trials, or you choose to divide tests into different *segments* across time (see Breaking tests into parts using the 'Segment of test' measure).

In programs such as SPSS and Statistica, repeated measures are plotted in columns - so for example, if you have 4 animals and each one was tested in 3 trials and you wanted to analyse the Distance the animals travelled, SPSS would require the data to be organised as 4 rows (one per animal) with 3 columns, one for the distance travelled in the first trial, one for the distance travelled in the second trial and one for the distance travelled in the third trial. The Data page, however, will normally show this data as 12 rows, one per trial, with a single column for distance travelled.

To address this difference, ANY-maze includes two options which can be used to have repeated trials and/or test segments plotted in columns.

Setting repeated measures to be shown in columns

To have repeated measures shown in columns, you need to:

1. Click the  *Select data* button in the *Actions* section of the Data page's ribbon bar - this will cause the 'Data settings' page to be shown.
2. Below the list of measures (which is at the top of the page) are two check boxes - *Show repeated trials in different columns* and *Show test segments in different columns* - you should simply check the box(es) appropriate to your data.

Using these options can have some effects which you may not be expecting. First, choosing the option to *Show repeated trials in different columns* will cause the *Stage* and *Trial number* measures to be removed from the main measures list. This is because these measures are going to be automatically used to divide the data into different columns, so you can't choose whether or not to include them.

Similarly, if you choose the *Show test segments in different columns* option, then the *Segment of test* measure will be removed from the main measures list.

How the data is presented when repeated measures are shown in columns

If you select the option to *Show repeated trials in different columns,* then ANY-maze will organise the data with one *Animal* per row. It will then include a series of columns for each measure you have chosen to analyse, showing the measure's result for each trial. For example, if each animal was tested in 4 trials and you choose to analyse the *Distance travelled* and the *Maximum speed,* you would see four columns for *Distance travelled* and four for *Maximum speed.* These would be titled as *1/1 - Distance travelled*, *1/2 - Distance travelled*, etc. where '1/1' means Stage 1, Trial 1 and '1/2' means Stage 1, Trial 2, and so on.

In some experiments, the animals may not all have had the same number of trials - for example, they might have been trained to achieve some goal, with some animals requiring more trials to achieve the goal than others. In this case ANY-maze will show enough columns for the maximum number of trials any animal received. So, for example, we might have 3 animals, two of which had 3 trials and one of which had 5. In this case, ANY-maze will show 5 columns for each measure, and for the animals that didn't have trials 4 and 5 the data will be shown as '#N/A' (Not Available).

If you choose to *Show test segments in different columns* then the data will be formatted in a similar way - i.e. with one column for each segment, with the number of columns being determined by the longest test duration. In this case the columns are titled as *0-30; Distance travelled*, where '0-30' indicates the test segment from time 0 to time 30 seconds.

You can, of course, choose to show *both* repeated trials *and* test segments in different columns. In this case, the number of columns per measure will be the maximum number of trials multiplied by the maximum number of time segments. The columns will be titled as *1/1; 0-30; Distance travelled,* etc., meaning the distance travelled in the time from 0-30 seconds in the first trial of the first stage.

As you will appreciate, the number of columns shown when data is presented in this way can quickly become quite large, especially if both repeated trials and test segments are plotted in different columns. For example, if an experiment has 4 trials, each of which lasts 5 minutes, and you choose to divide the tests into 30-second segments then for *each* measure you include, there will be 40 columns.

ANY-maze doesn't have a limit to the *number* of columns that can be shown, but Windows has a limit to the overall width of the Data page spreadsheet. This limit means that very large numbers of columns cannot be displayed. If the data you select will exceed this limit, then ANY-maze will warn you and will only actually include the columns that fit - even if you asked for more.

A final point - not all measures relate to individual trials. For example, the animal *Treatments* are specific to animals and not to trials, and therefore these measures will only ever be shown in a single column, even if repeated measures are shown in separate columns.

See also:

- Selecting the measures (columns) shown
- Limiting the tests (rows) shown
- Breaking tests into parts using the 'Segment of test' measure
- Transferring data to SPSS
- Transferring data to Statistica

Limiting the tests (rows) shown

In brief

To alter which tests (i.e. rows) are included in the spreadsheet:

1. Click the *Select data* button in the *Actions* section of the ribbon bar - this will cause the 'Data settings' page to be shown.
2. At the bottom of this page is a list containing various measures such as *Treatment* and *Trial*. Clicking a measure's name expands the list to show the individual values of the measure.
3. Select entries for the tests you want included. For example, to just include tests performed in the 2nd stage of the experiment you'd select the *Stage* measure and then, in the list of stages shown, check just the 2nd stage.

Details

Each row of the Data page spreadsheet shows a single test (or a segment of a test), but in some situations you may not want all the experiment's tests to be shown. For example, you might want to exclude some 'training' tests, the results of which are of little interest to you.

At the bottom of the 'Data settings' is a list which shows various measures, and you can use this list to select which tests should be shown - see figure 1.

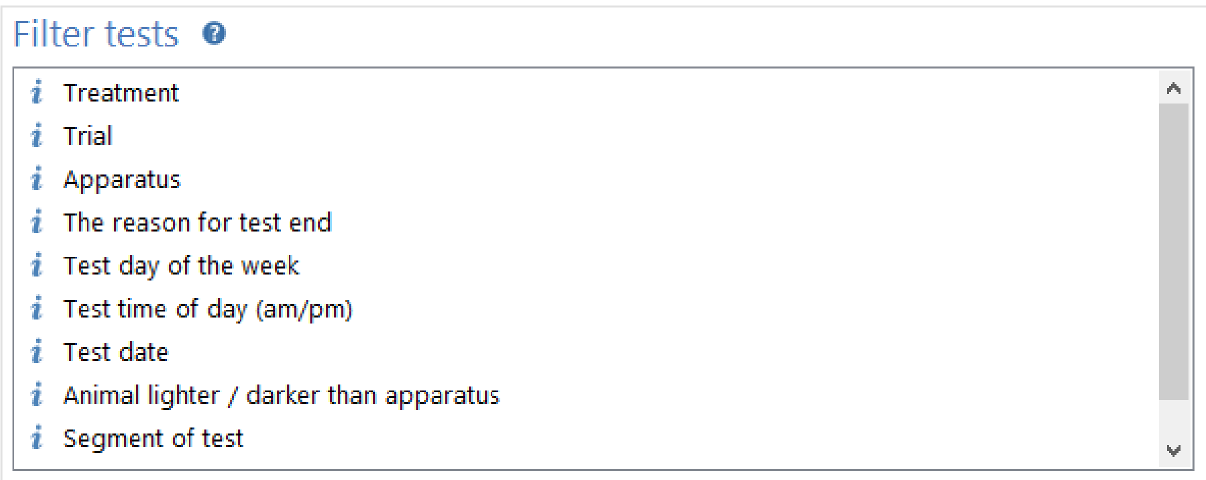


Figure 1. The Data page settings include a list of measures which you can use to

limit the tests shown on the Data page spreadsheet.

Clicking a measure's name expands the list to show the individual values of the measure. For example, one of the measures listed is *Treatment* - clicking it will show a list of the experiment's treatments - see figure 2.

Figure 2. In this case, the spreadsheet will just include those tests performed on animals which received the treatments 'Saline' or 'Compound X 1.0 mg/kg'.

To limit the tests, you should select items you want included - for example, in figure 2 the spreadsheet would include all the tests performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg.'

You can choose to limit the tests based on more than one measure - for example, you could also open the *Trial* measure and select just the 'Retest' trial. In this case, only the tests which match the selected treatments AND the selected trials will be included. So, in our example, just the 'Retest' trials performed on animals which received the treatments 'Saline' OR 'Compound X 1.0 mg/kg' would be included in the spreadsheet.

If you want to select a block of items *within the same group* then you can use a 'shift-click' as follows: Click the first item you want to select, then hold down the shift key on the keyboard and click the last item you want to select. All the items from the first to the last (inclusive) will change to the same state as the LAST item now has. This works whether you're selecting or deselecting items - all the items simply change to have the same state as the LAST item you click.

Selecting trials

In fact, limiting the tests to just those which match certain trials can be a little less obvious than it appears. Imagine you have a 'Training' stage, in which animals have repeated trials until they achieve some training goal, or have had a maximum number of trials - perhaps 6.

In this 'Training' stage, some animals might have 4 trials, some might have 5 etc. but what would probably interest you would be an animal's performance in its last training trial, irrespective of which trial this happened to be. For example, you might want to compare its performance in the last training trial to its performance in a trial performed after some type of treatment.

For this reason, when listing the *Trials* in a stage, ANY-maze includes not just the individual trials (trial 1, trial 2, etc.) but also *Last trial* which will match the last trial for each animal, irrespective of which trial it happened to be. (For obvious reasons, the *Last trial* isn't listed for stages which include just a single trial).

Excluding retired animals

As you may know, ANY-maze allows you to mark an animal as Retired. This means that the results of any tests the animal has already had are maintained, but the animal doesn't receive any further tests.

Retiring animals is common in experiments which include some type of training - animals which fail to achieve the training goal are retired.

Ordinarily, tests performed on retired animals are still included in the Data page spreadsheet, but you can choose to exclude them by checking the box shown at the bottom of the Data page settings titled *Don't include retired animals in the spreadsheet.*

Clearing the list

If you want the spreadsheet to go back to showing all tests (the default) then you should 'un-check' all the entries in the list. A quick way to do this is to right click over the list and select *Deselect all* from the menu which appears.

See also:

- Selecting the measures (columns) shown
- Showing repeated measures in columns
- Breaking tests into parts using the 'Segment of test' measure

Breaking tests into parts using the *Segment of test* measure

In brief

If you include the *Segment of test* measure in the Data page spreadsheet, then the spreadsheet will include one row for each test's segment rather than for each test.

Details

ANY-maze includes the ability to break tests into discrete, fixed-length segments and to analyse measures for each segment individually. This allows you to see how a measure alters during the test.

If you include the *Segment of test* measure as a column on the Data page (see Selecting the measures (columns) shown on the data page), then the spreadsheet will automatically show one row for each time segment (unless you choose to show segments in different columns - see Showing repeated measures in columns). For example, with 30-second segments, a five minute test would be shown in 10 rows, with the first row showing results for the first 30 seconds of the test, the second row showing the results for the second 30 seconds, and so on.

Breaking tests into segments in this way is referred to as 'Analysis across time', and you can choose how long the segments are in the *Analysis across time* element of the protocol. For some measures, such analysis affects how the measure is calculated - indeed some measures can't actually be analysed across time at all.

See the following topics for full details:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures

- Syringe pump measures
- Laser controller measures
- OPAD measures
- Procedure measures
- Event measures
- Virtual switch measures

See also:

- Selecting the measures (columns) shown
- Showing repeated measures in columns
- Limiting the tests (rows) shown

Sorting the data

Overview

You can sort the Data page spreadsheet using any of the columns which are shown. This can be very useful if you want to copy or save just a selection of cells, as it can help you to organise the rows so the relevant cells are contiguous. For example, if you just want to copy the data for *Male* animals then sorting by *Gender* will ensure that they're all grouped together.

By default, the spreadsheet is sorted on *Test number*.

Details

To sort the spreadsheet, simply choose the column you want to sort on using the first drop-down list in the *Sort order* section of the Data page's ribbon bar - see figure 1.

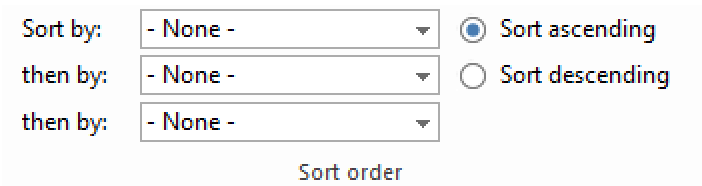


Figure 1. You can sort on up to three columns using the Sort order *settings.*

If you want to, you can perform a hierarchical sort by selecting second and third sort columns too. In this case, the rows will first be sorted on the first column; where the data in this column is the same, it will be sorted on the second column; and where the data in both the first and second columns is the same, it will be sorted on the third column.

You can also choose whether the sort order is ascending or descending.

See also:

- Choosing what data is shown
- Selecting cells, columns or rows
- Copying and saving data

ANY-maze help topic H0798

Selecting cells, columns or rows

Overview

You can easily select groups of cells, columns or rows in the Data page spreadsheet using the mouse.

Test	Animal	Treatment	Code	Stage	Trial	Reason ended	Distance	Island : latency to first entry
1	1	Saline	A	Training	1	Test duration	30.667m	120.0s
2	2	Compound X 5.0 mg/kg	C	Training	1	Test duration	28.585m	120.0s
3	1	Saline	A	Training	2	Island entry	2.034m	26.9s
4	2	Compound X 5.0 mg/kg	C	Training	2	Island entry	0.319m	2.5s
5	1	Saline	A	Training	3	Island entry	2.854m	11.6s
6	2	Compound X 5.0 mg/kg	C	Training	3	Island entry	6.008m	26.1s
7	3	Compound X 1.0 mg/kg	B	Training	1	Island entry	12.950m	52.5s
8	3	Compound X 1.0 mg/kg	B	Training	2	Island entry	1.803m	7.8s
9	4	Compound X 10.0 mg/kg	D	Training	1	Island entry	8.175m	43.8s
10	4	Compound X 10.0 mg/kg	D	Training	2	Island entry	21.408m	91.2s

Figure 1. To select rows, click and drag the mouse in the extreme left-hand edge of the spreadsheet.

Details

To select cells

Click the left mouse button and, holding it down, drag the mouse - the cells between where you first clicked and where the mouse is now will be selected. Dragging the mouse past the edge of the table will cause the table to scroll automatically.

To select columns

Clicking the left mouse button in a column title will select the entire column. Holding down the left button will allow you to drag the mouse to select multiple columns. Again, dragging the mouse past the edge of the table will cause the table scroll automatically.

Alternatively, you can select multiple columns by clicking one column, then holding down the 'Shift' key on the keyboard while clicking another column. All columns between these two will be selected.

To select rows

Move the mouse to the left-hand edge of the spreadsheet - the pointer will change to an arrow which points to the right. Clicking the left mouse button will now select the entire row. Holding down the left button will allow you to drag the mouse to select multiple rows - see figure 1. Again, dragging the mouse past the edge of the table will cause the table scroll automatically.

Alternatively, you can select multiple rows by clicking in the left-hand margin of the first row, then holding down the 'Shift' key on the keyboard while clicking in the margin of another row. All rows between these two will be selected.

Having selected some cells, columns or rows you can copy them to the clipboard or save them to a file , by selecting the appropriate option in the Data page's ribbon bar.

To remove a selection, just click anywhere in the spreadsheet.

Copying and saving data

Overview

The main use of the Data page is to allow you to transfer data to other programs such as Microsoft Excel, Statistica or SPSS. There are two ways to do this - you can either copy data to the clipboard and then paste it into another program, or you can save it to a file and then open the file in the other program.

Copying data to the clipboard

Copying data to the clipboard is extremely simple. To copy the entire spreadsheet, simply click the *Copy* button in the *Clipboard* section of the Data page's ribbon bar. To copy just part of the spreadsheet, first select the cells, columns or rows you want to copy before clicking the *Copy* button.

Notes:

- When you copy the entire spreadsheet or a selection of *columns*, the column titles are included; when you copy a selection of *cells* or *rows*, they are not.
- Some columns, such as 'Distance travelled', include units (e.g. '2.45m'). When copied, the units are omitted.

Saving data to a file

Although copying and pasting data is the quickest method to transfer data, saving to a file can often be a better technique. The reason for this is because most analytical programs (Excel, SPSS and Statistica included) have 'file import wizards' which will ask you about the data you're importing. For example, in SPSS this allows you (amongst other things) to tell the program that the first row of the file contains variable names.

Saving data is very easy. To save the entire spreadsheet, just click the *Save* button in the *Spreadsheet* section of the ribbon bar. To save just part of the spreadsheet, first select the cells, columns or rows you want to save and then click the *Save* button.

Either way, the Save data window will open, where you can specify a name and location for the file to save to, and also choose the file format to use.

Notes:

- When you save all or part of the spreadsheet, the column titles are always included, even if you're only saving a selection of cells.

- Some columns, such as 'Distance travelled', include units (e.g. '2.45m'). When saved, the units are omitted.
- When you save data to a SYLK file, ANY-maze will ask you if you want to use *variable names* as the column titles - for full details about this option, see Using 'Variable names' to title columns when saving data to a file.
- When you save data to a dBase file, ANY-maze will automatically use short variable names as the field names.

Formats in which data can be saved

Symbolic link format	Symbolic link format (also known as SYLK format) was designed by Microsoft for transfer of data between spreadsheet programs. As you'd expect, it's supported in many spreadsheet programs, most notably Excel. Saving data in SYLK format has the advantage that formatting (such as fonts, and column widths) is preserved.
Comma-separated values	The Comma-separated values format (also known as CSV format) is a plain-text format, widely used by spreadsheet and database programs. The disadvantage of this format is that formatting is lost.
dBase III/IV format	dBase format was developed for use with the once popular dBase database program. As this program was once the de facto standard database, almost all database and statistics programs support this format. This is definitely the best format to use if you want to manipulate ANY-maze data using a program like Microsoft Access, although this format is only supported up to Access 2010 and <u>not</u> in versions of Access from 2013 onwards.
Tab-separated format	Tab-separated format is another common plain-text data format, and is also widely supported by spreadsheet and database programs. In this case the file will be saved with a .txt extension, so when reading the file, you should select the option to 'Open a text file'. In fact, apart from the different file extensions, the only real difference between tab- and comma-separated formats is (as the names imply) the fact that in one there are commas between each value, whereas in the other the values are separated by tabs.

See also:

- Transferring data to Microsoft Excel

- Transferring data to Microsoft Access
- Transferring data to SPSS
- Transferring data to SigmaStat
- Transferring data to Systat
- Transferring data to Statistica

Using 'Variable names' to title columns when saving data to a file

Introduction

One of the main reasons for saving the ANY-maze Data page spreadsheet to a file is so it can be read into statistics programs such as SPSS, Systat or Statistica.

As you'd expect, these programs view the columns of the spreadsheet as 'variables', and therefore the column titles are taken to be the names of those variables. However, most stats programs impose limits on the length and contents of variable names, and the standard column titles used in ANY-maze don't tend to satisfy them. For example, if you read a SYLK file into SPSS and the file uses the standard column titles, then many of the variable names will be truncated and others will just be set to values like 'V1' or 'V2'.

To overcome this problem, ANY-maze can save files in which the columns are titled with special 'variable names' rather than the standard titles shown on the spreadsheet. The variable names used have been designed to be compatible with SPSS, but since the SPSS rules are so severe, they should work fine with any stats program.

When ANY-maze uses variable names

ANY-maze may use variable names when saving data, depending on the format you choose to save as:

- When you save data in SYLK format, ANY-maze will ask you if you'd like to use the existing column titles, or whether to replace them with column titles converted to variable name, or long or short variable names.
- When saving in dBase format, short variable names will be used as 'field' names.
- ANY-maze does not include the ability to use variable names in CSV or TXT files - these always use the standard column titles.

SPSS variable name rules

ANY-maze variable names adhere to the SPSS variable naming rules. Quoting from the SPSS help:

The following rules apply to variable names:

- *Each variable name must be unique.*
- *The maximum variable name length is 64 characters (version 12 and higher) or 8 characters ((pre version 12).*
- *The first character must be a letter or the character @.*
- *Middle characters of the variable name can be any combination of letters, numbers, @, #, $, comma (,), period (.) or underscore (_).*
- *The last character of a variable name cannot be a period (.) or underscore (_).*
- *Spaces are not allowed in variable names.*

The standard column titles used by ANY-maze usually violate at least the 'no spaces' rules.

Variable names in ANY-maze

ANY-maze includes three types of variable names - column titles converted into variable names, long variable name and short variable names.

Column titles converted into variable names

ANY-maze can simply convert the column titles shown on the Data page, into variable names which it does by:

- Replacing all spaces with underscores
- Replacing all 'illegal' characters (such as : / - etc.) with a dot
- Adding an @ character to the start of any names which don't begin with a letter
- Chopping off names that are longer than 64 characters

This mechanism usually generates easily understood variable names and it's the option we recommend when you're saving data you plan to analyse in SPSS.

Note that it is possible for this system to create variable names which are not unique. This is because ANY-maze doesn't force you to use unique names for protocol elements such as zones, points, sequences, etc. Thus you could have two different zones in ANY-maze, both of which are called "Centre". This would mean that the Data page could have *two* columns both titled "Centre : time" (reporting the time in the zone) and you'd therefore end up with two variable names called "Centre_._time". Clearly you can avoid this problem by ensuring that you use unique names for your zones, points, sequences, etc.

Another potential issue is that a name might be truncated by ANY-maze (to keep it within the 64 character limit) and *that* might make it non unique and/or make it unintelligible. The best way to avoid this issue is to keep the names of zones, points, sequences, etc. reasonably short.

Long and short variable names

Long and short variable names use an encoded name rather than the column title. For example, the column which shows the time the animal spent in the 'Centre' zone will have a title of 'Centre : time', but its variable name would be 'Z1_T' ('Z' for zone, '1' for the first zone defined in the protocol and 'T' for time). As you can see, this system generates short names which are guaranteed to be unique, but it can sometimes be hard to interpret what the variable names mean (a full list of variable names is given below).

Although ANY-maze has 'long' and 'short' variable names they are, in fact, the same, except when repeated measures are shown in different columns (see Showing repeated measures in columns). In this case, column titles may need to show not only the name of a measure, such as 'Distance travelled', but also a Stage number, a trial number and the start and end times of a test segment - clearly, packing all this information into 8 characters is impossible.

Long variable names

Long variable names adhere to the SPSS naming rules for version 12 and above of SPSS - i.e. they are never more than 64 characters long. The names are formatted as something like '1.2_0.30_DISTANCE', where the '_' character separates sections of the name. So in this example, the first section '1.2' means stage 1, trial 2; the second section 0.30 means the period of the trial from time 0 to time 30 seconds; and 'DISTANCE' is the standard variable name (see below) for the measure shown in the column.

Short variable names

Short variable names are the 'standard' variable names in ANY-maze, and are described below. The problem with short names is that they are limited to just 8 characters, and so they can't include details of stage/trials or time segments. For this reason, if your data table includes different columns for repeated measures, and you save data using short variable names, the columns will simply be named VAR00001, VAR00002, etc. Note that this ONLY occurs when repeated measures are shown in different columns - in all other cases, short variable names *are* meaningful.

Variable name abbreviations used by ANY-maze

The most demanding aspect of the variable name rules is the 8-character limit which they impose. To help create variable names which meet these criteria, ANY-maze employs some standardised abbreviations:

Zx	Zone x, where x is the zone's index. For example, the first zone in the protocol is Z1, the second one Z2 and so on. If you have more than 9 zones then they are referred to as ZA (the tenth zone), ZB (the eleventh) and so on. All zones above ZZ (the 34th zone) are called Z# (i.e. you can't differentiate them).
Px	Point x, where x is the point's index. For example, the first point in the protocol is P1, the second one P2 and so on. If you have more than 9 points then they are referred to as PA (the tenth zone), PB (the eleventh) and so on. All points

above PZ (the 34th point) are called P# (i.e. you can't differentiate them).

SQx	Sequence x, where x is the sequence's index. For example, the first sequence in the protocol is SQ1, the second is SQ2 and so on. If you have more than 9 sequences then they are referred to as SQA (the tenth sequence), SQB (the eleventh) and so on.
Kx	Key x, where x is the key's index. For example, the first key in the protocol is K1, the second is K2 and so on. If you have more than 9 keys then they are referred to as KA (the tenth key), KB (the eleventh) and so on.
Ix	On/off input x, where x is the input's index. For example, the first on/off input in the protocol is I1, the second is I2 and so on. If you have more than 9 on/off inputs then they are referred to as IA (the tenth input), IB (the eleventh) and so on.
SGx	Signal x, where x is the signal's index. For example, the first signal in the protocol is SG1, the second is SG2 and so on. If you have more than 9 signals then they are referred to as SGA (the tenth input), SGB (the eleventh) and so on. *Note that for signal measures within zones, 'Gx' is used rather than 'SGx'.*
Sx	Sensor x, where x is the sensor's index. For example, the first sequence in the protocol is S1, the second is S2 and so on. If you have more than 9 sensors then they are referred to as SA (the tenth sequence), SB (the eleventh) and so on.
Bx	Photobeam x, where x is the photobeam's index. For example, the first photobeam in the protocol is B1, the second is B2 and so on. If you have more than 9 photobeams then they are referred to as BA (the tenth photobeam), BB (the eleventh) and so on.
Ax	Photobeam array x, where x is the photobeam array's index. For example, the first photobeam array in the protocol is A1, the second is A2 and so on. If you have more than 9 photobeam arrays then they are referred to as AA (the tenth array), AB (the eleventh) and so on.
Ex	Rotary encoder x, where x is the rotary encoder's index. For example, the first rotary encoder in the protocol is E1, the second is E2 and so on. If you have more than 9 rotary encoders then they are referred to as EA (the tenth rotary encoder), EB (the eleventh) and so on.
MVx	Movement detector x, where x is the movement detector's index. For example, the first movement detector in the protocol is MV1, the second is MV2 and so on. If you have more than 9 movement detectors then they are referred to as MVA (the tenth movement detector), MVB (the eleventh) and so on.
Ox	Output switch x, where x is the output switch's index. For example, the first output switch in the protocol is O1, the second is O2 and so on. If you have more than 9 output switches then they are referred to as OA (the tenth switch), OB (the eleventh) and so on.

SPx	Syringe pump x, where x is the syringe pump's index. For example, the first syringe pump in the protocol is SP1, the second is SP2 and so on. If you have more than 9 syringe pumps then they are referred to as SPA (the tenth syringe pump), SPB (the eleventh) and so on.
Rx	Result variable x, where x is the result variable's index. For example, the first result variable in the protocol is R1, the second is R2 and so on. If you have more than 9 result variables then they are referred to as RA (the tenth variable), RB (the eleventh) and so on.
VSx	Virtual switch x, where x is the virtual switch's index. For example, the first virtual switch in the protocol is VS1, the second is VS2 and so on. If you have more than 9 virtual switches then they are referred to as VSA (the tenth virtual switch), VSB (the eleventh) and so on.
EVTx	Event x, where x is the Event's index. For example, the first Event in the protocol is EVT1, the second is EVT2 and so on. If you have more than 9 Events then they are referred to as EVTA (the tenth event), EVTB (the eleventh) and so on.
PLGx	Plug-in x, where x is the plug-in's index. For example, the first plug-in in the protocol is PLG1, the second is PLG2 and so on. If you have more than 9 plug-ins then they are referred to as PLGA (the tenth plug-in), PLGB (the eleventh) and so on. *Note that for plug-in measures within zones, 'Px' is used rather than 'PLGx'.*
H	Head. Used to introduce measures that relate to the position of the animal's head. For example Z1_HN is the number of entries (N) of the animal's head (H) into the first zone (Z1).
RR	Rearing. Used to introduce a measure that relates to rearing. For example, Z1_RRN is the number (N) of rears (RR) in the first zone (Z1).
N	Count. For example, Z1_N is the count of entries into the first (Z1) zone.
T	Time. For example, Z1K2_T is the time that the 2nd key (K2) was pressed while the animal was in the first zone (Z1).
SPD	Speed. For example, Z3_SPD is speed in the third zone (Z3).
DST	Distance.
LE	Latency to an entry. For example, Z1_LE is the latency to enter the first zone (Z1).
LX	Latency to an exit.
LP	Latency to a key press.
LR	Latency to a key release.
LS	Latency to start. For example, S1_LS is the latency to the start of the first sequence (S1).
LF	Latency to finish.

LA	Latency to activation. For example, I1_LA is the latency to activation of the first on/off input (I1).
LDA	Latency to deactivation.
MBL	Mobile.
IMBL	Immobile.
ACT	Active.
IACT	Inactive.
1ST	First. For example, ZONE_1ST is the first zone the animal entered.
MX	Maximum. For example, Z2_MXT is the maximum time/duration (T) of a visit to the second zone (Z2).
MN	Minimum.
AV	Average. For example, Z1K2_AVT is the average press time (T) of the second key (K2) in the first zone (Z1).
FRQ	Frequency.
ANGL	Absolute turn angle.
F	Distance from a zone border (when outside the zone) or from a point.
B	Distance to a zone border (when inside the zone) or to a point.
MT	Movement towards a zone or point.
MA	Movement away from a zone or point.
OT	Orientation towards a point.
OA	Orientation away from a point.
GC	Getting closer to a zone. For example, Z1_TGC is time (T) getting closer (GC) to the first zone (Z1).
GF	Getting further from a zone.
IHE	Initial heading error.
AHE	Average heading error.
CW	Clockwise.
ACW	Anti-clockwise.
LC	Line crossing.

In some cases, more complete names are used where space allows. For example, the overall distance travelled in the test is called DISTANCE rather than just DST.

Full list of ANY-maze measure variable names

A full list of all the variable names used by ANY-maze when saving the Data page spreadsheet is given below:

Information measures:

TEST	Test number
ANIMAL	Animal number
TREAMNT	Treatment name
TRMTCODE	Treatment code
STAGE	Stage name
TRIAL	Trial number
STGTRIAL	Stage name and trial number, for example 'Training 1'
APPARTUS	Apparatus name
END_RSN	The reason the test ended
TIME	Test time
DAY	Test day of the week
AM_PM	Test time of day (am/pm)
DATE	Test date
TESTER	Name of the tester
TESTNOTE	Test notes
ANMLNOTE	Animal notes
SEGMENT	Segment of the test
LGHT_DRK	Animal lighter/darker than the apparatus
PERIOD	The time period, for example '0-30'
Zx_LOCTN	The location of the Zx zone.

For more information on any of these measures, see Information measures.

Apparatus measures:

DURATION	Test duration
DISTANCE	Total distance travelled
SPEED	Overall average speed
ZONE_1ST	First zone entered
MBL_T	Total time mobile
IMBL_T	Total time immobile

MBL_N	Total mobile episodes
IMBL_N	Total immobile episodes
ACT_T	Total time active
IACT_T	Total time inactive
ACT_N	Total active episodes
IACT_N	Total inactive episodes
MXACT_T	Longest active episode
MNACT_T	Shortest active episode
MXIACT_T	Longest inactive episode
MNIACT_T	Shortest inactive episode
CROSSING	Number of line crossings
ANGLE	Absolute turn angle
H_DST	Distance travelled by the animal's head
MAXSPEED	Maximum speed
RTNS	Rotations of the animal's body
RTNS_CW	Clockwise rotations of the animal's body
RTNS_ACW	Anti-clockwise rotations of the animal's body
PATH_EFF	Path efficiency
MBL_LT	Latency to start of first mobile episode
IMMBL_LT	Latency to start of first immobile episode
FRZ_N	Total freezing episodes
FRZ_T	Total time freezing
FRZ_LT	Latency to start of first freezing episode
AV_FRZ	Average freezing score
ZN_LIST	Visited zone list
POSREV_N	On/off inputs positive reversal
NEGREV_N	On/off inputs negative reversal
REAR_N	Number of rears
REAR_T	Total time rearing
REAR_LS	Latency to first rear
REAR_AVT	Average duration of a rear
REAR_MXT	Maximum duration of a rear
REAR_MNT	Minimum duration of a rear

For more information on any of these measures, see Apparatus measures.

Apparatus key measures:

The following variable names are all prefixed with Kx, where x is the key's index: 1 = first key, 2 = second key and so on.

Kx_N	Number of presses
Kx_T	Time pressed
Kx_LP	Latency to first press
Kx_LR	Latency to first release
Kx_MXT	Longest press
Kx_MNT	Shortest press
Kx_AVT	Average press duration
Kx_FRQ	Press frequency
Kx_DFP	Distance travelled by the animal before first press
Kx_LST	List of durations of each key press

For more information on any of these measures, see Key measures.

Apparatus on/off input measures:

The following variable names are all prefixed with Ix, where x is the on/off input's index: 1 = first on/off input, 2 = second on/off input and so on.

Ix_N	Number of activations
Ix_T	Time activated
Ix_LA	Latency to first activation
Ix_LDA	Latency to first deactivation
Ix_MXT	Longest activation
Ix_MNT	Shortest activation
Ix_AVT	Average activation duration
Ix_FRQ	Activation frequency

For more information on any of these measures, see On/off input measures.

Apparatus signal measures:

The following variable names are all prefixed with SGx, where x is the signal's index: 1 = first signal, 2 = second signal and so on.

SGx_AVG	Average value

SGx_MAX	Maximum value
SGx_MIN	Minimum value
SGx_MXT	Time of maximum value
SGx_MNT	Time of minimum value
SGx_BL	Baseline value
SGx_BDV	Baseline deviation
SGx_BSD	Baseline standard deviation
SGx_BLT	Time of the end of the baseline period
SGx_PDT	Time of first positive deviation from baseline
SGx_PDR	Time of first return to baseline from positive deviation
SGx_NDT	Time of first negative deviation from baseline
SGx_NDR	Time of first return to baseline from negative deviation
SGx_PIN	Integral above baseline
SGx_NIN	Integral below baseline

For more information on any of these measures, see Signal measures.

Apparatus sensor measures:

The following variable names are all prefixed with Sx, where x is the sensor's index: 1 = first sensor, 2 = second sensor and so on.

Sx_AVG	Average value
Sx_MAX	Maximum value
Sx_MIN	Minimum value
Sx_VAL	Initial value
Sx_CHG	Change from initial value

For more information on any of these measures, see Sensor measures.

Apparatus rotary encoder measures:

The following variable names are all prefixed with Ex, where x is the rotary encoder's index: 1 = first rotary encoder, 2 = second rotary encoder and so on.

Ex_N	Number of rotations
Ex_NC	Number of clockwise rotations
Ex_NAC	Number of anti-clockwise rotations
Ex_NRV	Number of reversals of direction

Ex_NHF	Number of half-rotations
Ex_NQT	Number of quarter-rotations
Ex_MXR	Maximum RPM
Ex_MNR	Minimum RPM
Ex_AVR	Average RPM
Ex_CD	Number of degrees of clockwise rotation
Ex_ACD	Number of degrees of anti-clockwise rotation
Ex_DST	Distance

For more information on any of these measures, see Rotary encoder measures.

Apparatus movement detector measures:

The following variable names are all prefixed with MVx, where x is the movement detector's index: 1 = first detector, 2 = second detector and so on.

MVx_N	Count of beam breaks
MVx_T	Time moving
MVx_LA	Latency to first movement (beam break)

For more information on any of these measures, see Movement detector measures.

Apparatus syringe pump measures:

The following variable names are all prefixed with SPx, where x is the syringe pump's index: 1 = first pump, 2 = second pump and so on.

SPx_VI	Volume infused
SPx_VW	Volume withdrawn

For more information on any of these measures, see Syringe pump measures.

Apparatus result variable measures:

The following variable names are all prefixed with Vx, where x is the result variable's index: 1 = first variable, 2 = second variable and so on.

Rx_AVG	Average value
Rx_MAX	Maximum value
Rx_MIN	Minimum value
Rx_SUM	Sum of values
Rx_N	Count of values
Rx_VAL	Value *(if the result variable is noted only once at the end of the test)*

For more information on any of these measures, see Procedure measures.

Apparatus virtual switch measures:

The following variable names are all prefixed with Vx, where x is the virtual switch's index: 1 = first virtual switch, 2 = second virtual switch and so on.

Vx_N	Number of activations
Vx_T	Latency to first activation
Vx_LDA	Latency to first deactivation
Vx_MXT	Longest activation
Vx_MNT	Shortest activation
Vx_AVT	Average activation duration
Vx_FRQ	Frequency of activations
Vx_DFA	Distance travelled before first activation
Vx_DAC	Distance travelled while active

For more information on any of these measures, see Virtual switch measures.

Apparatus OPAD measures:

Some of the following variable names are prefixed with OPD_x, where x is the index of the temperature of interest: 1 = first temperature of interest, 2 = second temperature of interest and so on.

OPD_TCB	Temperature when contact broken
OPD_xTC	Temperature of interest: time in contact
OPD_xNB	Temperature of interest: number of times contact broken
OPD_xNM	Temperature of interest: number of times contact made
OPD_xNL	Temperature of interest: number of licks
OPD_NLCT	Number of non-lick contacts
OPD_LTCB	Left thermal element; temperature when contact broken
OPD_xLTC	Left thermal element; temperature of interest: time in contact
OPD_xLNB	Left thermal element; temperature of interest: number of times contact broken
OPD_xLNM	Left thermal element; temperature of interest: number of times contact made
OPD_xLNL	Left thermal element; temperature of interest: number of licks
OPD_RTCB	Right thermal element; temperature when contact broken
OPD_xRTC	Right thermal element; temperature of interest: time in contact

OPD_xRNB	Right thermal element; temperature of interest: number of times contact broken
OPD_xRNM	Right thermal element; temperature of interest: number of times contact made
OPD_xRNL	Right thermal element; temperature of interest: number of licks

For more information on any of these measures, see OPAD measures.

Apparatus RAPC measures:

RAPC_T1	Number of RAPC type 1 errors
RAPC_T2	Number of RAPC type 2 errors
RAPC_SEQ	Door sequence

For more information on any of these measures, see Performing experiments and analysing results for RAPC.

Apparatus plug-in measures:

The following variable names are all prefixed with Px, where x is the plug-in's index: 1 = first plug-in, 2 = second plug-in and so on.

The following variable names apply to RT_ONOFF result type measures:

PLGx_N	Number of activations
PLGx_T	Time active
PLGx_LON	Latency to first activation
PLGx_LOF	Latency to first deactivation
PLGx_MXT	Longest activation
PLGx_MNT	Shortest activation
PLGx_AVT	Average activation duration
PLGx_FRQ	Activation frequency

The following variable names apply to RT_VALUE result type measures:

PLGx_AVG	Average value
PLGx_MAX	Maximum value
PLGx_MIN	Minimum value

The following variable names apply to RT_CUMULATIVE result type measures:

PLGx_VAL	Cumulative value

For more information on any of these measures, see Plug-in measures.

Zone measures:

The following variable names are prefixed with Zx, where x is the zone's index: 1 = first zone, 2 = second zone and so on.

Zx_N	Number of entries to the zone
Zx_1ST	Was first zone entered
Zx_T	Time in zone
Zx_DST	Distance travelled in the zone
Zx_LE	Latency to first entry to the zone
Zx_LX	Latency to first exit to the zone
Zx_SPD	Average speed in the zone
Zx_MXT	Longest visit to the zone
Zx_MNT	Shortest visit to the zone
Zx_AVT	Average duration of visit to the zone
Zx_MBLT	Time mobile in the zone
Zx_IMBLT	Time immobile in the zone
Zx_IMBLN	Immobile episodes in the zone
Zx_ACTT	Time active in the zone
Zx_IACTT	Time inactive in the zone
Zx_IACTN	Inactive episodes in the zone
Zx_AVF	Average distance from the zone
Zx_MXF	Maximum distance from the zone
Zx_MNF	Minimum distance from the zone
Zx_AVB	Average distance to the zone border
Zx_MXB	Maximum distance to the zone border
Zx_MNB	Minimum distance to the zone border
Zx_TGC	Time getting closer to the zone
Zx_TGF	Time getting further away from the zone
Zx_ANGL	Absolute turn angle while in the zone
Zx_HT	Time the animal's head was in the zone
Zx_HN	Number of entries of the animal's head into the zone
Zx_HDST	Distance travelled by the animal's head in the zone
Zx_HLE	Latency to the first entry of the animal's head into the zone
Zx_HLX	Latency to the first exit of the animal's head from the zone

Zx_HAVF	Average distance of the animal's head from the zone
Zx_HMXF	Maximum distance of the animal's head from the zone
Zx_HMNF	Minimum distance of the animal's head from the zone
Zx_HAVB	Average distance from the animal's head to the zone border
Zx_HMXB	Maximum distance from the animal's head to the zone border
Zx_HMNB	Minimum distance from the animal's head to the zone border
Zx_RRN	Number of rears in the zone
Zx_RRT	Total time rearing in the zone
Zx_RRL	Latency to first rear in the zone
Zx_RRAVT	Average duration of a rear in the zone
Zx_RRMXT	Maximum duration of a rear in the zone
Zx_RRMNT	Minimum duration of a rear in the zone
Zx_IHE	Initial heading error to the zone
Zx_AHE	Average absolute heading error to the zone
Zx_TMT	Time moving towards the zone
Zx_TMA	Time moving away from the zone
Zx_DFE	Distance travelled until first entry into the zone
Zx_LLE	Latency to last entry to the zone
Zx_MXSPD	Maximum speed in the zone
Zx_EXIT	Number of exits from the zone
Zx_CIPL	Corrected integrated path length
Zx_INDST	Initial distance from the zone
Zx_CUMD	Cumulative distance from the zone
Zx_T_OR	Time oriented towards the centre of the zone when inside zone
Zx_HICO	Time the animal's head was in the zone when its centre was outside the zone
Zx_PEFF	Path efficiency of first entry to the zone
Zx_FRZN	Freezing bouts in the zone
Zx_FRZT	Time freezing in the zone
Zx_SIHE	Signed initial heading error to the zone
Zx_WCT	Time spent in Whishaw's corridor
Zx_WCD	Distance travelled in Whishaw's corridor
Zx_LST	List of the duration of each visit to the zone
Zx_AHTA	Absolute head turn angle in the zone

Zx_LC	Number of line crossings in the zone

For more information on any of these measures, see Zone measures.

Zone key measures:

The following variable names are prefixed with ZxKy, where x is the zone's index and y the key's index. For example Z1K2 is a measure for the 2nd key in the 1st zone.

ZxKy_N	Number of presses while the animal is in the zone
ZxKy_T	Time pressed while the animal is in the zone
ZxKy_LP	Latency to first press while the animal is in the zone
ZxKy_LR	Latency to first release while the animal is in the zone
ZxKy_MXT	Longest press while the animal is in the zone
ZxKy_MNT	Shortest press while the animal is in the zone
ZxKy_AVT	Average press while the animal is in the zone
ZxKy_FRQ	Press frequency while the animal is in the zone

For more information on any of these measures, see Key measures.

Zone on/off input measures:

The following variable names are prefixed with ZxIy, where x is the zone's index and y is the on/off input's index. For example, Z1I2 is a measure for the 2nd on/off input in the 1st zone.

ZxIx_N	Number of activations while the animal is in the zone
ZxIx_T	Time activated while the animal is in the zone
ZxIx_LA	Latency to first activation while the animal is in the zone
ZxIx_LDA	Latency to first deactivation while the animal is in the zone
ZxIx_MXT	Longest activation while the animal is in the zone
ZxIx_MNT	Shortest activation while the animal is in the zone
ZxIx_AVT	Average activation duration while the animal is in the zone
ZxIx_FRQ	Activation frequency while the animal is in the zone

For more information on any of these measures, see On/off input measures.

Zone signal measures:

The following variable names are prefixed with ZxGy, where x is the zone's index and y is the signal's index. For example, Z1G2 is a measure for the 2nd signal in the 1st zone.

ZxGy_AVG	Average value while the animal is in the zone
ZxGy_MAX	Maximum value while the animal is in the zone

ZxGy_MIN	Minimum value while the animal is in the zone
ZxGy_PIN	Integral above baseline while the animal is in the zone
ZxGy_NIN	Integral below baseline while the animal is in the zone
ZxGy_MX	Average maximum for each visit to the zone
ZxGy_MXT	Average time to maximum for each visit to the zone
ZxGy_MI	Average minimum for each visit to the zone
ZxGy_MIT	Average time to minimum for each visit to the zone
ZxGy_MEN	Average value at zone entry
ZxGy_MEX	Average value at zone exit

For more information on any of these measures, see Signal measures.

Zone sensor measures:

The following variable names are prefixed with ZxSy, where x is the zone's index and y is the sensor's index. For example, Z1S2 is a measure for the 2nd sensor in the 1st zone.

ZxSy_AVG	Average value while the animal is in the zone
ZxSy_MAX	Maximum value while the animal is in the zone
ZxSy_MIN	Minimum value while the animal is in the zone
ZxSy_VAL	Initial value while the animal is in the zone

For more information on any of these measures, see Sensor measures.

Zone rotary encoder measures:

The following variable names are prefixed with ZxEy, where x is the zone's index and y is the rotary encoder's index. For example, Z1E2 is a measure for the 2nd rotary encoder in the 1st zone.

ZxEy_N	Number of rotations while the animal is in the zone
ZxEy_NC	Number of clockwise rotations while the animal is in the zone
ZxEy_NAC	Number of anti-clockwise rotations while the animal is in the zone
ZxEy_NRV	Number of reversals of direction while the animal is in the zone
ZxEy_NHF	Number of half-rotations while the animal is in the zone
ZxEy_NQT	Number of quarter-rotations while the animal is in the zone
ZxEy_MXR	Maximum RPM while the animal is in the zone
ZxEy_MNR	Minimum RPM while the animal is in the zone
ZxEy_AVR	Average RPM while the animal is in the zone
ZxEy_CD	Number of degrees of clockwise rotation while the animal is in the zone

ZxEy_ACD	Number of degrees of anti-clockwise rotation while the animal is in the zone
ZxEy_DST	Distance while the animal is in the zone

For more information on any of these measures, see Rotary encoder measures.

Zone syringe pump measures:

The following variable names are prefixed with ZxSPy, where x is the zone's index and y is the syringe pump's index. For example, Z1SP2 is a measure for the 2nd syringe pump in the 1st zone.

ZxSPy_VI	Volume infused while the animal is in the zone
ZxSPy_VW	Volume withdrawn while the animal is in the zone

For more information on any of these measures, see Syringe pump measures.

Zone result variable measures:

The following variable names are prefixed with ZxRy, where x is the zone's index and y is the result variable's index. For example, Z1R2 is a measure for the 2nd result variable in the 1st zone.

ZxRy_AVG	Average value while the animal is in the zone
ZxRy_MAX	Maximum value while the animal is in the zone
ZxRy_MIN	Minimum value while the animal is in the zone
ZxRy_USM	Sum of values while the animal is in the zone
ZxRy_N	Count of values while the animal is in the zone
ZxRy_VAL	Value (if the result variable is noted only once at the end of the test), if the animal is in the zone at the end of the test

For more information on any of these measures, see Procedure measures.

Zone virtual switch measures:

The following variable names are prefixed with ZxVy, where x is the zone's index and y is the virtual switch's index. For example, Z1V2 is a measure for the 2nd virtual switch in the 1st zone.

ZxVy_N	Number of activations while the animal is in the zone
ZxVy_T	Latency to first activation while the animal is in the zone
ZxVy_LA	Latency to first deactivation while the animal is in the zone
ZxVy_LDA	Longest activation while the animal is in the zone
ZxVy_MXT	Shortest activation while the animal is in the zone
ZxVy_MNT	Average activation duration while the animal is in the zone
ZxVy_AVT	Frequency of activations while the animal is in the zone
ZxVy_FRQ	Distance travelled before first activation while the animal is in the zone

ZxVy_DAC Distance travelled while active while the animal is in the zone

For more information on any of these measures, see Virtual switch measures.

Zone plug-in measures:

The following variable names are prefixed with ZxPy, where x is the zone's index and y is the plug-in's index. For example, Z1P2 is a measure for the 2nd plug-in in the 1st zone.

The following variable names apply to RT_ONOFF result type measures:

ZxPy_N	Number of activations while the animal is in the zone
ZxPy_T	Time active while the animal is in the zone
ZxPy_LON	Latency to first activation while the animal is in the zone
ZxPy_LOF	Latency to first deactivation while the animal is in the zone
ZxPy_MXT	Longest activation while the animal is in the zone
ZxPy_MNT	Shortest activation while the animal is in the zone
ZxPy_FRQ	Activation frequency while the animal is in the zone

The following variable names apply to RT_VALUE result type measures:

ZxPy_AVG	Average value while the animal is in the zone
ZxPy_MAX	Maximum value while the animal is in the zone
ZxPy_MIN	Minimum value while the animal is in the zone

The following variable names apply to RT_CUMULATIVE result type measures:

ZxPy_VAL	Cumulative value while the animal is in the zone

For more information on any of these measures, see Plug-in measures.

Point measures:

The following variable names are prefixed with Px, where x is the point's index: 1 = first point, 2 = second point and so on.

Px_AVF	Average distance from the point
Px_MXF	Maximum distance from the point
Px_MNF	Minimum distance from the point
Px_TMT	Time moving towards the point
Px_TMA	Time moving away from the point
Px_IHE	Initial heading error to the point
Px_AHE	Average heading error to the point
Px_HAVF	Average distance of the animal's head from the point

Px_HMXF	Maximum distance of the animal's head from the point
Px_HMNF	Minimum distance of the animal's head from the point
Px_HTMT	Time the animal's head was moving towards the point
Px_HTMA	Time the animal's head was moving away from the point
Px_HTOT	Time the animal was oriented towards the point
Px_HTOA	Time the animal was oriented away from the point

For more information on any of these measures, see Point measures.

Sequence measures:

The following variable names are prefixed with Sx, where x is the sequence's index: 1 = the first sequence, 2 = second sequence and so on.

SQx_LS	Latency to first sequence start
SQx_LF	Latency to first sequence end
SQx_N	Number of sequences
SQx_T	Time spent performing sequences
SQx_AVT	Average time to perform a sequence
SQx_MXT	Longest time to perform a sequence
SQx_MNT	Shortest time to perform a sequence
SQx_DST	Distance travelled while performing sequence
SQx_AVD	Average distance travelled per sequence
SQx_MXD	Maximum distance travelled in a sequence
SQx_MND	Minimum distance travelled in a sequence
SQx_AVSP	Average speed during the sequence
SQx_BRKN	Number of broken sequences

For more information on any of these measures, see Sequence measures.

Fields and calculations

The variable names of fields and calculations are the first 8 characters of the element's full name, with any illegal characters replaced by an underscore - this mainly applies to spaces. If the resulting name *ends* with a dot or an underscore, then the last character is replaced with a hash sign (#).

This method of naming fields and calculation variables means that names CAN be identical, for example the calculations 'Percentage of time in open arms' and 'Percentage of time in closed arms' would both be given variable names of 'PERCENTA'. It's worth considering this when naming fields and calculations, although getting variables names right isn't *hugely* important, as you can just change them in SPSS.

Test data variable names

A full list of all the variable names used by ANY-maze when saving the Test data report is given below:

Test control

PAUS_ON	Pause on
PAUS_OFF	Pause off
TEST_END	Test end

Animal data

COG_X	The x coordinate of the centre of the animal (COG = Centre Of Gravity)
COG_Y	The y coordinate of the centre of the animal (COG = Centre Of Gravity)
HEAD_X	The x coordinate of the animal's head
HEAD_Y	The y coordinate of the animal's head
TAIL_X	The x coordinate of the base of the animal's tail
TAIL_Y	The y coordinate of the base of the animal's tail
HDN_ON	Hidden on (the animal has become hidden)
HDN_OFF	Hidden off (the animal has ceased to be hidden)
SPEED	Speed of the animal
MBL	Mobile (the animal has become mobile)
IMBL	Immobile (the animal has become immobile)
ROT_CW	Clockwise rotation
ROT_ACW	Anti-clockwise rotation
FRZ_SCR	The animal's freezing score
FRZ_ON	Freezing on (the animal freezes)
FRZ_OFF	Freezing off (the animal stops freezing)
LENGTH	The length of the animal
REL_LEN	The relative length of the animal (relative to its mean length)

Zone data

The following variable names are prefixed with Zx, where x is the zone's index: 1 = the first zone, 2

= second zone and so on.

Zx_ENTRY	The animal is in the zone
Zx_EXIT	The animal is not in the zone
Zx_DST	The distance of the animal from the zone (when outside the zone)
Zx_BDR	The distance of the animal from the zone border (when inside the zone)
Zx_H_DST	The distance of the animal's head from the zone (when the head is outside the zone)
Zx_H_BDR	The distance of the animal's head from the zone border (when the head is inside the zone)
Zx_HDERR	The animal's heading error to the zone

Point data

The following variable names are prefixed with Px, where x is the point's index: 1 = the first point, 2 = second point and so on.

Px_DST	The distance of the animal from the point
Px_H_DST	The distance of the animal's head from the point
Px_HDERR	The animal's heading error to the point

Sequence data

The following variable names are prefixed with SQx, where x is the sequence's index: 1 = the first sequence, 2 = second sequence and so on.

SQx_STRT	The start of the sequence
SQx_END	The end of the sequence
SQx_BRKN	The sequence being broken

Key data

The following variable names are prefixed with Kx, where x is the key's index: 1 = the first key, 2 = second key and so on.

Kx_ON	The key is active
Kx_OFF	The key is inactive

On/off input data

The following variable names are prefixed with Ix, where x is the on/off input's index: 1 = the first on/off input, 2 = second on/off input and so on.

Ix_ON	The on/off input is active
Ix_OFF	The on/off input is inactive
I_SEQ	On/off input sequence - the indexes of inputs that were activated
I_RV_POS	On/off input sequence positive reversal
I_RV_NEG	On/off input sequence negative reversal

Signal input data

The following variable names use an x to represent the signal's index: 1 = the first signal, 2 = second signal and so on.

SGx	The value of the signal

Sensor data

The following variable names use an x to represent the sensor's index: 1 = the first sensor, 2 = second sensor and so on.

SNSx	The value of the sensor

Photobeam data

The following variable names are prefixed with Bx, where x is the photobeam's index: 1 = the first photobeam, 2 = second photobeam and so on.

Bx_ON	The photobeam is active
Bx_OFF	The photobeam is inactive

Rotary encoder data

The following variable names are prefixed with REx, where x is the rotary encoder's index: 1 = the first rotary encoder, 2 = second rotary encoder and so on.

REx_CWP	Clockwise pulse of the rotary encoder
REx_ACWP	Anti-clockwise pulse of the rotary encoder
REx_RPM	Speed of the rotary encoder (RPM)

Movement detector data

The following variable names use an x to represent the movement detector's index: 1 = the first movement detector, 2 = second movement detector and so on.

MOVDETx	A beam of the movement detector has been broken

Output switch data

The following variable names are prefixed with Ox, where x is the output switch's index: 1 = the first switch, 2 = second switch and so on.

Ox_ON	The output switch is active
Ox_OFF	The output switch is inactive

Speaker data

The following variable names use an x to represent the speaker's index: 1 = the first speaker, 2 = second speaker and so on.

SPKx	Speaker activation
SPKVOLx	The volume of the speaker

Lighting controller data

The following variable names use an x to represent the lighting controller's index: 1 = the first lighting controller, 2 = second lighting controller and so on.

LGHTONx	The lighting controller is activated
LGHTLUXx	The lighting controller light level

Syringe pump data

The following variable names use an x to represent the syringe pump's index: 1 = the first syringe pump, 2 = second syringe pump and so on.

SPRUNx	Syringe pump is running
SPVOLIx	Volume infused
SPVOLWx	Volume withdrawn

Shocker data

The following variable names are prefixed with SKx, where x is the shocker's index: 1 = the first shocker, 2 = second shocker and so on.

SKx_ON	Shocker activation
SKx_CRNT	Shocker current

Result variable data

The following variable names use an x to represent the result variable's index: 1 = the first variable, 2 = second variable and so on.

VARx	Result variable value

Virtual switch data

The following variable names are prefixed with VSx, where x is the virtual switch's index: 1 = the first virtual switch, 2 = second virtual switch and so on.

VSx_ON	The virtual switch is active
VSx_OFF	The virtual switch is inactive

Event data

The following variable names use an x to represent the event's index: 1 = the first event, 2 = second event and so on.

EVTx	The event occurs

Plug-in data

The following variable names are prefixed with PLGx, where x is the plug-in's index: 1 = the first plug-in, 2 = second plug-in and so on.

PLGx_ON	The on/off-type plug-in is active
PLGx_OFF	The on/off-type plug-in is inactive
PLGx	The value of the value-type plug-in

See also:

- Information measures
- Apparatus measures
- Zone measures
- Point measures
- Sequence measures
- Key measures
- On/off input measures
- Signal measures
- Sensor measures
- Rotary encoder measures
- Movement detector measures
- Output switch measures

- Syringe pump measures
- Laser controller measures
- OPAD measures
- Performing experiments and analysing results for RAPC
- Procedure measures
- Event measures
- Virtual switch measures
- Plug-in measures
- Transferring data to Microsoft Excel
- Transferring data to Microsoft Access
- Transferring data to SPSS
- Transferring data to SigmaStat
- Transferring data to Systat
- Transferring data to Statistica

ANY-maze help topic H0808

Transferring data to specific programs

Introduction

Transferring ANY-maze data to other programs is generally straightforward, but to save you time, we've included some topics which describe the process in detail for a few popular spreadsheet, database and statistics programs.

How to transfer data to:

- Microsoft Excel
- Microsoft Access
- SPSS
- SigmaStat
- Systat
- Statistica

Transferring data to Microsoft Excel

Introduction

Microsoft Excel is the world's de-facto standard spreadsheet program. ANY-maze fully supports transfer of data to Excel, either via the Windows clipboard or by saving data to a file.

Copy and paste

The easiest way to transfer to Excel is to simply copy and paste; however the formatting of the data (font, and column widths) will be lost.

Saving data to a file

Excel supports all the file formats in which ANY-maze can save data (SYLK, dBase, CSV and text) so you can save in any of them. Nevertheless, SYLK (which was developed by Microsoft) is the best one to use, as it preserves the format of the data.

When you save data to a SYLK file, ANY-maze will ask you if you want to use **variable names** *for the column titles. When transferring data to Excel, the standard column titles are fine so you can simply select Use column titles* when asked.

How to open a SYLK file in Excel

1. Click the *File* menu, or the *File* tab in versions of Excel from 2013 onwards.
2. Select *Open...*.

   For Excel 2013 onwards, you'll also need to select the *Computer* option and then *Browse...*.

   The *Open file* window will appear.
3. At the bottom right of the window is a list of the types of file you can open. Scroll down the list (it's quite long) until you find 'SYLK' and select it.
4. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
5. Click the *Open* button.

See also:

- Selecting cells, columns or rows

- Copying and saving data
- The save data window

Transferring data to Microsoft Access

Introduction

Microsoft Access is a database program that's part of the popular Microsoft Office suite. The easiest way to transfer data to earlier versions of Access (up to and including Access 2010) is simply to save it as a dBase file; however versions of Access from 2013 onwards no longer support this file type, so for newer versions of Access, you'll need to transfer by saving to a comma-separated (CSV) or tab-separated (text) file.

Copy and paste

It's not possible to copy and paste data from ANY-maze to Microsoft Access.

Saving data to a file

Microsoft Access supports all the file formats in which ANY-maze can save data, except SYLK. The easiest format to use for versions of Access up to Access 2010 is dBase - although, because field names in dBase are limited to 10 characters, this format will use ANY-maze's short variable names for field names. If you'd prefer to use the standard column titles that appear on the Data page spreadsheet as the field names in the Access database, or you have a version of Access from 2013 onwards, then you should save the data as a CSV file.

How to open a dBase file in Access

⚠ *Note that this will only work in versions of Microsoft Access up to and including Access 2010. Versions of Access from 2013 onwards do not support dBase files.*

1. Open the Access database you want to import the data into - or create a new database.
2. Click the *File* menu.
3. Select *Get external data* | *Import...*. The *Import* window will appear.
4. At the bottom of the window is a list titled *Files of Type*. Select 'dBase IV'.
5. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
6. Click the *Import* button.

How to open a CSV file in Access

1. Open the Access database you want to import the data into - or create a new database.
2. Click the *File* menu (or the *File* tab, in versions of Access from 2013 onwards).
3. In versions of Access up to Access 2010, select *Get external data | Import*.... The *Import* window will appear.

   For Access 2013 onwards, select *Open*, then select the *Computer* option and click *Browse*.... The *Open* window will appear.
4. At the bottom right of this window is a list of the types of file you can open. Select 'Text files' from this list.
5. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
6. Click the *Import* button (or the *Open* button in Access 2013 onwards) - the *Import text wizard* will open (from Access 2013, this is called the *Link text wizard*).
7. On the first page of the wizard, make sure that the *Delimited* option is selected, and click the *Next* button.
8. On the next page check the box titled *First row contains field names*, then click *Next*.
9. Proceed through the rest of the wizard selecting the options you want to use - these will depend on what you plan to do with the resulting Access database.

See also:

- Selecting cells, columns or rows
- Copying and saving data
- The save data window

Transferring data to SPSS

Introduction

SPSS is a popular statistical analysis program. Transferring data from ANY-maze to SPSS is relatively simple - the best method being to save data to a SYLK file.

If your data includes repeated measures, you should use the options to Show repeated measures in columns, as this is the format that SPSS expects.

Copy and paste

Copying data and pasting into SPSS does work, but textual data will be lost unless you've already set up the SPSS variables correctly to specify which columns contain text and which contain numbers.

Saving data to a file

SPSS supports all the file formats provided by ANY-maze (SYLK, dBase, CSV and text) but the best ones to use are SYLK and dBase.

The main issue when transferring data to SPSS is variable naming. SPSS will take the column titles from the ANY-maze spreadsheet and use them to create variable names - the problem is that SPSS variable names are limited to 8 characters (in versions prior to V12) and cannot contain spaces and certain other characters. ANY-maze's column titles generally break these variable naming rules.

You can overcome this when you save a file in SYLK format, because ANY-maze will ask you how you want to save the column titles. For older versions of SPSS (before version 12) you can select *Use short variable names*; for later versions, you can select *Use long variable names*. This will ensure that the column titles are saved as SPSS-compatible variable names.

In fact, saving data as a dBase file is even easier. In this case ANY-maze *always* uses short variable names, because dBase *field titles* (which are like column titles) are limited to 10 characters.

How to open a SYLK or dBase file in SPSS

These instructions are for SPSS version 10.0.7.

1. Click the *File* menu.
2. Select *Open | Data...*; the *Open file* window will appear.
3. At the bottom of the window is a list titled *Files of Type*. Select either 'SYLK' or 'dBase', depending on the type of file you wish to open.

4. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
5. Click the *Open* button.
6. When you open a SYLK file, the *Open file options* window will appear. Click the box labelled *Read variable names* and then click *OK*.

See also:

- Selecting cells, columns or rows
- Copying and saving data
- The save data window

Transferring data to SigmaStat

Introduction

SigmaStat is a popular statistical analysis program. Transferring data from ANY-maze to SigmaStat is very simple - just save the data to transfer to a dBase file.

Copy and paste

Copying data and pasting into SigmaStat works perfectly, although you should note that if you copy the entire ANY-maze spreadsheet (i.e. without selecting anything) then when you paste it into SigmaStat, the column titles will appear as the first row of *data*. You can avoid this by either deleting this row, or by selecting all the rows in the Data page spreadsheet before copying them - see Selecting cells, columns or rows for details.

In any case, if you want to transfer the entire spreadsheet to SigmaStat then the better option is to save it as a dBase file, as this way the columns will be titled for you.

Saving data to a file

SigmaStat can read data from CSV, tab-separated and dBase files. However, in the first two formats it doesn't take the first row as being column titles (i.e. variable names) and so the column titles appear as the first row of data.

The best format to use is dBase - this works perfectly.

How to open a dBase file in SigmaStat

These instructions are for SigmaStat for Windows version 3.00.0

1. Click the *File* menu.
2. Select *Open*; the *Open file* window will appear.
3. At the bottom of the window is a list titled *Files of Type*. Select 'dBase'.
4. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
5. Click the *Open* button.
6. The *Select file type* window will open - ensure 'dBase III or higher' is selected and click *OK*.

In fact, you can also use the *Import* command from the *File* menu. This is almost identical to using the

Open command (as described above), except that you can choose to import just a portion of the dBase file rather than all of it.

See also:

- Selecting cells, columns or rows
- Copying and saving data
- The save data window

Transferring data to Systat

Introduction

Systat is a popular statistical analysis program. You can transfer data from ANY-maze to Systat either by copying and pasting, or by saving the data in a dBase file (recommended).

Copy and paste

Copying and pasting data into Systat works correctly, but you need to be a little careful.

If you copy the entire ANY-maze Data page spreadsheet (i.e. without selecting anything), then when you paste it into Systat, the column titles will appear as the first row. As Systat uses the contents of the first row to determine what type of data is in each column, it will think everything is text (the column titles being textual) and will create its variables accordingly.

Fortunately this is easy to overcome - just *select* the entire table before copying it. When ANY-maze copies a selection, it doesn't include the column titles in the copied data.

Saving data to a dBase file

The easiest way to transfer data into Systat is to save it as a dBase file and then open the file in Systat. There's one downside to this - if your data includes any dates, then according to the Systat 10.2 help file, they'll be converted to numbers (quote from the Systat help):

> *Date fields are converted to numeric values. For example, if your dBase file contained a value for November 2, 1987, it would be stored as 19871102. Importing converts this to 0.19871102E + 8.*

Incidentally, Systat doesn't import ANY-maze CSV or tab-separated files correctly, as it appears to treat spaces as separators - therefore the only one of ANY-maze's formats which it does correctly support is dBase.

How to open a dBase file in Systat

These instructions are for Systat version 10.2

1. Click the *File* menu.
2. Select *Open*; the *Open file* window will appear.
3. Towards the bottom of the window is a list titled *Files of Type*. Select 'dBase files'.

4. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
5. Click the *Open* button.

See also:

- Selecting cells, columns or rows
- Copying and saving data
- The save data window

Transferring data to Statistica

Introduction

Statistica is a popular statistical analysis program. Transferring data from ANY-maze to Statistica is a little complex, as all the possible methods involve some degree of compromise.

If your data includes repeated measures, you should use the options to Show repeated measures in columns, as this is the format that Statistica expects.

Copy and paste

Copying data is probably the best way to transfer data to Statistica. However, you should note that if you copy the entire ANY-maze spreadsheet (i.e. without selecting anything), then when you paste it into Statistica, the column titles will appear as the first case. You can avoid this by either deleting this case or by selecting all the rows in the Data page spreadsheet before copying them - see Selecting cells, columns or rows for details.

Another thing to be aware of is that by default, Statistica creates *New* data tables with just 10 variables and 10 cases (i.e. 10 columns and 10 rows). If you paste more data than this into a new table, the excess columns and/or rows are simply ignored. To overcome this, you should add enough variables and cases to accommodate all your data before you paste. In fact, adding too many rows or columns isn't a problem, so it's often easiest to simply add, say, 100 variables and 1000 cases.

Saving data as comma (CSV) or tab (TXT) separated files

Statistica can read comma (CSV) or tab (TXT) separated text files, so saving data in either format and then opening the file in Statistica is another way to transfer data.

However, this doesn't work as well as you might expect, because Statistica truncates any text data to 8 characters. For example, if you have two treatments called 'Compound ABC 1mg/kg' and 'Compound ABC 2mg/kg', they will both be chopped off at 'Compound' - in fact Statistica will call one of them 'Compound' and the other 'Compoun2', but knowing which is which is still rather difficult. Nevertheless, if you plan read data into Statistica then you can just take care to use short names.

Another more serious issue is that variable names (which are taken from column titles) are also truncated - this can lead to some very confusing names, and for this reason using dBase format files is better.

Saving data in a dBase file

Saving a Data page spreadsheet in dBase format and then reading it into Statistica is better than using text format, because ANY-maze will title the dBase fields with 8 character short 'variable names' which will help you differentiate the variables. However, despite the fact that the dBase format specifies column widths, Statistica will still insist on truncating strings to 8 characters, as it does when reading text files - again, keeping names short is the only viable solution to this problem.

How to open a CSV, TXT or dBase file in Statistica

These instructions are for Statistica version 6.0

1. Click the *File* menu.
2. Select *Import Data | Quick...*; the *File import* window will appear.
3. Towards the bottom of the window is a list titled *Files of Type*. Select either 'Text' or 'dBase', depending on the type of file you wish to open. (Note: for CSV files, select 'Text').
4. In the main part of the window, navigate to the file you saved from within ANY-maze and select it.
5. Click the *Open* button.
6. When you open a text file, the *Quick import from text* window will appear. If the file is a CSV file, then select *Comma* as the *Field separator*; otherwise select *Tab*. At the bottom of the window, select the option to *Get variable names from the first row*.
7. When you open a dBase file, the *Quick import from dBase* window will appear. You can just click *OK*.

See also:

- Selecting cells, columns or rows
- Copying and saving data
- The save data window

The I/O page

Unlike the pages on the left side of the ribbon bar, you can access the I/O page without having to open an experiment first.

Introduction

I/O stands for *input/output* and is a term used in ANY-maze in a rather generic way to refer to any input or output that doesn't rely on a camera or the experimenter. For example ANY-maze input/output includes such things as, detecting switch closures, sensing photobeam breaks, turning devices, such as shockers, on and off, etc.

Of course to do any of the above, ANY-maze needs to be physically connected to a switch, a photobeam, a shocker, etc. and this is achieved using an *input/output device*. ANY-maze supports various I/O devices, including some which we have specifically developed for use with the system - the full list can be found here.

Whatever I/O device you use you will almost invariably want to configure and test it before using it in an experiment, and this is what the I/O page is for. Here you can alter the configuration of a device, for example, specifying when ANY-maze should consider the device to be 'on', and you can test that the entire chain of connections from ANY-maze through your I/O device to the source (or target) apparatus is working correctly - for example, when ANY-maze activates a pellet dispenser, is a pellet actually dispensed?

- An introduction to I/O in ANY-maze
- Working with the I/O page
- Making I/O connections
- Testing I/O
- The ANY-maze Watchdog

See also:

- I/O devices supported by ANY-maze

An introduction to I/O in ANY-maze

Introduction

ANY-maze is primarily a video tracking system, but there are clearly benefits to incorporating other *inputs* with the tracking data. For example, a set of photobeams positioned part way up the sides of an open-field could be used to automatically count rearing. And better still, as the tracking tells ANY-maze where the animal is, the results can tell you not just how often the animal reared, but where it reared - in the centre, near the walls, etc.

Giving ANY-maze the ability to *output* signals can also be very useful. For example, an output might control a shocker, giving ANY-maze the ability to switch it on and off depending on conditions in the experiment - perhaps if the animal enters a certain zone, a shock should be administered.

The point is, that by using the input and output (I/O) facilities of ANY-maze, you can get new information about how animals are behaving, and you can create experimental designs that would otherwise have been difficult, or impossible to achieve.

Getting connected

Let's imagine you have a lever in a cage - the animal can press the lever and this will cause a switch to close... but how do you connect this equipment to your computer? The answer is that you will need some type of *I/O device* that will act as an interface between your computer and the switch - see figure 1.

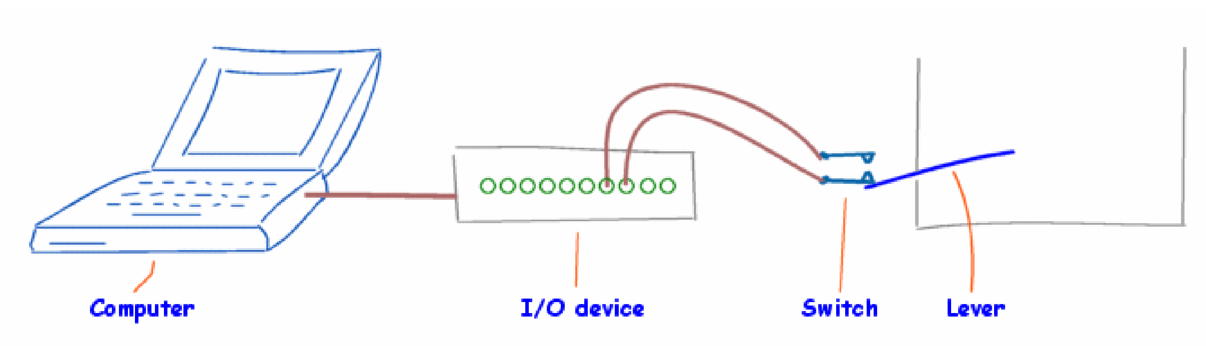


Figure 1. The I/O device acts as an interface between the switch and your computer.

ANY-maze supports a range of such I/O devices, some of which have been designed specifically for use with the system.

So you need an I/O device to connect anything to your computer - but how does it actually connect to the switch, shocker, photobeam or whatever? The answer, as you might expect, is 'it depends', but in many cases it's very simple, for example you can connect a simple push button switch to two of the *pins* on one of the *I/O ports* of an AMi-2 Digital interface and ANY-maze will be able to tell when the switch is pressed and when it's released. There's an entire section in this help dedicated to this subject of making I/O connections.

Configuring I/O devices and testing things out

Okay, let's assume you've now connected the switch in figure 1 to your computer via some kind of I/O device. The first thing you'll probably want to know is, does it work?

To find out you can use the I/O page. Here you can select the I/O device (and port) you connected the switch to, and see the state of the inputs - see figure 2.

Figure 2. The I/O page showing the state of the Input switches on an AMi-2 Digital interface - the 'Input switch on port 1' is active.

When you press the lever down you should find that the relevant input is shown active. So you try it and find that it works "backwards" - when the lever is *not* pressed the switch shows as active and when it is pressed it shows as inactive, what's happening? Well, you've encountered one of the vagaries of I/O interfacing - specifically, what counts as 'active'.

In fact this is easy to resolve using a second function of the I/O page, which is to configure your I/O devices. In this particular case you will need to alter the state that the device considers to be active so that it agrees with you - i.e. when the lever is *pressed* the switch is *active* and not vice versa.

Using I/O devices in an experiment

Finally your lever is working, when it's pressed it's 'active' and ANY-maze can detect this. At this point you'll want to actually include the lever in your protocol so that ANY-maze will score the lever presses during a test and report them in your results.

In this particular case, you would do this by adding an *On/off input* element to your protocol. I don't intend to go into all the details about this here, as full details can be found in the An introduction to on/off inputs topic.

But what about outputs, let's say we wanted to dispense a pellet each time the animal pressed the lever, how would that be done? In this case we would obviously need to connect the pellet dispenser via an I/O device to the computer and configure and test it using the I/O page. Then, to include the pellet dispenser in an experiment, we would need to add an *Output switch* into the protocol and finally add a *Procedure* to actually react to level press by switching the pellet dispenser on.

The different types of I/O supported by ANY-maze

As well as *On/off inputs* and *Output switches,* ANY-maze supports 10 other types of I/O, you'll find information about them all in these topics:

- An introduction to on/off inputs
- An introduction to signals
- An introduction to sensors
- An introduction to rotary encoders
- An introduction to movement detectors
- An introduction to output switches
- An introduction to speakers
- An introduction to analogue outputs
- An introduction to temperature controllers
- An introduction to lighting controllers
- An introduction to syringe pumps

- An introduction to shockers

Wrapping things up

In summary, ANY-maze's I/O facilities allow you to connect other equipment to your computer and score and/or control it from within the program. An I/O device is required to make the physical connection and the I/O page can be used to configure and test things. To actually use the equipment in an experiment, you need to add elements to the protocol, such as *On/off inputs, Output switches* and *Procedures.*

See also:

- Making I/O connections
- Working with the I/O page
- I/O devices supported by ANY-maze
- An introduction to procedures

Working with the I/O page

Introduction

The I/O page in ANY-maze is designed to let you do two things, configure the I/O devices connected to your computer and to test them. It's important to understand that this page is not actually used when performing an experiment, as the I/O functions are then all controlled through your protocol - rather it provides a place to get your I/O set up before you begin testing your animals.

- Setting up the I/O devices you want to use with ANY-maze
- Configuring individual devices
- Renaming devices
- Renumbering devices
- Testing devices

Setting up the I/O devices you want to use with ANY-maze

ANY-maze works with a wide range of I/O devices and will, *generally*, detect them automatically when it starts-up - it will then begin using any that it finds. But what if you have a device connected to your computer which you *don't* want ANY-maze to use? And what about devices which ANY-maze doesn't detect automatically - devices which connect to a serial port, syringe pumps for example, can't be detected automatically - how will ANY-maze know they're there?

These issues are addressed using the I/O ports and devices set-up window shown in figure 1. Here you can specify which ports and devices ANY-maze should use and, in the case of serial ports, what devices are attached to them.

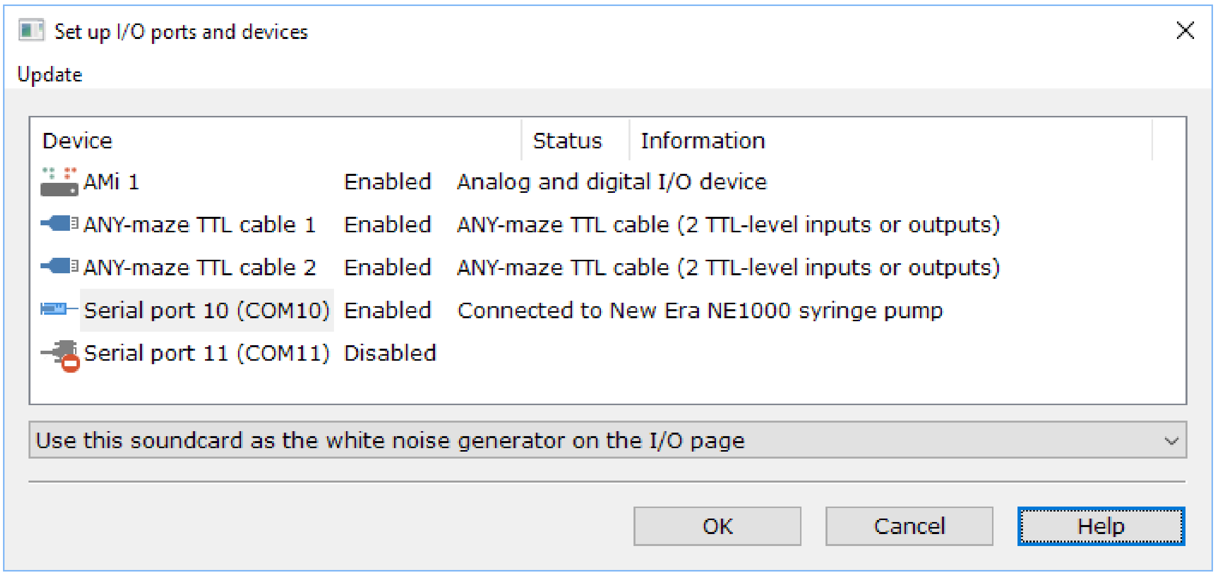


Figure 1. The I/O ports and devices set-up window used to specify what ports and devices ANY-maze should use.

The *I/O ports and devices set-up window* is accessed by clicking the *Setup devices* button shown in the ribbon bar when the I/O page is selected; the window is described in detail here.

Configuring individual devices

Most of the I/O devices supported by ANY-maze are, to some extent or other, configurable. For example, the ANY-maze TTL cable just includes two TTL ports, but should these ports be *inputs* or *outputs?* Well, this is something you can configure for yourself.

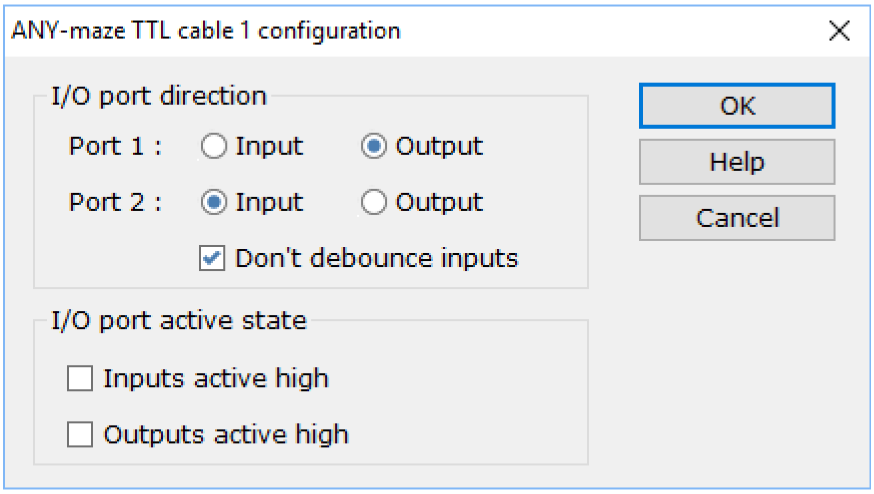


Figure 2. The ANY-maze TTL cable configuration window

Clearly, the configuration of the ANY-maze TTL cable is quite simple, but more complex devices, such as the ANY-maze interface (AMi), have many configuration options - for example, you each of the 16 AMi General purpose I/O ports can be configured to work in many different ways.

To actually configure a device, you should first select the device in the list on the left-side of the I/O page, and then click the *Configure this device* button in the ribbon bar. This will open the device's configuration window. Full details about each device's configuration options can be found in the relevant device's section of the I/O devices help.

In some cases, when you select a device on the I/O page, the ribbon bar may include a *Configure all these devices* button. As the name implies, selecting this button will configure *all* of the selected type of devices to have identical configurations. This can be a real time saver when, for example, you have eight identical devices that you'll be using in your experiment.

Renaming devices

Many devices have rather technical names which make it difficult to remember which is which. However, by selecting a device in the list on the I/O page and then clicking the *Rename this device* button, you can change the name to something more meaningful. For example, you may have some Skinner boxes that contain levers and pellet dispensers which are connected to a device called 'AMi-1', Changing the device's name to 'Skinner box interface' would make it much clearer what this device is for.

Renumbering devices

ANY-maze automatically numbers I/O devices, so for example, if you have four OPAD cages connected to your computer they will be numbered 1 to 4. This is fine, but imagine what might happen if you have just set up eight OPAD cages in your lab: you've carefully arranged the cage on a bench so they're in a row and you then turn everything on; ANY-maze will recognise the cages and will number them 1 to 8, but, there's no guarantee it will number them in the same order as you've put them on the bench (indeed it's very unlikely), so you'll probably end up with them numbered something like 4,1,8,3,6,2,5,7. Of course this won't stop them from working, but it would be much better if you could renumber them to be in order.

In fact, you can usually change a device's number using its configuration window, but this means you'd have to renumber each device individually, instead you can use ANY-maze's *renumber* function. This function isn't available for all devices, but when a device supports it (OPAD does, for example) then you will see a button in the ribbon bar like the *Renumber OPAD cages* button. In the case of OPAD, Selecting this button opens the Renumber OPAD cages window which guides you through the process of quickly renumbering all your OPAD cages at once (essentially you just push a button on the front of each cage, the first cage whose button you press will be cage 1, the second one, cage 2, and so on). Other devices work in a similar way.

See also:

- An introduction to I/O in ANY-maze
- I/O devices supported by ANY-maze
- Making I/O connections

Setting up the I/O devices to use with ANY-maze

To access the I/O devices set-up window, select *Setup devices* button in the ribbon bar on the I/O page.

Introduction

ANY-maze works with a wide range of I/O devices and will, generally, detect them automatically when it starts-up - it will then begin using any that it finds. But what if you have a device connected to your computer which you *don't* want ANY-maze to use? And what about devices which ANY-maze doesn't detect automatically - for example, devices such a syringe pumps, which connect to a serial port can't be detected automatically - how will ANY-maze know they're there?

These issues are what the *I/O ports and devices set-up window* addresses; it allows you to specify which ports and devices ANY-maze should use and, in the case of serial ports, what devices are attached to them - see figure 1..

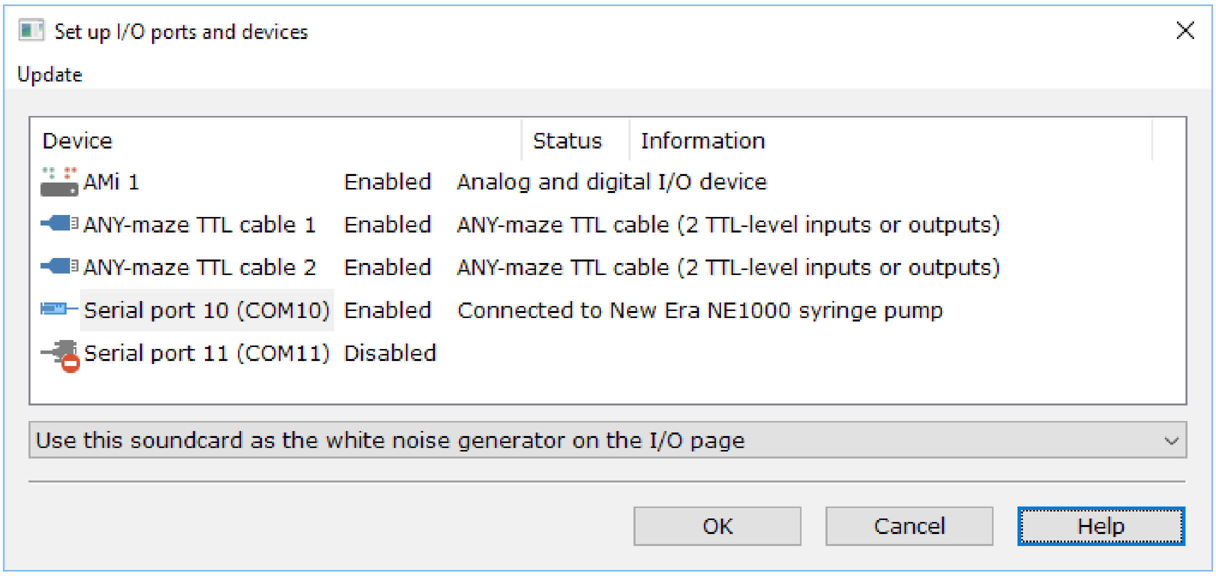


Figure 1. The I/O ports and devices set-up window.

Choosing which devices to enable

By default serial ports (and if you have any, parallel ports) will be disable for use in ANY-maze, while all other I/O devices will be enabled. To alter this you simply need to select the relevant device in the list and then alter its state in the drop-down list at the bottom of the window.

Specifying the device connected to a serial port

In the case of a serial port, you will find that the drop-down list doesn't simply allow you to enable or disable the port, rather it shows a list of all the serial devices that ANY-maze supports and you can choose which of them is physically connected to your port. In fact the device needn't be connected at the time, but what you're telling the system is that it can assume that the port will either have the specified device attached to it, or no device at all. ANY-maze will then detect when the device is physically present.

Configuring individual devices

Note that this window is not used to configure the individual devices you have connected (for that you should see the information in the Working with the I/O page topic), rather its purpose is simply to specify what devices will and will not be available for ANY-maze to use.

The Update menu

As can be seen in Figure 1, above, the *I/O ports and devices set-up window* includes an Update menu. This contains options that are used to update the Ugo Basile Fear Conditioning system from the original version 1 to version 2.

See also:

- Working with the I/O page

Renaming an input/output device

To access the Rename I/O device window, select the relevant device on the I/O page and then click the Rename device link in the Actions panel.

Introduction

Most of the I/O devices supported by ANY-maze have rather technical name which make it hard to remember exactly what they're connected to. To overcome this you can rename a device, for example you might choose to alter the device called "AMi 1" to be "Step-down boxes", because that's the equipment it physically connects to.

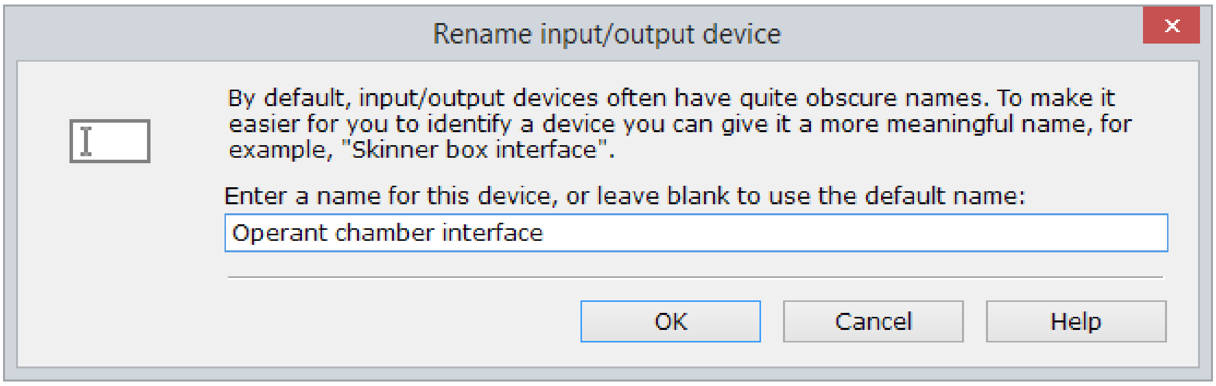


Figure 1. The Rename I/O device window.

Changing a device's name

To change the name of device you should simply delete the current name and type in the name you want to use. There's a limit of 63 characters, but otherwise you can enter whatever you like. Note, that this does mean you can give two devices the same name if you want to.

To restore the default name, simply delete the entry from the name field, so it's blank, and then click OK.

See also:

- Working with the I/O page

Making I/O connections

Introduction

ANY-maze can connect to wide range of different things - levers, photobeams, shockers, pellet dispensers, lasers, running wheels, etc. As explained in the introduction to I/O, to actually connect these things you'll need some sort of *I/O device*, which acts as the interface between them and your computer (after all, there's nowhere on your computer to plug in a pellet dispenser).

In this section I'll describe how you would physically connect these different items to ANY-maze. In most cases I'll describe how the connection would be made using the AMi-2 family of devices as this usually provides the simplest solution, but I have also included details of how to make connections to other more generic digital I/O devices such as the Switch and Sense 8/8.

How to connect to:

- Levers, switches, push buttons, etc.
- Photobeams and nose-poke holes
- Photobeam arrays
- Speakers
- Pellet dispensers
- Lights
- LEDs
- Optogenetic lasers
- Shockers
- Motors (for example, to control doors), pumps, etc.
- Syringe pumps
- Running wheels
- Light sensors
- Temperature sensors
- Movement detectors
- Voltage inputs from other equipment - for example, gas concentrations, ECG, EEG
- Voltage outputs to other equipment
- Digital inputs from other equipment
- Digital outputs to other equipment

This is not a definitive list - the ANY-maze interface is extremely flexible and can interface to a wide range of items, if you want to connect to something not listed here, just contact ANY-maze technical support (techsupport@anymaze.com) and we'll be glad to help.

Connecting levers, switches, push buttons, etc.

Details

Levers, switches, push buttons, etc. are all essentially the same - they consist of two terminals which are connected together (or sometimes separated) when the device is 'activated'. So all we need to do here is connect these two terminals to an I/O device.

The AMi-2 Digital interface switch input ports are perfect for this, you simply have to connect the two terminals to pins 1 and 2 (odd numbered ports) or pins 3 and 4 (even numbered ports) - see figure 1.

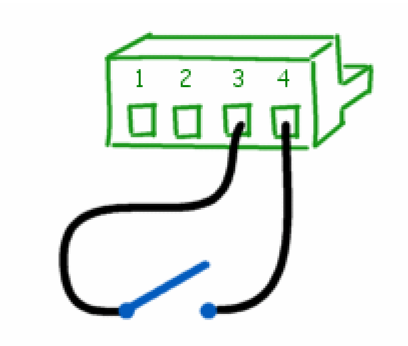


Figure 1. Any two-terminal switch simply connects between pins 1 and 2, or 3 and 4, of an AMi-2 Digital interface switch input port.

As well as making this connection, you will also need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used for a switch input. You should then test that the switch is working correctly using the I/O page.

To actually use the switch in an experiment, you'll need to add it to the protocol as an on/off input.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface switch input ports
- Configuring the AMi-2 Digital interface
- An introduction to on/off inputs
- Debouncing inputs

ANY-maze help topic H0813

Connecting photobeams and nose-poke holes

Details

Photobeams usually consist of an infra-red LED (the emitter) and an infra-red detector. These are arranged so that the LED shines a beam of light onto the detector. When the animal breaks the beam, the light no longer reaches the detector, and this is sensed by the I/O device.

Nose-poke holes are generally just small holes with a photobeam pre-installed across them, so when the animal pokes its nose into the hole, the beam is broken.

The AMi-2 Digital interface photobeam ports can be connected directly to a photobeam made specifically for AMi. In fact, photobeams from other manufacturers may also work, but you should contact ANY-maze technical support (techsupport@anymaze.com) and ask us to check this for you, before making such a connection.

Connecting a photobeam to a GPIO port is very easy - you simply have to connect the LED to pins 1 and 2 and the detector to pins 3 and 4 - see figure 1 below.

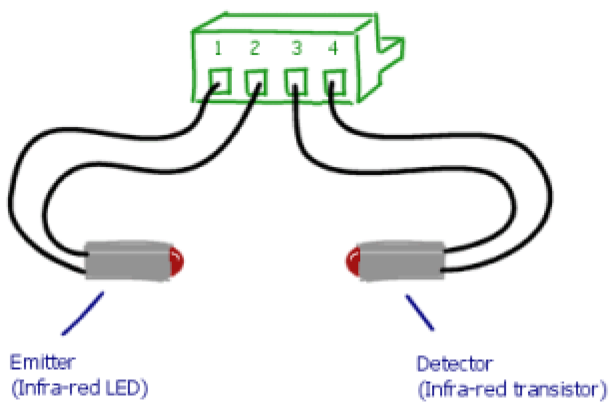


Figure 1. A photobeam connected to an AMi-2 Digital interface photobeam port.

The LED needs to be connected the right way round in order to work, specifically you should connect the cathode of the LED to pin 1 and the anode to pin 2. If you connect the LED backwards it won't be damaged, it just won't work, so if you're not sure how to connect it, try one way and then try the other.

The photobeam detector is an infra-red photo-transistor and, again, it has to be connected the right way round; in this case the collector connects to pin 3 and the emitter to pin 4. Like the LED, connecting the detector the wrong way round won't damage it, it just won't work - so if you're not sure which way round to connect it try one way and then try the other.

After connecting the beam, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used for a *photobeam*. You should then test that the beam is working correctly using the I/O page.

To actually use the photobeam in an experiment, you'll need to add it to the protocol as an on/off input.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface photobeam ports
- Configuring the AMi-2 Digital interface
- An introduction to on/off inputs

Connecting photobeam arrays

Details

Photobeams arrays consist of a number of photobeams mounted in parallel. A typical use for an array is to detect rearing; here the array is positioned horizontally above the apparatus floor such that the animal will only break beams in the array when it rears.

The AMi-2 Digital interface photobeam array port can be connected directly to photobeam arrays made specifically for AMi. These arrays are available in a number of sizes, including 40cm x 40cm and 100cm x 100cm. Unfortunately, arrays from other manufacturers won't work with ANY-maze.

To connect the array you just need to plug it in it to a port on the AMi-2 Digital interface (the array is supplied with a connector already wired-up).

After connecting the array, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used for a *photobeam array*. You should then test that the array is working correctly using the I/O page.

To actually use the array in an experiment, you'll need to add it to the protocol as an on/off input.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface photobeam array ports
- Configuring the AMi-2 Digital interface
- An introduction to on/off inputs

Connecting speakers

Details

ANY-maze can play sounds through the AMi-2 Audio interface, which supports up to six independent speakers. To connect a speaker you just need to connect wires from the speakers terminals to the two screw-terminals of an AMi-2 Audio interface speaker port (the order doesn't matter).

After connecting the speaker, you should test that it is working correctly using the I/O page.

To actually use the speaker in an experiment, you'll need to add it to the protocol as a speaker output. You'll may also want to add a procedure, as you'll then be able to control the speaker *during* the tests - for example, starting or stopping play-back, changing what it plays, or adjusting the volume.

See also:

- The AMi-2 Audio interface
- Setting up the AMi-2 Audio interface
- Configuring the AMi-2 Audio interface
- An introduction to speakers

Connecting pellet dispensers

***IMPORTANT:** Not all pellet dispensers work as described here, so you should carefully check your device's manual before connecting it.*

Details

Most pellet dispensers include an 'operate' connection which, when connected to the dispenser's power cause it to dispense a pellet. Usually the operate signal needs to last for just a short duration, for example half a second, otherwise multiple pellets are dispensed.

So here what we need is something like a *computer controlled switch*, connected between the dispenser's *operate* and *power* inputs - the computer can then briefly close this 'switch' to dispense a pellet.

The AMi-2 Relay interface includes eight *computer controlled switches* just like this (they're called relays). So, to connect a pellet dispenser you simply need to connect its *operate* line to one input on an AMi-2 Relay interface port and its *power* line to the other, as shown in figure 1, below.

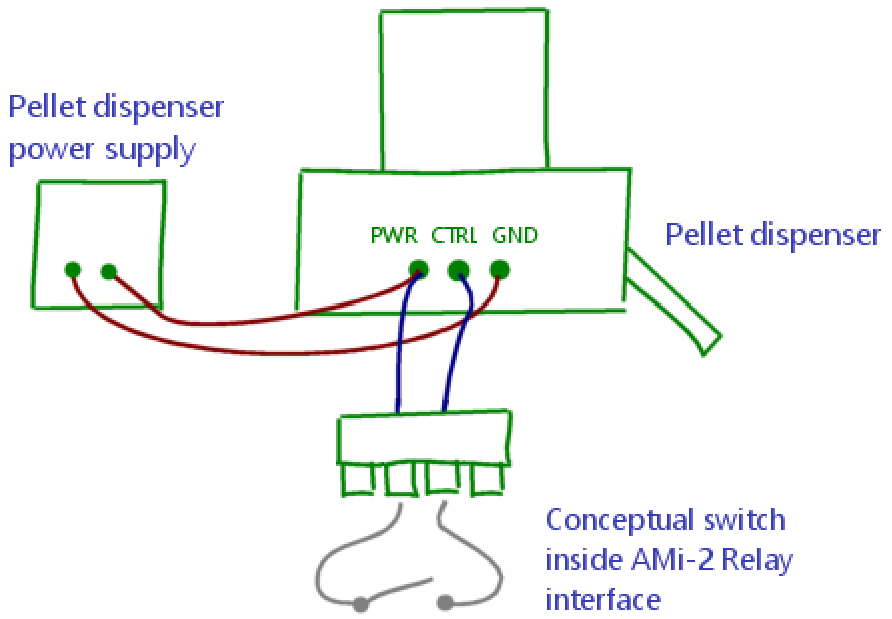


Figure 1. Controlling a pellet dispenser through an AMi-2 Relay interface. Note that not all pellet dispensers connect in this way - you should check how your device works before connecting it.

After connecting the pellet dispenser, you should test that it is working correctly using the I/O page. Here you should quickly turn the relevant switch output port on and then off, a pellet should be dispensed.

To actually use the pellet dispenser in an experiment, you'll need to add it to the protocol as a switch output. You'll find that switch outputs can be set to automatically switch off a set period after they are switched on - so you can use this to have the switch turn off half a second after it comes on, so as ensure just one pellet is dispensed (the exact pulse duration should be in the pellet dispensers manual, or you can determine it through experimentation). You'll probably also want to add a procedure, as it's the procedure which will actually turn the pellet dispenser on (so it dispenses a pellet) during the tests.

See also:

- The AMi-2 Relay interface
- An introduction to Output switches

Connecting lights

Details

To switch a light on and off, what we need is something like a *computer controlled switch*. >The AMi-2 Relay interface includes eight of them (they're called relays).

So, to connect a light you need to build the same sort of circuit that you'd use if the light was going to be controlled by a manual switch, but connect to a *AMi-2 Relay interface port* instead - see figure 1. Note you can connect either wire to either pin 2 or pin 3 of the port (all the port does is connect the two pins together when the port is switched on).

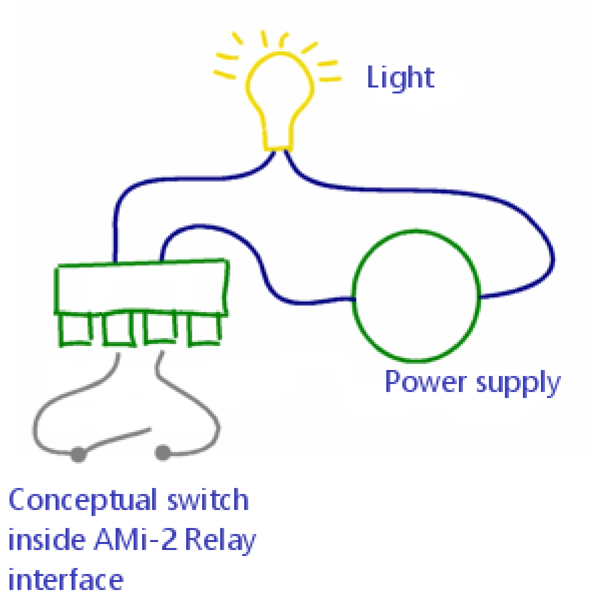


Figure 1. Each output of the AMi-2 Relay interface acts like a simple on/off switch. Here closing the switch will cause the light to turn on.

After connecting a light, you should test that it is working correctly using the I/O page.

To actually use the light in an experiment, you'll need to add it to the protocol as a switch output. You'll probably also want to add a procedure, as it's the procedure which will actually turn the light on and off during tests.

See also:

- The AMi-2 Relay interface
- An introduction to Output switches

Connecting LEDs

Details

To switch an LED on and off, you could use a similar circuit to the one used to control a light, (although you would need to include a current-limiting resistor), but provided the LED does not need more than 30mA current (and most don't), then you could connect it directly to a The AMi-2 Digital interface TTL output port (which would remove the need for an external power source).

Unlike incandescent bulbs, which can be connected *directly* to a power source (such as battery), LEDs need to include a *current-limiting resistor,* which, as the name implies, ensures that the current flowing through the LED is limited to an amount the LED can handle. This current limit is called the 'Forward current' (not to be confused with the 'Peak forward current') and will be reported in the LED's datasheet. Another value that the datasheet will report is the 'Forward voltage' and armed with these two values you can use ohm's law to calculate the required current-limiting resistor:

Resistor = (Power source voltage - LED Forward voltage) / Forward current

When connecting to an AMi-2 Digital interface TTL output port, the power source voltage is 5V and the Forward current should be the smaller of the LED's forward current and 30mA (which is the maximum the port can provide). So, typical values might be R = (5 - 1.8) / 0.025 = 128, so you'd need a 128 ohm resistor. Resistors are very cheap and can be found on-line or at shops like RadioShack. If you can't get the exact value required just get one that has a slightly higher resistance, perhaps 130 ohms in our example.

You'll actually connect the LED to pins 1 and 2 (odd numbered ports) or pins 3 and 4 (even numbered ports) of an AMi-2 Digital interface port, taking care to connect it the right way round (if you connect it the wrong way round, you won't damage it, it just won't work).

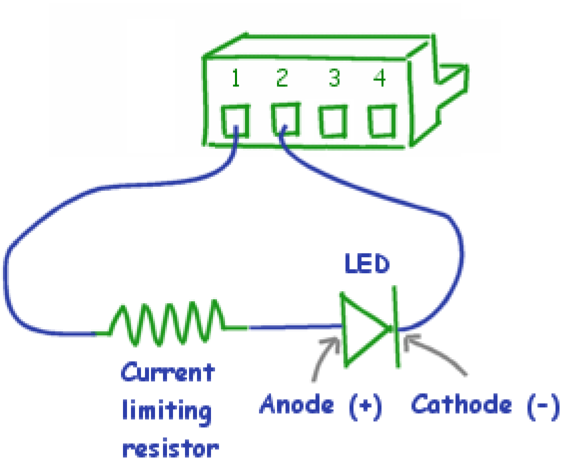


Figure 1. A circuit for connecting an LED to an AMi-2 Digital interface TTL output port.

After connecting the LED, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used to switch an LED on and off - specifically you'd set the port to be a *TTL output*. You should then test that the beam is working correctly using the I/O page, where you'd want to select the relevant ANY-maze interface's *Output switch* ports.

To actually use the LED in an experiment, you'll need to add it to the protocol as an output switch. You'll probably also want to add a procedure, as it's the procedure which will actually turn the LED on and off during tests.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface TTL output ports
- Configuring the AMi-2 Digital interface
- An introduction to Output switches

Connecting optogenetic lasers

Details

Many laser used in optogenetics include a 'remote control' input, which can be used to switch the laser on and off. These usually use TTL signals, which are a type of signal that the AMi-2 Optogenetics interface ports output.

To connect to an optogenetic laser's remote control input you simply need to connect the *remote control* line to one of the AMi-2 Optogenetics interface ports, using a BNC cable.

After connecting the laser, you should test that it is working correctly using the I/O page.

To actually use the laser in an experiment, you'll need to add it to the protocol as a laser controller. You'll find the laser controllers include options for outputting a pulse train, which are usually what you'd want to do with an optogenetics laser. You'll probably also want to add a procedure, as it's the procedure which will actually turn the laser on and off during tests.

See also:

- The AMi-2 Optogenetics interface
- Configuring the AMi-2 Optogenetics interface
- Connecting the AMi-2 Optogenetics interface to a laser and testing it
- Using the AMi-2 Optogenetics interface in tests

Connecting shockers

Introduction

There are two ways to control a shocker in ANY-maze, either using the ANY-maze shocker cable, or using one of the AMi-2 family of devices.

Using the ANY-maze shocker cable

The ANY-maze shocker cable is designed to control the San Diego Instruments programmable shocker. The cable simply plugs into a USB port on your computer and into a port on the rear of the shocker. It allows ANY-maze to switch the shocker on and off *and* to control the level of shock that is delivered.

After connecting a shocker cable you will want to check it is working correctly using the I/O page.

To actually use the shocker in an experiment, you'll need to add a shocker element to your protocol and also a procedure too, as it's the procedure which will actually switch the shocker on and off during tests.

Using an AMi-2 family device

If you don't own a San Diego Instruments programmable shocker, you won't be able to use the ANY-maze shocker cable, however, most shockers include a 'remote control' input which can be used to turn the shocker on and off and the AMi-2 family of devices can usually be connected to these inputs.

In fact there are three types of 'remote control' input commonly found on shockers - TTL inputs, switch inputs and opto-isolated inputs:

- TTL inputs can be connected to an AMi-2 Digital interface TTL ouptut port, the signal should connect to pin 1 and 2 (odd numbered ports) or pin 3 and 4 (even numbered ports). After connecting the shocker, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used to output a TTL signal - specifically you should set the port to be a *TTL output*.
- Switch inputs simply require that their two terminals are connected together in order for the shocker to deliver a shock. This type of input can be connected directly to pins 2 and 3 of a AMi-2 Relay interface port (it doesn't matter which way round you make the connection - so either terminal on the shocker can be connected to either terminal on the AMi-2 Relay interface port.)
- Opto-isolated inputs can be connected to an AMi-2 Digital interface too, but this is a

little more complicated, so we recommend you contact ANY-maze technical support (techsupport@anymaze.com) if your shocker has this type of input.

Once you have connected the shocker you should test that it is working correctly using the I/O page.

To actually use the shocker in an experiment, you'll need to add it to the protocol as an output switch. You'll need to add a procedure too, as it's the procedure which will actually switch the shocker on and off during tests at the appropriate times.

See also:

- The ANY-maze shocker cable
- The AMi-2 Digital interface
- Configuring the AMi-2 Digital interface
- The AMi-2 Digital interface TTL output ports
- The AMi-2 Relay interface
- An introduction to Output switches

Connecting motors

Details

Being able to have ANY-maze control a motor can be very helpful - for example, you might have doors in your apparatus which are controlled by small motors, so by being able to switch the motors on and off you can control which doors are open and therefore what parts of the apparatus the animal can access.

The AMi-2 Relay interface ports can be used to control low voltage motors - that is any motor that uses a voltage of 42V (AC or DC) or below and a current of 2A or below. To achieve this you simply need to connect the motor to the switch output port in the same way as you'd connect it to a manual switch, so something like the circuit shown in figure 1. By the way, if you are technically inclined, you may be aware that a motor is what's called an inductive load and that connecting this type of load to a relay requires some care - don't worry the ANY-maze interface switch output ports include a built-in 'snubber circuit' for exactly this reason.

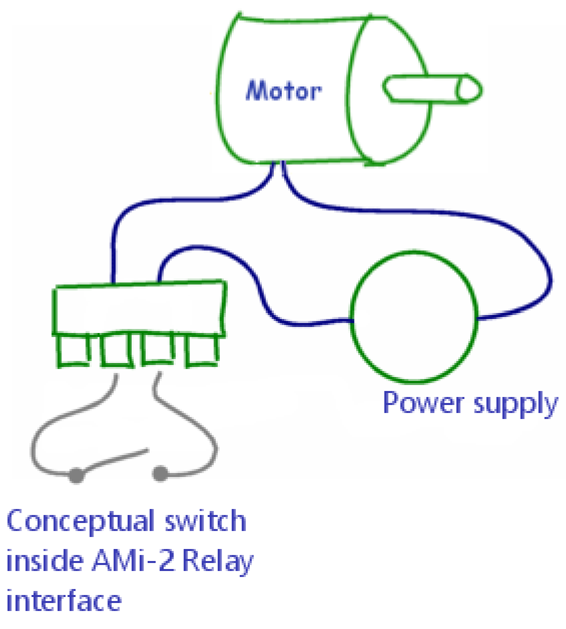


Figure 1. A simple circuit to connect a motor to an AMi-2 Relay interface port.
ANY-maze will then be able to turn the motor on and off.

Once you have connected the motor you should test that it is working correctly using the I/O page.

To actually use the motor in an experiment, you'll need to add it to the protocol as an output switch. Here you'll find that you can set up the output such that it turns on for a specific period of time and then turns off automatically. This can be very helpful if you just want the motor to run for a few seconds - as would be the case if it controls a door. You'll probably also want to add a procedure, as it's the procedure which will actually activate the motor during a test.

See also:

- The AMi-2 Relay interface
- Connecting the AMi-2 Relay interface to your apparatus and testing it
- An introduction to Output switches

Connecting syringe pumps

Details

There are two ways to connect to syringe pumps in ANY-maze. The first is to connect directly to the pump using a USB cable or a serial port; in this case ANY-maze sends commands to the pump which then executes them. This is the most flexible method and is supported by many pumps on the market. This type of connection is described in detail here.

The second method is to use a 'remote control' input on the pump; in this case ANY-maze will be able to switch the pump on and off, but it won't be able to alter its direction or speed. This type of connection is less common but is the only way to connect to the popular Razel pumps.

Most pumps which include a 'remote control' input, use TTL signals to control the pump. This type of input can be connected to an AMi-2 Digital interface TTL output port. Specifically the *signal* line should connect to pin 1 and the *ground* line to pin 2 (odd numbered ports) or to pins 3 and 4 (even numbered ports).

After connecting the pump, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used to output a TTL signal - specifically you should set the port to be a *TTL output*. You should then test that the pump is working correctly using the I/O page, where you'd want to select the relevant AMi-2 Digital interface's *Output switch* ports.

To actually use a syringe pump that is being controller through a Output switch port, you'll need to add it to the protocol as an output switch (NOT as a syringe pump). You'll probably also want to add a procedure, as it's the procedure which will actually turn the pump on and off during tests.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface TTL output ports
- Configuring the AMi-2 Digital interface
- An introduction to Output switches

Connecting running wheels

Details

Clearly, you can't actually *connect* to a running wheel, as wheels don't have any type of electrical connection, but you can have the wheel turn a device called a rotary encoder and then connect the *encoder* to an ANY-maze interface. An example of a rotary encoder is shown in figure 1.

Figure 1. A typical rotary encoder.

As you can see, the encoder has an axle and what it does is to 'encode' rotations of this axle; so, if you were to mount a running wheel on the axle you could see how many rotations it makes.

The specific type of encoder which can be used with ANY-maze is one which outputs a TTL level *quadrature* signal (they're sometimes called quadratic encoders). These encoders have 4 connections: Power (5V), GND, Signal A and Signal B. They can be connected to the AMi-2 Digital interface rotary encoder ports where each encoder will connect as follows:

- Pin 1 : Signal A
- Pin 2 : Power
- Pin 3 : Signal B
- Pin 4 : Ground

After making these connections, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used by a rotary encoder - specifically you should set the port to be a '*Rotary encoder*'. You will also need to specify the number of pulses per rotation that the encoder outputs - this information will be available in its datasheet.

You should then test that the encoder is working correctly using the I/O page, where you'll want to select the relevant ANY-maze interface's *Rotary encoder* ports. You may find that when you turn the running wheel clockwise ANY-maze sees it as turning anti-clockwise. If this happens then you can go back to the AMi-2 Digital interface configuration, select the port and set the option to *Invert the* *direction of this encoder.* Another common issue is to find that when you turn the encoder through a complete turn ANY-maze only sees a half turn (or sometimes it sees two turns). In these cases you again need to go back to the AMi-2 Digital interface configuration and there adjust the number of pulses per rotation - in the first case you should halve the number you originally entered, in the second case you should double it.

To actually use a running wheel mounted on a rotary encoder, you'll need to add it to the protocol as a 'Rotary encoder input'.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface rotary encoder ports
- Configuring the AMi-2 Digital interface
- An introduction to rotary encoders

Connecting light sensors

Details

The ANY-maze interface (version 1) includes a light-meter add-on which can be used to read ambient light levels. This light-meter simply connects to one of the ANY-maze interface's sensor ports.

After connecting a light-meter you should test that it is working correctly using the I/O page, where you'd want to select the relevant ANY-maze interface's *Light sensor* ports.

To actually use the light sensor, you'll need to add it to the protocol as a Sensor input.

See also:

- The ANY-maze interface (AMi)
- The AMi sensor ports
- An introduction to sensors

Connecting temperature sensors

Details

The ANY-maze interface (version 1) includes a temperature sensor add-on which can be used to read ambient temperatures. This temperature sensor simply connects to one of the ANY-maze interface's sensor ports.

After connecting a temperature sensor you should test that it is working correctly using the I/O page, where you'd want to select the relevant ANY-maze interface's *Temperature sensor* ports.

To actually use the temperature sensor, you'll need to add it to the protocol as a Sensor input.

See also:

- The ANY-maze interface (AMi)
- The AMi sensor ports
- An introduction to sensors

Connecting movement detectors

Details

The ANY-maze movement detectors are, in fact, ANY-maze photobeam arrays and they connect in exactly the same way - which is described here.

The only difference between a photobeam array and a movement detector is how it is used within ANY-maze, a photobeam array just reports when any beam is broken, whereas a movement detector uses beam breaks to determine where the animal is and therefore how much it has moved. To use a photobeam as a movement detector you'll need to add it to the protocol as a movement detector input.

See also:

- The AMi-2 Digital interface photobeam array ports
- An introduction to movement detectors

Connecting voltage inputs from other equipment

Details

Many devices output a voltage to represent some value that they are detecting. For example, the CWE CapStar-100 analyses CO_2 levels and outputs 0 volts for 0% CO_2 and 1 volt for 10% CO_2. Alternatively a sound level meter might output 0.3 volts for 30 decibel sound and 1.3 volts for 130 decibels. Some telemetry systems also output a voltage representing whatever the telemetered sensor is detecting, for example, blood pressure or biopotentials.

You can connect these sorts of devices to an AMi-2 Analogue interface, allowing ANY-maze itself to read their values - thus the system would be able to determine CO_2 concentrations, sound levels, or the animal's blood pressure, during a test (and don't forget that this information could be analysed for different zones and across time).

Voltage outputs of this type connect to the AMi-2 Analogue interface input ports. These ports use screw terminal blocks and you should connect the voltage signal to pin 1 and the ground to pin 2.

After making these connections, you will need to Configure the AMi-2 Analogue interface to tell it two things: the range of voltages it can expect at the input and the sampling rate it should use. You should choose a voltage range which just covers all the expected values - for example, in the case of the CO_2 detector described above a voltage range of 0-3V would be good (the device having a maximum voltage of 1V). The sampling rate determines how often the ANY-maze interface will read the voltage and this will very much depend on how quickly the value can change. For example, for CO_2 detection a rate of 10 samples per second would probably be more than adequate, but for sound levels, which could change very quickly, you might want to sample at 100 samples per second. Note that higher sampling rates generate more data (and hence larger files), so avoid using a rate that's unnecessarily high.

After configuring the port you should test that it is working correctly using the I/O page, where you'd want to select the relevant AMi-2 Analogue interface's *Analogue input* ports. This will display a chart showing the *voltage* being read from the port.

To actually use the voltage data, you'll need to add a signal input to your protocol. Here you will find that you can specify conversion data, which means that ANY-maze will then report the actual value the voltage represents (such as % CO_2) rather than the voltage itself.

See also:

- The AMi-2 Analogue interface
- The AMi-2 Analogue interface input ports
- Configuring the AMi-2 Analogue interface

- Connecting the AMi-2 Analogue interface to your equipment and testing it
- Using the AMi-2 Analogue interface in tests

Connecting voltage outputs to other equipment

Details

Some devices include a voltage input which can be used to control them. The AMi-2 Analogue interface includes two ports which can output voltages, exact;ly for this type of situation. The AMi-2 Analogue interface output ports are both found on the same connector: pin 1 is the signal and pin 2 is ground, of port 1; while pin 3 is the signal and pin 4 is ground, of port 2.

After making the connections you will need to add an analogue output to your protocol. You probably also want to add a procedure as it will be the procedure that controls the output during a test.

After configuring the port you should test that it is working correctly using the I/O page, where you'd want to select the relevant ANY-maze interface's *Analogue output* ports. Here, you can alter the output voltage and check that the output really does change, using a voltmeter.

See also:

- The AMi-2 Analogue interface
- The AMi-2 Analogue interface output ports
- Connecting the AMi-2 Analogue interface to your equipment and testing it
- Using the AMi-2 Analogue interface in tests

Connecting digital inputs from other equipment

Details

Some devices provide digital outputs to indicate their state. For example, a proximity meter might output a high TTL signal whenever something is close to it.

The AMi-2 Digital interface TTL input ports can be used to connect to TTL (or CMOS) signals of this type. The signal should be connected to pin 1 and ground to pin 2 (odd numbered ports) or pins 3 and 4 (even numbered ports).

After connecting the signal, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used to read a TTL input signal - specifically you should set the port to be a *TTL input* (note that the signal is an *output at the device* and an *input* at the ANY-maze interface). You should then test that the input is working correctly using the I/O page, where you'd want to select the relevant AMi-2 Digital interface's *Input switch* ports.

To actually use the input in an experiment, you'll need to add it to the protocol as an on/off input.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface TTL input ports
- Configuring the AMi-2 Digital interface
- An introduction to on/off inputs

Connecting digital outputs to other equipment

Details

The AMi-2 Digital interface TTL output ports can be used to output digital TTL signals to other devices. For example, such an output might be used to switch another device on and off, or it might be used to synchronise the other device with the start of a test in ANY-maze.

When an ANY-maze interface General purpose I/O port is being used as a digital output, pin 1 is the signal and pin 2 is ground (odd numbered ports), or pin 3 is the signal and pin 4 is ground (even numbered ports).

After making the connections, you will need to configure the AMi-2 Digital interface so that it knows that the port you've connected to is being used as a TTL output - specifically you should set the port to be a *TTL output*. You should then test that the output is working correctly using the I/O page, where you'd want to select the relevant AMi-2 Digital interface's *Output switch* ports.

To actually use the output in an experiment, you'll need to add it to the protocol as an output switch. You'll probably also want to add a procedure, as it's the procedure which will actually turn the output on and off during tests.

See also:

- The AMi-2 Digital interface
- The AMi-2 Digital interface TTL output ports
- Configuring the AMi-2 Digital interface
- An introduction to Output switches

Connecting using generic digital I/O devices

Details

We generally recommend that you use the ANY-maze interface to connect inputs and outputs to ANY-maze, but the system does support a range of other (mainly legacy) generic devices too:

- The RTV-24 digitiser's I/O ports
- The Switch and Sense 8/8 (legacy)
- I/O ports on Axis network cameras (legacy)
- I/O ports on Picolo series digitisers (legacy)
- Devices that connect to the computer's serial port (legacy)
- The computer's parallel (printer) port (legacy)

Making connections to these devices is usually a little more complex than using the ANY-maze interface and I've described what's required in the following topics:

- Connecting an on/off input switch
- Debouncing inputs
- Connecting a photobeam
- Turning something on and off

Connecting an on/off input switch

The details given in this topic provide guidance on connecting a switch to generic I/O devices such as the Switch and Sense 8/8. Connecting a switch using the ANY-maze interface is much simpler - refer to the ***ANY-maze interface*** *topic for further details.*

Introduction

The purpose of this topic is to describe the process of attaching an on/off switch (like a light switch) to ANY-maze via some type of I/O device. Thus, ANY-maze will be able to sense when the switch is on and when it's off.

Although this sounds as if it should be trivially simple, you'll find that it can be a little more complex than you might imagine.

- Types of input - TTL and opto-coupler
- Connecting to TTL inputs
- Connecting to opto-couplers

Types of input - TTL and opto-coupler

The "inputs" found on the generic I/O devices supported by ANY-maze fall into two categories, *TTL level inputs* and *opto-coupler inputs* - but what do these names actually mean?

TTL inputs have this name because they connect directly to a TTL 'chip' (for the curious, TTL stands for Transistor-transistor logic). In practical terms this means that the chip will be able to tell when the input is at a "logic high" (around 5 volts DC) or a "logic low" (around zero volts DC) - this is the 'digital' part of computing, everything is *high* = 1 or *low* = 0. We'll come back to TTL in a minute, but what about those opto-couplers.

Opto-couplers, as their name implies, 'couple', i.e. connect, your input to the I/O device using light. The big benefit of an opto-coupler is that it *isolates* the I/O device, and hence the computer connected to it, from the equipment you attach. There are a number of technical benefits to this isolation, but they're beyond the scope of this help - suffice to say that quite a few I/O devices use opto-couplers so you may need to connect to them.

Connecting to TTL inputs

As mentioned above a TTL input should either be held at around 5V or 0V, so our switch needs to connect either 5V or 0V to the input - using a simple on/off switch this suddenly doesn't seem so easy after all.

Fortunately help is at hand, in the form a 'pull-up' resistor. Again, I won't get too technical, but if you look at the circuit in figure 1 you'll see that the resistor connects the TTL input to 5V so the input will be at 5V and the TTL chip will sense a logic 1. Now when the switch is closed the TTL input is connected to 0V (through the switch) and so the chip will sense a logic 0 - bingo, we have our switch connected.

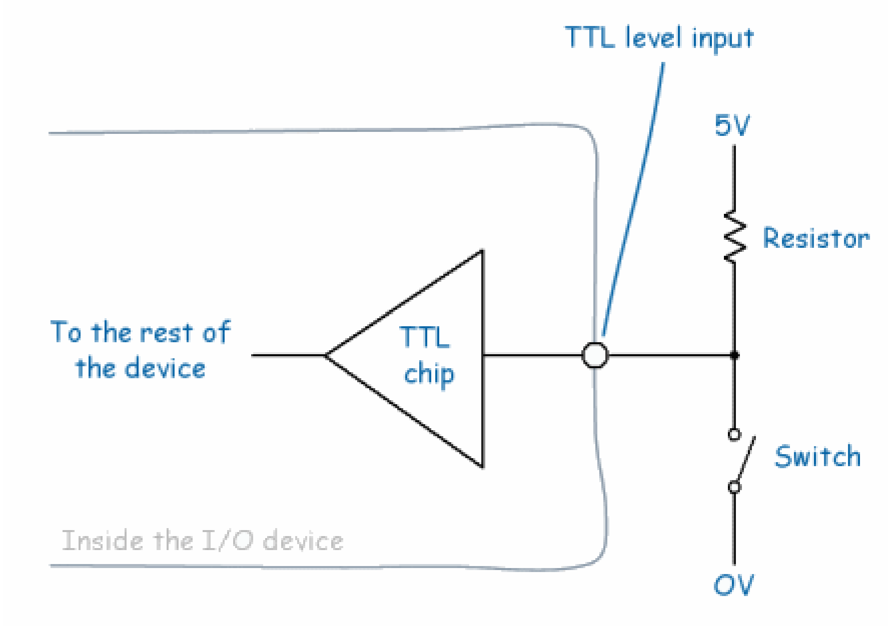


Figure 1. A circuit to connect an on/off switch to a TTL input

But this leaves a couple of questions - first why do we need the resistor, couldn't we connect directly to 5V, and second where are these 5V and 0V coming from?

The answer to the first question is that without the resistor we would be shorting-out the 5V and 0V lines whenever the switch was closed, this would create a very big current and something would go wrong. The resistor reduces this current to something small (around 1mA) so we can ignore it.

The answer to the second question, where are these 5V and 0V coming from, is that our I/O device will need to provide them for us. In fact I/O devices that have TTL inputs almost always have *pull-up* resistors built into them, so we won't actually need either a resistor or a connection to 5V. All we'll need is a connection to 0V, usually called GND (ground), and you'll always find a GND output on a connector with TTL level inputs.

In summary to connect an on/off switch to a TTL input you will need to use a pull-up resistor, but most I/O devices have these built in so you can just use a simple circuit like the one shown in figure 2.

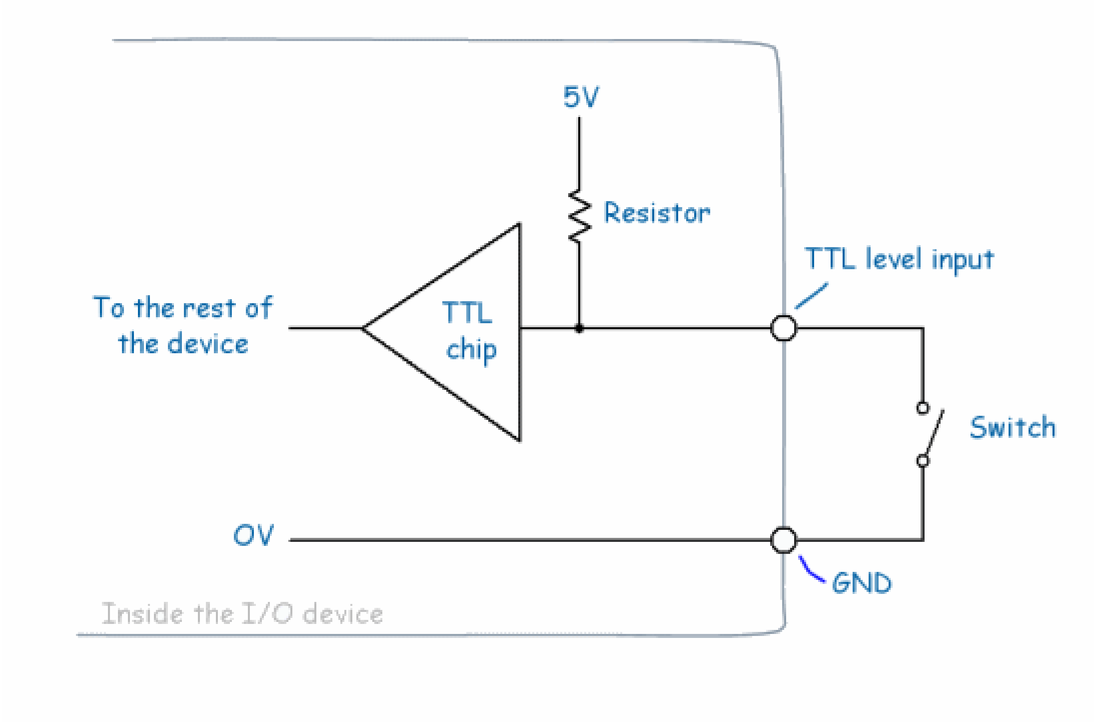


Figure 2. The real-world connections normally required to connect an on/off switch to a TTL input. Essentially the switch connects between the input and GND - this does assume that the input has a built-in pull-up resistor, but most do.

Connecting to opto-couplers

Inside an opto-coupler is a circuit like the one shown in figure 3. Don't worry about the details here, all you need to understand is that when a current flows through the LED it lights-up and this light is detected by the photo-transistor which the I/O device then interprets as meaning that the input is 'ON'.

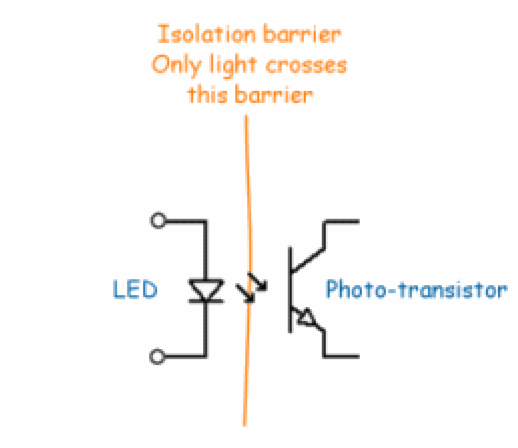


Figure 3. An opto-coupler. When current passes through the LED it lights-up and this is detected by the photo-transistor. Opto-couplers create electrical isolation between two circuits as only light passes between them.

So, what this means is that our on/off switch will need to control the current flow through the LED - so it really just becomes a switch in a lamp circuit, like the one shown in figure 4.

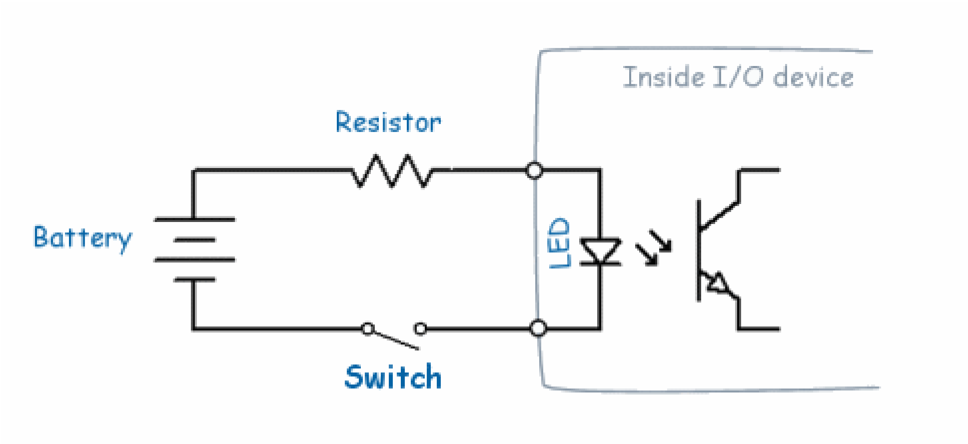


Figure 4. A circuit to connect an on/off switch to an opto-coupler. Although not shown as such, many I/O device have the resistor built-in.

Straight away you'll notice that this circuit includes a battery - does this mean we will need some additional power source (a battery or a power supply) to make our switch work? The answer is yes - and this is one of the disadvantages of using an opto-coupler input for a simple on/off switch.

The next thing you'll notice is that there's a resistor in the circuit, what's that for? The answer is that without the resistor the current that would flow through the LED in the opto-coupler would be very high and would almost certainly destroy it - so the resistor is rather important. Fortunately most device's actually have the resistor built (although it's not shown like that in figure 4).

Finally, you'll be left with one question, what voltage battery or power supply should I use? The answer to this will depend on the device you're using and will be found in its specification, but a typical value is 5VDC, although some devices accept a wide range of inputs, for example the isolated input module of the RTV-24 digitiser accepts any voltage from 5VDC to 24VDC.

Wrapping things up

In summary, it's usually much easier to connect a simple switch to a TTL level input than it is to connect it to an opto-isolated input because you don't need an external power source. This might make you wonder why opto-isolated inputs exist at all, and the answer to that is because they're better for connecting other types of inputs, they're just not too convenient for switches.

Connecting a photobeam

The details given in this topic provide guidance on connecting a photobeam to generic I/O devices such as the Switch and Sense 8/8. Connecting a switch using the ANY-maze interface is much simpler - refer to the ANY-maze interface topic for further details.

Introduction

The purpose of this topic is to describe the process of attaching a photobeam to a generic I/O device, so that ANY-maze can detect when the beam is broken. However, as off-the-shelf photobeams are quite hard to find, I've also included some details on how to build a simple photobeam using easy to source components.

Photobeams

Most photobeams consist of an infra-red emitter and an infra-red detector - as in figure 1.

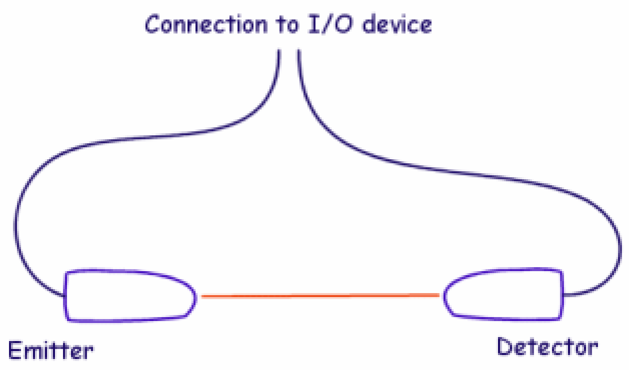


Figure 1. Schematic of an infra-red photobeam.

Photobeam emitter

The emitter in a photobeam is usually just an infra-red LED and all you need to do to make it work is connect it to a power source in such a way that a certain current flows through it. The required current depends on the emitter and will be quoted in its datasheet under *Forward current* - values from 20mA to 100mA are typical. To set this current you'll need to include a *current-limiting resistor*, as shown in the circuit in figure 2.

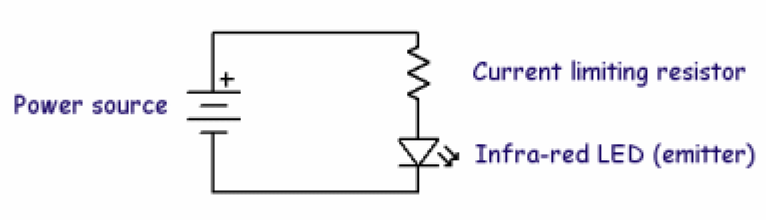


Figure 2. A simple infra-red photobeam emitter circuit.

To calculate the resistance required, you can simply apply ohms law although you will need to take the LED's *Forward voltage* into consideration (again this can be found in its datasheet).

Resistor = (Power source voltage - LED Forward voltage) / Forward current

Typical values might be R = (5 - 1.8) / 0.05 = 64 ohms. Something important to bear in mind is the power rating of the resistor. The power lost in the resistor will be P = current x current x resistance, which in my example would be 0.16 Watts. So you would need to use at least a 0.25W resistor.

Photobeam detector

A photobeam detector is usually a photo-transistor. In simple terms, this is a device that only conducts electricity between its two terminals when IR light strikes it. A simple circuit that will act as photobeam detector is shown in figure 3.

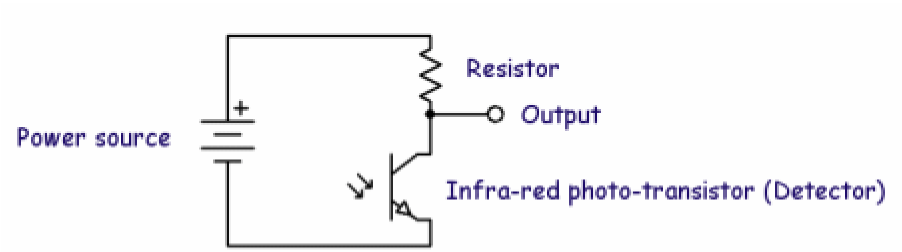


Figure 3. A simple infra-red photobeam detector circuit.

When there is no IR light striking the device it doesn't conduct, so the output will be held at 5V by the resistor. When IR does strike the device it conducts and the output will be connected to 0V - so the device is working like a switch.

In fact it's not really true to say the detector conducts or doesn't conduct, rather the amount of current that flows through it depends on the amount of IR light striking it. So with little light the current will be very low and it will increase as the IR light increases.

A complete photobeam design

Finding off-the-shelf photobeams is quite hard so you may want to build one yourself. As you can see from the above descriptions, this is quite simple and just requires a few components.

Figure 4 shows a complete circuit diagram for a simple photobeam that should work at distances of about 50cm, and below is a list of the parts you'll need - they're all available from Digi-key and the total cost should be under $2.00.

- L1, Infra-red LED - Fairchild QED123
- T1, Infra-red Photo-transistor - Fairchild QSD123
- R1, LED current-limiting resistor - 100 ohm 0.25W carbon film
- R2, Detector resistor - 22K 0.25W carbon film

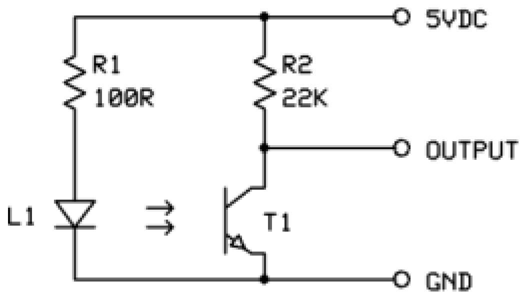


Figure 4. A complete infra-red photobeam circuit.

If you mount two or more of these simple photobeams close together you may experience cross-beam interference. This can be reduced by shielding the beams and/or by using a LED with a narrower angle. To completely avoid this problem you would need to use a more sophisticated (switched) beam design, which is what the ANY-maze interface does.

Connecting the photobeam to an I/O device

The "inputs" found on the I/O devices supported by ANY-maze fall into two categories, *TTL level inputs* and *opto-coupler inputs* - but what do these names actually mean?

TTL inputs have this name because they connect directly to a TTL 'chip' (for the curious, TTL stands for Transistor-transistor logic). In practical terms this means that the chip will be able to tell when the input is at a "logic high" (around 5 volts DC) or a "logic low" (around zero volts DC) - this is the 'digital' part of computing, everything is *high* = 1 or *low* = 0. We'll come back to TTL in a minute, but what about those opto-couplers.

Opto-couplers, as their name implies, 'couple', i.e. connect, your input to the I/O device using light. The big benefit of an opto-coupler is that it *isolates* the I/O device, and hence the computer connected to it, from the equipment you attach. There are a number of technical benefits to this isolation, but they're beyond the scope of this help - suffice to say that quite a few I/O devices use opto-couplers so you may need to connect to them.

Connecting a photobeam to a TTL input

This will be easy to do if the output from your photobeams is at TTL levels. For example, if the photobeam design given above is powered from a 5V supply then you could connect the output directly to a TTL input without any additional circuitry. However, some off-the-shelf devices may use different output levels in which case you will need use a 'level-shifting' circuit to interface to the TTL input. Contact ANY-maze technical support if you need help with this.

Connecting to an opto-coupler input

This is more complicated than connecting to a TTL input because the opto-coupler requires current to flow through it, so we need to make our photobeam output control this current. If you have some knowledge of electronics then you should be able to achieve this quite easily using a simple transistor circuit like the one shown in figure 5 - again you can contact ANY-maze technical support if you need help with this.

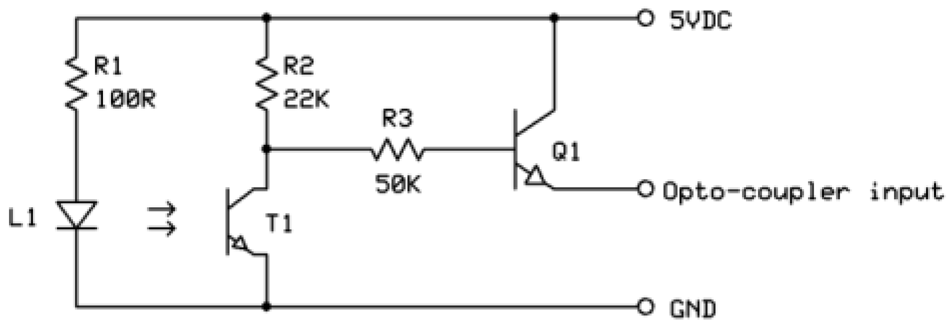


Figure 5. Circuit to interface a simple photobeam to an opto-coupler input. The transistor Q1 is controlled by the photobeam and will allow current to flow through the opto-coupler whenever the beam is broken. Note that depending on the device being used you may need to include a current limiting resistor in series with the output.

Turning something on and off

The details given in this topic provide guidance on controlling a device using a generic I/O device such as the Switch and Sense 8/8. Controlling a device using the ANY-maze interface is much simpler - refer to the ANY-maze interface *topic for further details.*

Introduction

The purpose of this topic is to describe how to use a generic I/O device to turn some equipment on and off. This might be a lamp, a buzzer, a shocker, etc.

- Types of output
- Relay outputs
- TTL outputs
- Open collector outputs

Types of output

The generic I/O devices supported by ANY-maze have three types of output, TTL level outputs, open collector outputs and relay outputs. Of these relay outputs are far and away the most useful, indeed other than specific situations, TTL and open collector outputs generally can't be used to switch anything on and off.

Relay outputs

As you may know, a relay is simply a mechanical switch that can be activated electronically. Thus your computer can send a signal to the I/O device which can close or open the switch. Most relays in I/O devices are of what's known as Form C relays meaning they have switch inside them, like the one shown in figure 1.

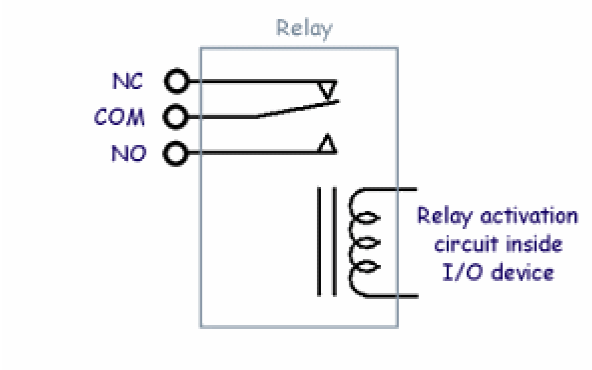


Figure 1. A Form C relay. When the relay is inactive the COM terminal is connected to the NC terminal, when activated the COM terminal is connected to NO.

As can be seen in figure 1, the relay has three terminals COM (common), NC (normally closed) and NO (normally open). When the relay is inactive the COM and NC terminals are connected together, when the relay activates the COM-NC connection is broken and the COM terminal connects to NO. So using a relay as a simple on/off switch is very easy, you just connect to the COM and NO terminals. In fact, in some cases you may wish to connect to COM and NC - that way when the relay is *inactive* your switch will be *on* and when the relay activates the switch will turn *off*.

Although relays are very easy to work with, there are a few things to watch out for:

- Relays have a certain voltage rating. So, for example, you can't use a relay with a 120VAC rating, to switch a mains-powered lamp in the UK, where the mains voltage is 240VAC.
- Relays have a certain current rating. This can vary widely depending on the device and you should pay close attention to it. For example, some relays in devices supported by ANY-maze have a current rating of 5 amps, while for others the rating is 0.5 amps.
- The ratings for AC and DC voltages will usually be different, with the ratings for DC voltages being significantly lower than for AC.
- As relays are mechanical switches they usually make a 'click' sound as they open and close. In some situations this may act as a cue to animals in which case you might want to consider using 'solid-state' relays instead. As the name implies, a solid-state relay isn't a mechanical switch, although it can still be viewed as one from the usage point of view. None of the generic I/O device supported by ANY-maze use solid-state relays, but the ANY-maze interface does.
- If you want to switch an inductive load (for example, a motor) using a relay then you should include a 'snubber' circuit across the terminals. This will suppress ('snub') the voltage transient that will appear across the relay terminals when the motor is switched off. For more details about snubber circuits contact ANY-maze technical support. Note that the ANY-maze interface has a built-in snubber circuit.

Although the circuit to switch a lamp or a buzzer is quite obvious, it may seem less obvious how to switch a shocker on and off, or how to switch a food hopper so it dispenses a food pellet; however, a little time studying the device's manual will usually provide an answer. For example, many shockers include a 'remote control' connector, which simply has two terminals - connect them together and the shocker comes on. So these two terminals can be connected to the relay and you're done. In the case a food hopper you may find, for example, that a pellet will be dispensed if 12VDC is applied to a certain input. In this case you can connect a 12VDC supply to one side of the relay and the input on the hopper to the other side. Now when the relay closes the hopper input will be connected to 12V and a pellet will be dispensed.

It's worth pointing out, that within ANY-maze it's very easy to create outputs that switch on momentarily, which is exactly what you'd want in either of these shocker and food pellet dispenser examples. You'll find more information about how to create momentary output switches here.

TTL outputs

As mentioned above, a TTL output on its own is not really able to switch anything, although you may find that some devices, such as shockers, have a TTL level input on them, in which case simply connecting the TTL output on your I/O device to this input (and connecting GND too) will allow you to control the shocker.

But if the device you want to control doesn't have a TTL input are you completely lost? Well, almost. Probably the easiest thing to do would be to change over to use a different, relay based, I/O device, but if you feel adventurous you could build a circuit to use your TTL output to switch either a mechanical or solid state relay. This isn't as complex as it might sound, and you can, of course, then choose exactly the right relay for your needs. If you're interested in further information then contact ANY-maze technical support.

Open collector outputs

An open *collector output* describes the type of output formed by a transistor that has its emitter connected to ground and its collector left 'open' - as in figure 2. Thus, when the transistor is conducting (i.e. active from ANY-maze's point of view) the 'output' is, effectively, connected to ground and when the transistor is inactive the 'output' is floating.

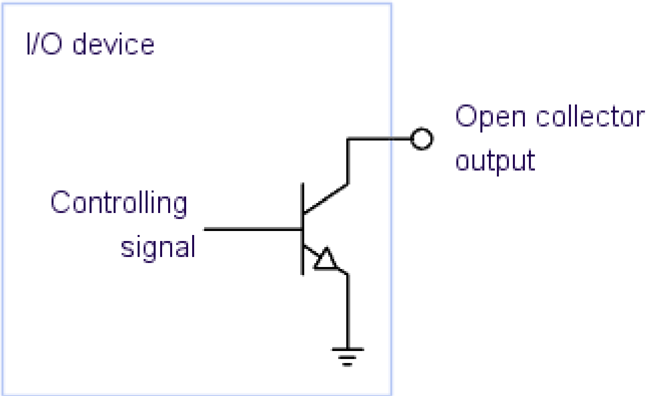


Figure 2. An I/O device with an open collector output.

In principle an open collector output is very useful, for example the circuits shown in figure 3 could be used to switch the LED on and off (figure 3a); to make the TTL input read a logic 1 or logic 0 (figure 3b); or activate the relay (figure 3c).

However, the problem with open-collector outputs from a input/output interfacing point of view is that they usually only work at relatively low voltages and currents - around 12VDC and 100 milliamps maximum.

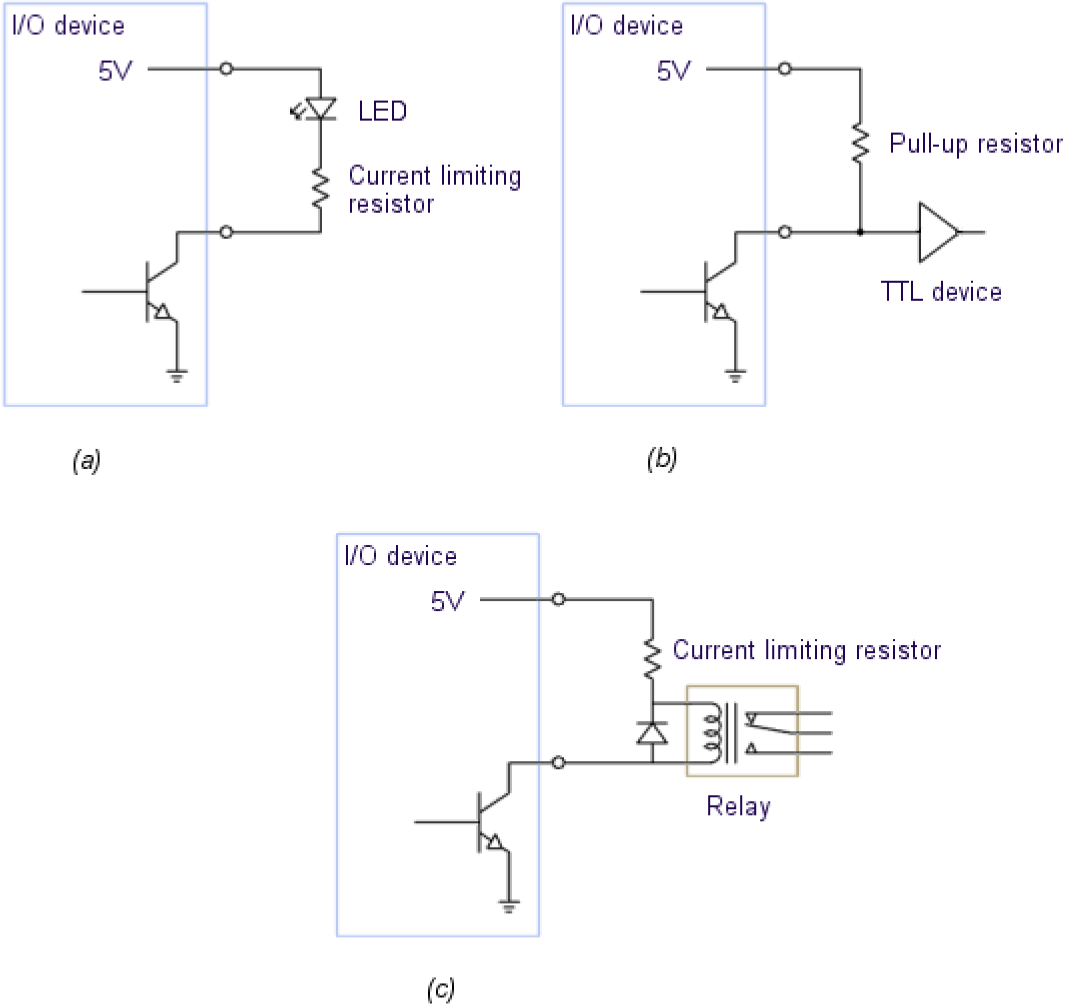


Figure 3. (a) An open collector output used to control an LED; (b) An open collector output used to provide an input to a TTL level input; (c) An open collector output used to control a relay.

So, to actually make use of an open collector output you will usually want to connect it to a relay (either mechanical or solid state) and have the relay switch the device you wish to control. The circuit in figure 3c can be used to do this. Here the diode is included to protect against voltage transients as the relay switches on and off.

Debouncing inputs

Introduction

One problem with switches is that they exhibit something called *contact bounce*. As the name implies, this is caused by the contacts bouncing apart and back together when the switch is closed. This bounce usually lasts a very short time and so it's often irrelevant in a simple switching circuit. However, in a situation where you want to count switch activations it can be very important - imagine that an animal pressed a lever and this caused some switch contacts to close but that they then bounced 5 times - ANY-maze would read 5 lever presses!

In fact you can also see 'bounce' type effects with other input devices such as photobeams. In this case as the beam is being broken it may momentarily fluctuate between being strong enough to count as 'made' and weak enough to count as 'broken'. Again, if you are counting beam breaks this could yield an unrealistically high score.

Solving the bounce problem

The usual solution to 'bounce' is to apply what's called a debounce interval, such that the state of the switch has to be stable for this interval before being registered as a change.

By default, ANY-maze will automatically apply a 20ms debounce interval to all switch inputs (whatever device they're generated by) to overcome this problem, but you can easily increase this by specifying a longer interval when you create a *Input switch* - this is described here.

See also:

- An introduction to on/off inputs
- Setting up an on/off input

Testing I/O using the I/O page

Introduction

An I/O device is usually an *interface* between your computer and some piece of apparatus. For example, you might use an ANY-maze interface to connect to a lever and to a pellet dispenser, but having made the physical connections, you will almost certainly want to test that the system is actually detecting lever presses and is able to cause a pellet to be dispensed. This is one of the principal functions of the I/O page.

Details

To test a device you should first select it in the list on the left side of the I/O page. When you do this you will find that the device *opens* to show a list of the *ports* it contains. For example, selecting an ANY-maze interface will show a list of ports like those in figure 1, below, (note that the exact ports shown may depend on the way the device has been configured).

To test one of the ports you just need to select it in the list, this will cause the rest of the I/O page to show a section entitled *Information* and a larger area showing the device's individual ports - as in figure 1.

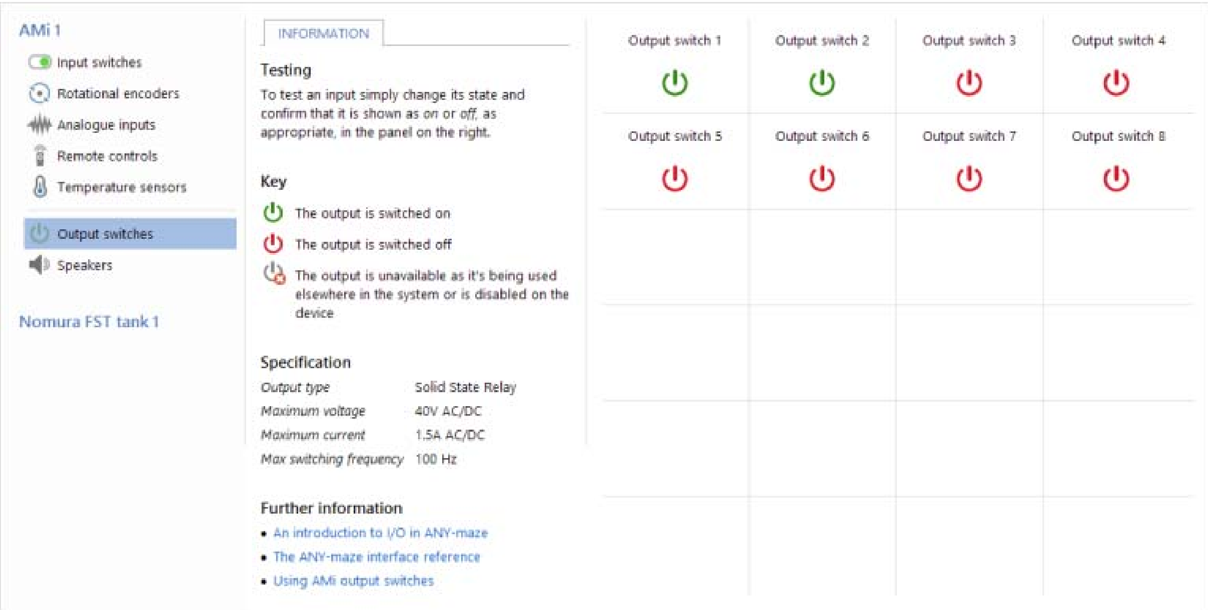


Figure 1. The I/O page showing the output switches of an ANY-maze interface

The way you actually test a port depends on the specific port type, but the *Information* section always provides detailed instructions. In the example in figure 1, where we're testing some output switches, you simply click the switch image for a port to turn the switch on and click it again to turn it off. In the figure, ports 1 and 2 have been turned on, which is why they're shown in green, and ports 3 to 8 are off, and therefore shown in red.

No matter what device you are using or which port you select, the Information panel will always include a section titled *Further information* which will include links to help topics specific to the selected device and port.

The ANY-maze Watchdog

Introduction

The ANY-maze watchdog is a utility feature which can be used to monitor light, temperature and humidity sensors and alert you if they go outside a predetermined range.

The watchdog operates independently of any experiments and can monitor any sensors at any time, even if the sensors are simultaneously being used in an experiment.

Scenarios in which you might want to use the watchdog include such things as: *Send an SMS message to my cell phone when the water in my water-maze reaches 34°C* or *Send me a message if the light level in my room goes below 50 lux at any time between 7am and 7pm.*

- The main watchdog window
- Adding a watchdog activity
- Editing a watchdog activity
- Deleting a watchdog activity
- Choosing which activities should be monitored
- Avoiding multiple alerts

The main watchdog window

If you have any light, temperature or humidity sensors connected to your computer, the ribbon bar on the I/O page will include a *Watchdog* section containing *Switch on watchdog* and *Watchdog set-up* buttons. Clicking the *Watchdog set-up* button opens the main *Watchdog window,* shown in figure 1, below.

Note that clicking the *Switch on watchdog* button, will switch the watchdog on and off. When off the watchdog won't monitor any sensors; when on it will monitor those sensors specified in the active watchdog activities.

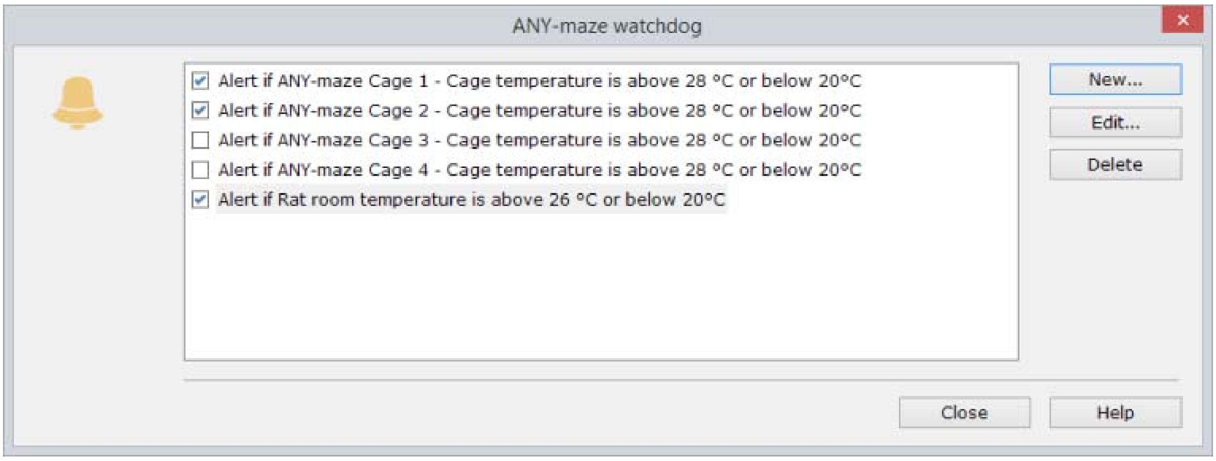


Figure 1. The main watchdog window, from which you can create new activities for the watchdog to monitor and control existing activities.

Adding a watchdog activity

To add a new watchdog activity you should simply click the *New...* button in the *Watchdog window*. This will cause the Set up watchdog activity window to open, where you will be able to specify the sensor to monitor, the limits to use, and the method by which the watchdog should alert you if the sensor goes outside the limits.

Editing a watchdog activity

To edit an activity, select it in the list and then click the *Edit...* button. This will cause the Set up watchdog activity window to open, where you will be able to edit all of the activity's details.

Deleting a watchdog activity

To delete an activity, select it in the list and then click the *Delete* button.

Choosing which activities should be monitored

Next to each activity in the list is a checkbox, selecting this will mean that the watchdog will monitor that activity - so in figure 1, above, for example, the watchdog is monitoring the temperature in ANY-maze cages 1 and 2, but not in cages 3 and 4.

Switching activities on and off is very useful as you can set up a selection of different things you *may* want to monitor and then select just those that are relevant to you at the moment. So, for example, if you want the watchdog to alert you when the water temperature in your water-maze reaches a certain value then you could switch that activity on and all other activities off.

It's important to understand that no activities at all will be monitored (whether they are 'checked' or not) when the watchdog itself is switched off.

Avoiding multiple alerts

If a temperature sensor, for example, has a limit of 27°C then when the temperature reaches 27.5°C the watchdog will raise an alert. But what happens if the temperature drops a little and then goes back up to 27.5° again, will another alert be sent?

The answer to this is, yes another alert will be sent, but only if the sensor dropped back to 27°C for 2 minutes and remained there before going over the limit again. Thus the 2 minute delay avoids multiple alerts if a sensor is oscillating around the limit value.

Of course, even with this delay a sensor could still cause multiple alerts, and to avoid this the watchdog will send no more than 4 alerts for any single activity in a period of 4 hours.

Should you need to reset these time delays, then you just need to toggle the watchdog on and off by clicking the *Switch on watchdog* button in the main toolbar.

See also:

- Setting up watchdog activities
- AMi sensor ports
- AMi - the ANY-maze interface

Setting up the ANY-maze Watchdog activities

Introduction

To set up a watchdog activity you just have to complete the fields in the window shown below:

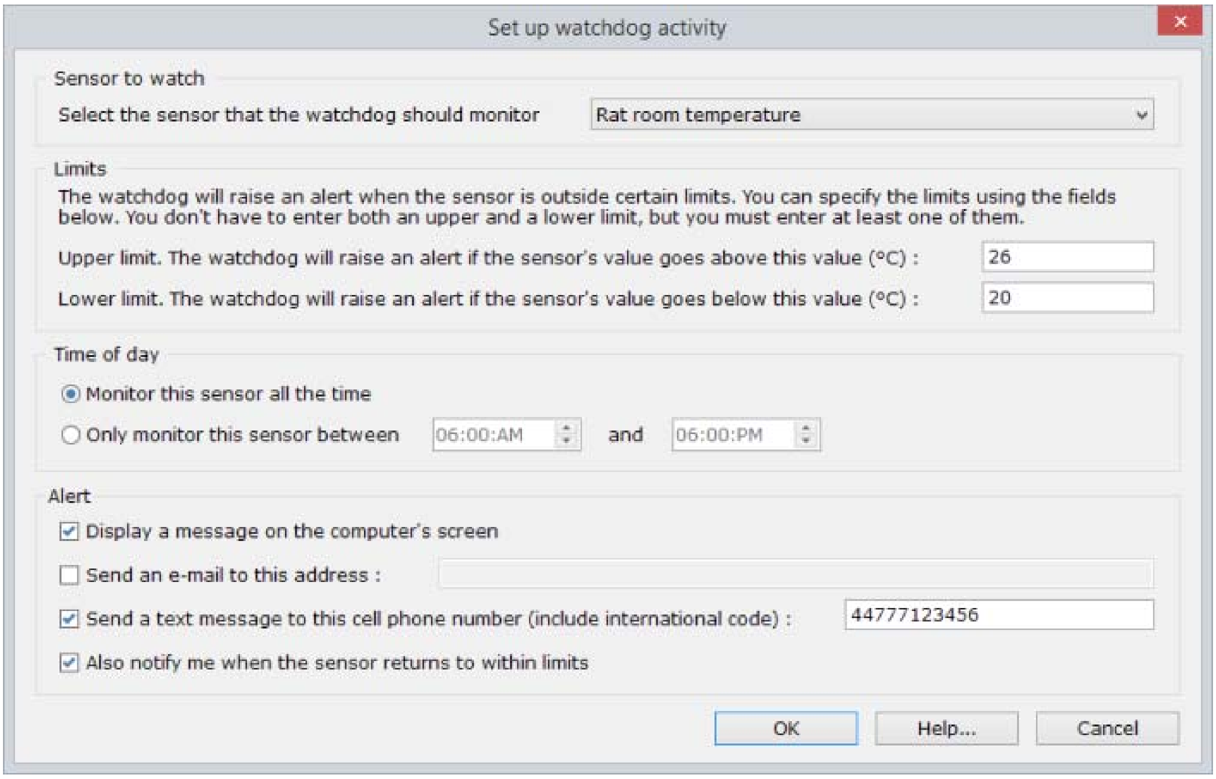


Figure 1. The Set up Watchdog Activity Window used to enter or edit the details of a watchdog activity.

Sensor — Use this drop-down list to choose the sensor that the watchdog should monitor. All light, temperature and humidity sensors that are *connected* to your PC will be listed as will the sensor currently being monitored by this activity (when editing) even if it's not actually connected.

It's very useful to understand that you can set up multiple activities to monitor the same sensor. There are many situations in which this is useful and there's no limit to how many times one sensor can be monitored.

Limits

Enter the limits that the sensor value must stay *inside*. For example, if you enter 30 (degrees) as the upper limit for a temperature sensor then the watchdog will alert you when the temperature is *greater* than 30°C. You don't have to enter both an upper and a lower limit, so if you don't care how cold the sensor gets, you just want to know if it goes above 30°, then you would leave the lower limit field blank.

The entry you make must be a whole number with no units. You can enter a minus sign if you want to.

Time of day

In most cases you will probably want to monitor a sensor all the time, but there may be occasions when you want to limit the monitoring to certain times of day. For example, imagine that you want to use the watchdog to confirm that the light/dark cycle in your animal room is running correctly. To do this you could create one activity that checks that the lights are on and another that the lights are off. You'd want the former to just monitor the light levels during the light period and the latter just during the dark period. In this case it would be wise to start monitoring about 10 minutes after the lights should come on and stop 10 minutes before they go off, just to avoid any problems with time differences between your computer's and your lighting system's clocks.

Alert

The watchdog has three ways it can alert you when a sensor goes out of its limits: By displaying a message on the screen, by sending you an e-mail or by sending a SMS message to your cell phone.

Message on screen

A message on the screen is the simplest alert, but of course it assumes you'll be near your computer. The message will pop up on top of all open windows and, if you've specified an Alert sound, a sound will play.

E-mail

To send an e-mail you simply need to enter an e-mail address. Your computer doesn't need to have any e-mail software installed although, of course, it does need to be connected to the Internet. You can enter more than one e-mail address if you wish, just separate multiple addresses with commas.

SMS message

Perhaps the best way to receive an alert is as an SMS message to your cell phone. In this case you need to enter the cell phone number **including** the international code and excluding any leading zeros. For example if you are in the UK (international code 44) and your number is 0777-1234-567 then you would enter 447771234567. In this case you can't enter multiple phone numbers, but you can work round this by creating two identical activities with different numbers - when the watchdog detects that the sensor has gone outside its limits it will process both activities and send two SMS messages.

In order to send SMS messages your computer must be connected to the Internet.

Return to limits As well as notifying when a sensor goes outside its limits, the watchdog can also notify you if it returns to within the limits. This can be useful as it allows you to wait and see whether a situation that's been reported, corrects itself. Note that the sensor must return to within limits and remain there for 2 minutes before a "Returned to limits" message will be sent.

The Video page

Unlike the pages on the left side of the ribbon bar, you can access the Video page without having to open an experiment first.

Introduction

The Video page provides some useful utility functions in ANY-maze. It lets you:

- View images from a capture device
- Record images from a capture device in ANY-maze video format
- Record images from a capture device in a standard video format
- Play a video file
- Play a DVD
- Convert an ANY-maze format video to a standard format

Viewing images from a capture device

Although you will normally use the images from a capture device to track animals in an experiment, there may be occasions when all you want to do is simply view the images - for example:

- You might just want to use a camera to observe an animal.
- You might be testing a device; or adjusting its zoom, focus etc.
- You might want to monitor the images from a shared device that's being used in an experiment on another computer.

To actually view the images from a device is very simple - just click the name of the device in the left-hand side bar of the Video page; the live image will then appear. To disconnect from the device, just click its name again.

The ability to view images from another computer on your network (which is sharing one of its capture devices) can be very useful, as it makes it possible to remotely monitor an experiment that's in progress - see the diagram below.

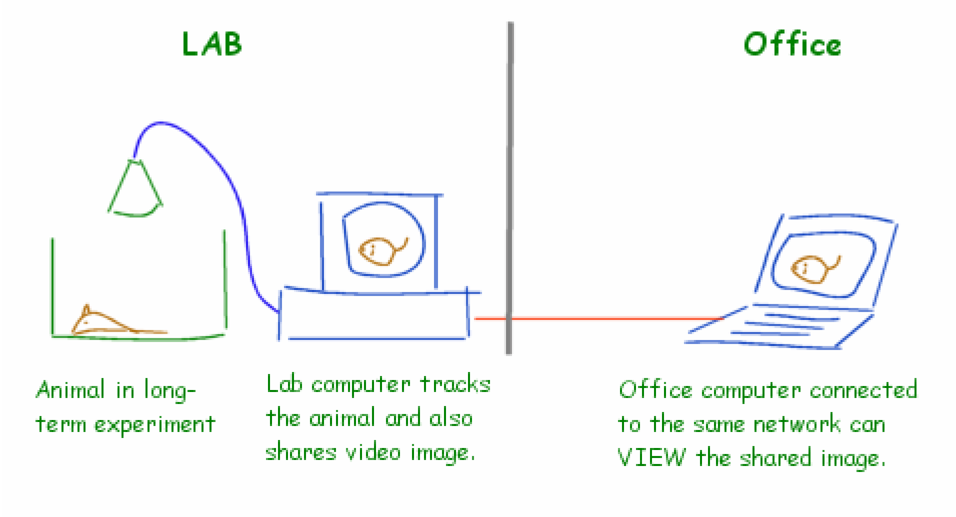


Figure 1. On the office computer, you could use the Video page to view the experiment that's in progress in the Lab.

Recording images from a capture device in ANY-maze video format

As you may know, you can record the images from any camera or digitiser using the VCR buttons on the Tests page. However, a drawback with this is that you must first set up an experiment in order for the Tests page to actually show the images. In many cases this won't be a problem, but there may be occasions when *all* you want to do is record a video.

A classic example of this is when you want to run two experiments at the same time, but you only have a single ANY-maze licence. In this case you could run one experiment on your licensed copy of ANY-maze, while at the same time recording a video of the other experiment using an unlicensed copy of ANY-maze on a second computer (remember you can install as many unlicensed copies of ANY-maze as you like). At some point in the future, you would simply copy the video to your licensed computer and then actually perform the tracking.

In situations like this, where all you want to do is record a video, then the easiest place to do it is on the Video page. You simply need to connect to the appropriate video capture device (see Viewing images from a capture device, above) and then use the VCR buttons shown above the image to make the recording.

By the way, if the intention is to *track* animals in the video you're recording, then the best format to use is ANY-maze video format; otherwise the best format is standard (AVI) format.

To record a video in ANY-maze format, you should:

- Select the video source you want to record from the left-hand side bar of the Video page - so the images are shown in the main section.

- Click the (Record) button at the right-hand side of the toolbar above the video image.
- In the window that opens, enter the name of the file you want to record the video to. You can also, optionally, change the compression settings used.
- Click *Save*.
- The recording will start, and the toolbar above the image will show the recording progress (i.e. the size of the file and the time it's been recording for).
- To end the recording, click the *(Stop)* button in the toolbar. This button has a drop-down arrow at its right-hand side; if you click this, you'll be given the option to *Stop video recording* (which will stop recording and save the video) or *Cancel video recording* (which will stop recording without saving the file). Just clicking the main body of the button will stop the recording and save the file.

If you switch to another page (such as the Protocol page or the Tests page) while recording a video on the Video page, the video recording will stop.

Recording images from a capture device in a standard video format

As well as recording images in ANY-maze's video format (which is best for subsequent tracking), the Video page can also be used to record images from any video source in a standard format (MP4 or WMV under Windows 7 and above, or AVI under Windows XP and Vista). This has the advantage that the video can subsequently be played in software such as Windows Media Player, or included in PowerPoint presentations.

To record a video in a standard format, you should:

- Select the video source you want to record from the left-hand side bar of the Video page - so the images are shown in the main section.
- Click the *Record a standard format video* button in the ribbon bar.
- In the window that opens, enter the name of the file you want to record the video to. You can also, optionally, choose the compression settings to alter the quality and size of the videos (for AVI videos, you can specify your own compression settings.
- Click *Save*.
- The recording will start, and a flashing red dot will be shown in the bottom right-hand corner of the video picture to indicate this.
- To end the recording, click the *Record a standard format video* button again.

Note that this feature can also be used to convert an ANY-maze format video to a standard format - instead of using a file, you can open and play an ANY-maze format video, saving it to a standard format as it is playing. See Converting ANY-maze format videos to a standard format for more details.

Playing a video

The Video page can be used to play a video of any format supported by ANY-maze and for which you have a video codec installed. This will usually include ANY-maze's own format videos, AVI videos, MPEG videos and Windows Media Player videos. This just provides a handy way to quickly and easily view a video without having to open an experiment.

To view a video, you should click the *Browse for a video file...* option in the *Video files* section of the left-hand side bar on the Video page. The Choose video window will open, where you can choose the video you want to play.

You'll notice that ANY-maze maintains a list of the videos you've recently played in the *Video files* section of the left-hand side bar. To play one of these videos, you simply need to click its name. Clicking it again will stop playback. If you no longer want to see a video in this list, right click on the video's name and select *Remove this video from the list.*

ANY-maze will only display videos in black and white, even if they're recorded in colour. Also, it won't play the video's sound track, even if it has one.

Adjusting video playback speed

When you play and *ANY-maze format video* (i.e. a .szv file), you'll find that a playback speed button is included in the video controls. Clicking this button opens a menu of options for faster and slower playback speeds - you can change the playback speed in any way at any time.

Moving through a video a frame at a time

Also, when playing an ANY-maze format video, you'll see that the playback timer shows the normal hours, minutes and seconds but also includes milliseconds. As you may know, if you roll the mouse wheel when the mouse pointer is over the playback timer, the video playback will jump forward or backward by one unit - for example, roll the mouse wheel forward when over the minutes and the video will jump forward by a minute, roll it backwards when over the hours and the video will jump backward by one hour, and so on. This also applies to the milliseconds; however this isn't really useful unless you first *pause* the video playback - then it becomes very useful, as it allows you to move backwards and forwards through the video *a frame at a time.*

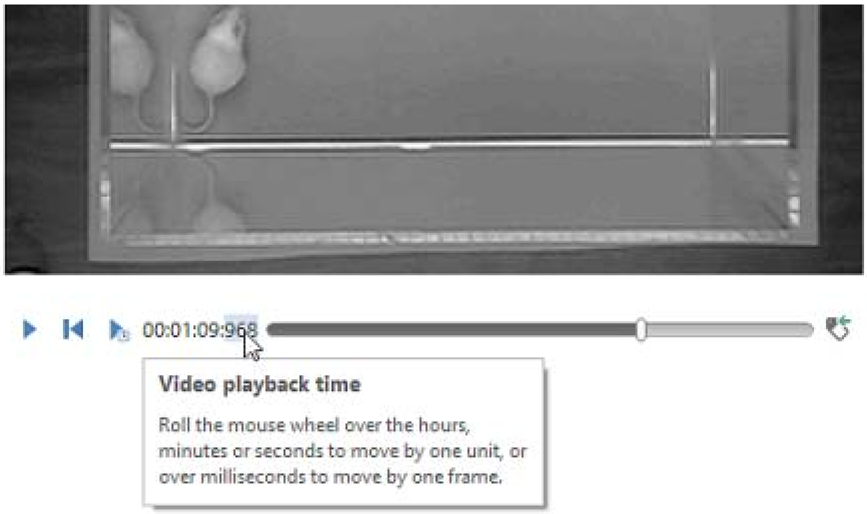


Figure 3. In a paused video, rolling the mouse wheel when the pointer is over the milliseconds part of the video playback timer will move through the video a frame at a time.

Playing a DVD

If your computer has a DVD drive, then you can use the Video page to play the DVD, provided you have a suitable video decoder installed on your computer. For full details about playing DVDs, refer to Playing DVDs in ANY-maze.

Setting up video devices

Overview

The *Set up video devices* window can be used to:

- Give 'user-friendly' names to the various video capture interfaces and cameras installed in your computer
- Disable specific devices
- Share devices with other users on your network

For more information on using this window, use the following links:

- The 'Set up video devices' window
- Naming video capture devices
- Disabling video capture devices
- Sharing video capture devices
- Adding network cameras
- Configuring devices

The *Set up video devices* window

You can alter the settings for video devices using the *Set up video devices* window. To open this window, simply click the *Set up video devices* button on the ribbon bar of the Video page.

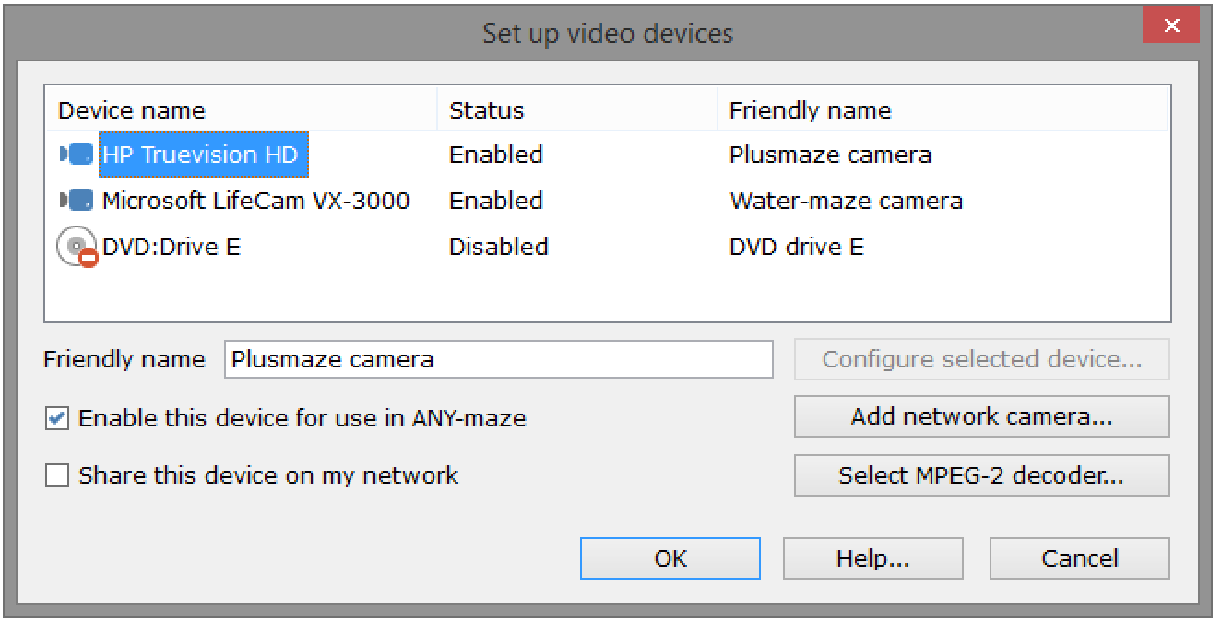


Figure 1. The Set up video devices *window.*

Naming video capture devices

The video capture devices installed in your computer, such as National Instruments digitisers or USB cameras, are given names by their manufacturers which are often a bit technical, or (in the case of NI digitisers) simply obscure.

To improve on this situation, it's usually a good idea to use this window to give the devices more user-friendly names, which ANY-maze will then use throughout the system.

To name a device, simply select it in the list and then type in a *Friendly name*. It's usually a good idea to base names on the location of the camera (for example, 'Camera Room 102') or on the name of the apparatus which the camera can see (for example, 'Water-maze camera').

Note that you're not obliged to provide user-friendly names if you don't want to, and if you only have a single camera, you might feel it's unnecessary.

Disabling video capture devices

ANY-maze automatically searches your computer for video capture devices that it can use, and then makes them all available to you. While this is normally what you'll want, there may be occasions when it would be inappropriate (or impossible) to use one or more of the devices that it finds.

For example, most notebook computers have cameras built into their lids, which you'd be unlikely to be able to use for tracking. Also, some older devices simply don't work in modern computers - such as the capture devices built into older ATI All-In-Wonder devices - so again, you'd want to disable them in ANY-maze.

To actually disable a device, simply select it in the list and then un-check the box labelled *Enable this device for use in ANY-maze*.

Sharing video capture devices

Normally, the video images captured by a digital camera or a digitiser card are only available to the computer to which the device is physically attached. However, by sharing a device, you can make the images available to any other computer on your network that's running ANY-maze.

Network cameras can't be shared from within ANY-maze, as they're already available as a shared network resource.

To share a device, you simply need to select it in the list and then check the box labelled *Share this device on my network*. The device will then automatically appear as an available video capture device on all the other computers on your network - you don't need to do anything else to make this happen.

Using a shared device is essentially the same as using one that's attached directly to your computer. The only difference relates to how ANY-maze resolves conflicts when two users want to share the same device:

- If you want to use a shared device in an experiment, then you'll use it as the *Source of video images* in one of your video sources. When you do this, ANY-maze will *lock* the device so that no one else can use it at the same time (well, almost - see below). This is important because otherwise a user on another computer could access the device and, for example, alter the brightness or contrast of the image - if they happened to do this in the middle of a test, they'd ruin your experiment. So, a device being used in an experiment is 'locked'.
- If you try to access a device that's been locked by a user on another computer (see the previous point), ANY-maze will display a message telling you that the device isn't available, and informing you which computer has locked the device.
- Although you will mainly use video capture devices in experiments, you can also view the images from a device on the Video page. The difference here is that ANY-maze *won't* lock a shared device that's displayed in the Video page, so any number of users can view the image at the same time. In fact, it will even let you view the image from a device that *is* locked (because on the Video page there's no way to alter the device's settings). This has the advantage that you can use the Video page to view images that are actually being used in a live experiment.

For example, imagine you have a computer in your lab that you're using to track an animal in its home cage for 72 hours. Clearly, you're not going to stay in the lab the whole time watching the experiment on the computer's screen. Nevertheless, you might like to be able to keep an eye on things - so you could simply use ANY-maze on the computer in your office to view the video images of the animal in the lab - see the diagram below.

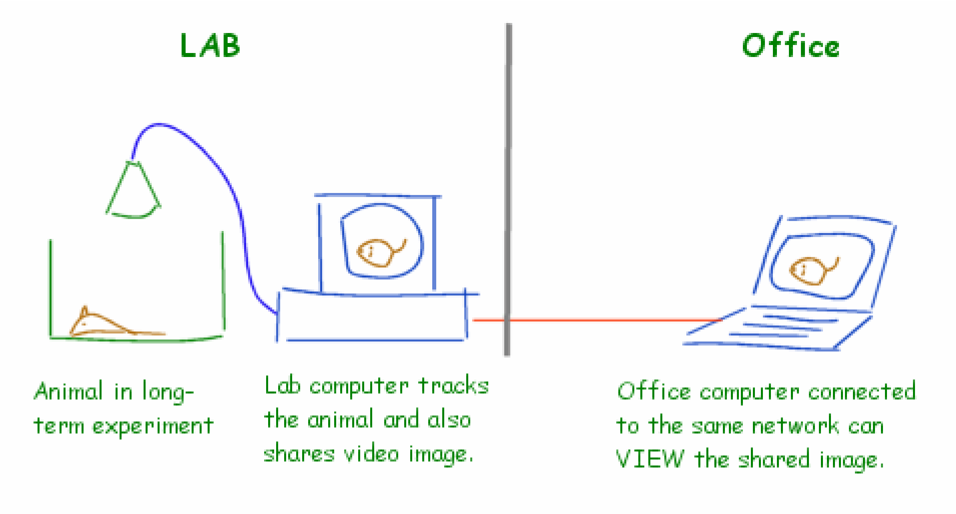


Figure 2. Using sharing to monitor an experiment at a distance.

As you'd expect, sharing images in this way means that the computer connected to the camera or digitiser has more work to do, and of course you'll also add some extra load to your network:

- Sharing works best on computers with higher-end processors, which are able to make light of the extra work involved. In this case, sharing an image typically adds about 10% to the computer's overall workload.
- Sharing requires relatively little bandwidth from your network. On a modern 100Mbps network, this is usually in the order of 5-10%.
- Sharing also works on wireless networks, although since these are slower than wired networks, sharing does consume a greater percentage of the available bandwidth.
- For the technically inclined: sharing compresses images before sending them across the network. Because the amount of compression depends on the image, the extra load on the processor and network can't be precisely defined.

Adding network cameras

If you have an Axis network camera connected to your network, then ANY-maze will be able to track animals in the images it shows.

However, ANY-maze can't automatically detect that network cameras are present, so you need to tell it about each camera you wish to use. This is very simple to do - just click the *Add network camera...* button and enter the camera's details into the *Axis camera configuration* window that opens.

Configuring devices

Some devices, such as RTV-24 digitisers and Axis cameras, can be configured within ANY-maze. For these devices, a button labelled *Configure selected device*... will be shown in this window. If you click it, a configuration window specific to the selected device will open.

Further details can be found in these topics:

- The RTV-24 configuration window
- The Euresys digitiser configuration window
- The Axis camera configuration window
- MPEG-2 decoder configuration

RTV-24 configuration

Overview

If you have one or more RTV-24 digitisers installed in your computer, then the *Set up video devices* window will include a button to configure the devices. The *Configure selected device...* button will open this window, where you can specify the default camera type you will use with the digitiser and which devices, if any, have RTV-E4 extension boards fitted.

Use the Setup video devices *button on the* **Video page** *to open the* Set up video devices *window.*

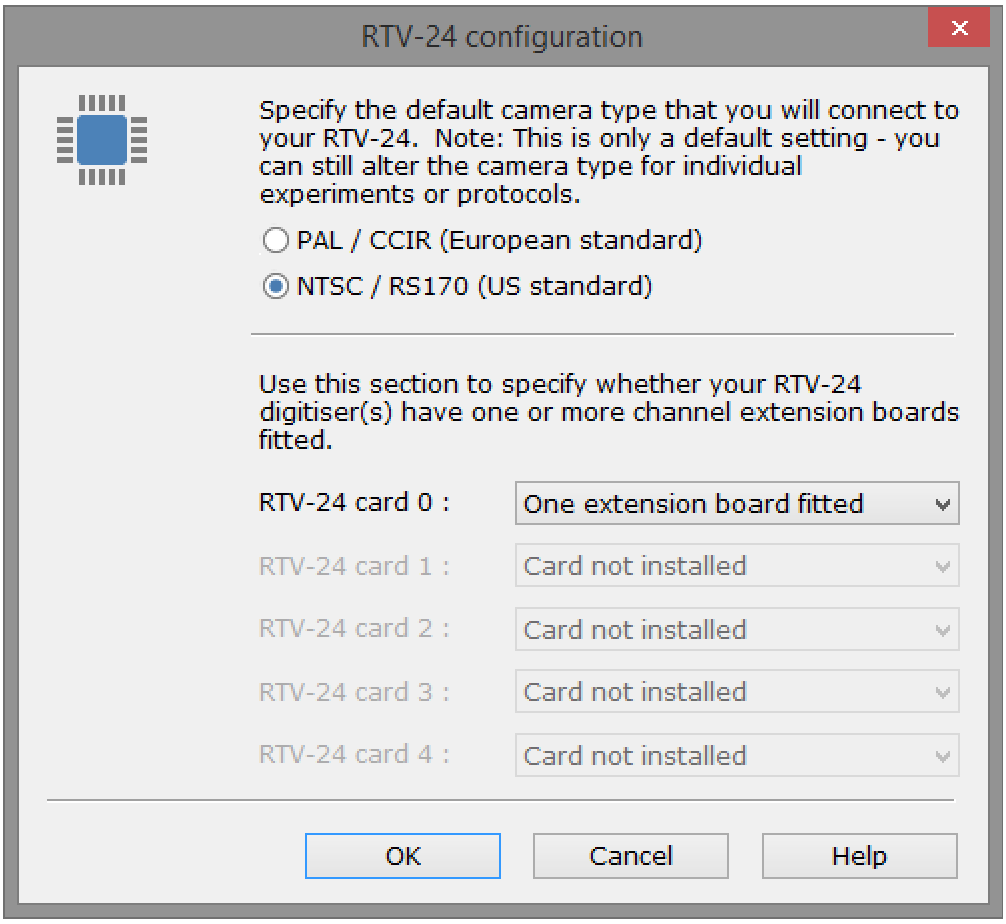


Figure 1. The RTV-24 configuration *window*

Specifying the default camera type

To function correctly, the RTV-24 needs to know whether the camera connected to it uses the PAL (European) or NTSC (US) video standard. You can use the options here to specify which camera type you will normally be using.

Note that this setting is just a default; you can still use the options on the *Video signal* page of the *Camera/Digitiser properties* window to set the camera type for individual experiments and protocols.

Specifying whether extension boards are fitted

Each RTV-24 can optionally be fitted with up to three RTV-E4 extension boards. These boards add multiplexed channels to the four ports of the RTV-24. For example, if you have 2 extension boards fitted to RTV-24 card 0, then you will have 12 video inputs:

- RTV-24 card 0 Port 0, channel 0
- RTV-24 card 0 Port 0, channel 1
- RTV-24 card 0 Port 0, channel 2
- RTV-24 card 0 Port 1, channel 0
- RTV-24 card 0 Port 1, channel 1
- RTV-24 card 0 Port 1, channel 2
- RTV-24 card 0 Port 2, channel 0
- RTV-24 card 0 Port 2, channel 1
- RTV-24 card 0 Port 2, channel 2
- RTV-24 card 0 Port 3, channel 0
- RTV-24 card 0 Port 3, channel 1
- RTV-24 card 0 Port 3, channel 2

It's important to understand that these extra channels are *multiplexed*. This means that the extra channels *share* the port - or to put it another way, each port can connect to any one of the channels, but it can only do this for one channel at a time. Thus in the above example, while you could connect 12 pieces of equipment to your computer, you could only actually use 4 of them at the same time.

By the way, you're required to specify the existence of the RTV-E4 boards manually using this window, because there is no method provided by the RTV-24 device for ANY-maze to discover programmatically whether an RTV-E4 board is installed or not.

Euresys digitiser configuration

Overview

If you have one or more Euresys digitisers installed in your computer, then the *Set up video devices* window will include a button to configure the devices. The *Configure selected device...* button will open this window, where you can specify the default camera type (PAL or NTSC) that you will use with the digitiser.

Use the Setup video devices *button on the* **Video page** *to open the* Set up video devices *window.*

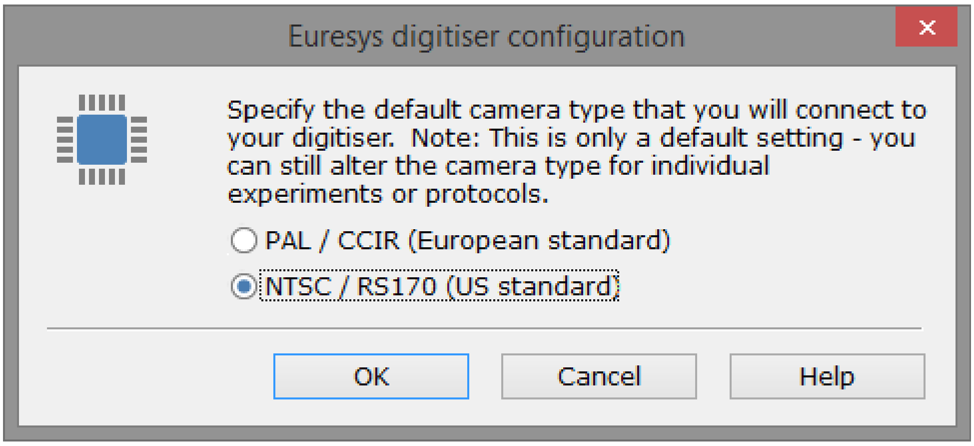


Figure 1. The Euresys digitiser configuration *window*

Specifying the default camera type

To function correctly, your digitiser needs to know whether the camera connected to it uses the PAL (European) or NTSC (US) video standard. You can use the options here to specify which camera type you will normally be using.

Note that this setting is just a default; you can still use the options on the *Video signal* page of the *Camera/Digitiser properties* window to set the camera type for individual experiments and protocols.

Axis camera configuration

Overview

The *Set up video devices* window includes an *Add network camera* button, which allows you to add a network camera to your list of video devices. Clicking this button opens the *Axis camera configuration* window, where you can specify the camera's IP address and the user name and password required to access it.

Use the Setup video devices *button on the* **Video page** *to open the* Set up video devices *window.*

For more information, use the following links:

- Adding a new camera
- Editing a camera's configuration
- Deleting a camera
- Types of users
- The effects of user type in ANY-maze

Adding a new camera

To add a network camera, simply click the *Add network camera* button on the *Set up video devices* window to open the Axis camera configuration *window:*

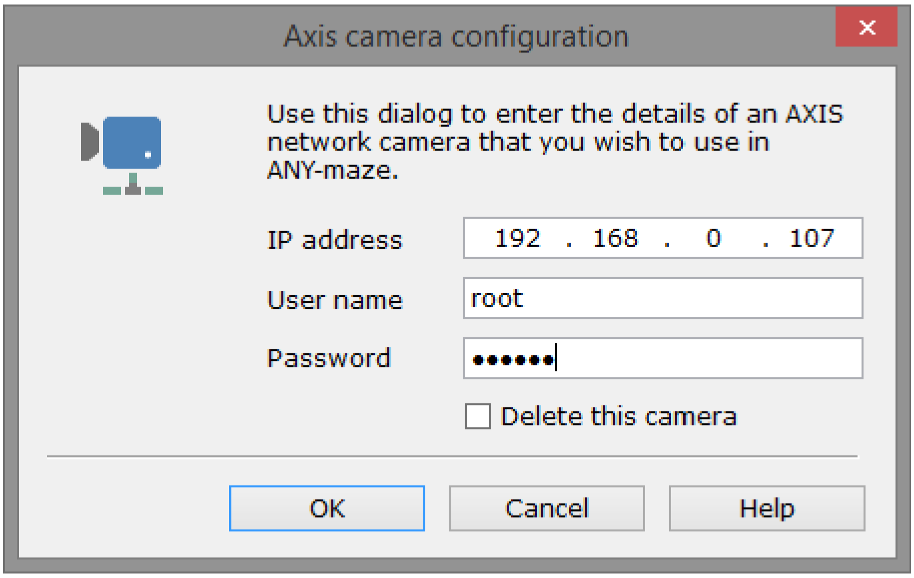


Figure 1. The Axis camera configuration *window*

When adding a new camera, you need to specify its IP address, a user name and a password.

The IP address is simply the address of the camera, and will either be specified in the camera set-up or it will be assigned by a DHCP server. Note that a camera does need to have a fixed IP address if it is to be used in ANY-maze. When you specify a camera's IP address, ANY-maze will simply accept your entry and then go off and check that a camera with the address actually exists; if the system is unable to find the camera, it will show it as *Unavailable* on the *Set up video devices* window.

To actually access an Axis network camera, you need a user name and password. By default, Axis cameras have a user name of 'root' and a password that you set the first time you use the camera - this default user is automatically an administrator. The Axis camera set-up allows you to add other users, each with their own name and password, and you can use any registered user to access the camera in ANY-maze. Indeed the cameras can even be set to allow 'anonymous viewer login', in which case you don't need to enter a user name or password at all - however, there are disadvantages to this; see Types of users below.

Editing a camera's configuration

Having added a camera, you can alter its details at any time by selecting it in the *Set up video devices* window and then clicking the *Configure selected device...* button. The camera's configuration window will open (see above), where you can alter the IP address, name or password.

Note that changes to the IP address will cause ANY-maze to re-confirm that the camera exists, and so it will be shown as having a status of *Checking...* in the list of devices. If ANY-maze is unable to confirm that the camera exists, then its status will change to *Unavailable,* otherwise it will change to

Enabled.

Deleting a camera

Having added a camera, you may wish to remove it - for example you might disconnect the camera from your network. To do this, simply select the camera in the *Set up video devices* window and then click the *Configure selected device*... button - the camera's configuration window will open (see figure 1, above). At the bottom of this window is a check box labelled *Delete this camera;* checking it and then clicking *OK* will cause the camera to be deleted from the list of video devices.

Types of users

Axis network cameras have three types (or groups) of user:

Viewer	These users can just view images. If your camera is set to allow *anonymous viewer login,* then anonymous users (with no user name or password) will be part of this group.
Operator	These users can view images and can also adjust camera parameters such as brightness, contrast, etc.
Administrator	These users can view images, adjust camera parameters, and alter all other camera settings too.

The effects of user type in ANY-maze

In ANY-maze, a *Viewer* user will be able to view images but won't be able to make any adjustments to the image, or use the camera's I/O functions. An *Operator* user will be able to adjust the image, but again, the camera's I/O functions won't work. An *Administrator* user will be able to use all the functionality of the camera, including camera I/O.

As can be seen from the above description, it's generally best to use an *Administrator* user within ANY-maze, although if you don't intend to use the camera's I/O functions (or if your camera doesn't include any I/O) then you can use an *Operator* user instead.

MPEG-2 decoder configuration

Overview

In order to play a DVD or MPEG-2 video file, ANY-maze needs something called an *MPEG-2 decoder*. As the name implies, this decodes the data from the DVD or file and converts it into video pictures.

Recent versions of Microsoft Windows (from Windows 7 onwards) are supplied with a built-in MPEG-2 decoder, but this doesn't always work with ANY-maze. Earlier versions of Windows (up to and including Windows Vista) are not shipped with a decoder, so you will need to install a third party decoder in order to play DVDs or MPEG-2 encoded files. There are a number of such decoders available, and some of them are more suited to playing DVDs and video files in ANY-maze than others. You can use this window to choose which of the decoders installed on your computer ANY-maze should use.

- Acquiring an MPEG-2 decoder
- Pros and cons of some common decoders

Acquiring an MPEG-2 decoder

As mentioned above, to play DVDs or MPEG-2 encoded video files you will probably need to install a decoder yourself. In fact, many computers are shipped with a free DVD player (such as CyberLink PowerDVD) and this will include a decoder - so there's a good chance you'll have a decoder installed already.

However, if you don't have a decoder, or you want to try a different one, then you'll need to download and install one from the internet - there are some links to free codec packs below.

- FFDShow MPEG-4 Video decoder
- K-Lite Mega Codec Pack

  (Although the K-Lite codec pack is a free download, you'll need to make sure you *Decline* any additional software offers during the installation, and un-check the choices for other software that it gives you the option to install).

These codecs should work correctly on all operating systems from Windows XP onwards, although they don't necessarily work well on all computers. Try each of them to see which works on your computer.

Playing DVDs in ANY-maze

Introduction

Increasing numbers of ANY-maze users are recording experiments directly to DVD discs for later tracking within the system. If you do this, and if your computer includes a DVD drive, then you can simply insert a DVD disc and then play it directly within ANY-maze.

DVD playback is a relatively new feature in ANY-maze, and we welcome your comments, suggestions and feedback on how well it works and what we can do to improve it.

- Requirements and restrictions for DVD playback
- Playing DVDs on the Video page
- Tracking from a DVD

Requirements and restrictions for DVD playback

There are two requirements for DVD playback. First, obviously, your computer must have a DVD drive; second, and less obviously, it must have DVD decoder software installed.

Microsoft Windows does not include a DVD decoder as standard, and nor does ANY-maze, so if you're running on either of those operating systems then you'll need to use a third-party decoder in order to play DVDs. This isn't usually a problem, because most computers that have DVD drives are shipped with some DVD playback software, such as WinDVD or CyberLink PowerDVD, and this includes a decoder that ANY-maze will be able to use. Even if your computer doesn't have such software already installed, you can download free decoders from the internet, for example:

- FFDShow MPEG-4 Video decoder
- K-Lite Mega Codec Pack

  (Although the K-Lite codec pack is a free download, you'll need to make sure you *Decline* any additional software offers during the installation, and un-check the choices for other software that it gives you the option to install).

These codecs should work correctly on all operating systems from Windows XP onwards, although they don't necessarily work well on all computers. Try each of them to see which works on your computer.

Even with a DVD drive and decoder, ANY-maze still won't play DVDs until you enable this functionality, as it is switched off by default. To enable DVD playback, do the following:

- Switch to the Video page.

- Click the *Set up video devices* button in the ribbon bar.
- You'll see your DVD drive listed; click it so it's selected.
- Click the check box *Enable this device for use in ANY-maze.*
- Click the *Configure selected device...* button. This will open the MPEG-2 decoder configuration window, where you will see a list of the DVD decoders installed on your computer. You need to make sure that there's at least one decoder listed. If there is more than one, you can choose which one to use, but you don't need to make a choice as the default entry of - *Choose automatically* - will just use the first decoder in the list if you don't make a selection.

  If there are no decoders listed, or you want help choosing one from those listed, then click the *Help* button - this will open a pop-out help window showing you a help topic which includes links to web sites where you can download decoders, as well as a section describing the pros and cons of various common decoders.

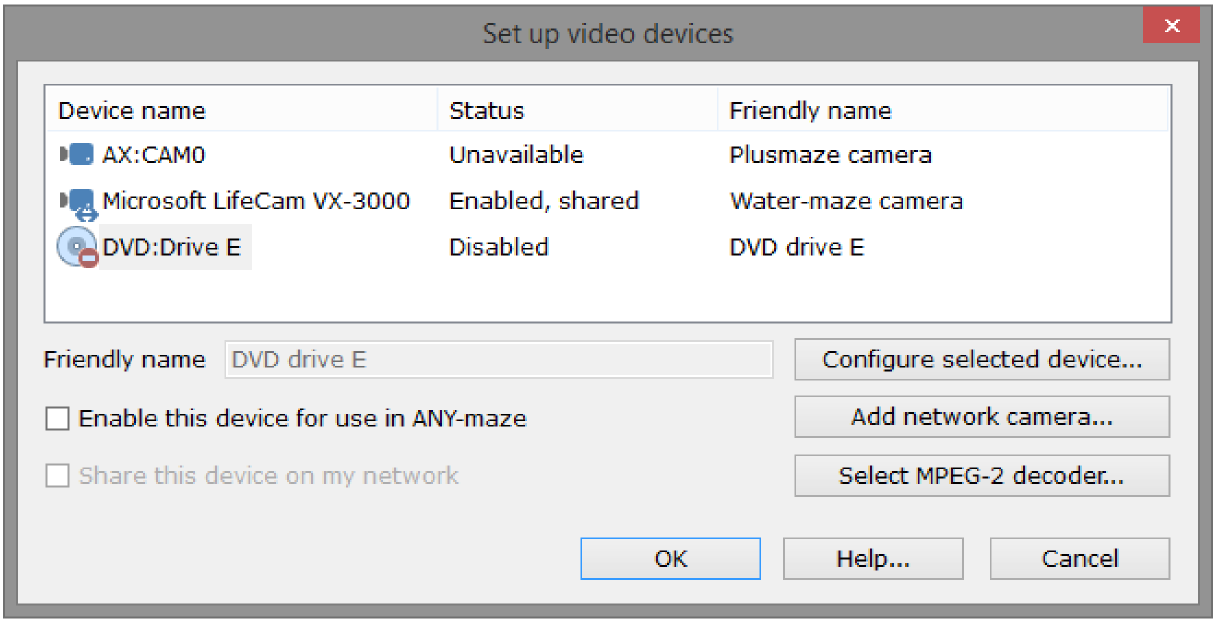


Figure 1. By default, video playback from DVD drives is disabled in ANY-maze, so you first need to enable it before you can play a DVD.

So, with a DVD drive fitted, a decoder installed and DVD playback enabled in ANY-maze, you're ready to play DVDs... well, almost. There is one restriction - ANY-maze can't play most commercial DVDs. This is because these discs usually include copy protection that prevents their images from being *captured* by ANY-maze - the images can only be *viewed*. However, this isn't really a problem, as you're unlikely to want to track anything in a commercial DVD.

Playing DVDs on the Video page

The easiest place to test DVD playback is on the Video page.

Here you will see your DVD drive in the list of available video capture devices, and simply clicking the entry will start playing the DVD in the drive. (If you don't see your drive listed, it's because you've not enabled it - see the previous section.)

When playing a DVD, the Video page will show the following controls:

- A button to pause playback

  (When paused, this will show a button to resume playback)
- A rewind button, to rewind playback to the start of the DVD
- A time counter, showing the current position in the playback
- A progress indicator, that can be dragged to change the current playback position
- A button, which displays a menu of all the titles and chapters on the DVD
- A button, to take you to the DVD's root menu

Navigating a DVD

To navigate within a DVD you can either use the menu button and select a title/chapter to watch, or you can use the DVD's own menu page by clicking the button.

If you use the DVD menu button to jump directly to a chapter on the DVD, you may see a 'No entry' sign in the top left of the video picture - this means that the DVD itself is preventing you from jumping directly to the chapter. There's nothing that can be done about this - it's something that was set on the DVD when it was authored. In this case, you'll need to navigate using the DVD's own menu page.

Most DVDs include a menu page where you can either choose a chapter to view, or you can just start playing the DVD from the start. Usually you navigate between these menu options using the up/down and left/right buttons on a remote control or, on a computer, you can click the options on the screen. Unfortunately, because ANY-maze is *capturing* the images from the DVD, rather than simply *displaying* them, clicking the options won't work; instead you can use the 'Remote control' buttons that ANY-maze displays below the video window when a DVD is menu is being shown - see figure 2.

In fact, if your mouse has a scroll wheel you can also navigate between the menu options by rolling the wheel, and clicking it to activate the selected option.

Figure 2. When a DVD menu is displayed, ANY-maze will show a set of 'Remote control'-style buttons in the toolbar below the video window. You can use these to navigate between the menu options.

Searching a DVD

If you record an experiment directly to a DVD, there's a good chance that the entire experiment will simply be recorded as a single chapter. In this case, you may want to be able to search through the chapter to find a specific moment in the experiment - you can do this using the search bar below the video window.

To search, simply drag the slider until you reach the desired point in the DVD. Note that this will search through all the chapters in a single title.

Tracking from a DVD

Of course, viewing a DVD is all very well, but what you'll actually want to do is *track* tests recorded on the DVD. To do this, you simply need to select the DVD as the *Source of video images* in a *video source* in your protocol.

When a DVD is used in a video source, all the options still work as normal - so you can crop the image, flip it and even zoom if you wish.

Having set up the protocol to use the DVD, you will then find that the Tests page shows the same DVD playback buttons as the Video page (see the previous section), so it's easy to pause playback, jump to different chapters etc.

Converting ANY-maze format videos to a standard format

Introduction

When videos of tests are recorded within ANY-maze, the system uses its own proprietary video format. This is because standard formats are computationally intensive to record, and if ANY-maze used them it would rapidly run out of processor resources when tracking and recording in multiple apparatus.

Although ANY-maze itself can play the videos it records, it is the only system which can, meaning that you can't include these videos in presentations or play them using any other software, such as Windows Media Player. However, it is possible to convert an ANY-maze format video to a standard format which *can* then be used in other programs.

Converting a video file

To convert an ANY-maze format video file to a standard format (or a number of files that all reside in the same folder on the computer), you should switch to the Video page and select the *Convert videos to standard format* button in the ribbon bar.

This page will allow you to select individual files (using the *Add file(s)...* button) or all files in a folder (using the *Add all files in a folder...* button). These will be added to a list, and when you select the *Start conversion* button, these files will be converted. This conversion will happen in the background, and you can carry on using ANY-maze while this conversion takes place.

You can choose where you want the converted files to be saved, as well as the quality of the conversion - better quality will take longer and result in larger file sizes. You usually won't need to change these settings from the default, which is 'Medium quality'.

The conversion process will occur in the background, allowing you to use ANY-maze while it's happening, and the progress will be indicated in the status bar at the bottom of the screen. Clicking on the status bar panel will show the current progress in more detail, and allow you to cancel the conversion if you want. Any errors that occur in the process will show in the status bar and a small window will pop up from this panel, showing any files that have not been converted.

Converting a video by recording it

Another way of converting an ANY-maze format video to a standard format is simply to record it to the standard format as you play it. To do this, you should follow these steps:

1. Switch to the Video page.
2. Click the *Record a standard format video* button in the ribbon bar - do this *before*

playing the video you want to convert.

3. A message will be displayed; read it and click OK. At this point ANY-maze will know you want to record the video as a movie, and it will be waiting for you to play the video.
4. Use the *Browse for a video file...* option in the *Video files* section of the left-hand side bar to open the ANY-maze format video that you want to convert.
5. When ANY-maze displays the first frame of this video, it will remember that you want to convert the file to an AVI movie and will pause playback. It will open a window where you can enter the name of file you want the movie to be saved to. You can also change the compression settings used if you wish.
6. When you click *Save* on this window, the video will start to play, and it will simultaneously be recorded in AVI format to the file you specified. While recording, a flashing red dot will be shown in the bottom right-hand corner of the video display.
7. When the video reaches the end, ANY-maze will automatically end the recording. If you want to end it before this, just click the *Record a standard format video* button in the ribbon bar again.

Converting a section of a video

The above instructions can be used to convert an entire video (or multiple video files), but you may wish to just convert a short section of an ANY-maze video, perhaps to show a typical behaviour that you want to mention in a presentation.

Here's how to do this:

1. Switch to the Video page.
2. Use the *Browse for a video file...* option in the left-hand side bar to open the ANY-maze format video that you want to convert.
3. Use the search bar, underneath the video image, to navigate to the start of the section you want to record.
4. Click the *Record a standard format video* button in the ribbon bar - playback will automatically be paused, and a window will open where you can enter the name of the file you want the movie to be saved in. You can also, optionally, change the compression settings used.
5. When you click *Save* on this window, playback will restart and the video recording will begin. While recording, a flashing red dot will be shown in the bottom right-hand corner of the video display.
6. When the video reaches the end of the section you want to record, click the *Record a standard format video* button in the ribbon bar again - or wait until the video reaches the end, at which point recording will stop automatically.

See also:

- Errors while converting videos

Errors while converting videos

Introduction

When using the Video page to convert ANY-maze videos to a standard format, there are a number of reasons that this conversion process might fail. This topic describes those reasons and why they might occur.

Details

The conversion process will occur in the background, and the current progress will be indicated in the status bar at the bottom of the screen. Clicking on the status bar panel will pop up a window showing the progress in more detail, and allowing you to cancel the conversion if you want.

Any errors that occur will also be shown in this pop-up window; these errors are described in more detail below.

Video files that use a legacy format

The proprietary format of videos used by ANY-maze has gone through some improvements over the years. Some very old versions of these files use a format that is not supported by the video conversion utility.

You can still convert these videos by playing them on the Video page and using the *Record a standard format video* option in the ribbon bar to record them to a standard format file.

Video files that cannot be opened

Occasionally, the ANY-maze video to be converted cannot be opened. This may be because it's in use, for example in a test currently being performed by ANY-maze. You'll need to make sure that the video is not in use before trying again.

Destination files that cannot be created

ANY-maze creates a new standard format video file to record the ANY-maze video into. For various reasons, it might not be possible to create this file - for example, the folder it's trying to create the file in might be read-only, or it may be on a network drive to which you don't have permission to save files. If you have converted the same video file before, and viewed it in a movie player (such as Windows Media Player), it might be that you've forgotten to close down the movie player and so it won't allow the file to be overwritten. You'll need to make sure that the video is not in use before trying again.

Other errors

There are a few other reasons that a file might fail to convert; these are given in detail under the file name. These could be temporary errors - such as the system not having enough memory available - or a specific problem with the video file itself. Try and convert these files again (after closing other applications to free up memory if necessary) and if the problem persists, contact ANY-maze technical support and send us the video - you can do this using the Send us files option on the Support page.

The Options page

Unlike the pages on the left side of the ribbon bar, you can access the Options page without having to open an experiment first.

Overview

The Options page in ANY-maze allows to you to alter many aspects of the software. Although some of the options, such as changing colours or fonts, appear to be purely cosmetic they can be extremely useful if, for example, the colour used to highlight zones doesn't show up too well on your apparatus or if the font is too small for you to read comfortably.

An important point about ANY-maze options is that they're saved on a per-user basis. This means that changes made by one user won't affect anyone else. For example, if you choose to alter the colours of your graphs and set a large font size, this won't inconvenience the other users.

Using the Options page

The Options page is divided into a number of different settings, each of which contains options related to a certain aspect of the program. To switch between the pages just click the item in the list on the left-hand side of the page - see figure 1.

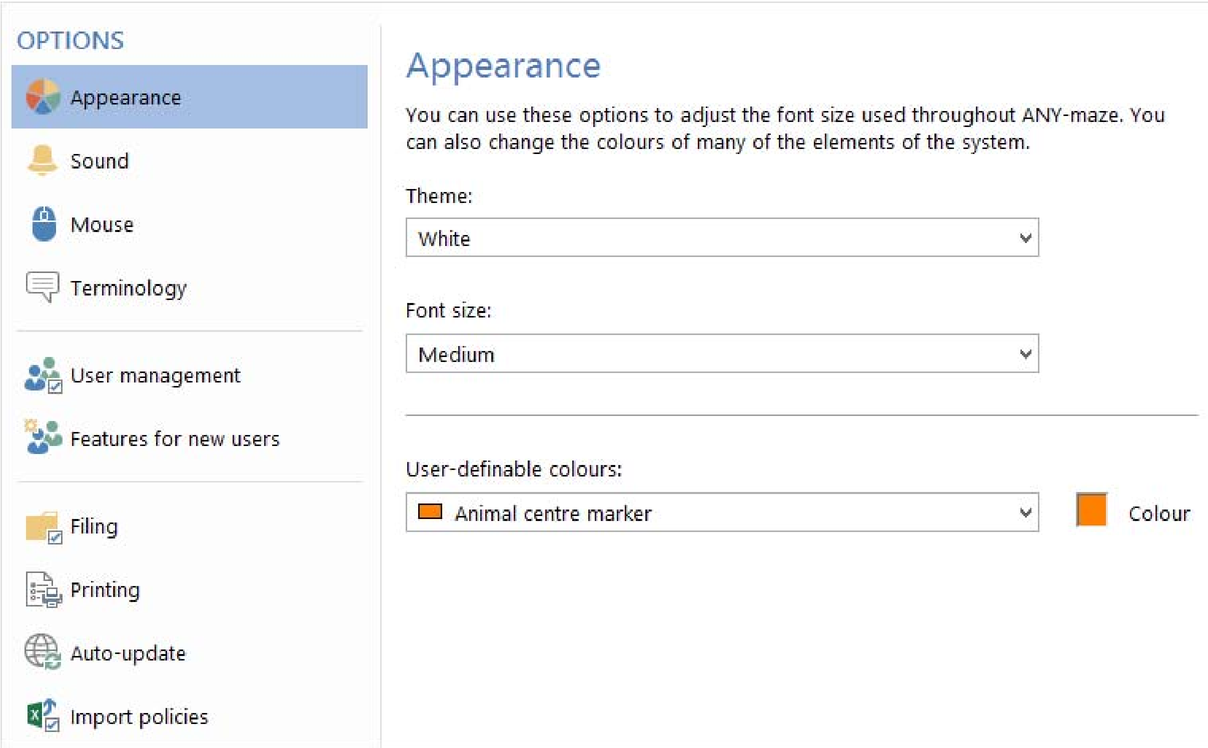


Figure 1. The ANY-maze Options page. Note that the exact options available depend on how your copy of ANY-maze is set up, and whether or not you're an ANY-maze administrator.

All changes that you make on the options page take effect immediately. For example, if you change the font size under the Appearance settings, the options page will change to be displayed using the new font size.

For more information about the specific options available, refer to the following help topics:

- Changing the appearance of ANY-maze
- Changing ANY-maze sounds
- Mouse options
- Changing ANY-maze terminology
- Resetting hidden messages

- Administrative options
- Managing users in ANY-maze
- Filing options
- Printing options
- Configuring automatic updates
- Settings for importing animals and tests
- Settings for reporting usage statistics

ANY-maze help topic H0851

Changing the appearance of ANY-maze

Overview

You can change the theme and font used by ANY-maze, as well as the colour of some visual elements such as graphs and tracking markers. Apart from having a cosmetic benefit this can be very useful - if, for example, the colour used to highlight the animal when tracking doesn't show up too well on your apparatus, then you can change it.

- Accessing the settings to change the appearance of ANY-maze
- Changing the current theme or font size
- Changing colours of visual elements

Accessing the settings to change the appearance of ANY-maze

To access the *Appearance* settings, select the Options page and then click the *Appearance* option in the left-hand side bar.

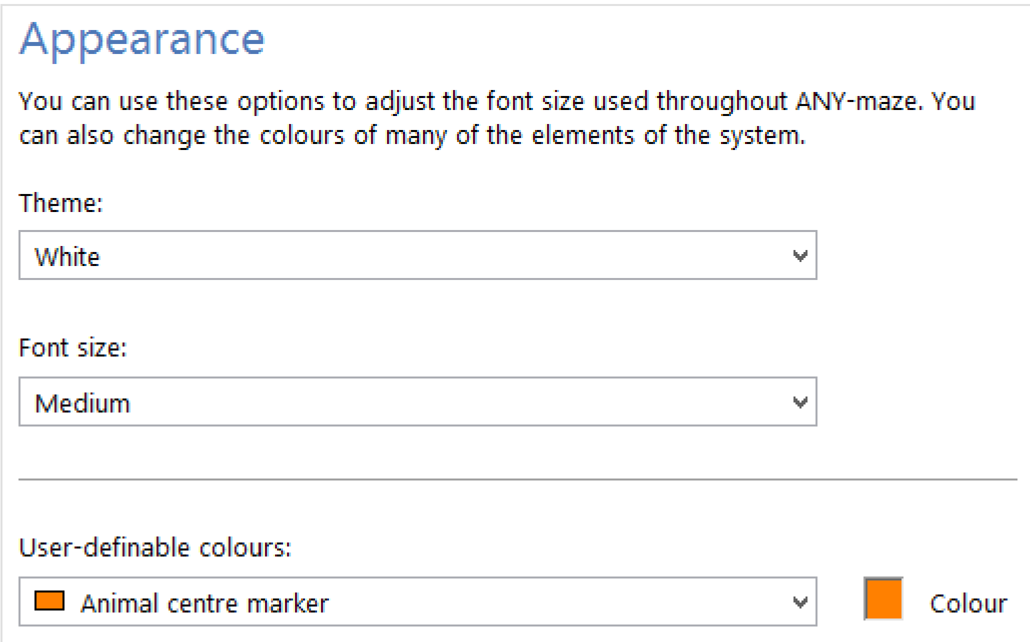


Figure 1. The Appearance *settings on the ANY-maze Options page.*

Changing the current theme or font size

The visual appearance of ANY-maze is controlled by a *theme*. This is a single setting which controls the font and colours of all the windows and pages in the ANY-maze software.

There are currently two themes available in ANY-maze; more may be added in future.

Although the font used is determined by the selected theme, the *size* of the font can be altered using the *Font size* drop-down list. The default size is medium (which is about a 10-point font on the default theme) but it can be changed to small or large (which are about 8-point and 12-point respectively).

Note that the font you choose will be scaled according to the current DPI settings in Windows; that is, if your Windows resolution settings are set to scale to 200% DPI, then the font (along with the size of other user-interface elements in ANY-maze) will double in size to take account of this.

Changing colours

Although the colours of the user-interface elements of ANY-maze (windows, fonts etc.) are determined by the current theme, there are a number of elements that are specific to the experiments that you'll perform whose colour you can change independently. This includes:

- The tracking markers that can be shown on the apparatus image (head, centre and tail)
- The shading colours for the animal, or the zones it enters
- The default colour for state channels on charts
- The colours of lines, columns or points on graphs on the Results page

To change the colour of a specific element of ANY-maze:

1. Select the element in the *User-definable colours* drop-down list - its current colour will then be shown in the colour box to the right of the list.
2. Click the colour box - the standard *Colour picker* window will open. You'll probably have seen this window before, as it's used in many Windows programs.
3. Select the colour you want to use, either by clicking one of the predefined colours shown on the left of the colour picker window or by clicking the colour swatch to choose the colour's Hue and saturation and then moving the slider in the colour gradient to select its luminescence.
4. Click OK - the *Colour picker* window will close and the element will be listed with the colour you chose.

Some of the elements are listed as 'Hue only'. This means that the only the 'type' of colour (blue, red, yellow etc.) can be changed, and not the brightness. This is because ANY-maze controls the *brightness of these elements itself - a good example of this is the colour used to highlight the active zone when tracking.*

If you wish to revert to the system's original settings, you can simply click the *Restore defaults* button in the ribbon bar.

See also:

- ANY-maze options
- Tracking the animal's head & tail
- What to display while testing
- Specifying zone highlighting
- An introduction to charts
- Setting up chart state channels

Changing ANY-maze sounds

Overview

ANY-maze includes the ability to play sounds when certain events occur, such as the end of a test or when an information message is displayed.

The sounds that the system can play are stored in '.wav' files, and the Windows operating system is supplied with a wide range of different sounds recorded in this format. In fact you can even record new sounds yourself using the Windows Sound recorder - so, for example, you could record a .wav file of you saying "Test ended" and then set the system to play this sound whenever a test ends.

By the way, although you need a sound card installed in your computer to take full advantage of the system's ability to play sounds, you can still use sounds without having a sound card - but you'll be limited to just a simple beep.

- Accessing the sound settings
- Choosing the sound to play when an event occurs

Accessing the sound settings

To access the *Sounds* settings, select the Options page and then click the *Sound* option in the left-hand side bar.

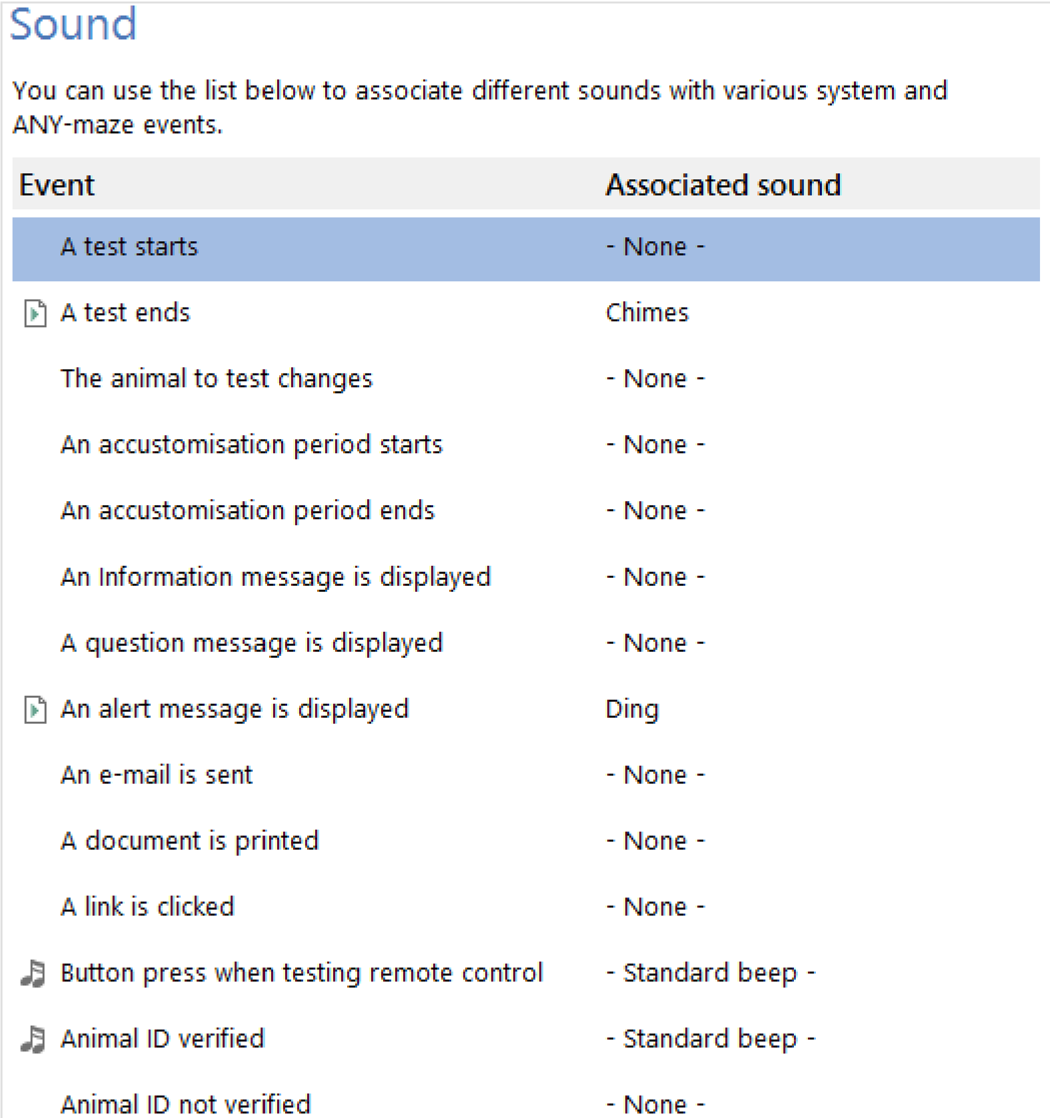


Figure 1. The Sound *settings on the ANY-maze Options page.*

Choosing the sound to play when an event occurs

ANY-maze includes several different events which can cause a sound to be played, and you can choose to associate any sound, or silence, with each of them. To do this:

1. Select the event whose sound you want to set, for example *A test ends*.
2. Click the *Choose sound* button in the ribbon bar to drop a list of available sounds that you can associate with the event. By default, this list includes all the sounds installed in the 'Media' folder of your computer. If you don't want any sound

associated with the event then choose the first item in the list: - *None* -.

3. As you select a sound in the list, it will play once to give you a preview. Once a sound is associated with the event, you can review it by selecting the sound in the list and clicking the *Play selected sound* button in the ribbon bar.
4. If the sound you want isn't listed in the sounds list, then you can select the *Browse for sound file...* option in the list to browse through the folders on your computer to find a .wav file to associate with the event. Note that when you select a sound file in this way, ANY-maze will automatically add all the sounds in the folder to the list of sounds. This makes it more convenient for you to choose other sounds from the same folder without having to 'Browse' each time.

You may find that not all of the sounds provided by Windows are installed on your computer. You can use the 'Add/Remove Windows components' option of the 'Add/remove programs' element of the Windows 'Control Panel' to add more sound schemes.

If you want to remove all associated sounds, i.e. you want all the listed events to happen silently, use the *Clear all sounds* button in the ribbon bar to set all the sounds to - *None* -.

Choosing a sound file

Introduction

You should use this window to choose a sound file you would like to associate with an event in ANY-maze.

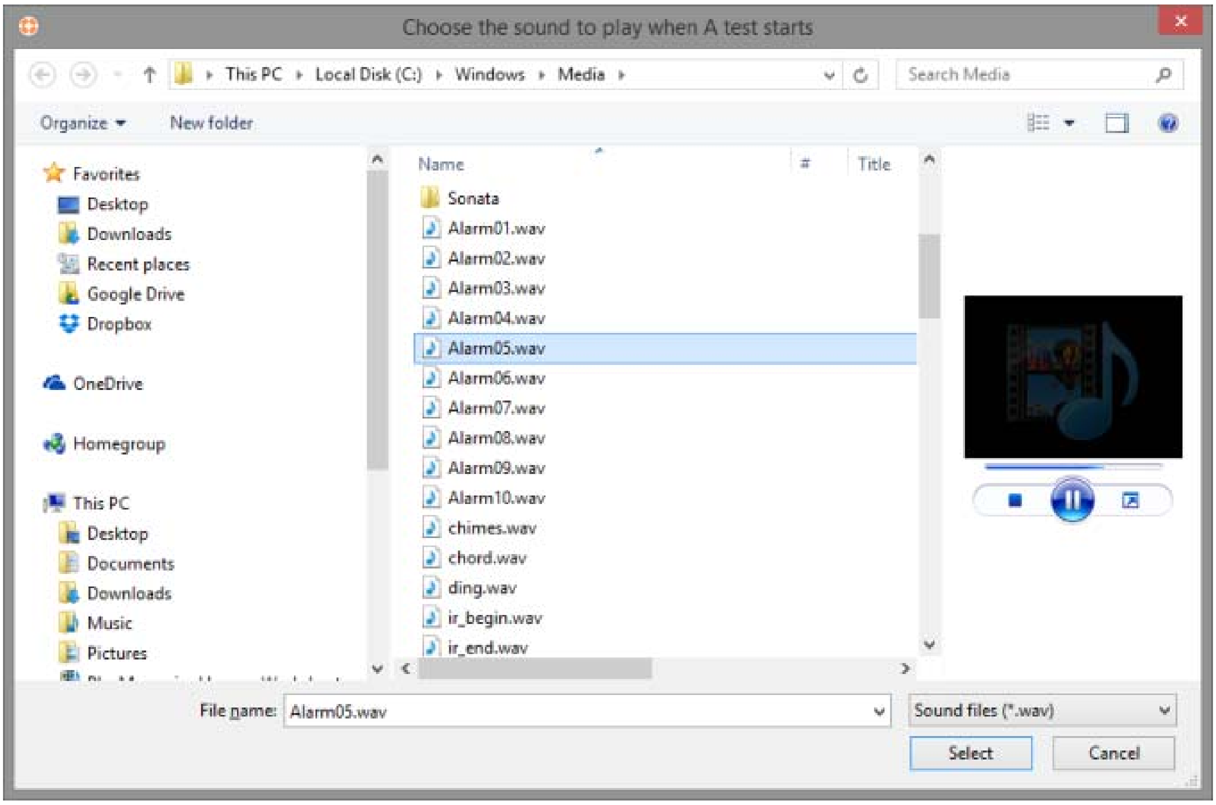


Figure 1. The 'Choose the sound to play' window.

Details

You will probably be familiar with this window already, as it's based on the standard 'open file' window used in almost all Windows software.

Files list	The files list shows the folders and sound (.wav) files stored in the current location. You can double click a folder to open it, or you can click a sound file to transfer its name to the *File name* field.
File name	You can use this field to type in the name of the sound file you want to use, but it's generally much easier just to choose it from the files list.
Type of file	In fact, there is only one choice here, 'Sound files', so you can ignore this option.

See also:

- Changing ANY-maze sounds
- ANY-maze options

Mouse options

Overview

ANY-maze includes an enhancement to scrolling using a mouse wheel, designed to make the system easier and more intuitive to use. However this enhancement is incompatible with the 'Auto scroll' or 'Universal scroll' features of Logitech mice.

Using this page you can choose whether or not to use the ANY-maze mouse wheel scrolling enhancement.

To access the *Mouse* settings, select the Options page and then click the *Mouse* option in the left-hand side bar.

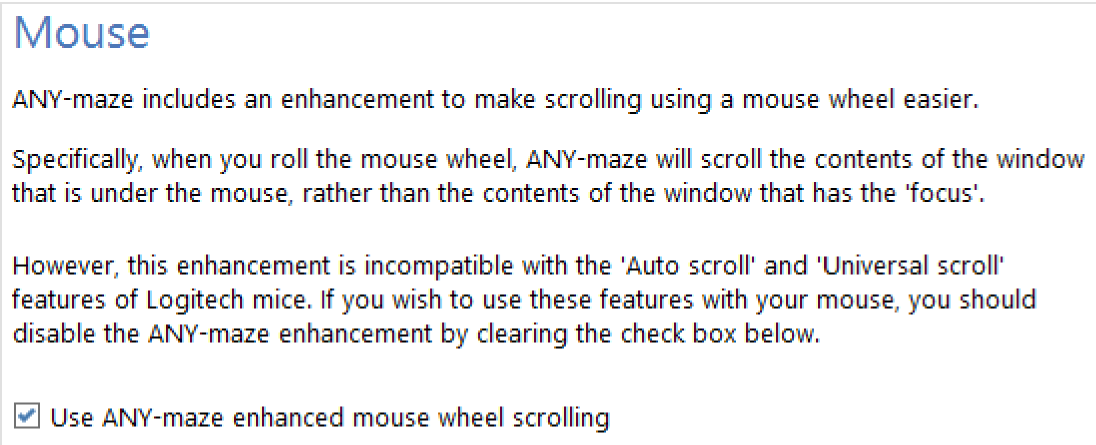


Figure 1. The Mouse *settings on the ANY-maze Options page.*

Details

Normally when you roll a mouse wheel, Windows will scroll the contents of the window that has the 'keyboard focus' - this is usually the window that contains a cursor where you can type, or a window that has some type of focus indicator, such as a high-lighted item, within it.

Although this is the standard behaviour for Windows, it can be rather irritating. For example, if you wish to scroll the protocol list in ANY-maze, you first need to click an item in the list (so the list has the focus) and then roll the mouse wheel.

To overcome this limitation, ANY-maze includes a simple enhancement to mouse wheel scrolling - rather than scrolling the window with the focus, ANY-maze scrolls the window under the mouse cursor - this is simpler and more intuitive.

However, Logitech mice (and perhaps other brands too) include their own scrolling enhancements such as *Auto scroll* and *Universal scroll,* and these are incompatible with this feature of ANY-maze. In general, if you try to use these Logitech features in ANY-maze the program closes immediately.

There are two solutions to this issue: you can either reconfigure your mouse so it doesn't use the Logitech features (which you can do using the Mouse item in the Windows right clicking on the Start button in the bottom left-hand corner of the screen.>'Control Panel'), or you can switch off the ANY-maze mouse wheel scrolling enhancement, which you can do using the option on this page.

See also:

- ANY-maze options

Changing ANY-maze terminology

Overview

ANY-maze is full of terms such as *Treatments, Apparatus, Zones, Measures,* etc. While the meaning of these terms is hopefully clear, it might be that you're accustomed to calling *Zones* 'Areas' or you might find it clearer if *Measures* were referred to as 'Parameters'.

While you can probably survive the inconvenience of having to adapt to new terminology, you might prefer it if the system could adopt your terminology rather than the other way round. Well it can - that's what the terminology settings on the Options page are for. Here, for example, you can tell the system that rather than calling a zone a *zone*, it should call it an 'area.' There are quite a few terms that you can redefine, and while you probably won't want to change them all, you may find it useful to alter a few.

By the way, changes to the terminology affect everything in the system - menus, windows, reports, and even these help topics!

- Accessing the terminology settings
- Changing terms

Accessing the terminology settings

To access the *Terminology* settings, select the Options page and then click the *Terminology* option in the left-hand side bar.

Terminology

You can use this window to change the terminology used by ANY-maze to match your own.

Original term	New term	Original term	New term
Accustomisation	Accustomisation	Retire	Retire
Accustomising	Accustomising	Retired	Retired
Animal	Animal	Schedule	Schedule
Apparatus	Apparatus	Segment	Segment
Blind	Blind	Sequence	Sequence
Block	Block	Sequential	Sequential
Cyclic	Cyclic	Speed	Speed
Delete	Delete	Stage	Stage
Deleted	Deleted	Test	Test
Episode	Episode	Time course	Time course
Experiment	Experiment	Time marker	Time marker
Immobile	Immobile	Travelled	Travelled
Immobility	Immobility	Treatment	Treatment
Latency	Latency	Trial	Trial
Measure	Measure	Undefined	Undefined
Mobile	Mobile	Virtual switch	Virtual switch
Mobility	Mobility	vs.	vs.
Protocol	Protocol	Zone	Area

Figure 1. The Terminology *settings on the ANY-maze Options page. You can see that the term for 'zone' has been changed to 'area'.*

Changing terms

To change a term you simply need to delete the entry shown in the relevant 'new term' cell and type in your term instead.

You don't need to worry about plural versions of terms, as ANY-maze will cope with this automatically. However, where a term is a particular word type, for example the past participle - as in *Deleted*, then you should ensure the new term is the same type, in this case perhaps 'Removed'.

You should match the case of the new term with that of the original term - in most situations this means that a new term should be in lowercase except the first letter, which should be a capital. For example, you'd want to replace the term *Deleted* with 'Removed' and not 'REMOVED' or 'removed'.

If you change some terms and later want to revert to the system's standard terminology, then you can simply click the *Restore defaults* button in the ribbon bar.

See also:

- ANY-maze options

Resetting hidden messages

Overview

In ANY-maze there are many messages which include an option for *Don't show this message again*. This option can be very useful, but it does mean that once you've chosen not to show a message again, it is hidden from you (as messages are hidden on a per-user basis) forever... well almost, the options to *Reset hidden messages* allow you to show these messages again.

Details

There are three types of messages which can be hidden in ANY-maze.

- Messages that include a *Don't show this message again* option. These are the most common type, and you've probably seen quite a few of them.
- *Useful to know* messages, that are shown above some reports and spreadsheets within ANY-maze. These messages include a close button, and if you select it the message will subsequently always be hidden from you.
- Balloon messages that describe the various formats in which times can be entered into ANY-maze. These are automatically shown to you just once for each time field in the system.

You can reset any of these messages - so they are shown to you again - using the options on this page. Simply check the box next to the message type(s) you want to reset and then click the *Reset* button at the bottom of the page.

See also:

- ANY-maze options

Managing users in ANY-maze

Overview

ANY-maze includes the concept of 'users', and the system records a range of information on a per-user basis. This includes things such as where files are stored, the layout of windows, the size of fonts, etc.

There are two ways to manage users - you can either just use your existing Windows user accounts, or you can manage specific ANY-maze user accounts from within the system itself. The pros and cons of the two approaches are described in the first of the topics below.

- An introduction to managing users
- Securing your ANY-maze system and data
- Switching from Windows to ANY-maze user management
- Managing ANY-maze user accounts
- Logging on
- Administrative options

An introduction to managing users

Overview

In most labs, ANY-maze will probably be used by more than one person. While this doesn't actually make any difference to the system, it could easily cause confusion or frustration if different users change the program's options and settings to those they prefer.

To avoid these types of problems, and to allow individuals to personalise ANY-maze so it suits their own way of working, ANY-maze stores options and other settings on a per-user basis. Of course, this means that the system has to know who's actually using it, and this is achieved by having user accounts - one for each user of the system.

- User accounts
- The settings which ANY-maze records for each user
- ANY-maze administrators
- Using ANY-maze or Windows to manage user accounts

User accounts

Although user accounts may sound a little complicated, they're actually very simple - essentially they're just a list of the names of the people who might use ANY-maze.

For example, imagine Dave, John and Mary all work in the same lab. They might all want to use ANY-maze, so you'd set them up to each have a user account. Then, each time one of them starts using ANY-maze, they'd simply have to identify who they are - ANY-maze would then be able to use their personal settings without interfering with those of the other users.

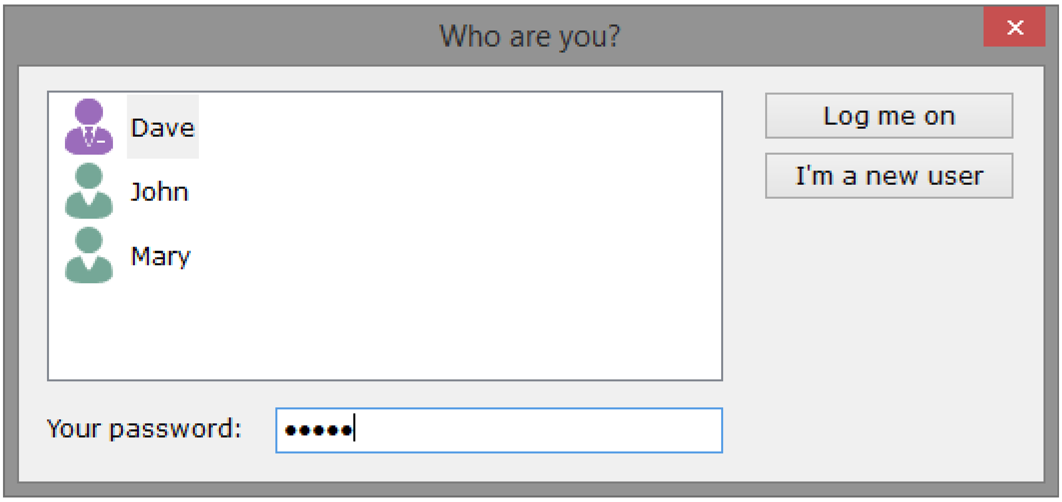


Figure 1. The ANY-maze log-on window - to start using the system, users simply have to identify themselves.

The settings which ANY-maze records for each user

Of course, the actual benefit of having different settings for each user depends on what those settings actually are. In ANY-maze, the system will maintain the following information separately for each user:

- The user's name.
- An optional password.
- The position and size of the ANY-maze window, the last time it was used (and that of the popped-out help window, if applicable).
- Some of the information entered on the Options page, for example:
    - The visual theme and font size used by ANY-maze.
    - The colours of some elements of the software.
    - Sounds which should be played when specific events occur - such as the end of a test.
    - The terminology used within ANY-maze - for example, you can ask the system to refer to *zones* as 'areas', or *measures* as 'parameters'.
    - Whether the user's password should be used to protect their experiment and protocol files.
    - The folder where the user will generally store his/her experiment files.
    - Whether the system should automatically save experiments when they're closed, and whether it should automatically keep backup copies.
    - The headers and footers to be printed on documents, and a note as to whether documents should be printed in colour or black and white.

ANY-maze administrators

To actually manage user accounts requires an administrator. As the name implies, this person administers the accounts - adding new ones, deleting old ones etc. (see figure 2, below). In fact, in ANY-maze an administrator has other abilities as well - specifically, administrators can:

- Activate / deactivate an ANY-maze licence.
- Update the ANY-maze software.
- Install and uninstall drivers.
- Set various global options which affect all users.

As you can see, an administrator doesn't just manage the ANY-maze user accounts, but administrates the entire system - for this reason it's normal to have just a single administrator.

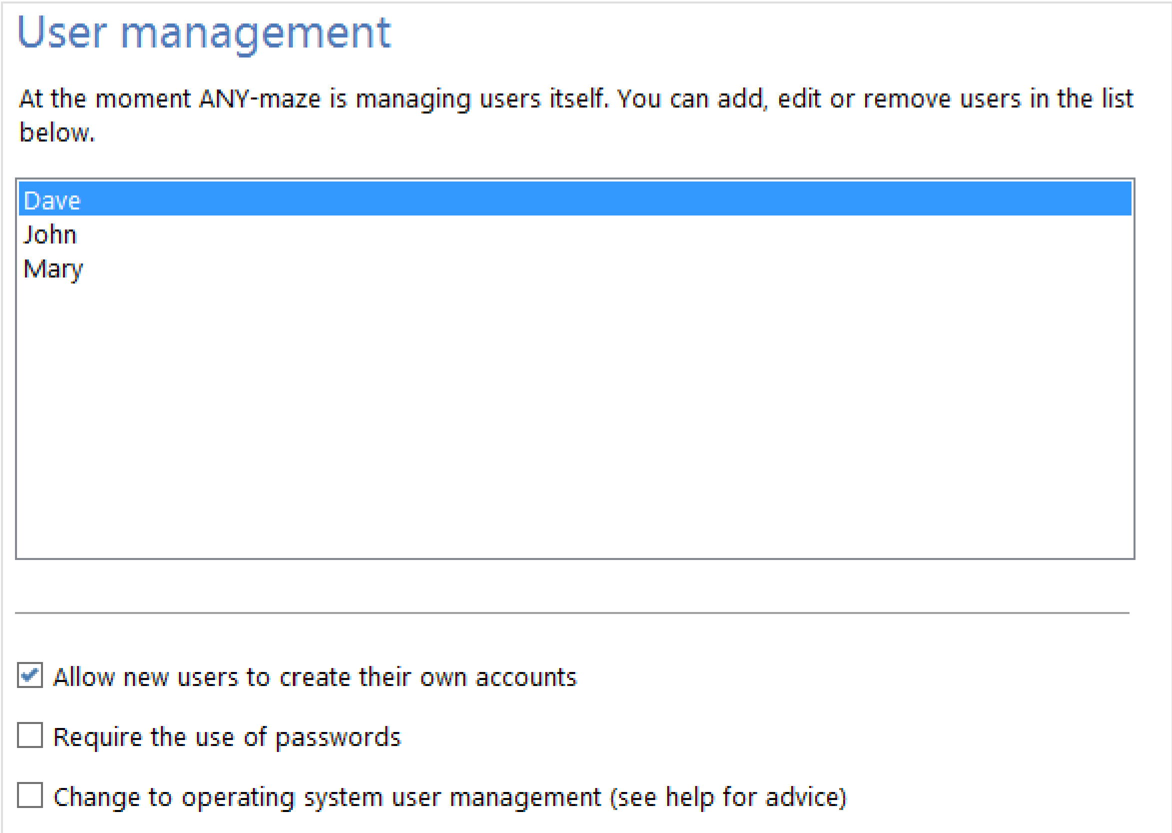


Figure 2. The User Management *section of the ANY-maze Options page. Using this page, an ANY-maze administrator can add, edit or delete users and set overall user policies.*

Using ANY-maze or Windows to manage user accounts

You're probably used to the fact that whenever you start Windows, you have to provide a 'log-on' name and password. This is because Windows itself has user accounts, and you can't start using the system without first identifying who you are.

Clearly then, in these versions of Windows, the system already knows who you are, so why do you need to identify yourself in ANY-maze too? Actually, you don't - if you wish to, ANY-maze can simply use these Windows user accounts to differentiate users, rather than maintaining user accounts itself.

So in effect, there are two ways to manage users in ANY-maze - either by using Windows user accounts, or by getting ANY-maze to manage its own accounts. Here are some notes to help you decide which option is best for you.

Advantages of using Windows user accounts

- Using Windows users is more secure. ANY-maze users are not intended to be highly secure - the intention being simply to provide a way to separate user information, rather than secure it.
- If you already have a well-defined policy for Windows users, with personal secured folders etc., then using Windows users will mean ANY-maze simply slots into your existing strategy.

Advantages of using ANY-maze user accounts

- The ANY-maze user log-on window is very simple and easily understood - see figure 1, above. Indeed the administrator can decide whether passwords are actually required or not.
- Managing users in ANY-maze is very simple - see figure 2, above.
- The administrator can decide whether new users can create their own accounts, or whether the administrator has to do this for them. Letting new users create their own accounts is a handy way to reduce bureaucracy and help prevent 'sharing' of user accounts (which would rather defeat the point of having them).
- Users can set the system to automatically protect their experiment and protocol files using their passwords - this helps prevent situations in which one users alters another's files. Note that you can protect files when using Windows users too - just not automatically.
- ANY-maze's self-repairing features, which automatically re-install ANY-maze system files if they get deleted, will work for all users - when you use Windows user accounts, they'll only work when an administrator is logged on.

In summary, unless you already use Windows user accounts and you're comfortable managing them, then it's probably best if you use ANY-maze to manage its own user accounts.

By the way, if you choose to use Windows user accounts then only *Windows administrators* will be able to perform the restricted operations described above.

See also:

- Managing users
- How to change from ANY-maze to Windows user management
- How to change from Windows to ANY-maze user management
- Editing your user name and password

Securing your ANY-maze system and data

This topic describes methods of securing ANY-maze using its own user accounts, and other security features. If you have chosen to use Windows user accounts, then you should also refer to the Windows documentation for details of how to secure your system.

Overview

ANY-maze is not intended to be a highly secure system, but it does include some security features that are intended to protect your ANY-maze settings and data from casual interference. For example, it's easy to protect an experiment file using a password, which will prevent another user from opening the file and altering something - either accidentally or intentionally.

However, if you have critical security concerns, such as the threat of commercial espionage, then the security features of ANY-maze won't provide sufficient protection and you should seek a more robust solution.

- Limiting access to ANY-maze
- Securing per-user settings
- Protecting experiment and protocol files
- Other security options

Limiting access to ANY-maze

Limiting access to ANY-maze is a simple and effective first line of defence against either inadvertent or malicious attempts to access your data.

ANY-maze includes its own system for managing users, and you can use this to limit access to a select group of individuals. If your concern is to secure the system, then you should ensure that the user policies are set so that new users CAN'T create their own accounts, and that passwords ARE required.

Note that ANY-maze user accounts are *casually* secure - a specialist could probably *crack* them with minimal effort.

You'll find more information in the following topics:

- Managing users - Setting user policies

Securing per-user settings

Securing a user's settings (such as those entered on the Options page) is not really a necessary action, as settings are unlikely to be confidential. Nevertheless, protecting settings can still be a good idea as it can help avoid frustration and confusion when one user either accidentally or mischievously alters another's settings.

To secure users' settings, you should ensure that the user policies are set so that passwords ARE required.

You'll find more information in the following topics:

- Managing users - Setting user policies

Protecting experiment and protocol files

Protecting ANY-maze experiment and protocol files is often highly desirable even if the files don't contain highly confidential data - if you've ever found yourself wondering how something was deleted from a document you were writing, or how your spreadsheet suddenly no longer works, then you'll know what I mean!

Files in ANY-maze can be protected using a password - this can be the same as your user password, or different - it's up to you. When you save a file, you supply the password you want to use - then whenever you open the file, you will need to enter the password before you can access it.

In fact, ANY-maze also encrypts password-protected files so they can't be read using any other software either. In this respect, the files really are quite secure (although they probably wouldn't pose much of a challenge to a true expert).

By the way, if you think you'll *usually* want to secure your files, then you can set ANY-maze to automatically protect them using your ANY-maze user password. This has the added benefit that ANY-maze won't require you to enter your password each time you open a protected file, provided you're logged on under your own user account (and you haven't changed your user password since you saved the file).

You'll find more information in the following topics:

- Saving experiments
- Saving protocols
- Opening a password-protected file
- Editing your user account details

Other security options

Although the above features provide a reasonable level of *casual* security for your ANY-maze system and files, they are by no means a complete solution. For example, while ANY-maze can protect your files so other users can't *open* them, it can't prevent someone else from moving, copying or deleting them.

You can improve security by using the Windows operating system's native capabilities to protect access to individual drives and folders. If you decide to do this, then you will want to set up individual Windows user accounts for all your ANY-maze users, in which case you should probably also change ANY-maze so it uses Windows users rather than managing its own user accounts.

You'll find more information in the following topics:

- Managing users - Changing to operating system user management
- An introduction to managing users

Switching from Windows to ANY-maze user management

The User management *settings are only included on the Options page if you're an administrator. The page described here is the page displayed when using Windows to manage users. To view details of the page displayed when using ANY-maze to manage users, see* Managing users.

Overview

When you're using Windows user accounts in ANY-maze, there are no user management functions available within the system - instead you should use the appropriate tools within Windows itself to create, edit and delete users.

Nevertheless, there is still a *User management* section on the Options page, which you can use to change from Windows user management to ANY-maze user management.

To access the *User management* settings, select the Options page and then click the *User management* option in the left-hand side bar.

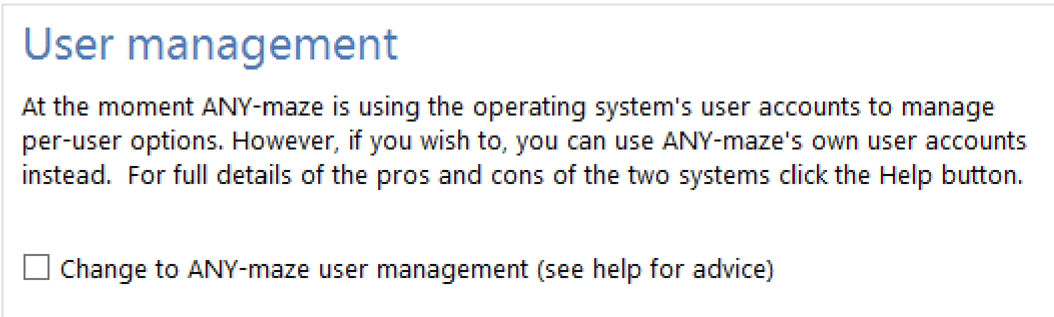


Figure 1. The User management *settings on the ANY-maze Options page, when Windows is being used to manage users.*

Changing to ANY-maze user management

Changing to ANY-maze user management is a good idea if users don't have individual Windows user accounts set up on the machine and/or you're not experienced with managing user accounts in Windows. There's a full discussion about the pros and cons of using ANY-maze or Windows user accounts in the help topic An introduction to managing users.

Managing ANY-maze user accounts

The User management *settings are only included on the Options page if you're an administrator. The page described here is the page displayed when using ANY-maze to manage users. To view details of the page displayed when using Windows to manage users, see* *Switching from Windows to ANY-maze user management*.

Overview

The principal role of an ANY-maze administrator is to manage ANY-maze user accounts. This involves creating new accounts, deleting old ones and setting the overall ANY-maze user policies.

For information on how to use this page to manage users, please see:

- Accessing the user management settings
- Adding, editing and deleting users
- Setting user policies
- Changing to operating system user management

Accessing the user management settings

To access the user management settings, select the Options page and then click the *User management* option in the left-hand side bar. If this option doesn't appear on the Options page, it's because you're not an ANY-maze administrator.

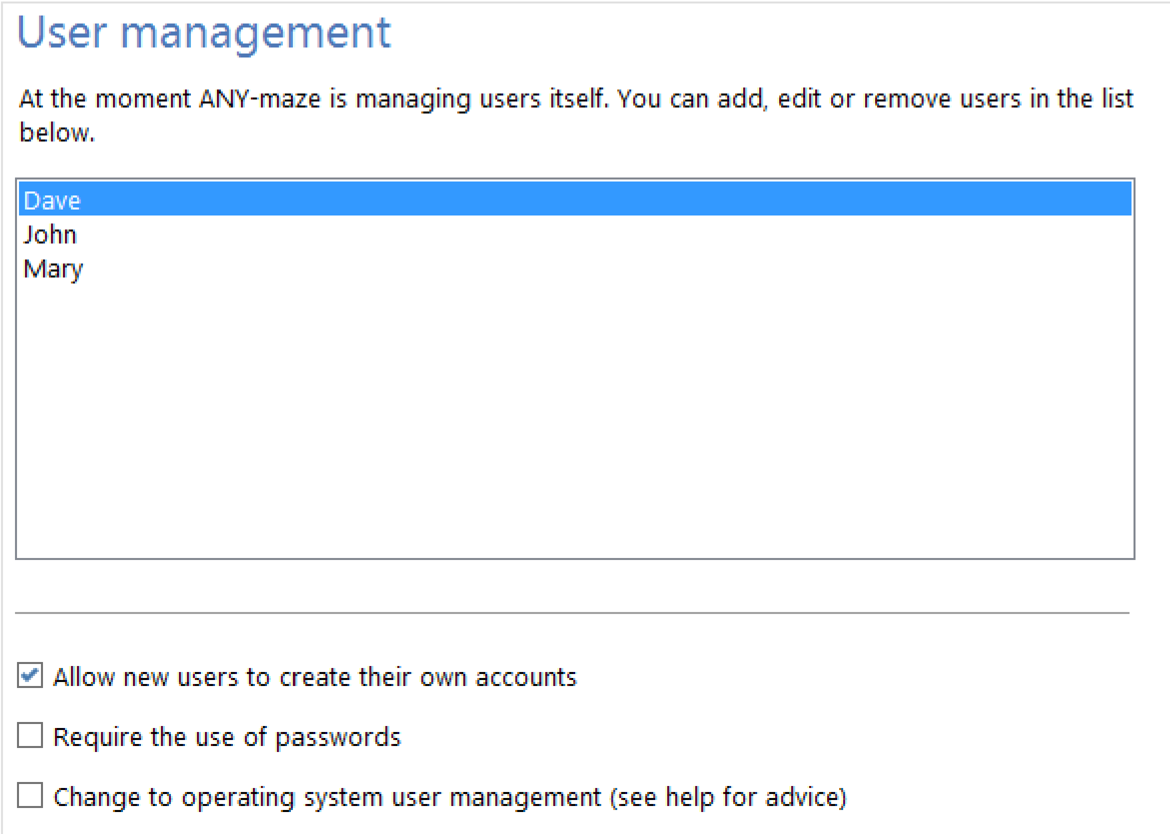


Figure 1. The User management *settings on the ANY-maze Options page, when ANY-maze is being used to manage its own users.*

Adding, editing and deleting users

Adding a user	To add a new user, simply click the *Add user* button in the ribbon bar. This will open the User settings window where you can enter the user's name and password.
Editing a user's details	To edit a user, first select the user in the list and then click the *Edit user* button in the ribbon bar. The *User settings* window will open where you can edit any of the user's details.
Deleting a user	To delete a user, first select the user in the list and then click the *Delete user* button. ANY-maze will ask you to confirm your decision before deleting the user. Note that deleting a user removes them from the user list and deletes all their settings; however it doesn't delete their personal ANY-maze folder or any experiment files it might contain.

Setting user policies

Allow new users to create accounts If you select this option, then ANY-maze will include a button labelled *I'm a new user* on the log-on window - see figure 2. If a user clicks this button then they will be asked to provide a user name and password, and ANY-maze will then create a new user account for them.

Whether you will want to use this facility will largely depend on who you expect to use ANY-maze. For example, in a commercial lab there will probably be a small, clearly defined group of ANY-maze users and allowing other individuals to create their own ANY-maze user accounts would probably be inappropriate. On the other hand, in an academic setting many individuals may wish to use ANY-maze, and by allowing them to create their own accounts the system is likely to remain better organised, as without this facility, lots of people would probably end up sharing accounts or trying to guess other users' passwords so as to gain access to the system.

Require the use of passwords Bearing in mind that the primary intention of ANY-maze user accounts is to provide *per-user* options rather than to secure the system, you may wish to relax the requirement for users to enter a password in order to use the system. Note that this doesn't prevent users from providing a password for their account if they wish to, it just doesn't make it obligatory.

Again, whether you wish to use this option or not will largely depend on the circumstances of your lab, although it's generally popular with users to avoid having 'yet another password'!

Note: If you change the policies from NOT requiring passwords to requiring them, then any users who don't have passwords will be asked to provide one the next time they log-on.

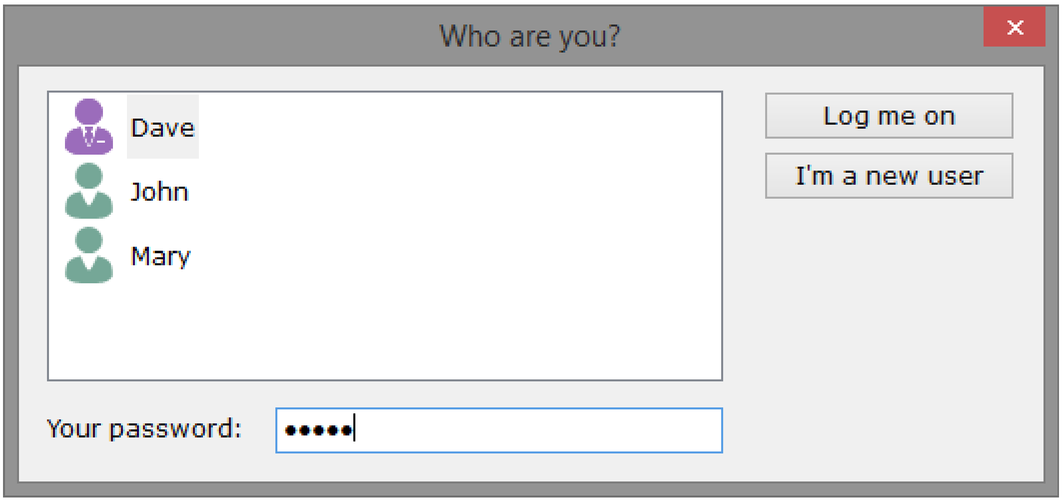


Figure 2. The ANY-maze log-on window, including the I'm a new user *button. By clicking it, new users can set up their own ANY-maze user accounts.*

Changing to operating system user management

Although ANY-maze can manage its own users, you may wish to use Windows user accounts instead. This is generally only a good idea if users already have individual Windows user accounts on the machine. There's a full discussion about the pros and cons of using ANY-maze or Windows user accounts in the help topic An introduction to managing users.

User settings

Overview

This window is used to create or edit an ANY-maze user's account.

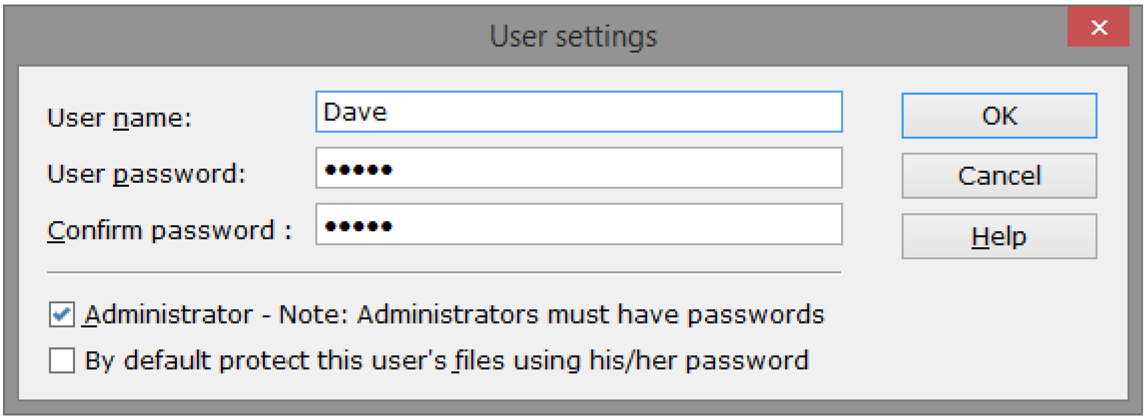


Figure 1. The User settings *window, where an administrator can enter or edit a user's account details.*

Details

User name	Enter the name of the user. You can enter anything you like up to a maximum of 24 characters, although the name must be unique amongst the names of existing users.
User password	Enter a password for the user's account. Depending on how you've set the user policies, you may not **have** to enter a password, but if you do then it must be at least 5 characters long and different to the user's name. Note that all leading and trailing spaces are automatically removed and aren't counted in the 5 character requirement - for example, entering ' dog' will not be accepted (because the two spaces at the start of the password would be ignored and the password would be seen to have just 3 characters). Spaces *within* the password, such as 'my dog', are acceptable.
Confirm password	Simply re-enter the password, to confirm that the entry you made is what you intended.

Administrator	If this user is to be an ANY-maze administrator, then check this box. Administrators **must** have passwords, irrespective of the user policies. Administrators can: • Create, edit and delete user accounts • Set user policies • Change administrative options • Install updates to the system • Activate/deactivate ANY-maze licences • Change the location of shared ANY-maze files In general, we recommend that there's only one ANY-maze administrator per system.
By default protect this user's files	When saving an Experiment file or Protocol file, users can optionally provide a password that will be used to 'lock' the file so other people can't open it - this facility is explained more fully in the Saving experiments and Saving protocols topics. If it is anticipated that a user will generally take advantage of this facility, then checking this box will set ANY-maze to automatically use the user's password to protect any files that they save. Of course, as this is only a default, users can still change an individual file to have a different password (or no password at all).

See also:

- Managing users
- Administrative options
- Installing ANY-maze updates
- The ANY-maze licence manager
- Video interfaces
- Filing options
- Saving experiments
- Saving protocols

Logging on

Overview

Before you can start using ANY-maze, you may need to identify yourself to the system. You do this by simply selecting your name from a list of ANY-maze users and, if required, entering your password - see figure 1.

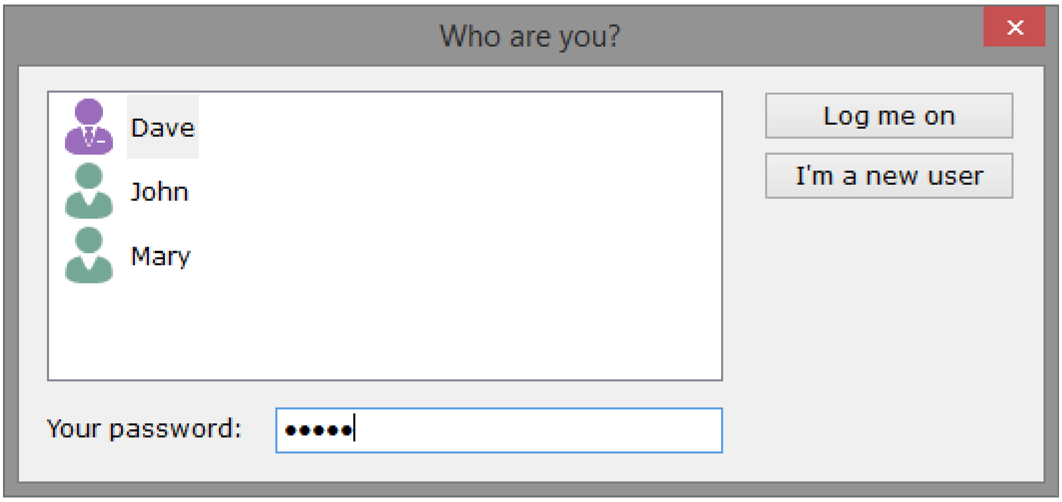


Figure 1. The ANY-maze log-on window.

Details

If you click the close button on this window, then ANY-maze will close.

If your account doesn't have a password, then you can log on most quickly by simply double clicking your name in the list of users.

The user list

This list shows all the users currently registered with the system. Administrators, such as Dave in the above figure, are shown with a purple coloured image.

If your name is not included in this list, then you've not been set up as a user on this copy of ANY-maze. However, depending on how the user policies have been set, you may be able to do this yourself. If you can see a button labelled *I'm a new user,* then clicking it will allow you to create your own ANY-maze user account. If this button is not shown, then you'll need to ask the ANY-maze

	administrator to create a user account for you.
Your password	If your user account is protected with a password, then you should enter it into this field. You won't need to enter a password if the ANY-maze user policies don't require passwords, and you haven't entered one on your account. If you enter an incorrect password, then ANY-maze will wait a few seconds before you can try again - each time you fail to enter the correct password, this delay will double. This is a security feature which makes it hard for someone to try guessing your password. NOTE: PASSWORDS ARE CASE-SENSITIVE.
Log me on	Once you've selected your name in the list, and entered your password (if required), then clicking the *Log me on* button will give you access to ANY-maze.
I'm a new user	This button will only be present if the user policies have been set to allow new users to create their own accounts. Clicking it will open a window where you can add yourself to the list of users registered with ANY-maze, by entering your name and specifying a password.

After logging on, you can log off at any time by selecting the File page and clicking the *Log off* option. This will re-open this Log-on window, for the next user to log on.

See also:

- An introduction to ANY-maze user management
- ANY-maze user management - this explains how to set user policies
- Editing your user name and password

Set user password

Overview

If the ANY-maze administrator changes the system's user policies so that all users must have passwords, then any user who until now hasn't used a password will be asked to provide one when they log on.

Details

My password	You can enter anything you like as a password, but it must be at least 5 characters long and different to your user name. Note that any spaces you enter at the start or end of your password are ignored, so for example, entering ' dog' would not be accepted (because the two spaces at the start of the password would be ignored, and the password would be seen to have just 3 characters). Spaces inside the password, such as 'my dog', are valid and will be counted as characters.
Confirm password	Just re-enter your password here, to confirm that what you entered is what you intended!
Protect my files...	When you save an Experiment or Protocol file in ANY-maze, you can optionally provide a password which will be used to 'lock' the file so that other people can't open it - this facility is explained more fully in the Saving experiments and Saving protocols topics. If you expect that you will generally use this facility, then you can check this box to have ANY-maze automatically use the password you've entered here to protect any experiment or protocol files that you save. Of course, as this is only a default, you will still be able to change an individual file to have a different password (or no password at all).

See also:

- An introduction to ANY-maze user management
- Editing your user name and password

Create new user

Overview

In order to use ANY-maze, you must have a **user account** which you can create for yourself using this window.

The benefit of having a user account is that ANY-maze will remember any options and settings which *you* set, and restore them whenever *you* use the program. Thus if you change the size of the ANY-maze font, for example, then this won't affect other users - nor will changes they make affect you.

Details

My name	You can enter anything you like as your user name (up to a maximum of 24 characters); the only limitation is that you can't use a name already taken by another user - ANY-maze will tell you if this is the case. Generally it's a good idea to simply use your first name, perhaps with an initial if there's another user with the same name.
My password	Depending on how ANY-maze has been set up, you may not need to enter a password at all (you can find out whether you need to by reading the text at the top of the window). However, whether it's required or not, using a password is a good idea as it protects your ANY-maze settings. You can enter anything you like, but it must be at least 5 characters long and different to your user name. Note that any spaces you enter at the start or end of your password are ignored, so for example entering ' dog' would not be accepted (because the two spaces at the start of the password would be ignored, and the password would be seen to have just 3 characters). Spaces inside the password, such as 'my dog', are valid and will be counted as characters.
Protecting your files	When you save an Experiment or Protocol file in ANY-maze, you can optionally provide a password which will be used to 'lock' the file so other people can't open it - this facility is explained more fully in the Saving experiments and Saving protocols topics. If you expect that you will generally use this facility, then by checking the box labelled *Protect my files by default...*, you can set ANY-maze to automatically use your user password to protect any experiment or protocol files that you save. Of course, as this is only a default, you will still be able to change an

individual file to have a different password (or no password at all).

Changing your user name, password etc. in the future

You can edit the details that you enter here at any time in the future. To do this, simply select the ANY-maze Options page and then click the *My account* option.

See also:

- Editing your user name and password
- An introduction to managing users
- Saving experiments
- Saving protocols

Administrative options

The Administrative options *are only included on the Options page if you're an administrator.*

Overview

This page allows ANY-maze administrators to enable or disable certain features in the software for other, non-administrator users.

To access the *Administrative options*, select the Options page and then click *Administrative options* in the left-hand side bar.

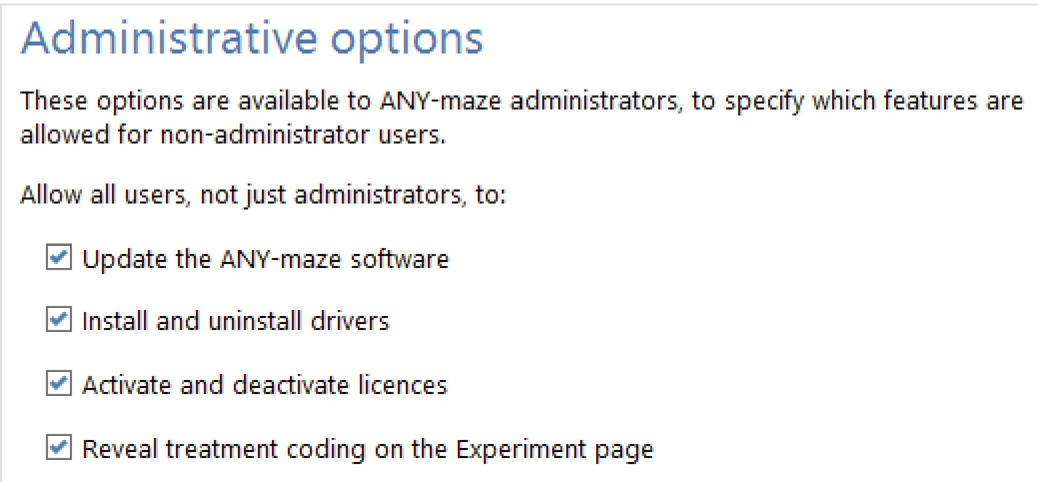


Figure 1. The Administrative options *on the ANY-maze Options page.*

On this page, an administrator can allow all users (not just administrators) to:

- Update the ANY-maze software
- Install and uninstall drivers
- Activate and deactivate licences
- Allow standard users to reveal treatment coding on the Experiment page

Most of these options are turned ON by default, to try and avoid the situation where the administrator who installed and set up the ANY-maze software has since left the lab, and no-one knows the administrator password. If these settings are left as their defaults, this situation doesn't prevent the software being updated, drivers being installed or licences being transferred.

Allow non-administrators to update the ANY-maze software

ANY-maze's Support page contains the option to check for updates to the software - and if there's a newer version available, to download and install it. By default, any user can perform this check, although due to Windows' security settings, the actual *installation* of the software can only be done by a Windows administrator.

If you want to, you can limit the ability to check for and download updates to ANY-maze administrators, and not standard users - if that's what you want, un-check this option.

Note that whether this option is turned on will also affect the automatic checking for updates which happens when the software is run - see the auto-update options.

Allow non-administrators to activate and deactivate licences

If you're in a big lab with fewer ANY-maze licences than there are computers, you may need to **transfer** an ANY-maze licence from one computer to another. To do this, you need to deactivate the licence on one computer, and activate it on another.

This option allows the transfer of licences to be done by any user, and not just administrators.

Don't forget that you only need an ANY-maze licence to actually perform tracking - if you only need to create an experiment or look at results, you can use an unlicensed version of the software.

Allow non-administrators to install and uninstall drivers

When you get a new camera or hardware to go with your ANY-maze software, you might need to install a driver for it on the Support page.

This option allows any user to install the necessary drivers (or uninstall a driver, if it's causing a problem). It's on by default, but you can turn it off to prevent non-administrator users from being able to install and uninstall drivers.

In most cases, a user will also need to be a Windows administrator in order to install or uninstall a driver.

Allow non-administrators to reveal treatment coding on the Experiment page

Protocols in ANY-maze can be set up to run experiments *blind*. This means that the different treatments used in the experiment are coded randomly; the code (rather than the treatment name) is then used throughout the experiment, and only revealed when testing is complete. The setting to run experiments blind is turned on by default, and can be changed in the *Treatment groups* element of the protocol page.

You'll use the Experiment page of ANY-maze to set up the treatments that you'll use in your tests, and then to allocate those treatments to animals (or, you can allow ANY-maze to allocate the treatments randomly). The ribbon bar on the Experiment page has the option to reveal the treatment coding - i.e. to show which treatments have been given which coding - but as an administrator, you might not want the person running the tests to be able to see the treatments.

To prevent non-administrators from being able to reveal the treatment coding, un-check the option on this *Administrative options* page to *Allow standard users to reveal treatment coding*...

For more details on the coding of treatments, please see the Treatment coding topic.

Editing your user account details

This option is only included on the Options page if you're using ANY-maze user accounts (rather than Windows accounts) and you're NOT an ANY-maze administrator. If you are an ANY-maze administrator, you can change your own user name and password on the ANY-maze user management page.

Overview

Although your ANY-maze user name and password will initially be given to you by the ANY-maze administrator, you can use the *My account* settings on the Options page to change them at any time. Indeed, it's a good idea to change your password at regular intervals - perhaps once a month.

You can also use this page to choose whether ANY-maze should, by default, protect your experiment and protocol files using your password so other people can't open them.

- Accessing your account settings
- Changing your user name
- Changing your password
- Protecting your files by default

Accessing your account settings

To access your account settings, select the Options page and then click the *My account* option in the left-hand side bar. If this option doesn't appear on the Options page, it's because your administrator has configured ANY-maze to use Windows user accounts.

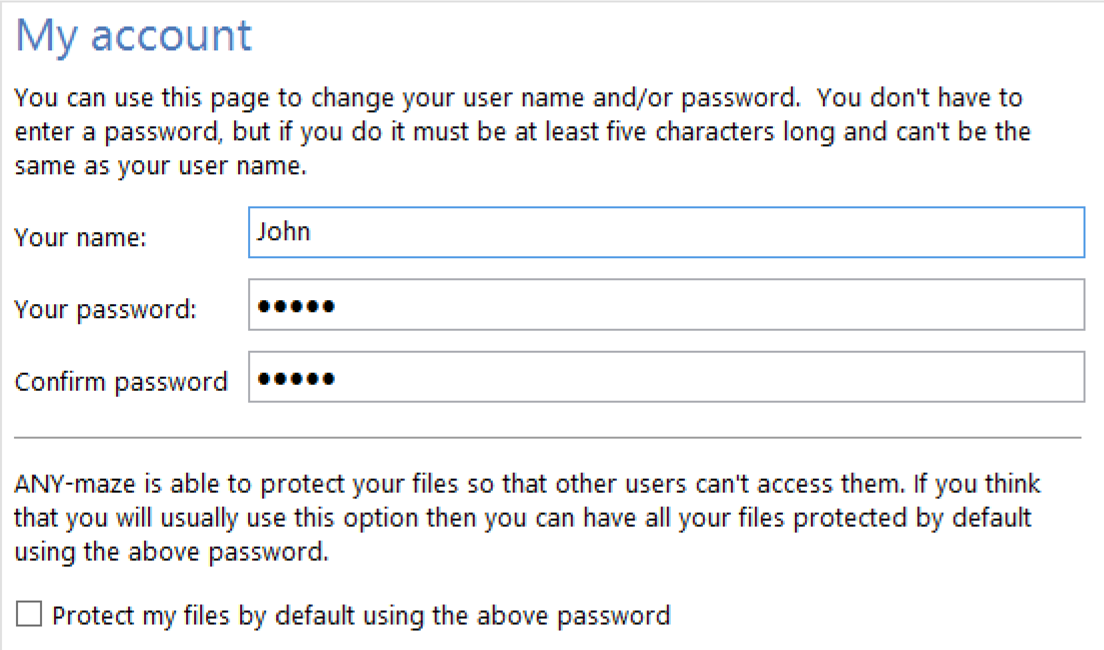


Figure 1. The My account *settings on the ANY-maze Options page.*

Changing your user name

You can change your user name by simply editing the name shown. You can enter anything you like up to a maximum of 24 characters. The only limitation is that you can't use a name already taken by another user - ANY-maze will tell you if this is the case.

Changing your password

Depending on how the ANY-maze administrator has set up the system, you may not need to enter a password at all. However, doing so is a good idea as it protects your ANY-maze settings. You can enter anything you like, but it must be between 5 and 15 characters long and different to your user name. Note that any spaces you enter at the start or end of your password are ignored, so entering ' dog' for example would not be accepted (because the two spaces at the start of the password would be ignored and the password would be seen to have just 3 characters). Spaces inside the password, such as 'my dog', will be counted as characters.

Protecting your files by default

When you save an Experiment file or Protocol file in ANY-maze, you can optionally provide a password that will be used to 'lock' the file so that other people can't open it - this facility is explained more fully in the Saving experiments and Saving protocols topics.

If you expect that you will generally use this facility, then by checking the box labelled *Protect my files by default...*, you can set ANY-maze to automatically use your user password to protect any experiment or protocol files that you save. Of course, as this is only a default, you will still be able to change an individual file to have a different password (or no password at all).

See also:

- ANY-maze options
- Managing users

Features for new users

Overview

ANY-maze includes a few features specifically designed to help you learn about the system. However, as you become more familiar with the program you may want to switch these features off, which you can do without affecting other users.

- Accessing the settings for new users
- Switching off specific features

Accessing the settings for new users

To access the settings for new users, select the Options page and then click the *Features for new users* option in the left-hand side bar.

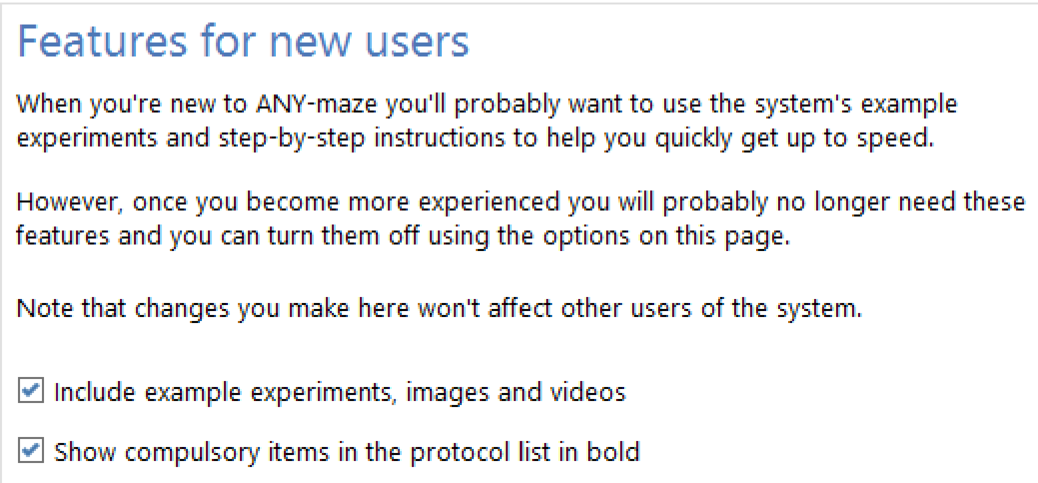


Figure 1. The Features for new users *settings on the ANY-maze Options page.*

Switching off specific features

Include example experiments, images and videos

ANY-maze is provided with two standard example experiments, and others may be included depending on the version of the software you have installed. These examples are accessed from the

File page, under the *Open example experiment* option. Turning off example experiments will remove this menu option, and will also exclude any example videos from the list shown on the *Video sources* page of the Protocol.

Show compulsory items in the protocol list in bold

Although there are quite a few items in the protocol list to consider when setting up a protocol for your experiments, only a very few of them are compulsory, or contain settings that must be specified. If you turn this option on, these items will be shown in **bold** text in the protocol list to make them obvious.

The compulsory items or settings in the protocol are:

- Under Treatment groups, you need to specify how the animals will be assigned to the groups (randomly, manually, or in a fixed order)
- At least one video source must be added.
- At least one apparatus must be added.
- The animal colour must be specified (i.e. whether the animal is lighter or darker than the apparatus background)
- The *Test duration* of each Stage of the experiment must be specified.

See also:

- ANY-maze options
- The Protocol page

Filing options

Overview

The filing options allow you to specify the location of your personal ANY-maze folder and to customise the way that ANY-maze saves and loads experiment files.

- Accessing the filing options
- Specifying file locations
- Temporary files
- Specifying other filing options
- Default locations for the shared and personal ANY-maze folders

Accessing the filing options

To access the filing options, select the Options page and then click the *Filing* option in the left-hand side bar.

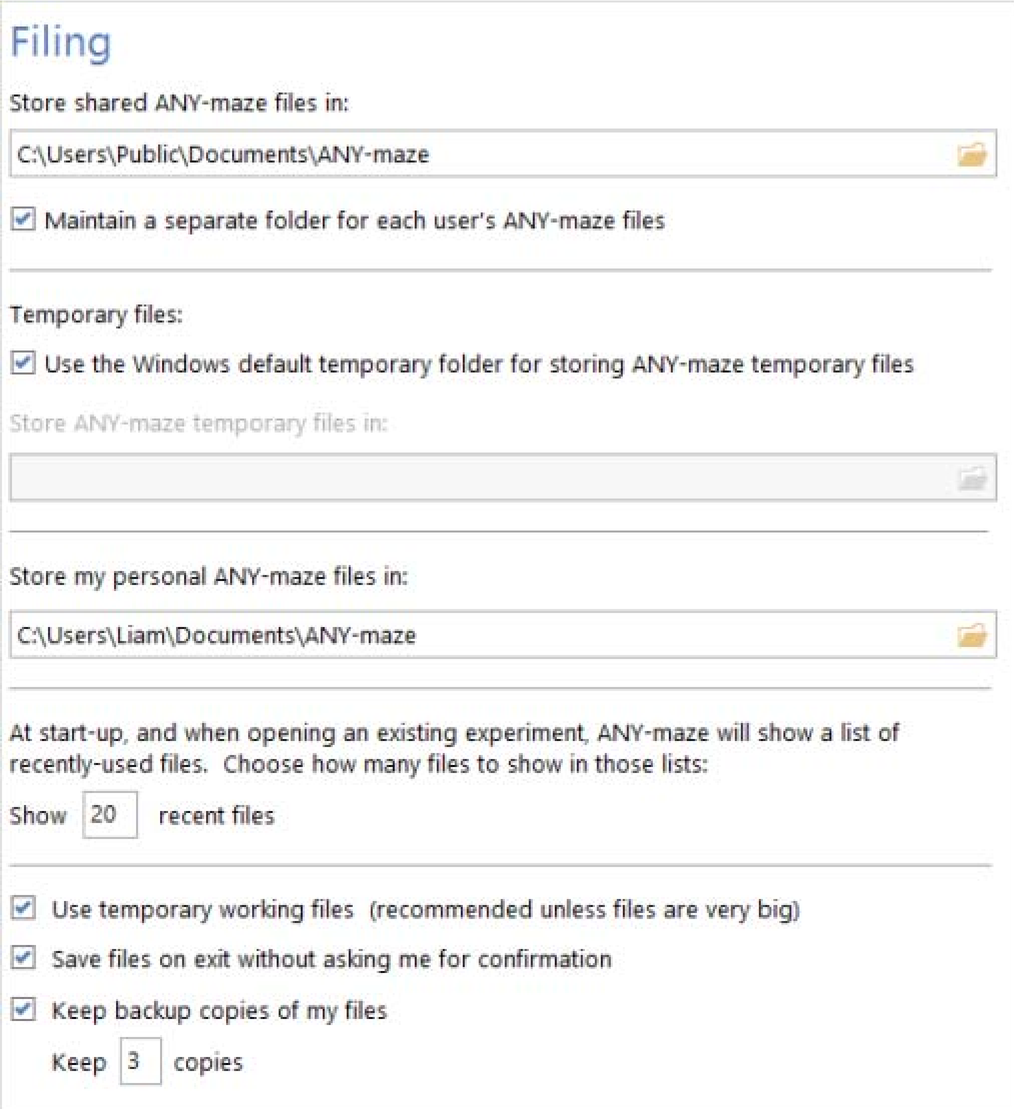


Figure 1. The Filing *settings on the ANY-maze Options page.*

⚠ *The exact options available on this page will depend on whether you're an ANY-maze administrator and if you're not, on how the ANY-maze administrator has configured the system.*

Specifying file locations

ANY-maze includes two special locations for files: the *shared ANY-maze folder* and users' *personal ANY-maze folders.*

The shared ANY-maze folder	This folder is available to all users and is intended to be used to save common files such as shared protocols, examples etc. Only an ANY-maze administrator can set the path of the shared

ANY-maze folder. When an administrator is logged on, the *Filing* options will include a field where the shared folder path can be typed in and a button which can be used to browse for the desired folder.

Personal ANY-maze folders

An ANY-maze administrator can configure the system to maintain a separate *personal* folder for each user - these folders are intended to be used for personal files such as experiments, personal protocols etc.

An ANY-maze administrator can specify that ANY-maze should maintain a personal folder for each user by checking the option to *Maintain a separate folder for each user's ANY-maze files*. This option only appears on the Filing page when an administrator is logged on.

Each user can specify where his or her personal ANY-maze folder is located, either by entering the path into the field titled *Store my personal ANY-maze files in* or by clicking the button next to the field and browsing for the folder.

Both the shared ANY-maze folder and the personal ANY-maze folder will automatically be included as 'Places' on the File page's *New experiment* and *Open experiment* options, as places to look for experiment and protocol files. For more details, see Using 'Places' to quickly access your files. Under Windows XP only, you'll find a shortcut to your personal folder is also included on all file open and file save windows in ANY-maze.

Note that you can specify network locations for both the shared ANY-maze folder and/or personal ANY-maze folders. This can be a good idea if your lab has a file server which is regularly backed-up.

Full details of the default locations for these folders are given at the end of this topic.

Temporary files

ANY-maze uses temporary files for various purposes during its operation; these files are created when necessary and deleted when they are no longer needed. If you wish, you can choose where these files should be located. Usually, you won't need to worry about this - ANY-maze will use the Windows folder for temporary files by default. However, if you want to specify a folder to use, you'll need to un-check the option *Use the Windows default temporary folder...* and make an entry in the *Store ANY-maze temporary files in...* field. You can do this by entering the full path of the folder, or by clicking the button next to the field and browsing for the folder.

Specifying other filing options

Apart from specifying file locations, there are four other filing options which are always available to all users:

Show ... recent files

When you start the ANY-maze software, the File page will show a list of the most recently-used experiment files, for quick access, together with options to create new experiments from a list of the most recently-used protocols. The *New experiment* and *Open experiment* options on the File page also contain lists of the most recently-used protocols and experiments, respectively.

You can choose how many files are displayed on these lists by making an entry here. You can enter any number from 4 to 30. Note that if there isn't enough room for all of the files to be displayed, then ANY-maze will only display those which it has space for, up to the number entered here.

Use temporary working files

Normally when you open an experiment file, ANY-maze actually makes a temporary copy of the file and opens that. This means that changes you make are applied to the temporary file and not to the experiment file itself. When you 'save' the experiment, what the system actually does is transfer the contents of the temporary file (where the changes you made have been stored) to the experiment file.

ANY-maze uses this apparently odd way of working because it stores all changes to an experiment as you make them, which makes the system more reliable. However, if all the changes were stored directly to the experiment file, there would be no method for you to 'Cancel' the changes - in other words you wouldn't be able to choose to **not** save your changes.

Normally you don't need to concern yourself about the fact that the system works this way, as it all seems quite intuitive. However, if the experiment file is very big and/or your computer is not very fast, then this copying of the experiment file when you open it can take an appreciable length of time. In this case you may wish to change the system so it doesn't use a temporary file, but instead alters the experiment file directly. That's what turning off this option will do.

Unless your experiment files are taking a long time to open, we strongly recommend you keep this option turned on.

Save files on exit...

Normally when you exit from ANY-maze (or when you load a new file) the system checks whether you have made any changes to the file which is currently open, and if you have, asks whether you want to save them - naturally you will almost always say 'Yes'.

By checking this box, you can instruct the system to automatically save your changes whenever you exit, without asking you if you want to. This makes the system 'cleaner' to use, and there's less chance that you'll fail to save your work. If at some point you DON'T want to save changes, then you have to be a little careful - just closing an experiment WILL save them. Instead you should select the option on the File page to *Revert to saved experiment*.

If you have chosen not to use temporary working files (see the

	previous option) then this option will be disabled, because all changes are saved as soon as you make them.
Keep backup copies of my files	If you check this box then ANY-maze will automatically create a backup copy of files when you save them. In general, this is a good idea, although it obviously uses up more space on your hard disk. There's more information about this in the Saving experiments topic. If you have chosen not to use temporary working files (see the option above) then this option will be disabled, because the backup file which the system keeps is actually the original experiment file before the temporary working file replaces it. Choosing this option will allow you to enter the number of backup copies to keep - the default is 3, although you can enter any number between 1 and 9.

Default locations for the shared and personal ANY-maze folders

Shared ANY-maze folder

ANY-maze will automatically select a location for the shared ANY-maze folder during installation. This will be a folder called **ANY-maze** in the Windows shared/public documents folder. The exact location depends on the version of Windows in use:

Windows Vista and above	This will generally be `C:\Users\Public\Documents\ANY-maze`
Windows XP	This will generally be `C:\Documents and Settings\All Users\Documents\ANY-maze`

Personal ANY-maze folders

The default location for personal ANY-maze folders depends on whether ANY-maze is using Windows user accounts or is managing users itself. If it's using Windows users, then the default location is a folder called **ANY-maze** in the user's **My documents** folder. Again the exact location depends on the version of Windows in use:

Windows Vista and above	This will generally be `C:\Users\[Name]\Documents\ANY-maze`
Windows XP	This will generally be `C:\Documents and Settings\[Name]\Documents\ANY-maze`

If ANY-maze is managing users, then the location is a folder with the same name as the user, which is located inside the shared ANY-maze folder - see above.

See also:

- ANY-maze options
- Working with ANY-maze files, reports and videos

- The Startup page
- Opening experiment files
- Saving experiment files

Printing options

Overview

Using the printing options you can set the headers and footers which ANY-maze will include on each printed page, and specify various options such as reverse order printing, which can make printing more efficient.

- Accessing the printing options
- Setting page headers and footers
- Other printing options

Accessing the printing options

To access the printing options, select the Options page and then click the *Printing* option in the left-hand side bar.

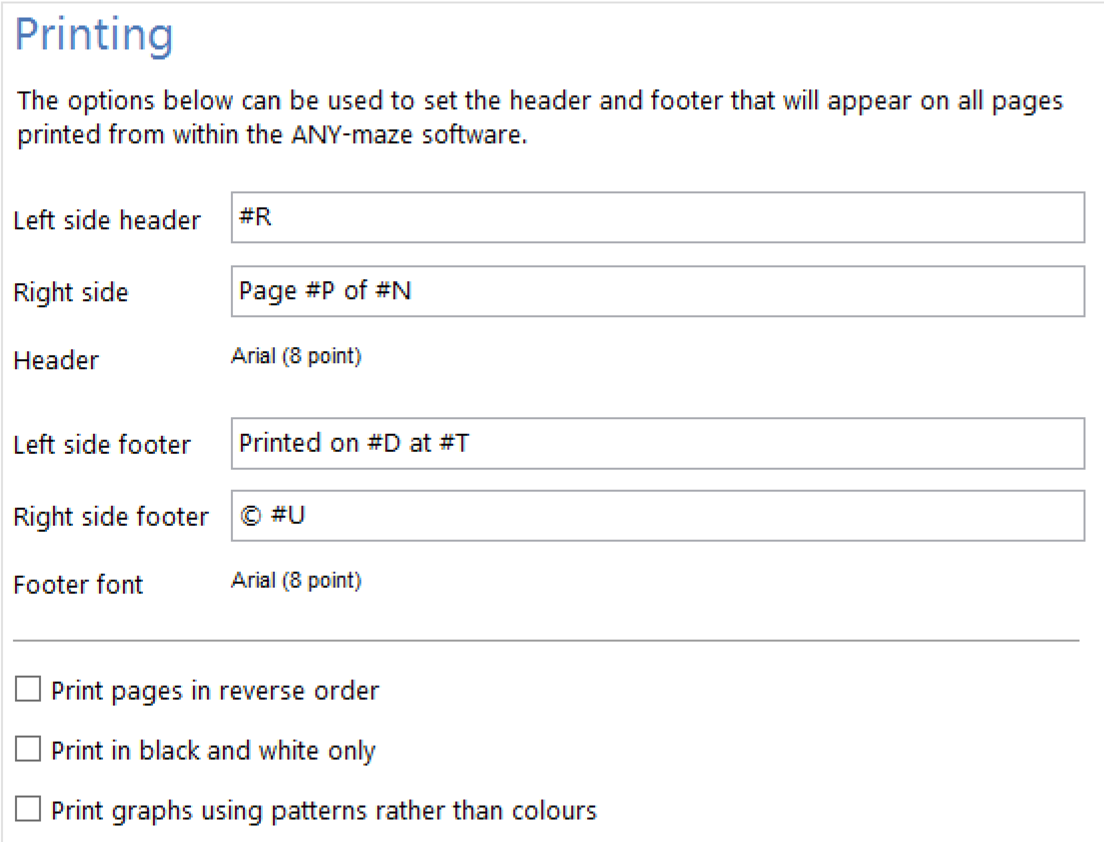


Figure 1. The Printing *settings on the ANY-maze Options page.*

Setting page headers and footers

When ANY-maze prints a document, it can include a header and/or a footer at the top and bottom of every page. You can use the fields on this window to enter the header and footer text you want to use. As well as entering text, you can also enter one or more of the following codes:

#R	The name of the document being printed.
#P	The page number.
#N	The number of pages in the document.
#D	The date the document was printed.
#T	The time the document was printed.
#U	The name of the user who printed the document.
#O	The name of your organisation. This is the name of the organisation that was entered when the software was installed.

#F The file name of the current experiment file.

#E The title of the current experiment - this is whatever is entered in the *Experiment title* field on the Experiment details page.

#A The date the experiment was created.

For example, if you enter 'Page #P of #N' as the right side header, then every printed page will include something like 'Page 2 of 4' printed in its top right corner. If you don't want ANY-maze to include a header or a footer on printed documents then you can simply leave the fields blank.

You can also choose the font used to print the header and footer by clicking the *Edit header font* or *Edit footer font* buttons in the ribbon bar. Once you've selected a different font, the name of the font will be displayed in that font, so you can see how it will appear on the document.

Other printing options

Print in reverse order Some modern printers, such as the HP DeskJet series, deposit pages in their out tray face up. This has the effect that when you print more than one page, they end up in reverse order and you have to manually reorganise them. If you check this box then the system will print the pages in reverse order, so that when the printer deposits them in the out tray they will actually end up sorted correctly. This option is only useful if your printer suffers from this type of problem.

Print in black and white only Most of the documents in ANY-maze include coloured elements which many modern printers are capable of printing. However, colour inks are generally expensive and you may want to print documents in black and white rather than colour to save money. While you can probably instruct your printer to print in black and white, you may need to tell it to do so every time you print. A simpler solution is to check this box and ANY-maze will always print in black and white irrespective of the printer's settings.

Warning: Due to incompatibilities in some print drivers, this option doesn't always work. If you have this type of problem then we suggest you try using a print driver from Microsoft rather than from the printer manufacturer. Microsoft print drivers can be downloaded for free from their web site at www.microsoft.com/downloads.

Print graphs using patterns... If you're using a printer that will only print in black and white, or you've checked the *Print in black and white only* box, then printing coloured graphs in ANY-maze will be rather unsatisfactory as the columns or lines will be printed in various shades of grey. To resolve this problem, you can use this option to choose to print graphs using patterns rather than colours. Note that this ONLY affects printed graphs - they'll still appear coloured on the screen.

See also:

- ANY-maze options
- Printing reports
- Altering page set up
- Printing documents

Choose header font

Use this window to choose the font which will be used to print headers on ANY-maze documents.

Simply choose the font face and font size using the two lists, and then optionally specify a bold or italic effect by checking either (or both) of the two check boxes. An example of the font you've chosen will be shown in the area at the bottom of the window.

Click *OK* to use the currently-selected font, or *Cancel* to ignore any changes you've made in this window and leave the header font as it is.

ANY-maze help topic H0872

Choose footer font

Use this window to choose the font which will be used to print footers on ANY-maze documents.

Simply choose the font face and font size using the two lists, and then optionally specify a bold or italic effect by checking either (or both) of the two check boxes. An example of the font you've chosen will be shown in the area at the bottom of the window.

Click *OK* to use the currently-selected font, or *Cancel* to ignore any changes you've made in this window and leave the footer font as it is.

ANY-maze help topic H0873

Configuring automatic updates

Only ANY-maze administrators can alter the settings described here.

Overview

We are continuously working on ANY-maze to add new features and refine those already included. To help you quickly get the benefit of upgrades to the system, ANY-maze can be set to automatically check our web-site for new versions of the software, and to download and install any it finds. Of course, your computer needs an internet connection for this to work.

By the way, don't worry that upgrades will just be installed without your consent - when a new version is detected, the system will display full details of what's included and ask you what action you would like to take.

- Accessing the auto-update options
- Specifying the auto-update settings
- Issues which may affect auto-update

Accessing the auto-update options

ANY-maze administrators can choose whether ANY-maze will automatically check for updates using the *Auto-updates* settings on the Options page. To access these settings, select the Options page and then click the *Auto-update* option in the left-hand side bar - if this option doesn't appear on the Options page, it's because you're not an ANY-maze administrator.

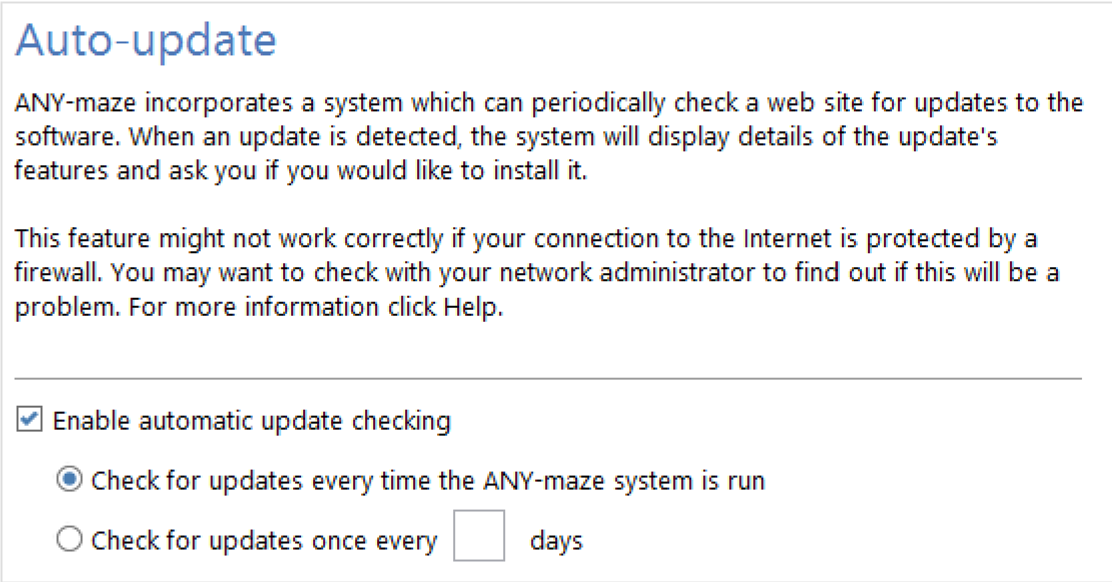


Figure 1. The Auto-update *settings on the ANY-maze Options page.*

Specifying the auto-update settings

Enabling auto-updates	If you would like ANY-maze to automatically check for updates then you should tick the *Enable automatic update checking* box. You should then specify the frequency of checks.
Specifying update frequency	If your computer has a permanent internet connection (via your lab's network for example) then you may as well choose to check for updates every time you run ANY-maze; the check only takes a couple of 2 seconds. If on the other hand you need to dial-up to connect to the internet, then you will probably prefer to choose to check less frequently, perhaps just every 14 days. You can enter any number of days from 1 to 99.

Issues which may affect auto-update

ANY-maze uses an HTTP connection to the ANY-maze web server (www.anymaze.com) to check for updates. The updates are contained in binary files with the file extension .udd (standing for 'update data'). These files are not natively executable, although they do contain executable code.

If you experience difficulties using the ANY-maze update feature, then you may want to show the above information to your network administrator - your lab's firewall might be blocking the updates because they're binary files and/or because they contain executable code.

If you're unable to use the update feature, don't worry. We e-mail all users when an update is released, and you can simply download the new version from our web site and install it manually.

See also:

- ANY-maze options
- Checking for updates to ANY-maze

Settings for importing animals and tests

Only ANY-maze administrators can alter the import policies.

Introduction

ANY-maze includes the ability to import lists of animals and tests from CSV files. This feature is described fully in the Importing lists of animals or tests topic.

When performing an import, ANY-maze may encounter certain problems - for example, an animal import file may include an animal that is already part of the experiment, or a test import may specify a test on an animal that has been retired. In these situations, ANY-maze will refer to these *import policies* to see how it should proceed.

To access the *Import policies* settings, select the Options page and then click the *Import policies* option in the left-hand side bar.

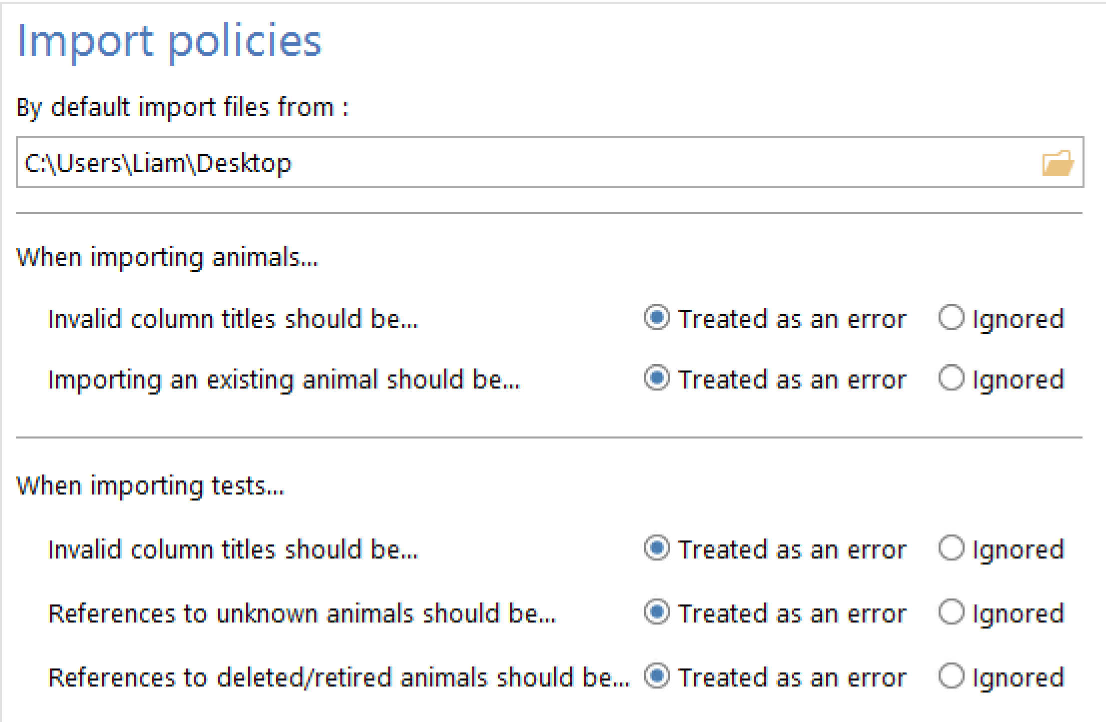


Figure 1. The Import policies *settings on the ANY-maze Options page.*

Default import location

The default import location simply specifies which folder ANY-maze should show by default when you perform an import. Specifying this folder is useful if you usually place the import files in the same location, as it avoids the need to navigate to the correct folder on each import. You can either enter the full path of the folder that the files will be in, or click the button to open the *Browse for folder* window.

Animal import policies

The animal import policies define what ANY-maze should do when it encounters specific situations during the preprocessing stage of an animal import:

Invalid column titles should be...

The column titles are contained in the *header row* of the CSV file (the first non-blank row in the file) and define what information is contained in the data rows. Valid entries are 'Animal', 'Treatment' and the names of any animal fields defined in the protocol. If any other entries are found, ANY-maze will use this policy to decide what to do.

If you choose *Ignored* then the data in the column will simply be ignored during the import processing. If you choose *Treated as an error* then the preprocessing will fail and ANY-maze will report the problem. In this case the import will be aborted, i.e. no animals will be imported.

Importing an existing animal should be...

If an animal import file includes an 'Animal' column, the data is the ID of each animal. Clearly no two animals can have the same ID, so this policy defines what ANY-maze should do if an import specifies an animal whose ID already exists.

If you choose *Ignored* then the entire data row for the animal will be ignored, i.e. the animal simply won't be added to the experiment. If you choose *Treated as an error* then the preprocessing will fail and ANY-maze will report the problem. In this case the import will be aborted, i.e. no animals will be imported.

Test import policies

The test import policies define what ANY-maze should do when it encounters specific situations during the preprocessing stage of a test import:

Invalid column titles should be...

The column titles are contained in the *header row* of the CSV file (the first non-blank row in the file) and define what information is contained in the data rows. Valid entries are 'Animal', 'Apparatus' and the names of any test fields defined in the protocol. If any other entries are found, ANY-maze will use this policy to decide what to do.

If you choose *Ignored* then the data in the column will simply be ignored during the import processing. If you choose *Treated as an error* then the preprocessing will fail and ANY-maze will report the problem. In this case the import will be aborted, i.e. no tests will be imported.

References to unknown animals should be...

Animals are specified in the import file either using their ANY-maze animal numbers or, if the protocol is set to use animal IDs, using their IDs.

This policy defines what ANY-maze should do if the animal number or ID simply does not exist in the experiment. For example, the import file specifies that a test should be performed on animal 54 but there are only 40 animals in the experiment.

If you choose *Ignored* then the animal will simply be skipped and no test for it will be added. If you choose *Treated as an error* then the preprocessing will fail and ANY-maze will report the problem. In this case the import will be aborted, i.e. no tests will be imported.

References to deleted/retired animals should be...

During an experiment animals may be *deleted* or *retired* (for an explanation of these terms, see the 'Altering an animal's status to deleted or retired' section of the Animal details report topic). For example, an animal may become ill during the experiment and therefore you choose to retire it. However, you may have prepared a number of test list files to be imported into the experiment as it progresses, and these will, assumedly, still refer to the retired animal. When you import such a list, ANY-maze will use this policy to decide what it should do.

If you choose *Ignored* then the animal will simply be skipped and no test for it will be added. If you choose *Treated as an error* then the preprocessing will fail and ANY-maze will report the problem. In this case the import will be aborted, i.e. no tests will be imported.

See also:

- Importing animals
- Importing tests
- Setting up treatment groups and animals

Browse for import folder

Introduction

ANY-maze includes the ability to import lists of animals and tests from CSV files. This feature is described fully in the Importing lists of animals or tests topic.

The *Import policies* settings on the ANY-maze Options page allow you to specify a default folder for files to import, and this window allows you to choose that folder.

Details

You should use this window to navigate to the folder that will hold the CSV (or text) files containing the lists of animals or tests to import. This folder can be anywhere on your computer or network, and can be called anything you like.

If the folder doesn't already exist, you can click the *Make New Folder* button to create a new folder - if you do this, make sure that the name of the new folder is noted in the *Folder* field before clicking *OK*.

Details of the format of the animal and test import files can be found in the following topics:

- Importing animals
- Importing tests

Settings for reporting usage statistics

Only ANY-maze administrators can alter whether usage statistics are reported.

Introduction

ANY-maze collects anonymous information about which features of the software you use. In order to help with future development, this information can be sent to the ANY-maze software team for analysis.

Details

We are continually improving ANY-maze and adding new features. In order to help us decide where best to target our efforts, ANY-maze can collect anonymous information about the features that you use. This information can then be sent to our software development team, where it will influence the areas of the software they focus their efforts on. For example, if we're thinking of improving a specific feature, then the priority we give to that feature will depend on the number of people using it.

What information is collected?

Note that we are *only* looking at how many people use different features, and the frequency with which they are used - we're not looking at anything specific to an experiment. For any given experiment, we gather quite a lot of information - things like how many Zones are used in the experiment, how many tests are run, how many apparatus you use simultaneously, or the type of results which you look at. The important thing to remember is that there is *nothing* personal in this information - for example, the experiment notes or names of treatments are deliberately **not** included.

If you want to see exactly what *will* be sent to us, you can look at the contents of the `C:\ProgramData\ANY-maze` folder on your computer; all the information collected is stored in files with a file extension of `.afl`. These files will be zipped up and sent to us when ANY-maze is run; this will happen in the background and won't affect the operation of the software.

Note that this folder is a 'hidden' folder by default under Microsoft Windows.

What if I don't want to send you any information?

If you'd rather not send information about your ANY-maze usage to the development team, that's fine - you can see how to turn off the data collection below. But bear in mind that if we don't have information about which features of ANY-maze people are using most, we won't be able to effectively target which areas to work on. In the worst-case scenario, we may decide in future to drop features from the software that we don't think anyone is using - so if you're using a little-used part of the software, and we don't know about it, then it may disappear in a future version of ANY-maze!

My computer isn't connected to the internet; what happens then?

If your computer is not connected to the internet, then ANY-maze will still collect the usage information - but it will never be sent to us. Periodically, the information that is collected will be deleted, to prevent it taking up too much space.

Changing whether information is sent to ANY-maze

You can access the *Usage statistics* settings, to specify what information is sent to us, by selecting the Options page and then clicking the *Usage statistics* option in the left-hand side bar.

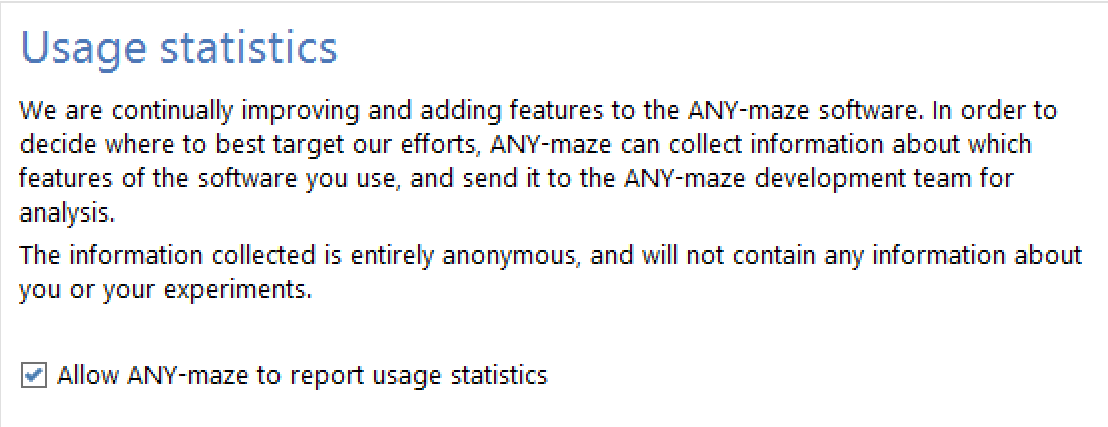


Figure 1. The Usage statistics *settings on the ANY-maze Options page.*

You can choose here whether or not to *Allow ANY-maze to report usage statistics.* Selecting this option will cause information about how you use the software to be collected, and periodically sent to the ANY-maze development team. If this option is not selected, then no usage statistics will be gathered or sent.

ANY-maze help topic H0876

The Support page

Unlike the pages on the left side of the ribbon bar, you can access the Support page without having to open an experiment first.

Introduction

The Support page provides some useful utility functions in ANY-maze. It lets you:

- Contact ANY-maze technical support for help
- Send files to ANY-maze technical support
- Activate, deactivate, purchase or renew an ANY-maze licence
- View information about the ANY-maze software or computer
- Check for ANY-maze software updates
- View or change the currently-installed drivers
- See the USB devices connected to the computer
- View and edit advanced support options

Using the Support page

The Support page is divided into a number of different sections. To switch between the sections just click the item in the list on the left-hand side of the page - see figure 1.

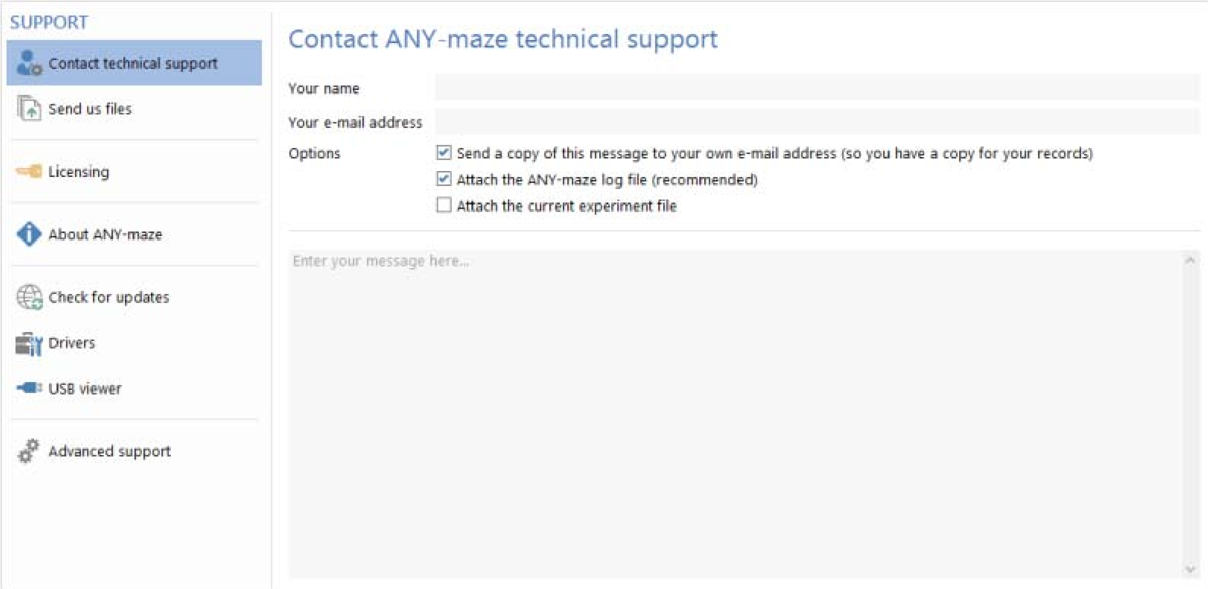


Figure 1. The ANY-maze Support page. Note that the exact options available depend on whether or not you're an ANY-maze administrator.

Contacting ANY-maze technical support

The ability to send an e-mail to our technical support team will only be available if your computer is connected to the internet. You can, of course, still contact us by sending a normal e-mail to techsupport@anymaze.com, or using the technical support page of our web-site: www.anymaze.com/support.htm

Introduction

A great feature of the ANY-maze's Support page is the ability to send an e-mail direct to our technical support staff. To do this, just select the Support page and then click the *Contact technical support* option in the left-hand side bar - see figure 1. Click this link to go there now.

You'll need to fill in your name and e-mail address, select the information you want to include with the message, and type in your message to our support engineers (see below).

- Types of support
- How to report a bug
- Attaching experiment files and log files
- Technical information sent with your message
- Sending the message

Figure 1. The Contact ANY-maze technical support *option on the Support page.*

Types of support

Please feel free to ask us for *any* type of help - we're happy to answer all enquires, from simple 'how to' questions to requests for advice on designing protocols. Note that you receive full technical support even if you're only evaluating the software, so please don't hesitate to get in touch if you need to!

If you need help with a 'bug' (where the software is not doing what you would expect) then please report the bug to technical support and we'll do our best to fix it as soon as we can. Please see details below for how to report a bug.

How to report a bug

If you find a bug in the software, **please** report it to us as soon as possible after it happens (so it's fresh in your memory) and try to give us as *much* information as possible.

Ideally, we'd like to know:

- Exactly what you were trying to do when the problem happened.

  If the problem is reproducible (i.e. it happens again when you repeat the same sequence of actions), then it would be really helpful if you could send us the experiment file you're working on, and the exact sequence of steps that cause the problem to happen.

- If the software displayed an error message, the exact wording of the error message.

  Screen shots are often useful; most modern keyboards have a 'Prt Scrn' key, which will copy the current desktop to the clipboard so you can paste it into an e-mail message.

- Make sure the *Attach the ANY-maze log file* option is checked, so we receive the log file with additional information in it.

- If you use this page to send us a bug report, then we'll automatically receive information about your computer as well, which might help us determine the cause of the problem. See below for the information that's included in the message.

We may need to contact you to ask you further questions about the problem.

Attaching experiment files and log files

When giving advice on protocol design and when resolving problems, it is often very helpful if we have a copy your experiment file. If relevant, you can check the *Attach the current experiment file* box to attach the current experiment to your message. If you're reporting a bug, then we'll probably need your ANY-maze log file as well - this might contain useful information about what ANY-maze was trying to do when the problem happened. The *Attach the ANY-maze log file...* option is checked by default.

The Attach the current experiment file *box will only be enabled if an experiment file is currently loaded.*

If you want to send files other than just the experiment file and log file, or you need to send us an experiment file that isn't currently loaded into the ANY-maze software, this can be done using the Send us files option on the Support page.

Technical information sent with your message

When trying to resolve support questions, it can be extremely helpful if we know certain technical information about your computer. For example, if the tracking seems rather poor, the problem may simply be that your computer isn't powerful enough. Because this information is so useful, the system will automatically include it in any e-mail you send using this support page.

Specifically, the information included is:

- The version of Windows installed on your computer.
- The quantity, type, speed and capabilities of the processors in your computer.
- The amount of memory (RAM) in your computer.
- Details of the type and capacity of the drives in your computer.
- Details of the display attached to your computer.
- Information about the type of internet connection available on your computer, whether the computer is attached to a network and whether it has a sound card and/or mouse installed.
- Details of any video interfaces (capture cards, cameras etc.) that ANY-maze can use to capture video images for tracking.
- The serial and licence numbers of your copy of ANY-maze, and the licence expiry date.

ONLY this technical information is sent in this process - ANY-maze won't transmit anything about you, or the contents of your files.

Sending the message

Once you've selected the options and entered your message, just click the *Send* button in the ribbon bar to send the message together with any associated files. Note that you can check the *Send a copy of this message to your own e-mail address...* box to send yourself an e-mail, so you can keep a copy for your own records - or to see exactly what information was sent to ANY-maze technical support.

Send us files

You can only send files to ANY-maze technical support if your computer is connected to the internet.

Overview

There may be occasions when you need to send us files in addition to the ANY-maze experiment file - for example, if you've got a problem with tracking in one of your apparatus, our support engineers might ask you to send a video of the animal in the apparatus, so we can try to ascertain the problem. The *Send us files* section of the Support page is where you can send us these additional files.

- Accessing the 'Send us files' settings
- Adding and removing files
- Sending the files

Accessing the *Send us files* settings

To send files to ANY-maze technical support, select the Support page and then click the *Send us files* option in the left-hand side bar. Enter your name, e-mail address and a message to send to our support engineers. You can click this link to go there now.

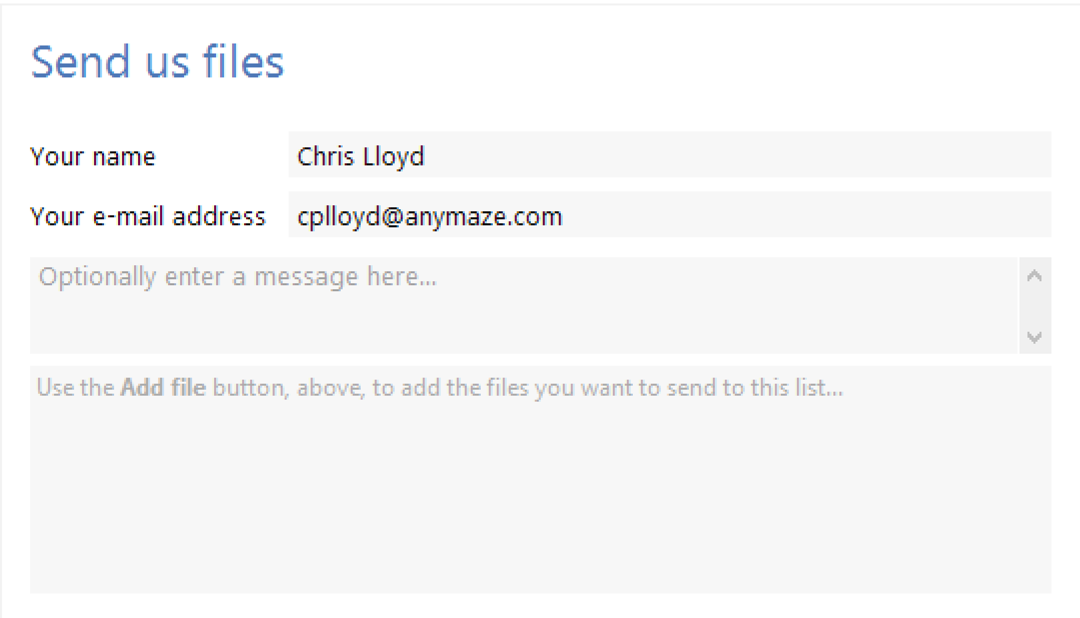


Figure 1. The Send us files *settings on the ANY-maze Support page.*

Adding and removing files

You can add as many files as you want to the message. To add a file, use the *Add file...* button in the ribbon bar; this will open the *Select file(s) to add* window. This is the standard Windows file selection window which you'll have seen in other Windows applications - just navigate to the file you want to add and click the *Add file(s)* button to select the file and close the window. You can select multiple files at a time by holding down the Ctrl key as you select them, or you can select a contiguous range of files by clicking the first file, and holding the Shift key as you select the last file. You can continue to add as many files as you want in this way.

If you want to remove a file from the list, simply select it in the list of files and select the *Remove file...* button in the ribbon bar.

File size limitations

ANY-maze compresses all the files to send into a single zip file before sending them. The largest zip file size that ANY-maze can send is 2GB, but of course this is the compressed size, so the largest uncompressed file size is significantly more than this. Exactly how large a file you can send will depend on the file type - ANY-maze experiment files will typically compress to at least half their original size, while video files won't compress very much. After compressing the files to send ANY-maze will check the resulting zip file size and if it is too large you will be notified. In this case you can still send us the file via the *Support* > *Send us files* page of the ANY-maze web site.

Sending the files

Once you've entered your details, the message, and selected the files to send, just click the *Send* button in the ribbon bar to send the message together with the associated files.

Note that files are sent in the background, so you can carry on working while this happens. The status bar at the bottom of the ANY-maze window will show the current progress of the files being sent:

Figure 2. The ANY-maze status bar, showing the progress of files being sent using the Send us files *page.*

At any time, you can click on this section of the status bar to show a small pop-up window containing a list of the current files being sent, with their progress. You can cancel the sending of an individual file using the button next to a file in this popped-up list.

You can carry on doing anything else in ANY-maze while these files are being sent - you can even start to send more files using the *Send us files* page! Note that if you try to exit from the ANY-maze software while files are still being sent, you'll be shown a message warning you of this and asking if you really want to exit and cancel the sending of the files.

The ANY-maze licence manager

Only ANY-maze administrators can activate and deactivate licenses, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Introduction

The ANY-maze licence manager is used to purchase, renew, activate or deactivate an ANY-maze licence.

In fact you only actually need a licence for the copy of ANY-maze which you'll use for tracking. You can install unlicensed copies on as many other computers as you like, and use them to set up experiments, analyse results, transfer data etc.

- How licensing works
- Accessing the licence manager
- Using the licence manager
- More information

How licensing works

Every time you install ANY-maze on a computer, it's given a serial number by the installation program. This number is unique to that specific copy of ANY-maze; this is the case even if you use exactly the same installation program to install ANY-maze on a number of different computers - each copy will have a different serial number.

When you buy an ANY-maze licence, what we actually supply you with is a licence number. You then use this number to activate your licence - this switches on the tracking system's ability to save results.

A critical thing to understand is that licence numbers are paired with serial numbers. In other words, a specific licence number will only switch on the ability to save tracking results in the copy of ANY-maze that has the serial number that is paired with that licence number. For this reason, when you buy a licence, you will need to supply us with the serial number of the copy of ANY-maze you actually want to license.

Accessing the licence manager

To open the ANY-maze licence manager, select the Support page and then click the *Licensing* option in the left-hand side bar - or you can click this link to go there now.

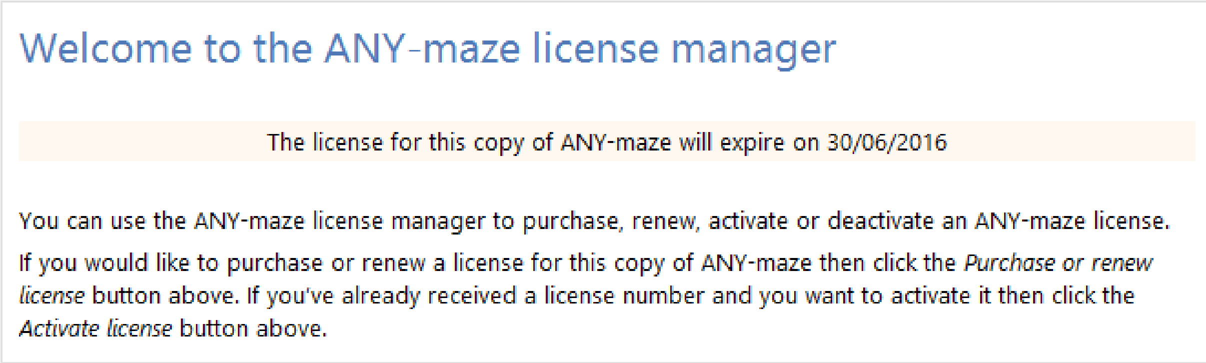


Figure 1. The licence management part of the ANY-maze Support page.

The licence manager will show a summary of your current licence (if you have one) and the ribbon bar will contain options for activating, deactivating, purchasing and renewing licences.

Using the licence manager

The ANY-maze licence manager allows you to:

- View your current licence details, including how much longer the licence has to run.

  This information is shown on the licensing page when it is first selected.

- Purchase or renew a licence.

  If your tracking computer is connected to the internet, you can use the *Purchase or renew licence* button in the ribbon bar, and the licence manager will take you through the necessary steps to purchase or renew a licence.

- Activate a licence.

  When you purchase a licence, what we actually supply you with is a licence number which you then need to activate by entering it into the licence manager. Use the *Activate licence* in the ribbon bar to activate a copy of ANY-maze for which you've been given the licence number.

- Deactivate a licence.

  Deactivating a licence is useful if you want to transfer a licence from one copy of ANY-maze to another - for example, when you purchase a new computer. In this situation, you simply deactivate the licence on the old machine, and we then supply you with a new licence number which you can activate on the new machine.

  To transfer an ANY-maze licence from one computer to another, use the *Deactivate licence* button in the ribbon bar to deactivate the licence on the first machine. Full details are included in the licence manager, but briefly, this entails:

1. Deactivating the licence on the original computer.
2. Sending an e-mail to ANY-maze technical support, quoting the serial number of the copy of ANY-maze on the original computer, together with the serial number of the copy of ANY-maze on the new computer.
3. ANY-maze technical support will send you a licence number for the new computer, which you'll need to enter using the licence manager on the ANY-maze software on that new computer.

More information

The licence manager itself includes comprehensive instructions and documentation - you can go directly to it by clicking this link.

ANY-maze information

Overview

The *About ANY-maze* option on the Support page brings together a range of information about ANY-maze and your computer. To access it, switch to the Support page and select the *About ANY-maze* option - or you can just click this link to go there now.

Details

The information available includes:

- The version number and release date of your copy of ANY-maze.
- The serial number of your copy of ANY-maze.
- Your licence number and its expiry date (if you've licensed your copy of ANY-maze).
- Acknowledgements and legal notices.
- Technical details about your computer, including the type and speed of the processor, the amount of memory, the type and capacity of your disk drives and information about any installed video digitisers and/or cameras that ANY-maze can use.

*The same information is available on the **Help page**.*

Checking for ANY-maze updates

Only Windows administrators can install ANY-maze updates, unless your administrator has specified in the Administrative options that all users, not just administrators, can perform updates.

Introduction

We're continuously working on ANY-maze to improve existing features and add new ones, and you'll probably want to update your system as new versions become available.

To make this process as simple as possible, ANY-maze includes the ability to check our web-site for updates and download and install any it finds.

- What software updates am I entitled to?
- Checking for an update
- What happens when an update is found
- Updating ANY-maze manually

What software updates am I entitled to?

When you buy ANY-maze, you are automatically eligible for all software updates that we release within one year from the date of purchase. Thereafter, you'll need to purchase a **Software Update contract**. This will allow you to install any new software updates that we release, for a period of one year.

The current price of a Software Update contract is $495 (price correct as of January 2019).

You'll receive technical support on ANY-maze forever, regardless of whether or not you purchase a Software Update contract.

Checking for an update

Checking whether there's an update available for ANY-maze is very easy and typically takes only a few seconds. Firstly, select the Support page and then select the *Check for updates* option - this will take you to the ANY-maze update manager. You can click this link to go there now.

The update manager itself includes detailed instructions, but essentially all you have to do is click the *Start* button. ANY-maze will connect to our web-site (dialling a connection if this is necessary) and will check whether you're using the most up-to-date version or not.

By the way, ANY-maze can also automatically check for updates on a regular basis, saving you the bother of having to remember to do this yourself.

What happens when an update is found

If ANY-maze finds an update, then it will display details of what's included in the new version and you'll be able to decide what action you'd like to take - see figure 1.

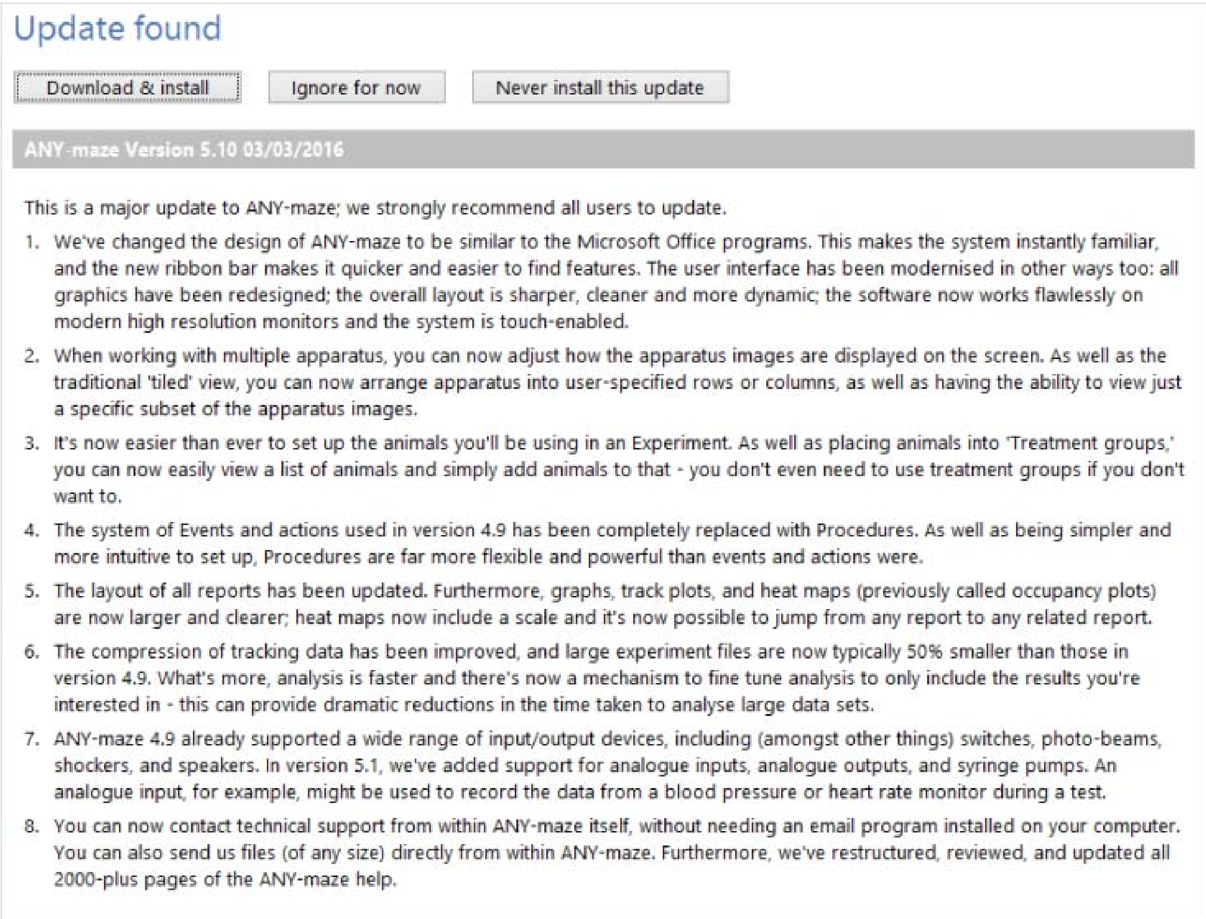


Figure 1. When an update is found, ANY-maze displays details of what it includes and asks you what action you would like to take.

You have three options:

Download & install	If, after reading the details about the update, you decide you want to install it then you simply need to click this button. ANY-maze will download the update to your computer and install it. You don't need to do anything - the entire process is automated and takes just a few minutes.
Ignore for now	If you're interested in the update but you don't want to perform the download and installation right away, perhaps because you don't have time, then you can choose this option. ANY-maze won't do anything more for now, but the next time you check for updates it will tell you about this one again.
Never install this update	You may decide, after reading details of the update, that you don't want to install it. If this is the case then you can use this option to tell ANY-maze that it should ignore this update and only inform you when a *newer* update becomes available.

The update program can only be run by a Windows administrator. Depending on the security settings of your Windows system, you may need to provide Windows administrator logon details in order to run the ANY-maze update.

Updating ANY-maze manually

If your ANY-maze computer isn't connected to the internet, then obviously you won't be able to use the update manager to check for and install updates - instead you'll need to install updates manually.

Whenever we release an update to ANY-maze, we'll automatically send you an e-mail informing you of this (unless you've specifically asked to be removed from our mailing list). When you receive the e-mail, you can go to our web site and download the update file onto any computer. You then simply need to copy this file to your ANY-maze computer (either using a network or a USB pen drive) and run it - this will install the update.

Drivers

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Overview

The *Drivers* page is used to install and uninstall various drivers used by ANY-maze, and to resolve driver-specific issues.

- The 'Drivers' section of the Support page
- Installing and uninstalling drivers
- Resolving problems
- Detailed information about all the ANY-maze drivers

The *Drivers* section of the Support page

You can manage ANY-maze drivers by selecting the Support page and then clicking the *Drivers* option. You'll be shown a list of all the drivers available for use with the devices you have connected - see figure 1.

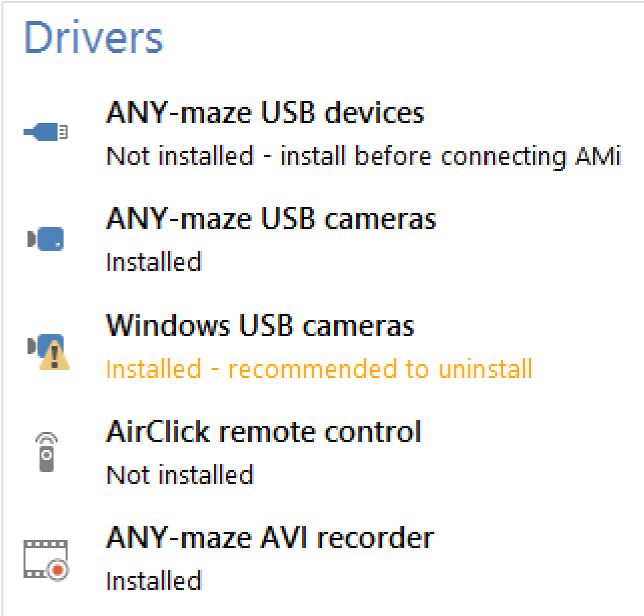


Figure 1. The Drivers *section of the ANY-maze Support page.*

Installing and uninstalling drivers

To install or uninstall a driver, you simply need to select it in the list of drivers and then click either the *Install driver* or *Uninstall driver* button in the ribbon bar.

However, you may find that the appropriate button is disabled (shown in grey), preventing you from performing the operation. This can occur for the following reasons:

- The driver is already installed - so the *Install driver* button will be disabled.
- The driver is not installed - so the *Uninstall driver* button will be disabled.
- You are not an administrator*. Only administrators can install and uninstall drivers.
- A test is currently being performed. You can't install or uninstall drivers while a test is running.
- The driver cannot be installed or uninstalled from within ANY-maze. For example, the entry for the 'Adlink RTV-24' driver reflects the *status* of the installed driver, but doesn't permit you to install or uninstall it. On the other hand, the entry for 'Microsoft MPEG-4 codec' reflects whether the codec is installed, and if not allows you to install it - but the codec can't be *uninstalled* from within ANY-maze.

* In order to install or uninstall drivers, you must be a **Windows** administrator, *and* running ANY-maze with administrator privileges.

Resolving problems

Under the name of each driver in the list is its status; normally this will read 'Installed' or 'Not installed'. However, in some situations an error or warning message may be displayed. Errors are shown in red text, and warnings in orange.

You can see more details about this message by selecting the driver in the list and clicking the *Driver details* button in the ribbon bar. This will open a pop-out help window displaying a help topic specific to that driver, which will include further information about the error or warning and how to resolve it.

Detailed information about all the ANY-maze drivers

For full details about the individual drivers that can be shown in the list, refer to the following topics:

Some drivers are only available under certain versions of Windows, so you may find that the list of drivers on your computer does not include all those shown here.

- ANY-maze USB device driver
- AirClick remote control driver
- ANY-maze USB camera driver
- Windows USB camera driver
- ANY-maze FireWire DCAM driver
- Windows FireWire DCAM driver
- Unibrain FireWire DCAM driver
- Adlink RTV-24 driver
- Data Translation digitiser SDK
- National Instruments IMAQ driver
- Microsoft MPEG-4 codec
- ANY-maze AVI recorder

ANY-maze USB device driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Introduction

The ANY-maze USB device driver is contained within the ANY-maze software, and is usually pre-installed by default when you install the ANY-maze system itself. You can use the *Drivers* option on the Support page to confirm whether the driver is installed, and to install and uninstall it as required.

The USB device driver is used for the following I/O devices:

- The ANY-maze interface (AMi)
- The ANY-maze remote start switch
- The ANY-maze TTL cable
- The ANY-maze shocker cable
- The Parallel Rod interface
- The Orofacial Pain Assessment Device (OPAD) cage
- The Waterwheel Forced Swim Test Tank

You should ensure this driver is installed BEFORE attaching any of the devices listed above to your computer.

How to install or reinstall the ANY-maze USB device driver

To install this driver, you should do the following:

1. Select the Support page.
2. Select the *Drivers* option in the left-hand side-bar.
3. Look at the entry for *ANY-maze USB devices* - the status should read 'Installed'.

Click here to open the *Drivers* settings now.

If the driver isn't installed, then you should simply select it in the drivers list and click the *Install driver* button in the ribbon bar.

How to uninstall the ANY-maze USB device driver

To uninstall this driver, you should do the following:

1. Select the Support page.
2. Select the *Drivers* option in the left-hand side-bar.
3. Look at the entry for *ANY-maze USB devices*. If the status reads 'Installed' then click the entry in the list (to select it) and then click the *Uninstall driver* button in the ribbon bar.

Note that if you uninstall the entire ANY-maze software, then this driver will, if necessary, be uninstalled automatically.

Installing an ANY-maze USB device

Once the *ANY-maze USB device driver* has been installed on your computer, installing one of the USB devices listed above, such as an ANY-maze interface device (AMi), is as simple as plugging it into a USB port. This will cause the Windows *Found new hardware wizard* to run, where you can just select the default options on every page.

If, during device installation, you see a warning telling you that the device driver has not passed Windows logo testing, you should select the *Continue anyway* option, otherwise your device won't be installed.

AirClick remote control

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Introduction

The Griffin AirClick remote control is a device that can be used to control tests remotely in ANY-maze. The AirClick is manufactured by a company called Griffin and, although it is not part of ANY-maze, ANY-maze supports it and we have included the required driver software within ANY-maze to make the set-up as quick and easy as possible.

As can be seen in figure 1, the AirClick consists of two parts - the remote control, which has 5 buttons, and a receiver, which connects to a USB port on your computer.

Figure 2. The Griffin AirClick.

Installing the AirClick

You can install the AirClick software from the *Drivers* option of the Support page of ANY-maze. Click here to open the *Drivers* settings now.

In this window you will see an entry for *AirClick remote control* and next to it a status of either 'Installed', 'Update required' or 'Not installed'.

If the status is 'Not installed' or 'Update required', then you should first click the entry in the list (to select it) and then click the *Install driver* button in the ribbon bar. ANY-maze will install the software, and the AirClick will be available for use in ANY-maze. (Note that the installation process will launch the AirClick's own installation program, provided by Griffin Technology.)

Using the AirClick in ANY-maze

Installing the AirClick in this way will program the buttons of the remote control as follows:

Play button	= F1
Fast forward button	= F2
Rewind button	= F3
Volume up button	= F4
Volume down button	= F5

In other words, the buttons are programmed from top to bottom as F1 to F5.

To actually control tests using the AirClick, you simply need to select one of the above 'F' keys as the Test control key in the Apparatus page of the protocol.

ANY-maze USB camera driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options *that all users, not just administrators, can do this.*

Introduction

ANY-maze is supplied with a driver for USB cameras that are supplied by ANY-maze/Stoelting, and used in devices such as the ANY-maze cage.

Note that this is **not** a generic USB camera driver, and won't be used with USB cameras provided by other manufacturers; for those cameras, you should install the driver supplied with the camera.

Installing the ANY-maze USB camera driver

By default, the ANY-maze USB camera driver is installed together with the ANY-maze system, and you probably won't need to do anything in order to use an ANY-maze USB camera.

To confirm whether the driver is installed on your computer, do the following:

1. Select the Support page.
2. Select the *Drivers* option in the left-hand side-bar.
3. Look at the entry for *ANY-maze USB cameras* - the status should read 'Installed'.

Click here to open the *Drivers* settings now.

If the driver isn't installed, then you should simply select it in the drivers list and click the *Install driver* button in the ribbon bar.

Problems with other drivers for USB cameras

Windows itself is provided with a generic USB camera driver, which Windows will sometimes erroneously use with ANY-maze USB cameras.

If this driver is installed on your computer, you will see an entry for it in the drivers list with a status of 'Installed - recommended to uninstall'. You can uninstall the driver by simply selecting it in the list and clicking the *Uninstall driver* button in the ribbon bar. This shouldn't cause a problem with other USB cameras, as they generally come with their own drivers.

Windows default USB camera driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options *that all users, not just administrators, can do this.*

Introduction

Windows is shipped with a default driver for USB cameras. However, this driver is quite limited and generally doesn't work well with ANY-maze.

Fortunately, USB cameras are almost always supplied with their own specific drivers, making the default driver obsolete. However, sometimes when you install a USB camera, Windows may automatically choose this default driver rather than the driver specifically designed for the camera. This limits the camera's functionality, and can be tedious to rectify. For this reason, it's a good idea to uninstall this default driver.

Uninstalling the default Windows USB camera driver

You can uninstall (and re-install) the Windows USB camera driver from the *Drivers* section of the ANY-maze Support page. Click here to open the *Drivers* settings now.

In this window you will see an entry for *Windows USB cameras* and next to it the status of the driver, either 'Installed' or 'Not installed'. If installed, the status will recommend that you uninstall the driver.

If the status is 'Installed', then you should first click the driver's name in the list (to select it) and then click the *Uninstall driver* button in the ribbon bar. ANY-maze will uninstall the driver, and the status will change to reflect this.

Reinstalling the default Windows USB camera driver

You can reinstall this driver at any time by simply selecting it in the drivers list and then clicking the *Install driver* button in the ribbon bar. Also note that if you completely uninstall the ANY-maze system from your computer, this driver will, if necessary, be reinstalled automatically.

ANY-maze FireWire DCAM driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Although many DV camcorders connect to a computer using a FireWire cable, they are not FireWire cameras as such. To use a DV camcorder with ANY-maze, you do not need to install this driver, indeed you don't need to install any additional software at all - they just work.

Introduction

ANY-maze is supplied with a driver for FireWire cameras that adhere to the DCAM specification. In practice, this includes all FireWire cameras that you are likely to encounter - other than DV camcorders, which don't need a driver at all.

Installing the ANY-maze FireWire driver

By default, the ANY-maze FireWire driver is installed together with the ANY-maze system, and you probably won't need to do anything in order to use a FireWire camera.

To confirm whether the driver is installed on your computer, do the following:

1. Select the Support page.
2. Select the *Drivers* option in the left-hand side-bar.
3. Look at the entry for *ANY-maze FireWire DCAM* - the status should read 'Installed'.

Click here to open the *Drivers* settings now.

If the driver isn't installed, then you should simply select it in the drivers list and click the *Install driver* button in the ribbon bar.

Problems with other drivers for FireWire cameras

Ideally, you will only have one driver for FireWire cameras installed on your computer - the ANY-maze driver. This will mean that when you plug a FireWire camera into your computer, Windows will use this driver automatically.

However, Windows itself is provided with a default driver for FireWire cameras, and it's also common for many cameras to be shipped with another driver from a company called Unibrain. If either of these drivers is installed on your computer, you will see an entry for them in the drivers list with a status of 'Installed - recommended to uninstall'.

You can uninstall these drivers by simply selecting them in the list and clicking the *Uninstall driver* button in the ribbon bar.

Windows default FireWire DCAM driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Introduction

Windows 2000 and XP are shipped with a default driver for FireWire DCAM cameras. However, this driver is very limited and can't be used by ANY-maze. Nevertheless, if it remains installed on your computer, Windows may automatically choose it as the driver to use with any FireWire cameras you install. For this reason it's a good idea to uninstall this driver.

Uninstalling the default Windows FireWire DCAM driver

You can uninstall (and re-install) the Windows FireWire DCAM driver from the *Drivers* section of the ANY-maze Support page. Click here to open the *Drivers* settings now.

Here you will see an entry for the *Windows FireWire DCAM driver* together with the status of the driver, either 'Installed' or 'Not installed'. If the status is 'Installed' then you should first click the driver's name in the list (to select it) and then click the *Uninstall driver* button in the ribbon bar. ANY-maze will uninstall the driver, and the status will change to reflect this.

Reinstalling the default Windows FireWire DCAM driver

You can reinstall this driver at any time by simply selecting it in the drivers list and clicking the *Install driver* button. Also note that if you completely uninstall the ANY-maze system from your computer, this driver will, if necessary, be reinstalled automatically.

Unibrain FireWire DCAM driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options *that all users, not just administrators, can do this.*

Introduction

A number of FireWire cameras, including the Fire-i, are shipped with a driver from a company called Unibrain.

Although this driver is recognised by ANY-maze, and you will probably be able to view images from a camera that uses it, it is not explicitly supported by the system. As a result a number of issues can arise, including an inability to view images from multiple cameras connected to the same computer.

By the way, the incompatibilities between the Unibrain driver and ANY-maze are not caused by problems with the Unibrain software (as far as we're aware, it's very good); rather they arise from the fact that ANY-maze has not been explicitly programmed to support the driver's features.

ANY-maze includes its own driver for FireWire cameras

Rather than using the Unibrain driver for a FireWire camera, we recommend that you use a driver that's included as part of ANY-maze itself. This driver is fully compatible with the system, and supports features such as viewing images from multiple cameras.

Changing drivers

If you have a camera that is currently using the Unibrain driver, then you can change over to the ANY-maze driver by first uninstalling the Unibrain driver and then reinstalling the camera using the ANY-maze driver.

Full instructions on how to do this are given in the How to uninstall the Unibrain Fire-i camera driver topic.

How to uninstall the Unibrain Fire-i camera driver

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options that all users, not just administrators, can do this.

Introduction

A number of FireWire cameras, such as the Fire-i, are supplied with a Windows driver developed by a company called Unibrain.

Although ANY-maze will recognise cameras that use this driver and will even display images from them, the system has not been explicitly developed to support it and there are a number of known incompatibilities.

Rather than using this driver, it's recommended that you use the FireWire camera driver that is supplied by ANY-maze itself. This topic explains how to uninstall the Unibrain driver, and how to install the ANY-maze driver in its place.

Installing the ANY-maze driver

Before uninstalling the Unibrain driver, it's a good idea to check that the ANY-maze driver is already installed on your computer. You can do this from the *Drivers* section of the ANY-maze *Support* page. Click here to open the *Drivers* settings now.

Here you will see an entry for *ANY-maze FireWire DCAM,* together with the status of the driver, either 'Installed' or 'Not installed'. If the status is 'Not installed' then you should first click the driver's name in the list (to select it) and then click the *Install driver* button in the ribbon bar. ANY-maze will install the driver, and the status will change to reflect this.

Uninstalling the Unibrain driver

With the ANY-maze FireWire DCAM driver installed, you're ready to uninstall the Unibrain driver. This is a relatively simple process, but for technical reasons it can't be performed by ANY-maze itself - instead you must use the Windows *Device manager.*

The exact process to follow depends on which version of Windows you are using:

- Uninstalling the Unibrain driver under Windows Vista and above
- Uninstalling the Unibrain driver under Windows XP

Uninstalling the Unibrain driver under Windows Vista and above

Part I:

1. Start the **Device Manager** - click here to do this now.
2. Double click the entry for *Imaging devices* (or click the little arrow next to this entry). A list of the cameras and other imaging devices installed on your computer will appear.
3. To check whether a camera is using the Unibrain driver, right click on it and select *Properties* from the menu that appears.
4. In the window that appears, select the *Driver* tab and check the *Driver provider*. If it is 'Unibrain', then you need to update the driver for the camera - see Part II.

Part II:

1. Having determined that you need to update the driver for a camera, you should right click on the camera's entry in the Device Manager and select *Update driver software* from the menu that appears.
2. In the window that opens, click *Browse my computer for driver software* (this may be titled *Let me pick from a list of device drivers on my computer,* depending on which version of Windows you are using).
3. On the next page, find the entry for *CMU 1394 Digital Camera Device* in the list of drivers and click on it. If the entry is not listed, it's because you have not installed the ANY-maze DCAM driver software yet. You should return to the *Drivers* section of the ANY-maze Support page and install the driver before proceeding - click here to go there now.
4. The *CMU 1394 Digital Camera Device* is not 'digitally signed', and you will see a warning to this effect.
5. Click the *Next* button and the camera's driver will be changed; you should then see a confirmation message to this effect.

If you have more than one camera connected to your computer that is using the Unibrain driver, then you should perform the above steps for each of them.

Uninstalling the Unibrain driver under Windows XP

Part I:

1. Start the **Device Manager** - click here to do this now.
2. Double click the entry for *Imaging devices* (or click the little plus sign next to this entry). A list of the cameras and other imaging devices installed on your computer will appear.
3. Look for entries that start with the word **'Unibrain'** - these are the cameras whose drivers you need to alter.

4. Right click a 'Unibrain' entry and select *Uninstall* from the menu that appears.
5. You will be asked to confirm the device removal; click *OK*.

Part II:

1. Next, select *Scan for hardware changes* from the Device Manager's *Action* menu. Windows should detect the camera you just uninstalled, and the *Found new hardware wizard* will start automatically.
2. If you are asked whether Windows can connect to Windows Update to search for a new driver, select *No, not this time*. Then click *Next*.
3. You will asked *What do you want the wizard to do*; select the option *Install from a list or a specific location,* then click *Next*.
4. On the next page choose the option *Don't search, I will choose the driver to install,* then click *Next*.
5. If you are asked to select the Hardware type to install, select the option for *Imaging devices,* then click *Next*.
6. You will now be asked which driver you want to install - select the option for *CMU 1394 Digital Camera Device*. (If this entry is not present in the list then it means you haven't installed the ANY-maze FireWire driver yet. Click *Cancel* and follow the instructions above for installing this driver, then start over with these instructions). Assuming the entry is present, select it and then click *Next*.
7. You may see a message telling you that the driver has not passed 'Windows Logo testing'; click the option to *Continue anyway*.
8. Windows will install the selected driver and then display a confirmation message. Click *Finished*. The Device Manager will update to reflect the change in the driver.

If you have more than one camera connected to your computer that is using the Unibrain driver, then you should perform the above steps for each of them, until the Device Manager's list of *Imaging devices* doesn't include any entries for 'Unibrain...'.

Adlink RTV-24 driver

*It isn't possible to install or to uninstall this driver from within ANY-maze. It is listed here only because certain early versions of the driver were incompatible with ANY-maze. To install/uninstall this driver, you should use the **Programs and features** option in the Windows 'Control Panel' (Under Windows XP, use the **Add or remove programs** option).*

Click here to open the Windows 'Control panel' now.

Introduction

The RTV-24 driver is used by the Adlink RTV-24 digitiser. Early versions of this driver were incompatible with ANY-maze, but this problem has been resolved in version 1.4.0.2 and above.

What to do if ANY-maze reports that the installed driver needs to be updated

If ANY-maze detects that the installed driver is incompatible, it will report the status of this driver as 'Updated driver required'. To resolve this, you simply need to download and install the latest driver.

Downloading the latest driver

You can download the latest driver from the Adlink web-site - note that you'll probably need to register with Adlink in order to access the file.

Installing the new driver

To install the driver, firstly ensure that ANY-maze isn't running and then run the file you downloaded. Note that to use the digitiser with ANY-maze, you only need to install the *driver*, you don't need to install the *software package*.

Once the new driver is installed, you can run ANY-maze. If you open the *Drivers* section of the ANY-maze Support page, you should see that the RTV-24 driver has a status of 'Installed'. Click here to open the *Drivers* settings now.

See also:

- RTV-24 configuration

Data Translation digitiser SDK

Only Windows administrators can install and uninstall drivers. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options *that all users, not just administrators, can do this.*

The procedure described here installs the Data Translation digitiser software development kit (SDK); it does ***not*** *install the Data Translation digitiser drivers. These must be installed from the Data Translation CD that was provided with your hardware.*

Introduction

When you install a Data Translation digitiser, you will typically install the drivers for the device you have. Although this will allow Windows to see the device, and some Data Translation programs (such as DT Acquire) will work correctly, ANY-maze won't be able to communicate with the digitiser and it simply won't be listed within the program.

The reason for this is that ANY-maze requires the **Data Translation digitiser software development kit (SDK)**, but this *isn't* installed as standard with the drivers. To overcome this you can simply install the SDK from within ANY-maze itself.

Installing the SDK

You can install the Data Translation SDK from the *Drivers* section of the ANY-maze Support page. On this page you will see an entry for *Data Translation SDK*, and next to it a status, either 'Installed' or 'Not installed.' Click here to open the *Drivers* settings now.

If the status is 'Not installed' then you should first click the entry in the list (to select it) and then click the *Install driver* button in the ribbon bar. ANY-maze will install the SDK, and your Data Translation digitiser will then be visible to the program.

You can't uninstall *the SDK from within ANY-maze. However, if you* install *it from within ANY-maze, then it will be uninstalled if you uninstall the entire program.*

National Instruments IMAQ driver

*It isn't possible to install or uninstall the NI-IMAQ driver from within ANY-maze. It is listed here because versions of NI-IMAQ prior to 3.0 are incompatible with ANY-maze. To install or uninstall this driver you should use the **Programs and features** option in the Windows 'Control Panel' (Under Windows XP, use the **Add or remove programs** option).*

Click here to open the Windows 'Control panel' now.

Introduction

ANY-maze was originally developed to work with version 2.6 of the National Instruments (NI) IMAQ driver. However, with the release of version 3.0 of IMAQ, NI updated their driver design, and in doing so they altered some functions in such a way that they are no longer compatible with applications written for version 2.6. To address this, ANY-maze was updated to work only with version 3.0 of IMAQ and above.

What to do if ANY-maze reports that a different driver needs to be installed

If ANY-maze detects that the installed driver is incompatible, it will show the status of this driver as 'Different driver required'. To resolve this, you simply need to download and install the latest IMAQ driver.

Downloading the latest IMAQ version driver

You can download the IMAQ drivers from the National Instruments web-site - note that you may need to register with NI in order to access the file. This is the latest version at time of writing (March 2016).

Installing the driver

Before installing the driver, you **MUST** first fully uninstall any IMAQ software you already have installed. You can do this using the **Programs and features** option in the Windows 'Control Panel' (Under Windows XP, use the **Add or remove programs** option). Click here to open the Windows 'Control Panel' now.

Once you've uninstalled any existing driver, you simply need to run the downloaded file to install the updated version of IMAQ. If you then open the *Drivers* option on the ANY-maze Support page, you should see that the IMAQ driver has a status of 'Installed'. Click here to open the *Drivers* settings now.

Microsoft MPEG-4 codec

Only Windows administrators can install and uninstall drivers and codecs. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options *that all users, not just administrators, can do this.*

Introduction

If you wish to record standard format movies from within ANY-maze, then you will need to have suitable video codecs installed on your computer.

A *codec* is a piece of software that encodes and decodes the compressed video pictures that go to make up a movie. All digital videos are encoded in some way or another, and to decode them you need an appropriate codec. In fact, most videos are encoded using one of a number of standard systems, such as MPEG or Indeo, and most versions of Windows include codecs to read them.

However, most of the codecs provided with Windows (particularly more recent versions) are designed to decode (read) videos but not to encode (record) them. As a result, you may find that any videos you record within ANY-maze use a rather old, inefficient encoding system, as the program will be limited to using whatever codecs it can find.

To overcome this, if you're running Windows XP, you can install the Microsoft MPEG-4 codec, which produces small files with good image quality.

Installing the codec

To install the codec, select the Support page, and then the *Drivers* option in the left-hand side-bar. Click here to open the *Drivers* settings now.

On this page, you will see an entry for *Microsoft MPEG-4 codec* and next to it the status of the codec, either 'Installed' or 'Not installed'. If the status is 'Not installed' then you should first click the codec's name in the list (to select it) and then click the *Install driver* button in the ribbon bar. ANY-maze will install the codec, and will then use it by default in any movie files you record.

You can't uninstall *the MPG4 codec from within ANY-maze. This is because it is likely to be used by other programs, such as media players.*

Operating systems other than Windows XP

For operating systems from Windows Vista onwards, we recommend you download and install one of the following codec packs, which contain MPEG-4 encoders:

- Microsoft Media Video 9 VCM codec
- Xvid codec

These codecs should work on all operating systems from Windows XP onwards, although they don't necessarily work well on all computers. Try each of them to see which works on your computer.

ANY-maze AVI recorder

Only Windows administrators can install and uninstall drivers and codecs. You'll also need to be an ANY-maze administrator, unless your administrator has specified in the Administrative options *that all users, not just administrators, can do this.*

Introduction

If you wish to record standard format movies (as AVI files) from within ANY-maze, then you will need to install the ANY-maze AVI recorder. (Note: you will also need a suitable codec, such as the Microsoft MPEG-4 codec.)

For the technically minded, the ANY-maze AVI recorder is actually implemented as a DirectShow 'push-source' filter.

Installing the AVI recorder

You can install the ANY-maze AVI recorder using the *Drivers* option on the ANY-maze Support page. Click here to open the *Drivers* settings now.

In this window you will see an entry for *ANY-maze AVI recorder* and next to it the status of the recorder, either 'Installed' or 'Not installed'. If the status is 'Not installed', then you should first click the recorder's name in the list (to select it) and then click the *Install driver* button in the ribbon bar.

You can't uninstall *the AVI recorder from within ANY-maze, because it will simply be reinstalled automatically whenever you restart the program. However, if you uninstall the entire ANY-maze system, then the recorder will be uninstalled too.*

Updating drivers under Windows XP and Vista

An easy way to get back to this topic is to search for the words 'update driver XP' or 'update driver Vista'. You can use words like 'updating' and 'drivers' too, so for example, 'Updating drivers for Windows XP' would also work.

Introduction

As you probably know, Microsoft have stopped supporting Windows XP and Vista and this means that the drivers ANY-maze uses for USB devices, such the ANY-maze interface and the ANY-maze USB camera, are no longer 'signed' to work with *these* operating systems. However, the drivers *are* signed, and they do work correctly with these older operating systems.

What this means in practical terms, is that Windows will 'prefer' old signed drivers over the more up-to-date unsigned drivers, and so even after installing the updated drivers it will continue to use the older drivers. In this topic I'll show you how to convince it to use the new drivers.

Step-by-step instructions

The process of changing Windows XP or Vista to use the newer unsigned drivers is simple, but a little long-winded. In the instructions below I explain how to do this for XP, but the steps are the same for Vista, although the screens might look a bit different:

Step 1

First, close all running applications, including ANY-maze (so you'll probably want to print these instructions).

Step 2

Next, right click on the 'My computer' icon on your desktop and select 'Properties' from the menu which appears - see Figure 1.

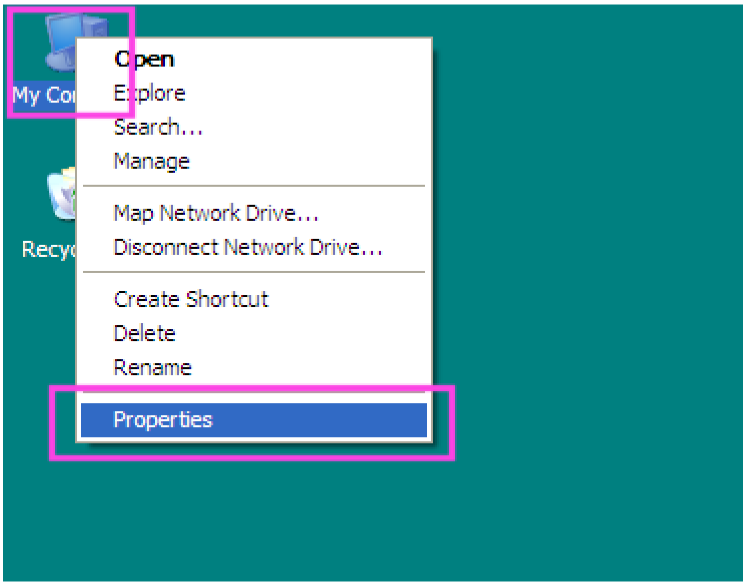


Figure 1. Right click on 'My computer' and select 'Properties'

Step 3

The 'System properties' window will open. Select the 'Hardware' tab and then click on the 'Device manager' button - see Figure 2. [Under Vista select the 'Device Manager' link shown in the top-left of the window that opens.]

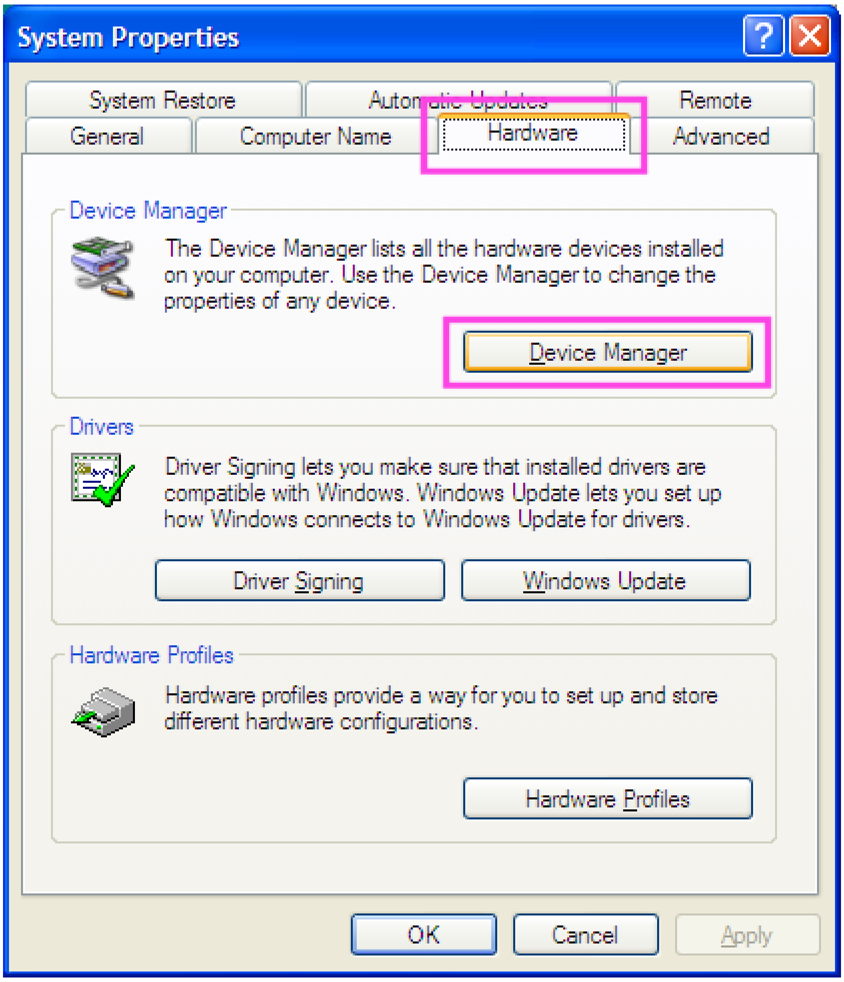


Figure 2. Select the 'Device manager' button on the 'Hardware' tab

Step 4

The Windows 'Device manager' window will open. Look for 'Imaging devices' in the list and click to open it. Then look for any items which start with 'The Imaging Source Europe GmbH', right click the first of them and select 'Update driver...' from the menu that appears - see figure 3. If you can't find any entries like this don't worry, it just means you don't have this driver installed and you can jump to step 10.

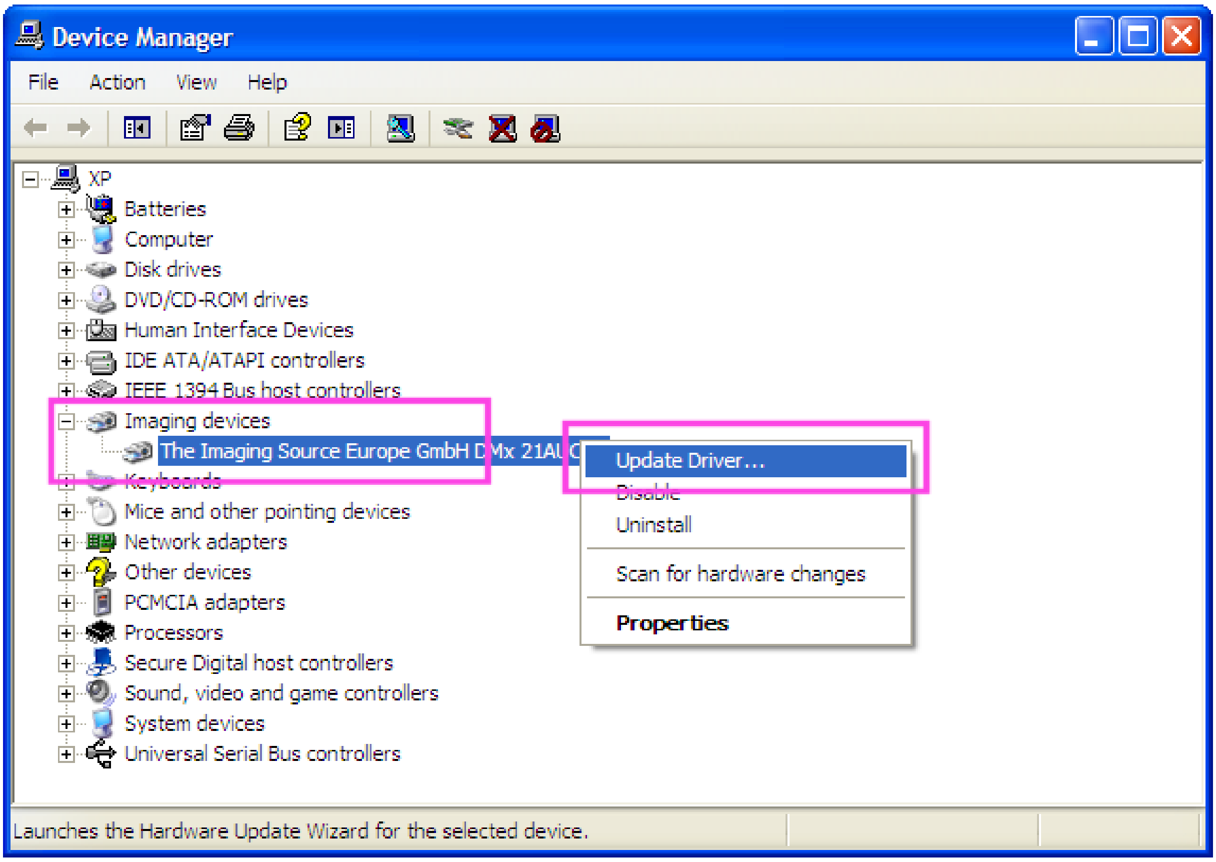


Figure 3. Choose the option to 'Update driver...' for the first 'The Imaging Source Europe GmbH...' 'Imaging device'

Step 5

The 'Hardware update wizard' will open. Choose the option to 'Install from a list or a specific location'. This will *not* be the option that's selected by default, so you will have to change it - see figure 4. Then click next.

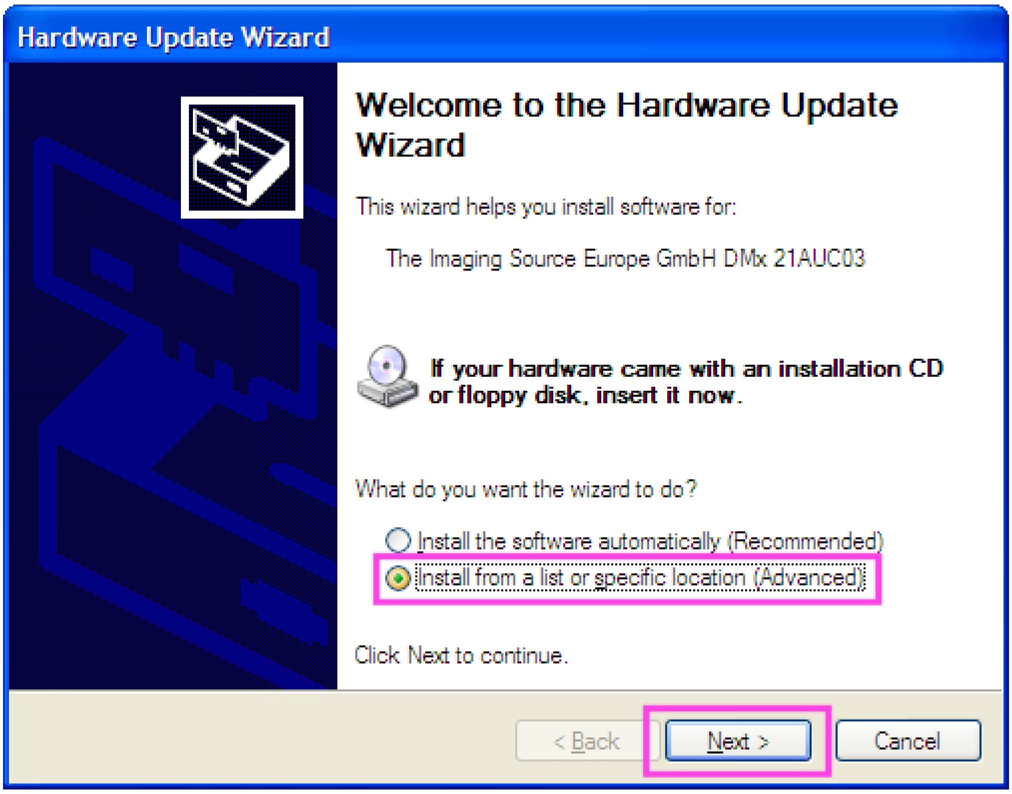


Figure 4. On the 'Hardware update wizard' make sure you choose the option to 'Install from a list or a specific location'

Step 6

On the next page select the option 'Don't search. I will choose the driver to install.' - see figure 5. Then click next.

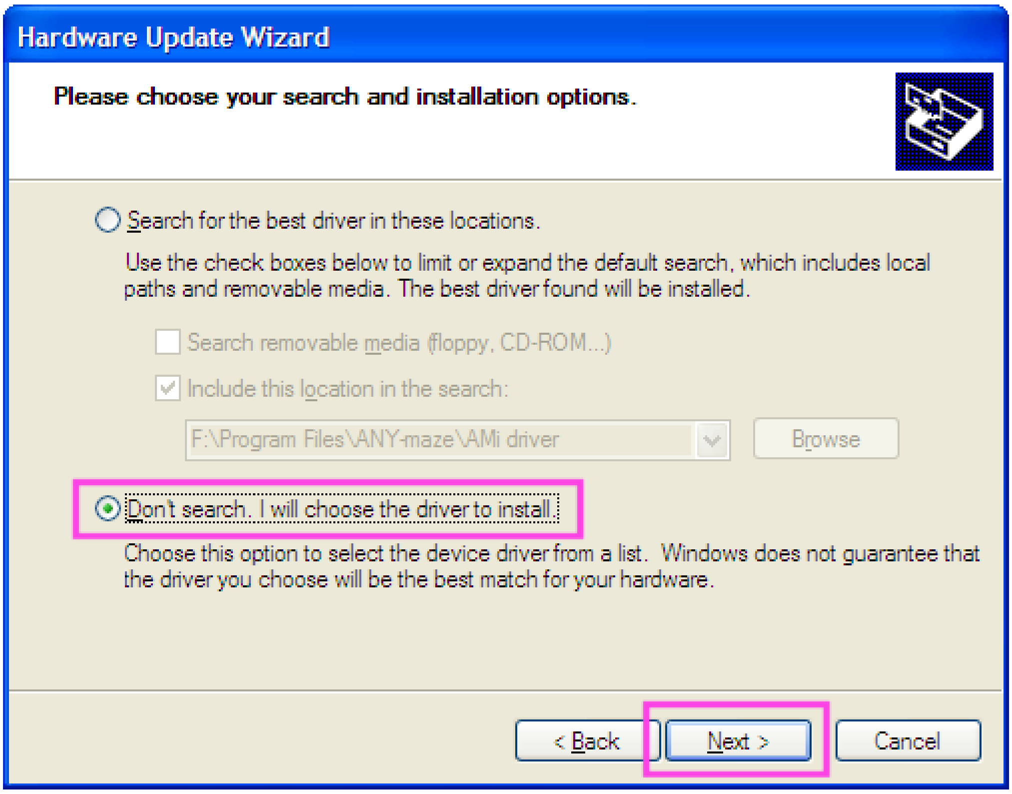


Figure 5. Select the option 'Don't search. I will choose the driver to install.'

Step 7

On the next page make sure the option to 'Show compatible hardware' is selected. You should then see a list of drivers. Select the one with the **highest version number**, for example, this is '2.8.9.1466' in figure 6, below. When you select this driver you will see a warning 'This driver is not digitally signed' - this is expected.

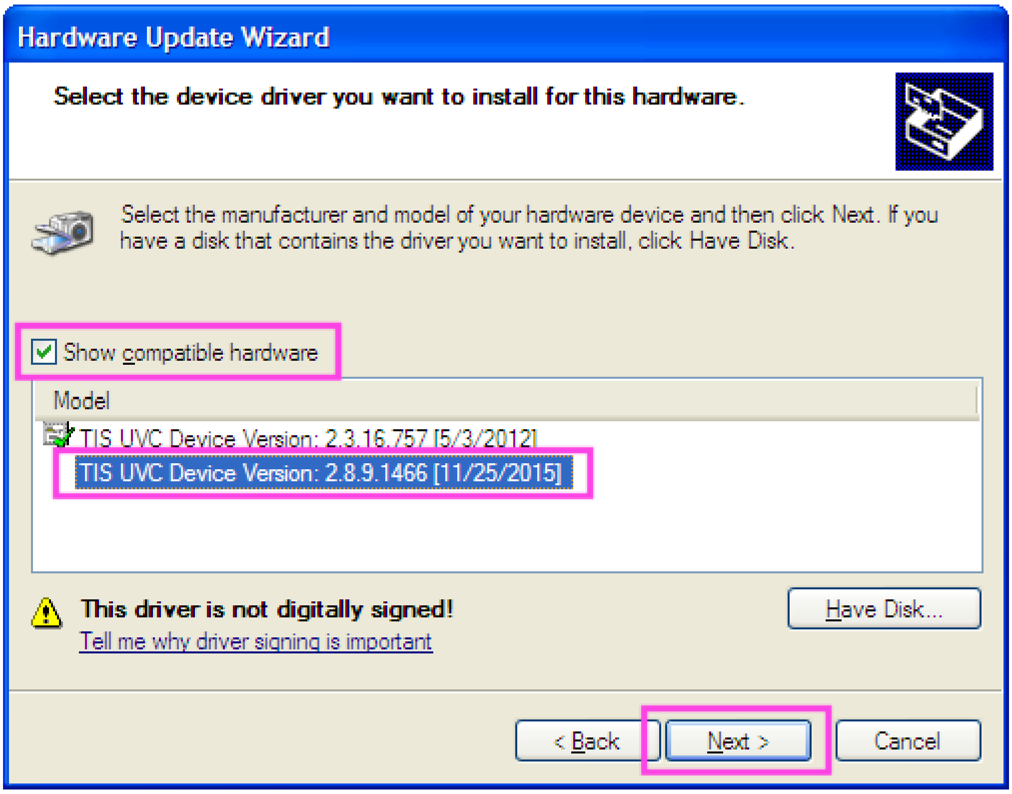


Figure 6. Select the driver from the list that has the highest version number. Don't worry that this says it is not digitally signed.

Step 8

Click 'Next' and the driver will be updated, this may take a little while.

Step 9

If there was more than one entry in the list on 'Imaging devices' that started with 'The Imaging Source Europe GmbH' (see step 4), then you should now repeat steps 4 - 8 for each of them.

Step 10

You should now be back at the 'Device manager' window. Look for 'Universal Serial Bus Controllers' in the list and click to open it. Then look for an item called 'ANY-maze USB devices', right click on it and select 'Update driver...' from the menu which appears - see figure 7.

If you can't find this item don't worry, it just means you don't have this driver installed... and so you're done!

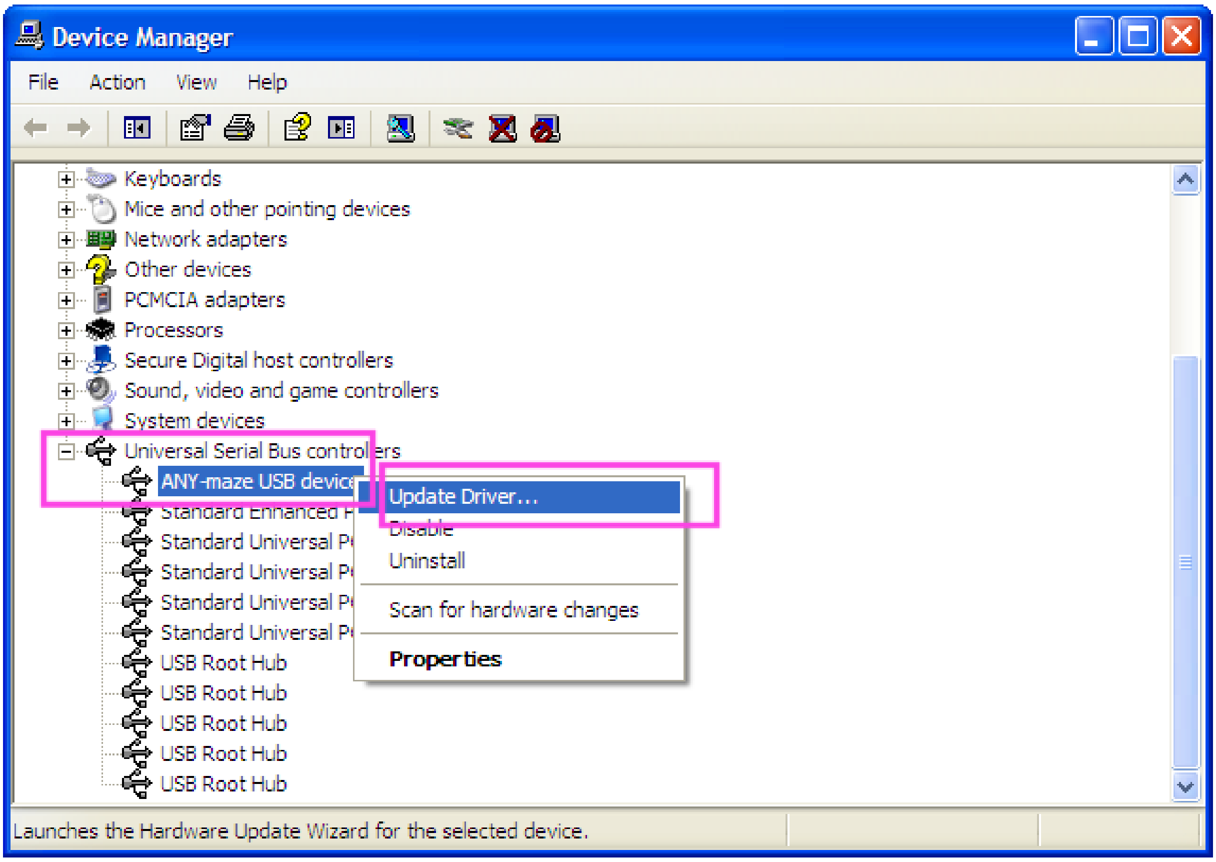


Figure 7. Choose the option to 'Update driver...' for the 'ANY-maze USB devices' driver under 'Universal Serial Bus Controllers'

Step 11

The 'Hardware update wizard' will open. Choose the option to 'Install from a list or a specific location'. This will *not* be the option that's selected by default, so you will have to change it - see figure 8. Then click next.

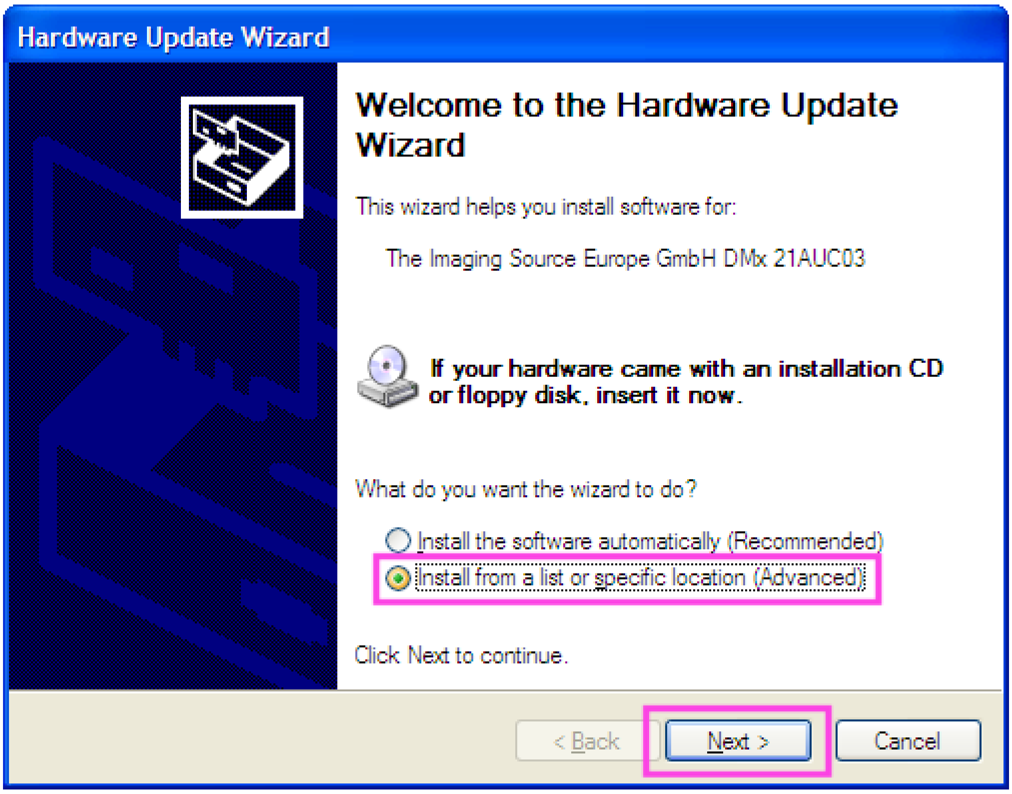


Figure 8. On the 'Hardware update wizard' make sure you choose the option to 'Install from a list or a specific location'

Step 12

On the next page select the option 'Don't search. I will choose the driver to install.' - see figure 9. Then click next.

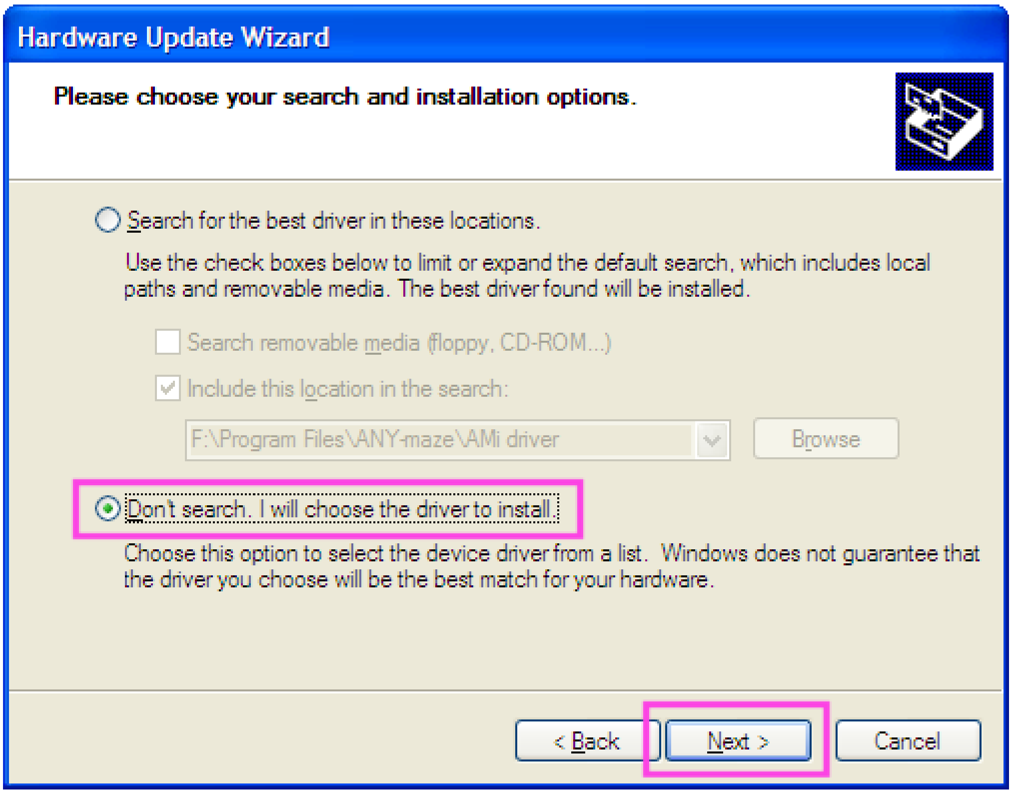


Figure 9. Select the option 'Don't search. I will choose the driver to install.'

Step 13

On the next page make sure the option to 'Show compatible hardware' is selected. You should then see a list of drivers. Select the one with the **highest version number**, for example, this is '2.12.18.0' in figure 10, below. When you select this driver you will see a warning 'This driver is not digitally signed' - this is expected.

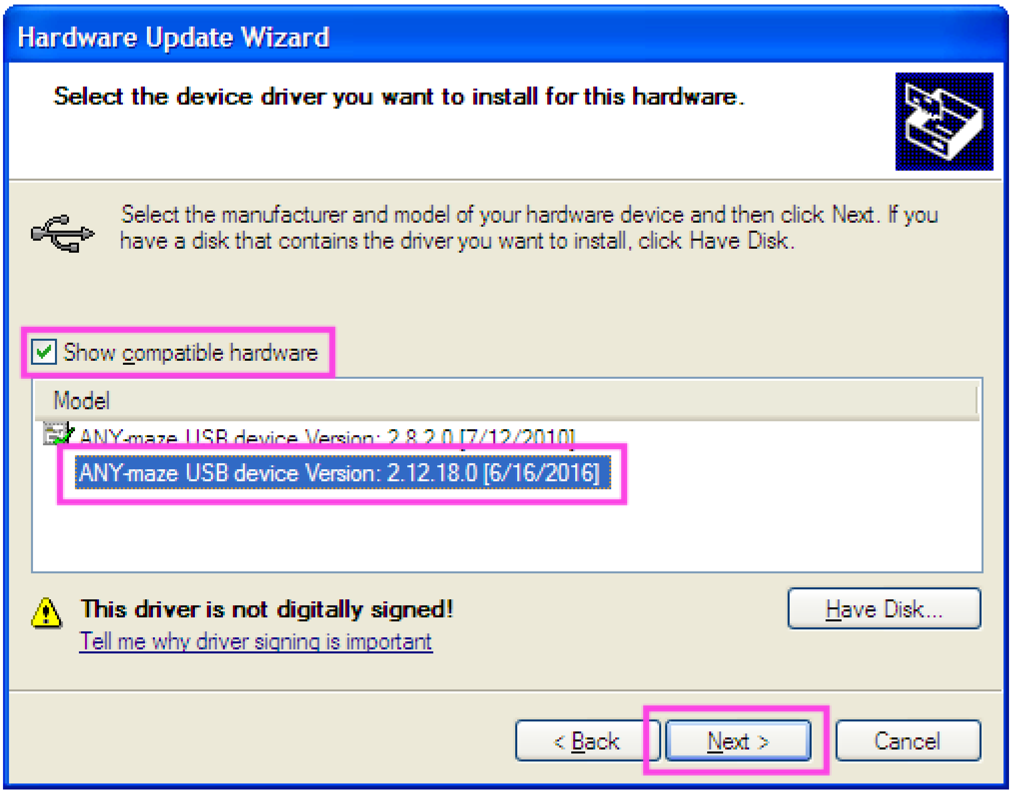


Figure 10. Select the driver from the list that has the highest version number. Don't worry that this says it is not digitally signed.

Step 14

Click 'Next' and the driver will be updated, this may take a little while.

That's it, you can close all open windows. It's best to then restart your computer.

USB Viewer

Overview

The *USB viewer* on ANY-maze's Support page allows you to see the USB devices attached to the computer, and which USB controllers they are attached to. This is particularly useful if you are using more than one USB camera to track animals in multiple apparatus.

Details

ANY-maze can use multiple cameras to allow simultaneous tracking in more than one apparatus at a time. However, if the cameras used are attached to the ANY-maze computer via USB, then problems may arise if more than a few cameras are used. The reason for this is because of a limit to USB *bandwidth* - i.e. the amount of data that can be transferred across a USB connection in a given period of time.

Although it might seem that different cameras use different USB ports, so it shouldn't be a problem, in reality the bandwidth limit applies to the USB *controller* inside the computer, and not the port. So a notebook computer, for example, may have 3 USB ports, but internally they could all connect to the same controller and therefore they share the same bandwidth.

Usually, you can connect around 3 USB cameras to a single *controller* before you start to run into problems, so if you need more than 3 cameras, you'll need to ensure that they are on different controllers. Fortunately, most computers have multiple USB controllers in them, so you just need to make sure that each of your computer's USB controllers has no more than 3 cameras attached to it.

So how do you know which ports are on which controllers? Well, that's what the *USB viewer* on the Support page is for. It will show you a list of all the USB ports on your computer, and more importantly, which USB *controllers* they are connected to:

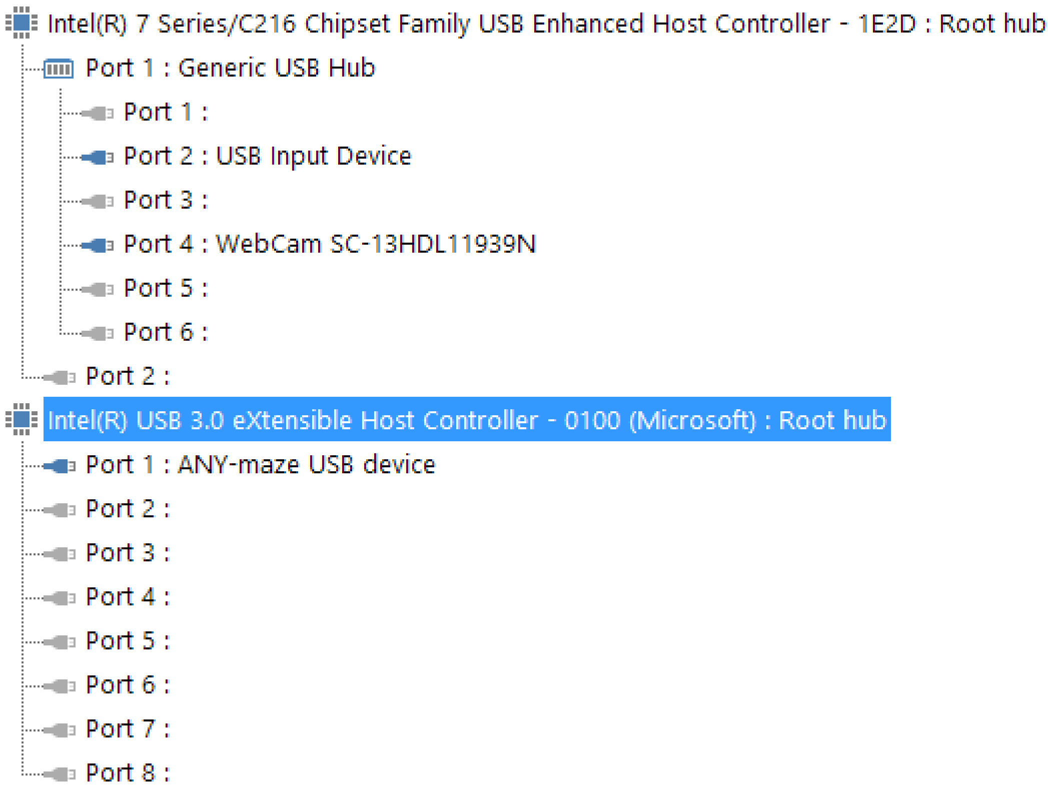


Figure 1. The USB viewer on the ANY-maze Support page. Note that although there's a total of 15 available USB ports, these only reside on two controllers.

The USB viewer will dynamically update as devices are inserted and removed, so you can easily plug and unplug your cameras into different USB ports to see which controllers they are connected to.

Click this link to open the USB Viewer now.

Further information

There are more details on using multiple cameras, and what to do if you don't have enough USB connectors in your computer, in the Connecting multiple cameras topic.

See also:

- Cameras and videos

Advanced Support

Overview

ANY-maze is an extremely complex software system, and there can be occasions when things don't work quite as expected. When this happens, ANY-maze engineers may need access to some advanced diagnostic and debugging tools.

The *Advanced support* section of the Support page is where such tools are kept. However they are not options that should be accessed under general use. For this reason, they're not described in any detail here.

*These advanced support options should **not** be used unless you are explicitly instructed to do so by ANY-maze technical support. Incorrect usage could prejudice your experiments!*

The Help page

Unlike the pages on the left side of the ribbon bar, you can access the Help page without having to open an experiment first.

Overview

ANY-maze includes well over 1000 help topics, covering all aspects of the system. You can find information either by searching, or by navigating a table of contents. Individual topics, or entire sections of the help, can be printed and topics can easily be copied to the Windows clipboard, saved to files or e-mailed to other people.

Accessing ANY-maze help

There are a number of ways of accessing help in the ANY-maze software. You can simply change to the Help page, by clicking the *Help* tab at the right-hand end of the ribbon bar, or you can click the button near the right-hand end of ANY-maze's title bar, which will open a separate pop-out help window. Many parts of ANY-maze also include *Help* buttons or links which take you directly to relevant topics.

- Getting help on whatever you're doing now
- Searching help
- Browsing the help contents
- Printing help topics
- Copying, saving and e-mailing help topics
- Toggling whether help is context-sensitive
- Showing help in a separate window

Getting help on whatever you're doing now

A common reason for wanting help is, of course, because you're stuck - for example, you're looking at a window which contains options you don't understand. Because the ANY-maze help is *context-sensitive* by default, getting help in these situations is as simple as switching to the Help page - the help will display a topic relevant to whatever it is you are doing.

For example, if you're creating a protocol and you've just added a new *Zone,* then switching to the Help page will show the 'An introduction to zones' help topic - this provides advice on how to divide your apparatus into zones, as well as describing the exact steps you need to perform to do this.

Once you've read the help topic, you can quickly return to whatever you were doing using the *Back* button in the ribbon bar.

Searching help

Searching ANY-maze help is straightforward but powerful. The system responds both to simple word searches such as 'Zone', as well as to questions like 'How do I create a zone?'.

Search results are ranked, with the topic ANY-maze thinks is most likely to answer your question shown first. The results will include brief abstracts of the topics, making it easier for you to quickly identify those you want to read. The Searching ANY-maze help topic will give you more information on searching the help, including some hints to help you find the information you need.

Browsing the help contents

The ANY-maze help is divided into a number of different 'books', and each of these has its own button in the ribbon bar:

- *What's new* describes the changes made in this version of the software.
- *Getting started* will get you up and running when you first start using the software.
- *Using ANY-maze* is a task-oriented introduction to using the software, to help you quickly learn how to perform some common operations such as setting up an experiment, running tests and viewing results.
- *Cameras and videos* contains details about the cameras and video file formats supported by ANY-maze.
- *I/O devices* lists the I/O devices currently supported by ANY-maze, with links to specific topics for each one.
- *ANY-maze reference* is a full, comprehensive description of every part of the ANY-maze software.
- *Troubleshooting* gives solutions to some common problems encountered when using ANY-maze.
- *About ANY-maze* brings together a range of information about ANY-maze and your computer.

Selecting one of these books will cause the first page of that book to open, and the contents of the book to be shown on the left-hand side of the page. To navigate within the contents, simply click the name of the topic - the topic will be shown in the main area of the help page and, if the topic includes any sub-topics, these will be shown in the contents below the selected topic. Note that if you follow a link from one help topic to another, and the new help topic is in a different book, then the new book will become selected in the ribbon bar and the list of contents on the left-hand side will change to reflect the contents of the new book. If at any point you want to go back to the previous topic, you can just use the *Previous topic* button to go to the last help topic you were looking at;

alternatively the Back can be used, but this works across the whole system and not just the help, so if the last thing you were looking at was a different page, then the *Back* button would take you to that page.

Both the Previous topic *and* Back *buttons work within a topic as well as between topics, so if you follow a link that takes you to a different position within the same topic, then these buttons will return you to the original position in the topic.*

Selecting the *Help contents* button will take you to a main 'contents' page for the entire help system, allowing you to choose one of the books from a list.

Printing help topics

To print a help topic, simply click the *Print* button. This will display a message asking if you'd like to print just the current topic, or its sub-topics too. For example, if you're reading a topic about how to set up a piece of apparatus in a protocol, then there will be sub-topics which describe additional information such as how to draw the apparatus map. If you're printing the topic because you want to sit down quietly and read it, then it would make sense to print these sub-topics too.

In fact if you print the 'root' page of any of the ANY-maze books, then printing the topic *and* its sub-topics will print the entire book. This is a great way to create a printed manual for ANY-maze - although be warned, some of the books (particularly the *ANY-maze reference*) - contain a lot of pages!

Copying, saving and e-mailing help topics

You can copy a topic, or a selection from a topic, to the Windows clipboard simply by clicking the *Copy* button. (To select only a portion of a topic, you need to move the mouse cursor to the left-hand margin of the topic, then you can drag up and down to select the lines of text adjacent to the cursor). Copied topics can be pasted directly into Microsoft Word, or other major word processing programs, without losing any of their formatting.

Likewise, you can save a topic to a file by clicking the *Save* button. Topics can be saved as RTF, HTML, plain text or web archive files - see the Saving documents topic for more details.

If you want to e-mail a help topic to a colleague then you can do so by simply clicking the *Send as e-mail* button - the *Send e-mail* window will open, where you can enter the recipient's e-mail address and add a message to go with the topic. See Sending documents by e-mail for more details.

Toggling whether help is context-sensitive

By default, the Help page is context-sensitive - that is, wherever you are in the ANY-maze software, if you select the Help page then the help will open on the relevant help topic for that part of the software. If you wish, you can turn this off by un-checking the *Synchronise topic with current context* box at the right-hand end of the ribbon bar. This will mean that whichever topic you select on the Help page will still be selected when you return to the Help page, regardless of what you've done in the meantime.

Showing Help in a separate window

It's all very well being able to switch to the Help page to show the help on whatever you're currently doing, but this can be inconvenient - for example if you're running through a series of steps and want to look at the help for each of those steps as you do so. You'd have to repeatedly switch to the Help page, read the help, and switch back to the previous page to perform each step.

To make this easier, you can open the Help in a separate 'pop-out' window if you need to. This window can then be moved around the screen to a convenient location, and you can use one of the other ANY-maze pages while still being able to see the help - without having to repeatedly change between the two.

You can open the 'pop-out' help window in one of the following ways:

- Using the button in the ANY-maze title bar
- From the Help page itself, using the *Pop out* button in the *Window* section of the ribbon bar

For more details, see The pop-out help window.

Searching ANY-maze help

Introduction

You will often find the quickest way to get the answer to a question is to search the ANY-maze help. You can do this by simply clicking the *Search topics* button in the Help page's ribbon bar to display the search options - see figure 1.

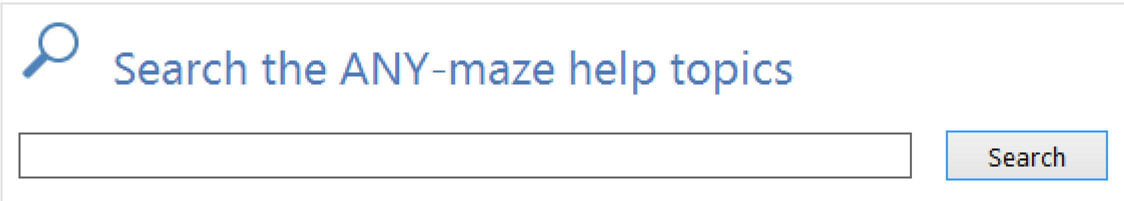


Figure 1. The 'Search' options on the Help page.

- Simple searches
- Asking ANY-maze a question
- Topic abstracts
- Advanced search features

Simple searches

You can search ANY-maze help by entering any number of words on the subject that you need help on. You can use a single word, for example 'Camera', and all topics that contain that word will be listed. However, this might turn up rather a lot of topics, so you might want to try more than one word - like 'USB camera'. This will help narrow down your search, as ANY-maze first tries to find topics that contain *all* your search terms. ANY-maze will treat similar words in the same way, so searching for the word 'camera' or 'cameras' will find the same topics. Similarly, for example, 'Configuring an AMi-2 device' will return the same topics as 'How to configure an Ami2 device', because ANY-maze knows that 'configure' and 'configuring' mean the same thing, and 'AMi-2' and 'Ami2' are equivalent. Searches are case-insensitive as well, so it doesn't matter if you use capital letters or not.

Topics that contain your search terms in their title, or which ANY-maze thinks are particularly relevant to the word(s) you've searched for, will be listed first. After this, ANY-maze will list topics that contain all the words *somewhere* in the topic's text. Priority will be given to topics where the words are closer together in the text. Further topics will be listed that contain *any* of the words you've searched for; again, the closer the words are in the topic, the higher the topic will be in the list of search results.

A maximum of 25 topics will be listed. Occasionally, you may find that the answer to your question isn't the first topic listed, although it will usually be in the top three or four - reading the

- topic abstracts generally makes it easy to find the right one. If the topic you're looking for is not listed at all in the search results, you could try changing the wording slightly, or adding words to narrow down what you're looking for.

Asking ANY-maze a question

Although you can search the help by asking it questions, such as 'Can I use a USB camera with ANY-maze?', ANY-maze will ignore words which aren't particularly relevant to the help topics themselves. In the example, the words 'Can', 'I', 'a' and 'with' will actually be ignored (as will the word 'ANY-maze'), and ANY-maze will simply search on the terms 'USB' and 'Camera'. So although you *can* enter an English-language question to search on, ANY-maze will pick out the important words and search for those. So 'Can I use a USB camera with ANY-maze' will be treated in the same way as 'Using USB cameras with ANY-maze' or 'USB camera'.

Topic abstracts

To help you quickly identify the topic which answers your question, ANY-maze will display short *abstracts,* which briefly describe the topics' contents. This can make it much easier to find the help topic you're looking for:

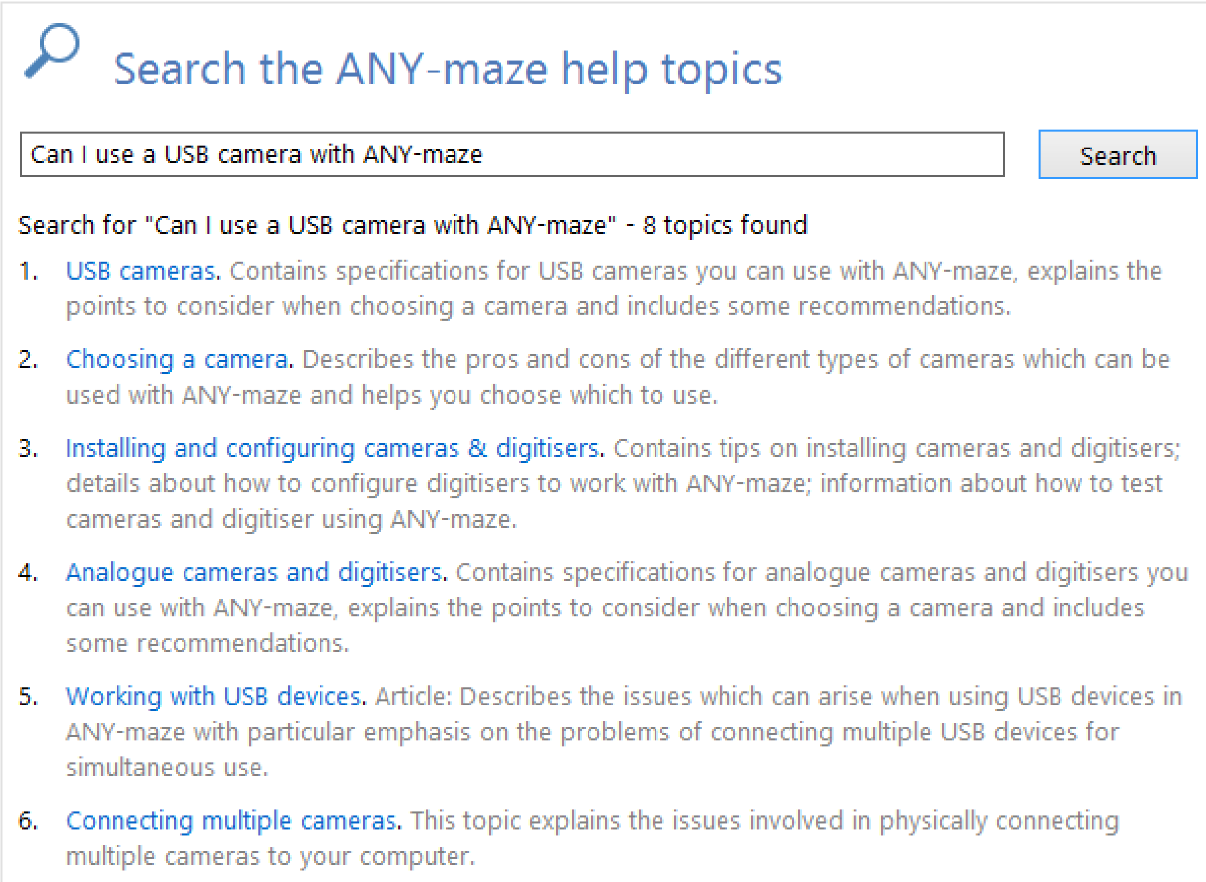


Figure 2. Topic abstracts give a brief summary of each page, which can make it much easier to find what you're looking for in the list of pages that ANY-maze has found.

Advanced search features

There are some more advanced features of searching the help index in ANY-maze, which you might be aware of if you've used them in other search engines - Google, for example. They are listed here:

- If you want to search for a specific phrase, then you can enclose it in double-quotes ("..."). This will only find topics where that exact phrase is found.
- If you want to search for topics that contain a certain word but *not* another word, you can use '+' and '-'. For example, searching for '+analogue -digital' will find all topics that contain the word 'analogue' but *not* the word digital, for example.

About ANY-maze

Overview

The *About ANY-maze* option on the Help page brings together a range of information about ANY-maze and your computer. To access it, just click the *About ANY-maze* button in the ribbon bar of the Help page - or you can simply click this link to go there now.

Details

The information available includes:

- The version number and release date of your copy of ANY-maze.
- The serial number of your copy of ANY-maze.
- Your licence number and its expiry date (if you've licensed your copy of ANY-maze).
- Acknowledgements and legal notices.
- Technical details about your computer, including the type and speed of the processor, the amount of memory, the type and capacity of your disk drives and information about any installed video digitisers and/or cameras that ANY-maze can use.

*The same information is available on the **Support page**.*

The pop-out help window

Overview

Help can be accessed from within ANY-maze at any time, simply by switching to the Help page. However this can be inconvenient if you want to see the help at the same time as the page you're currently working on.

ANY-maze allows you to see both things at once by 'popping out' a dedicated help window that you can position anywhere on the screen and continue working, without having to switch between pages. This is particularly useful if you've got more than one monitor connected to your computer, as you could have the main ANY-maze software open on one screen, and the popped-out help window on another.

Note that unlike the built-in Help page, the pop-out help window is *not* synchronised to the current context, so whichever topic you select in this window will remain selected, regardless of what you do in the ANY-maze software.

- Opening the pop-out help window
- Showing expanded or reduced controls
- Auto-sizing the popped-out window
- Showing the help topic for the current context

Opening the pop-out help window

You can open the pop-out help window in one of the following ways:

- Using the button in the ANY-maze title bar
- From the Help page itself, using the *Pop out* button in the *Window* section of the ribbon bar

If you use the button in the title bar, and the pop-out help window is already open but not visible, it will be brought into view and the topic shown will be the one relevant to whatever you are currently working on.

Showing expanded or reduced controls

By default, the pop-out help window will show a minimal window, with no ribbon bar and only a small selection of controls. This is to try and reduce the amount of screen space that the window takes up, only showing the relevant help topic and nothing else.

If you want to show a full ribbon bar and help contents, use the button to change the window to a 'full' window. Note that the size of the window doesn't change, so you might need to resize it to see everything that you need to. To change back to the minimal window, use the *Show reduced controls* button in the *Window* section of the ribbon bar.

Once you're viewing the window with full controls, you can toggle whether or not to view the help contents using the *View contents* button in the ribbon bar.

Note that all details to do with the layout of the help window, such as whether the window is being shown full or reduced, and whether the contents side-bar is being displayed, are stored so that the pop-out help window will open in the same way the next time it is used. These layout settings are stored separately for each ANY-maze user.

Auto-sizing the popped-out window

If you've got the help window popped-out, you can ask ANY-maze to automatically set it to a sensible size using the *Auto-size window* button. This will tile the ANY-maze window and the help window on the screen, such that the help window takes up a small column of space to the right of the screen (enough to see a sensible amount of the current help topic), and ANY-maze fills the rest of the screen.

Showing the help topic for the current context

As I mentioned above, the pop-out help window is not synchronised to the current context. Whichever topic you select in this window will remain selected, regardless of what you do in the ANY-maze software. However you can quickly ask it to display the relevant help topic for the current context, using the *Synchronise now* button.

Working with files and reports

ANY-maze experiment files

In ANY-maze, each experiment you work with is stored in its own individual file, and so to do anything useful, you first have to create or open a file.

In fact, as every experiment is stored in a file, ANY-maze tends to just refer to **experiments** rather than **experiment files,** but the two terms in this context are synonymous.

For full details about working with experiment files, refer to the following topics:

- Creating new experiments
- Opening existing experiments
- Saving experiments
- Filing options

ANY-maze protocol files

Although on a day-to-day basis you'll be working with **experiment** files, as described above, all experiments are based on a **protocol,** and this protocol must be set up before you can perform an experiment.

Once a protocol has been set up for an experiment, you can save it (separately from the experiment itself) as a **protocol file,** which means that you can reuse this protocol for as many experiments as you like. In future, instead of having to set up your experiment from scratch again, you can just create a new experiment based on an existing protocol, which is obviously much quicker!

For full details about working with protocol files, refer to the following topics:

- Saving and loading protocols
- Creating new experiments
- Filing options

Working with reports and data

As you'd expect, reports that you can view on screen can also be printed, copied to the Windows clipboard, saved in files or even sent in e-mails. In the latter two cases, ANY-maze can create HTML versions of its reports - this means that the report can be viewed using a web browser on any type of computer, including Apple Macs.

- Printing reports
- Saving reports and data

* Copying reports and data
* Sending reports and data by e-mail

Printing reports

Overview

Printing reports in ANY-maze is as simple as clicking the *Print* button in the ribbon bar. This will drop a short menu, giving you the following options:

- *Quick print* will print the currently-displayed report directly to your default printer, using that printer's current settings.
- *Print...* will give you more control over how a document will be printed. This will open the print window, allowing you to select which printer to use, how many copies, and the print settings (for example, the quality of the printout or whether to print in colour or black and white).
- *Page set-up...* will allow you to alter the page set-up (paper size, margins etc.) of the document to be printed.

For more information about printing reports, refer to the following topics:

- Using the print window to control how a document is printed
- Altering paper size, orientation and margins
- Setting headers and footers to print on documents
- Setting options to print pages in reverse order and/or in black and white.

Printing documents

Overview

The easiest way to print a document in ANY-maze is to click the *Print* button in the ribbon bar, and select *Quick print* from the menu that appears. A single copy of the current document will be sent directly to the default printer (the name of the default printer will be shown under the *Quick print* option in the menu).

However, there may be occasions when you want a little more control over how a document is printed, in which case you should use the *Print...* option from the *Print* button's menu. This will cause the *Print* window to open.

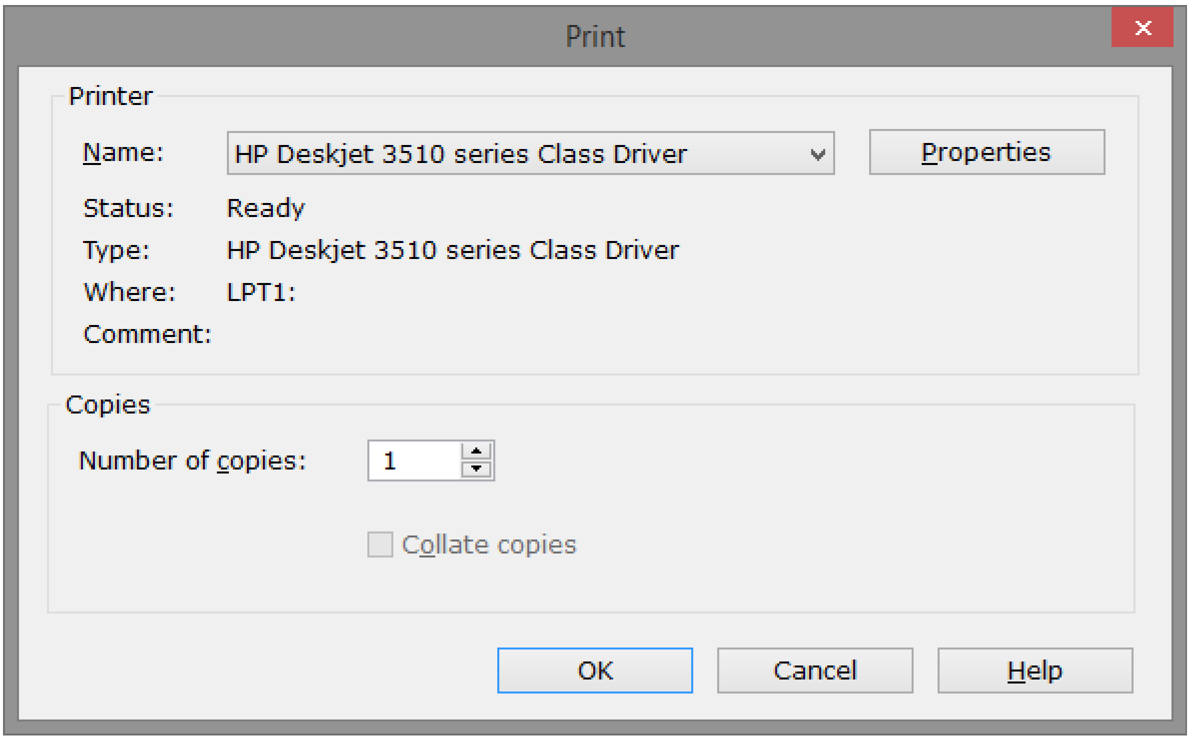


Figure 1. The Print *window, which you can use to change which printer a document will be sent to, and choose how many copies you want to print.*

Details

Using the *Print* window, you can choose where a document is printed, how many copies are printed and how multiple copies are collated. You can also use the *Properties* button to access the printer's property pages, which may include other useful options such as print quality.

Selecting the printer to use	By default, documents you print in ANY-maze will be sent to the Windows default printer. However, you can override this default setting for an individual document by choosing a different printer from the list shown on this window.
Changing a printer's properties	Most printers include their own property settings which can be used to control such things as print quality, paper size, print order etc. - the exact options depend on the individual printer. By clicking the *Properties* button on the *Print* window, you can access the printer's own settings window and alter these properties for the document you're about to print. You should take care with properties which duplicate those already available in ANY-maze. For example, some printers include the ability to print documents in reverse order, but so does ANY-maze - if you set them both, then the document will NOT be printed in reverse order because ANY-maze will reverse the document and the printer will reverse it back again! Other duplicated options, such as paper size, will usually override the settings you've made in ANY-maze, but only for the document you're about to print.
Printing multiple copies	If you want to print more than one copy of a document, then you can simply alter the setting on the *Print* window for *Number of copies.* When you do this, you can also choose whether or not the pages are collated - i.e. an entire copy of the document is printed followed by another entire copy. Note that the *Collate copies* option is disabled when a document contains just a single page.

See also:

- Altering paper size, orientation and margins
- Setting headers and footers to print on documents
- Setting options to print pages in reverse order and/or in black and white.

Altering page set-up

Overview

The *Page set-up* window can be used to alter the paper size, orientation or margins. The changes you make will be retained by the system and applied to all documents that **you** print - note that other users have their own settings, so any changes you make won't affect them.

The *Page set-up* window is opened by clicking the *Print* button in the ribbon bar, and selecting the *Page set-up...* option from the menu that appears.

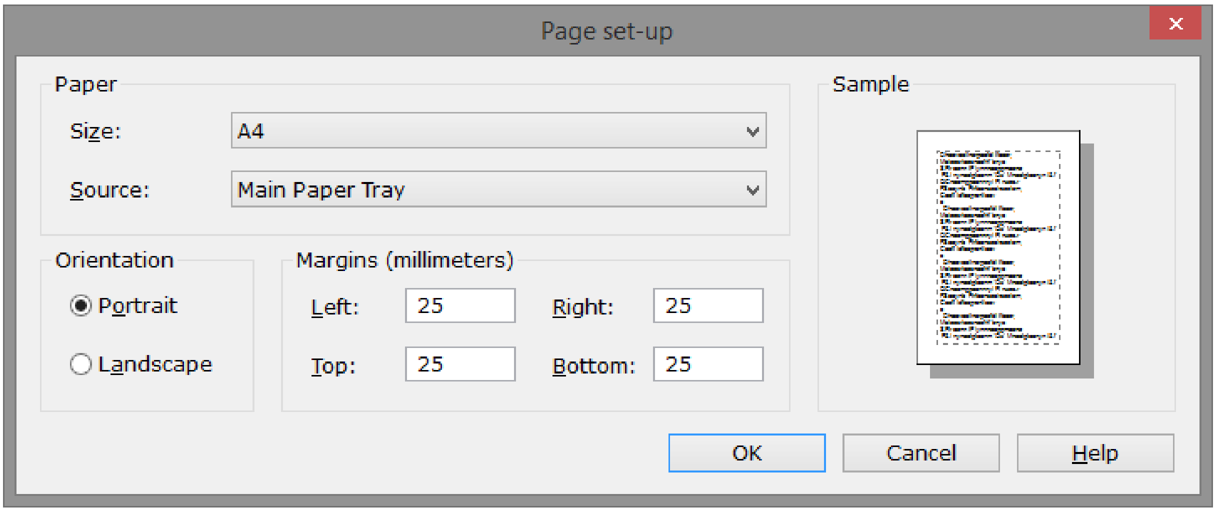


Figure 1. The Page set-up *window, where you can set the paper size, orientation and margins of documents printed by ANY-maze.*

Details

Setting the paper size	You should check the size of the paper that's actually in your printer, and then use the *Paper size* list to tell ANY-maze what it is - the most common sizes are Letter and A4, but there's a whole range of common sizes to choose from. By the way, if you choose

	the wrong size then documents may get cut off at the bottom or the sides.
Setting the paper source	A lot of modern printers have more than one paper tray. If this is the case for your printer, you can choose which tray to use by selecting it in the *Paper source* list. If this is set to 'Auto' (which may be the only option, if your printer doesn't have multiple trays) then it's usually best to leave this setting as it is.
Setting the page orientation	If you wish to, you can pick the orientation in which documents will be printed - either portrait or landscape. You should note, however, that a few documents in ANY-maze will automatically set their own orientation, irrespective of what you choose here.
Setting the page margins	You can alter the margins on all sides of the page by simply entering the desired margin sizes (in millimetres) into the respective fields. If you plan to hole-punch or bind the printed pages, then it's usually a good idea to set the left margin bigger than the right. If you include headers or footers on your documents, then ANY-maze will start the header or footer at the margin you set here, and will then use its own standard sized margins between the header/footer and the body of the document.

See also:

- Using the print window to control how a document is printed
- Setting headers and footers to print on documents
- Setting options to print pages in reverse order and/or in black and white.

Saving reports and data

Overview

ANY-maze includes a wide range of different reports, any of which can be saved to a file by selecting the *Save* button in the ribbon bar, or by right clicking on the report and selecting *Save to file...* from the menu which appears.

Reports can be saved in a number of different formats, including RTF (which is used by Microsoft Word) and HTML (which is used by web browsers). This means you can open the saved file on almost any computer, including Apple Macs.

As well as saving individual reports, ANY-maze can save the contents of the Data page spreadsheet using a format which will open directly in Microsoft Excel. You can also save individual graphs and track plots, which can be very useful if you'd like to include them in presentations or word-processed documents.

Full details about these features are included in the following topics:

- Saving documents
- Saving data
- Saving graphs
- Saving track plots and heat maps
- Saving charts
- Copying and saving data from the Data page

Saving documents

In ANY-maze, reports are just one example of documents, and the techniques described here apply to all documents - not just reports. For example, this information also applies to saving a help topic to a file.

Overview

You can save any report in ANY-maze simply by clicking the *Save* button in the ribbon bar, or by right clicking somewhere over the report and selecting *Save to file...* from the menu which appears. This will cause the *Save document* window to be displayed, where you can specify a name and location for the file and choose the format it should be saved in.

- The 'Save document' window
- Formats in which documents can be saved
- Saving just part of a report

The *Save document* window

You will probably be familiar with the *Save document* window already, as it's based on the standard 'save' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging the bottom right corner with the mouse.

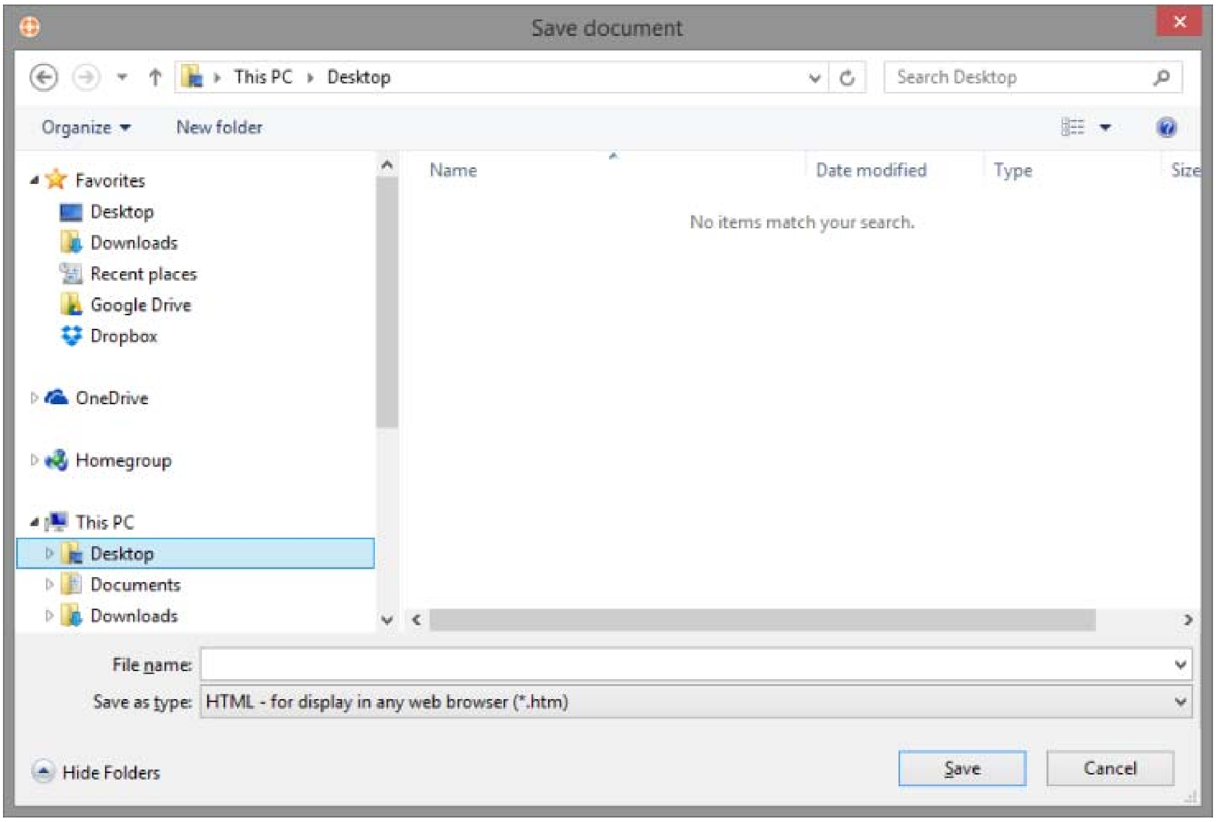


Figure 1. The Save document *window.*

Files list	The files list shows the folders and files of the selected type (see *Save as type,* below) stored in the current location. You can double click a folder to open it, or you can click a file to transfer its name to the *File name* field.
File name	Use this field to enter a name for the file. You can enter up 255 characters, but your entry can't contain any of the following characters: \ / : * ? " < > |.
Save as type	You can save documents in a number of different formats, and you should choose the one you wish to use from the list displayed here. The formats themselves are described in detail below.

Formats in which documents can be saved

HTML HTML is the format used for web pages. Saving a document in this format

	has two big advantages - firstly, all formatting and graphics will be preserved (i.e. the saved document will look exactly as it does on the screen) and secondly, you'll be able to open the resulting file on any computer which is equipped with a web browser, such as Internet Explorer or Google Chrome. However, there is one disadvantage to this format - any graphs, track plots/heat maps or other graphical images shown in the report will be saved in individual files - in other words, rather than ending up with one file, you may end up with lots. This can make it rather tedious to move or copy the resulting files.
Web archive	Web archive format (also called .mht format) has almost all the advantages of HTML format (see above) but without the disadvantage of creating lots of files - a document saved in web archive format is always saved as a single file. However, there is a downside - this format can only be read by Microsoft Internet Explorer; other web browsers don't support it. Nevertheless, if you expect to view the saved file using Internet Explorer, then this is the best format to use.
Rich Text Format	Rich Text format (also called RTF format) is supported by Microsoft Word and many other word processing programs. If you intend to *edit* a saved file, or you want to cut and paste parts of it, then this is the best format to use. As for HTML and web archive files, the report's formatting and all graphics are preserved by this format. By the way, you may know that Word can also *edit* HTML documents, so you might be tempted to save a report as HTML and then open it in Word for editing. Although this *is* possible, we don't recommend it - you'll get better results using RTF.
Text Format	Text format (also called TXT format) has the advantage of being almost universal - if you want to read a saved report using a program which doesn't support one of the previous formats, then this is the one to use. However, the big disadvantage of text format is that most formatting and all graphics are lost.

Saving just part of a report

It's easy to save just a part of a report by following these steps:

1. Move the mouse to the report's left-hand margin, where the mouse pointer will change from a left-pointing arrow to a right-pointing one - see figure 2.
2. Click and hold down the left mouse button.
3. Drag the mouse up and down to select the lines of the report which you want to save.
4. Click the right-hand mouse button and select *Save to file...* from the menu which appears. Just the area you selected will be saved.

Animal's results

You can use the Results, reports and data section of the Protocol page to select the results shown here.

Stage / trial number	1 / 1	1 / 2	1 / 3	1 / 4	2 / 1
Test duration (s)	23.4	44.4	6.0	6.8	35.7
Total distance travelled (m)	5.612	13.419	1.136	1.120	6.522
Average speed (m/s)	0.240	0.302	0.189	0.165	0.183

Figure 2. Using the mouse pointer, you can select just part of a report to save.

Saving data

Overview

You can save the spreadsheet shown on the Data page by selecting the *Save* button in the Data page's ribbon bar, or by right clicking anywhere in the spreadsheet and selecting *Save to file...* from the menu that appears.

Doing this will cause the *Save data* window to be displayed, where you can specify a name and location for the file and choose the format it should be saved in.

- The 'Save data' window
- Formats in which data can be saved
- Saving just part of the spreadsheet

The *Save data* window

You will probably be familiar with the *Save data* window already, as it's based on the standard 'save' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging the bottom right corner with the mouse.

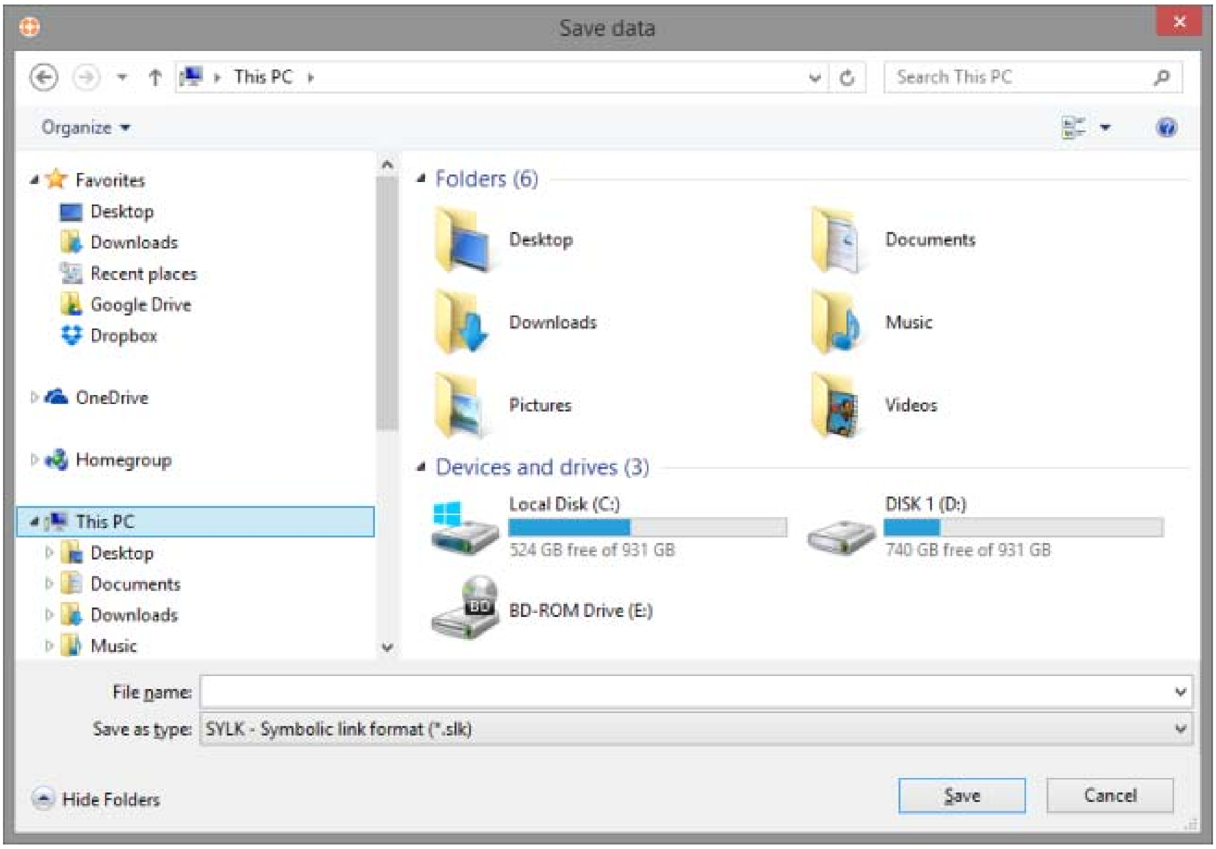


Figure 1. The Save data *window.*

Files list The files list shows the folders and files of the selected type (see *Save as type,* below) stored in the current location. You can double click a folder to open it, or you can click a file to transfer its name to the *File name* field.

File name Use this field to enter a name for the file. You can enter up 255 characters, but your entry can't contain any of the following characters: \ / : * ? " < > |.

Save as type You can save data in a number of different formats, and you should choose the one you wish to use from the list displayed here. The formats themselves are described in detail below.

Formats in which data can be saved

Symbolic link format Symbolic link format (also known as SYLK format) was designed by Microsoft for transfer of data between spreadsheet programs. As you'd expect, it's supported in many spreadsheet programs, most notably Excel.

	Saving data in SYLK format has the advantage that formatting (such as fonts and column widths) is preserved.
Comma-separated values	The Comma-separated values format (also known as CSV format) is widely used by spreadsheet and database programs. The disadvantage of this format is that formatting is lost.
dBase III/IV format	dBase format was developed for use with the once popular dBase database program. As this program was once the de facto standard database, almost all database and statistics programs support this format. This is definitely the best format to use if you want to manipulate ANY-maze data using a program like Microsoft Access (although this format is only supported in versions of Access up to Access 2010).
Tab-separated format	Tab-separated format is another common data format, and is also widely supported by spreadsheet and database programs. In this case, the file will be saved with a .txt extension so, when reading the file, you should select an option to 'Open a text file'. In fact, apart from the different file extensions, the only real difference between tab- and comma-separated formats is (as the names imply) the fact that in one there are commas between each value, whereas in the other the values are separated by tabs.

Saving just part of your data

It's easy to save just a part of the Data page spreadsheet, for example just a particular group of cells, or just a certain number of rows - the techniques for doing this are described in the Selecting cells, columns or rows topic.

More details

For more details on saving and copying data from the Data page, see Copying and saving data from the Data page.

Saving graphs

Overview

Although you can save an entire report which contains a graph (see Saving documents), there may be occasions when all you actually want to save is the graph itself. This is easily done - just right click anywhere over the graph itself and select *Save graph...* from the menu which appears. This will cause the *Save graph* window to be displayed, where you can specify a name and location for the file and choose the format it should be saved in.

- The 'Save graph' window
- Formats in which graphs can be saved

The *Save graph* window

You will probably be familiar with the *Save graph* window already, as it's based on the standard 'save' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging the bottom right corner with the mouse.

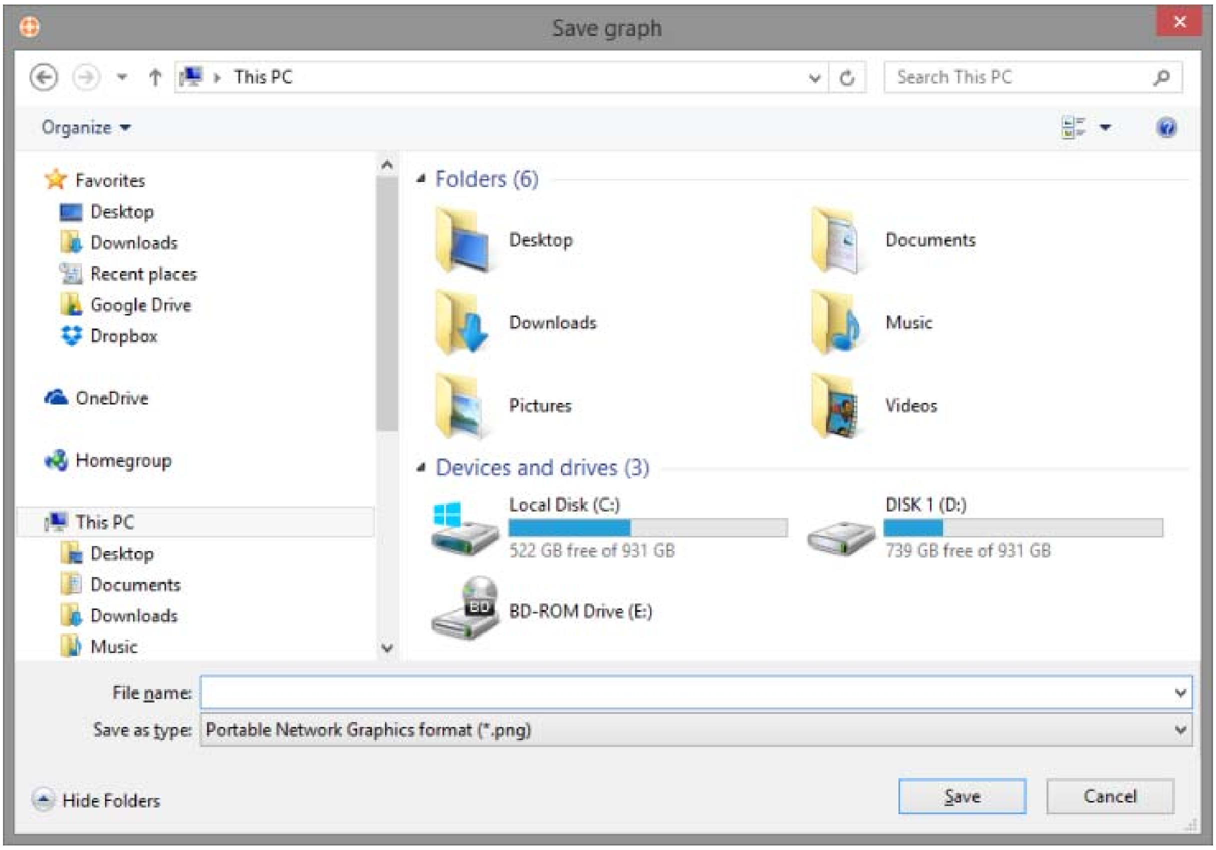


Figure 1. The Save graph *window.*

Files list	The files list shows the folders and files of the selected type (see *Save as type,* below) stored in the current location. You can double click a folder to open it, or you can click a file to transfer its name to the *File name* field.
File name	Use this field to enter a name for the file. You can enter up 255 characters, but your entry can't contain any of the following characters: \ / : * ? " < > |.
Save as type	You can save graphs in a number of different formats, and you should choose the one you wish to use from the list displayed here. The formats themselves are described in detail below.

Formats in which graphs can be saved

Windows bitmap format	Windows bitmap format (also known as BMP format) is supported by just about every Windows program, but it has two major disadvantages - firstly, scaling a bitmap can't be done without losing quality (because it's a raster graphics format), and secondly, bitmap files tend to be rather large.
Graphics interchange format	Graphics interchange format (also known as GIF format) is also widely supported and is one of the graphics formats used on web pages. GIF images saved by ANY-maze are compressed, so they don't take up too much space, although being a raster graphics format, they can't be scaled without losing image quality.
JPEG format	JPEG format is another raster graphics format used for web pages, and is also widely supported in many programs (Word and PowerPoint, for example). JPEG has the advantage that files are compressed, but the technique used is really designed to work well with photographs - graphics images (such as graphs) tend to look a little blurred.
Tagged Image File format	Tagged Image File format (also known as TIFF or TIF format) is widely used in the publishing and graphic design industries, and was designed to allow images to be shared across different operating systems (e.g. Windows and Macs). Again, it's a raster graphics format, which means it can't be easily scaled without losing image quality.
Portable Network Graphics format	Portable Network Graphics format (or PNG format) is a vector graphics format that supports lossless compression. It is the most-used lossless image compression format on the internet, and PNG images can be scaled without losing quality - so it is probably the best option to use when saving graphs.
Enhanced metafile format	Enhanced metafile format (also known as EMF format) is another vector graphics format, which means that EMF images can be scaled without losing quality. The EMF format was designed by Microsoft, but is not supported by as many programs as the other formats described here. However, if you want to save a graph and later include it in a Word document, this might be a good format to choose.

Saving track plots and heat maps

Overview

Although you can save an entire report which contains track plots or heat maps (see Saving documents), there may be occasions when all you actually want to save is one of the track plots or heat maps themselves. This is easily done - just right click over the relevant track plot (or heat map) and select *Save plot...* from the menu which appears. This will cause the *Save plot* window to be displayed, where you can specify a name and location for the file and choose the format it should be saved in.

- The 'Save plot' window
- Formats in which track plots and heat maps can be saved

The *Save plot* window

You will probably be familiar with the *Save plot* window already, as it's based on the standard 'save' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging the bottom right corner with the mouse.

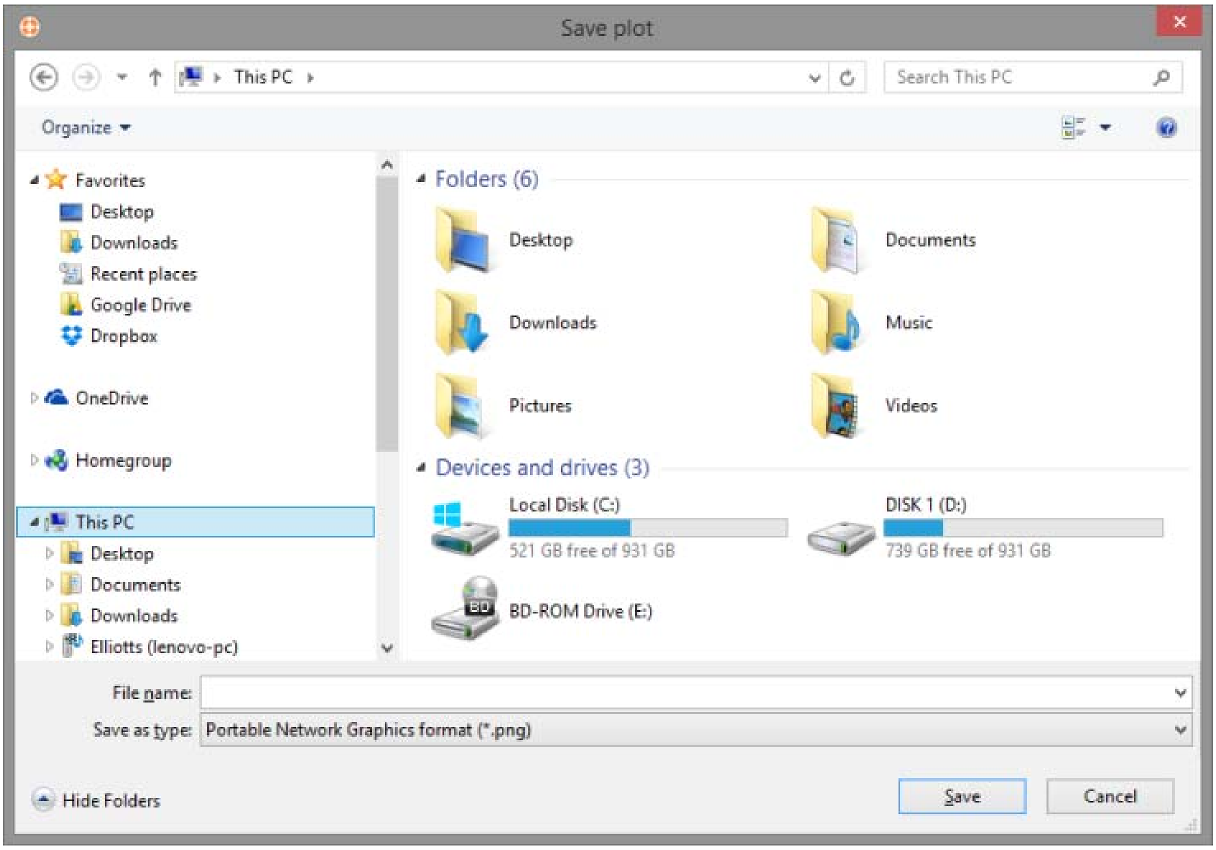


Figure 1. The Save plot *window.*

Files list	The files list shows the folders and files of the selected type (see *Save as type,* below) stored in the current location. You can double click a folder to open it, or you can click a file to transfer its name to the *File name* field.
File name	Use this field to enter a name for the file. You can enter up 255 characters, but your entry can't contain any of the following characters: \ / : * ? " < > |.
Save as type	You can save track plots and heat maps in a number of different formats, and you should choose the one you wish to use from the list displayed here. The formats themselves are described in detail below.

Formats in which track plots and heat maps can be saved

Windows bitmap format	Windows bitmap format (also known as BMP format) is supported by just about every Windows program, but it has two major disadvantages - firstly, scaling a bitmap can't be done without losing quality (because it's a raster graphics format), and secondly, bitmap files tend to be rather large.
Graphics interchange format	Graphics interchange format (also known as GIF format) is also widely supported and is one of the graphics formats used on web pages. GIF images saved by ANY-maze are compressed, so they don't take up too much space, although being a raster graphics format, they can't be scaled without losing image quality.
JPEG format	JPEG format is another raster graphics format used for web pages, and is also widely supported in many programs (Word and PowerPoint, for example). JPEG has the advantage that files are compressed, but the technique used is really designed to work well with photographs - graphics images (such as track plots) tend to look a little blurred, although it should be OK for heat maps.
Tagged Image File format	Tagged Image File format (also known as TIFF or TIF format) is widely used in the publishing and graphic design industries, and was designed to allow images to be shared across different operating systems (e.g. Windows and Macs). Again, it's a raster graphics format, which means it can't be easily scaled without losing image quality.
Portable Network Graphics format	Portable Network Graphics format (or PNG format) is a vector graphics format that supports lossless compression. It is the most-used lossless image compression format on the internet, and PNG images can be scaled without losing quality - so it is probably the best option to use when saving track plots and heat maps.
Enhanced metafile format	Enhanced metafile format (also known as EMF format) is another vector graphics format, which means that EMF images can be scaled without losing quality. The EMF format was designed by Microsoft, but is not supported by as many programs as the other formats described here. However, if you want to save a track plot and later include it in a Word document, this might be a good format to choose (it shouldn't really be used for heat maps though, as there are better options available).

Saving charts

Overview

Although you can save an entire report which contains a chart (see Saving documents), there may be occasions when all you actually want to save is an individual chart. This is easily done - just right click over the relevant chart and either select *Save this chart to a file* or *Save all charts to a file* from the menu which appears. The former option will save just the chart you actually clicked on, whereas the latter will save all the charts in the set. Selecting either option will cause the *Save chart* window to be displayed, where you can specify a name and location for the file and choose the format it should be saved in.

- The 'Save chart' window
- Formats in which charts can be saved

The *Save chart* window

You will probably be familiar with the *Save chart* window already, as it's based on the standard 'save' window used in almost all Windows software.

It's often very useful to enlarge this window so you can see more files. You can do this by dragging the bottom right corner with the mouse.

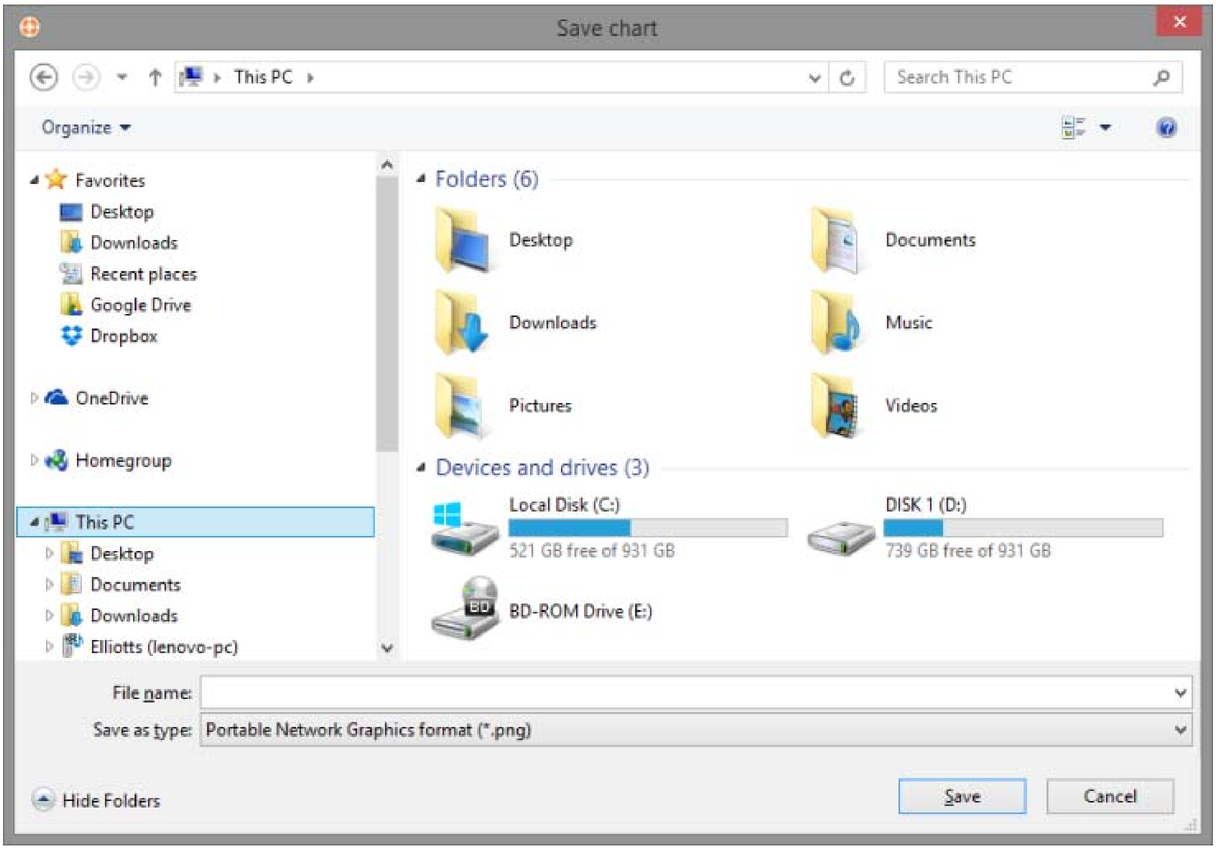


Figure 1. The Save chart *window.*

Files list	The files list shows the folders and files of the selected type (see *Save as type,* below) stored in the current location. You can double click a folder to open it, or you can click a file to transfer its name to the *File name* field.
File name	Use this field to enter a name for the file. You can enter up 255 characters, but your entry can't contain any of the following characters: \ / : * ? " < > |.
Save as type	You can save charts in a number of different formats, and you should choose the one you wish to use from the list displayed here. The formats themselves are described in detail below.

Formats in which charts can be saved

Windows bitmap format	Windows bitmap format (also known as BMP format) is supported by just about every Windows program, but it has two major disadvantages - firstly, scaling a bitmap can't be done without losing quality (because it's a raster graphics format, and secondly, bitmap files tend to be rather large.
Graphics interchange format	Graphics interchange format (also known as GIF format) is also widely supported and is one of the graphics formats used on web pages. GIF images saved by ANY-maze are compressed, so they don't take up too much space, although being a raster graphics format, they can't be scaled without losing image quality.
JPEG format	JPEG format is another raster graphics format used for web pages, and is also widely supported in many programs (Word and PowerPoint, for example). JPEG has the advantage that files are compressed, but the technique used is really designed to work well with photographs - graphics images (such as charts) tend to look a little blurred.
Tagged Image File format	Tagged Image File format (also known as TIFF or TIF format) is widely used in the publishing and graphic design industries, and was designed to allow images to be shared across different operating systems (e.g. Windows and Macs). Again, it's a raster graphics format, which means it can't be easily scaled without losing image quality.
Portable Network Graphics format	Portable Network Graphics format (or PNG format) is a vector graphics format that supports lossless compression. It is the most-used lossless image compression format on the internet, and PNG images can be scaled without losing quality - so it is probably the best option to use when saving charts.
Enhanced metafile format	Enhanced metafile format (also known as EMF format) is another vector graphics format, which means that EMF images can be scaled without losing quality. The EMF format was designed by Microsoft, but is not supported by as many programs as the other formats described here. However, if you want to save a chart and later include it in a Word document, this might be a good format to choose.

Copying reports and data

Overview

You can copy all reports, graphs, spreadsheets, etc. to the Windows clipboard and paste them directly into other programs such as Microsoft Word, Excel or PowerPoint - all formatting will be retained.

Details

To copy an entire report

Click the *Copy* button in the ribbon bar, or right click on the report and select *Copy* from the menu which appears.

To copy just a part of a report

First, select the part of the report you want to copy by following these steps:

1. Move the mouse to the report's left-hand margin where the mouse pointer will change from a left-pointing arrow to a right-pointing one.
2. Click and hold down the left mouse button.
3. Drag the mouse up and down to select the lines of the report you want to save.

Then, simply click the *Copy* button (or right click on the report and select *Copy* from the menu which appears). Just the lines you've selected will be copied to the clipboard.

To copy a graph

Right click on the graph and select *Copy graph* from the menu which appears.

To copy a track plot or heat map

Right click on the track plot or heat map and select *Copy plot* from the menu which appears.

To copy the Data page spreadsheet

Click the *Copy* button in the ribbon bar of the Data page, or right click anywhere on the spreadsheet and select *Copy* from the menu which appears.

To copy part of the spreadsheet	First select the cells, rows or columns you want to copy (see Selecting cells, columns or rows) and then click the *Copy* button in the ribbon bar, or right click anywhere on the spreadsheet and select *Copy* from the menu which appears.

Sending reports and data by e-mail

The ability to send a report, document or spreadsheet by e-mail will only be available if your computer is connected to the internet.

Overview

You can send reports, spreadsheets, etc. directly from ANY-maze by simply clicking the *Send as e-mail* button shown in the ribbon bar. Alternatively, you can right click anywhere on the report and select *Send as e-mail...* from the menu which appears. The *Send this document by e-mail* window will open, where you can address your e-mail, write a message and choose the format in which the document will be sent.

- The 'Send document' window
- Formats in which documents can be sent

The *Send this document by e-mail* window

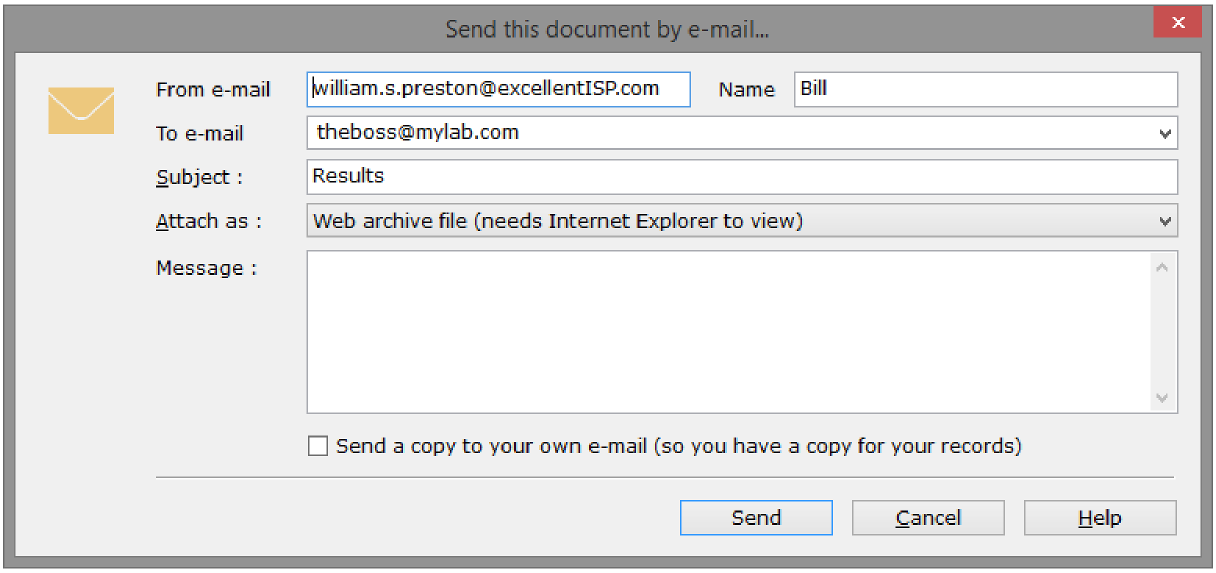


Figure 1. The Send this document by e-mail *window*

From e-mail	Enter the e-mail address that you want to send the e-mail from. If the recipient replies to the e-mail, the reply will be sent to this e-mail address.
Name	Enter your name here. This name will be combined with the *From e-mail* to show the recipient who the e-mail is from.
To e-mail	Enter the e-mail address to send the document to (for example, techsupport@anymaze.com). The e-mail addresses that you send documents to will be stored, and the next time you open this window, will be available for selection by clicking the drop-down arrow next to this field.
Subject	Optionally enter the subject of the e-mail. ANY-maze may fill this in with the title of the document you're sending, but of course you can change this if you wish.
Attach as	The document you're sending will be attached to your message as a file. Use this list to select the format for the file. The options available are described in detail below.

Message — Optionally enter a message for the e-mail. If you don't enter anything, then the e-mail will appear to be blank, although the document will still be attached correctly.

Send — Creates the message, attaches the file(s) from the document, and sends both the message and the attached file to the recipient.

Note that files are sent in the background, so you can carry on working while this happens. The status bar at the bottom of the ANY-maze window will show the current progress of the files being sent:

Figure 2. The ANY-maze status bar, showing the progress of files being sent.

At any time, you can click on this section of the status bar to show a small pop-up window containing a list of the current files being sent, with their progress. You can cancel the sending of an individual message using the button next to an e-mail in this popped-up list.

You can carry on doing anything else in ANY-maze while this message is being sent - you can even start to send another report or document using the *Send as e-mail* button! Note however that if you try to exit from the ANY-maze software while files are still being sent, you'll be shown a message warning you of this and asking if you really want to exit and cancel the sending of the files.

Cancel — Closes this window, without sending the document.

Help — Opens this help topic.

Formats in which documents can be sent

When you send a document (or data) from within ANY-maze, the document is actually attached to your message as a file. ANY-maze supports the following formats for such attached files:

Formats for text documents:

HTML — HTML is the format used for web pages. Sending a document in this format has two big advantages - all formatting and graphics will be preserved (i.e. the attached document will look exactly as it does on the screen), and the recipient will be able to open the resulting file on any computer which is equipped with a web browser, such as Internet Explorer or Google Chrome (including an Apple Mac).

However, there is one disadvantage to this format - any graphs, track plots or other graphical images shown in the report will be saved in individual files; in other words, rather than attaching just one file, you may end up attaching lots. Although all files will be zipped into a single file, when they are unzipped by the recipient, this can be rather confusing and can make it hard for them to move or copy the received document.

Web archive	Web archive format (.mht format) has almost all the advantages of HTML format (see above) without the disadvantage of creating lots of files - a document attached in web archive format is always placed in a single file. However, there is a downside - this format can only be read by Microsoft Internet Explorer; other web browsers don't support it. Nevertheless, if you know the recipient has Internet Explorer, then this is the best format to use.
Rich Text Format	Rich Text format (also called RTF format) is supported by Microsoft Word and many other word processing programs. If you expect that the recipient will want to edit the document, then this is the best format to use. As for web archive format, documents will be attached as just a single file. By the way, you may be tempted to send documents which the recipient is only expected to *view* in RTF format, or to send documents which will be *edited* in HTML format. While both ideas would work, we don't recommend them - HTML produces a more accurate rendering of the original document and so is better for viewing, whereas RTF produces a version which is far easier to edit.
Text Format	Text format (TXT format) has the advantage of being almost universal - the recipient should be able to read the document no matter what computer or software they're using. However, the BIG disadvantage of text format is that most formatting and all graphics are lost.

Formats for data (i.e. spreadsheets):

Symbolic link format	Symbolic link format (also known as SYLK format) was designed by Microsoft for transfer of data between spreadsheet programs. As you'd expect, it's supported in many spreadsheet programs, most notably Excel. Saving data in SYLK format has the advantage that formatting (such as fonts and column widths) is preserved.
Comma-separated values	The Comma-separated values format (also known as CSV format) is widely used by spreadsheet and database programs.
dBase III/IV format	dBase format was developed for use with the once popular dBase database program. As this program was once the de facto standard database, almost all database and statistics programs support this format. This is definitely the best format to use if you expect that the recipient will want to manipulate ANY-maze data using a program like Microsoft Access.

	However, bear in mind that versions of Access from 2013 onwards no longer support this file type, so for newer versions of Access, you'll need to transfer by saving to a comma-separated (CSV) or tab-separated (text) file.
Tab-separated format	Tab-separated format is another common data format, and is also widely supported by spreadsheet and database programs. In this case data is saved as a text file with 'tab' characters between each item.

Note that whichever format you use to send the document, it will be compressed by being zipped into a single file.

ANY-maze plug-ins

Plug-ins are designed to allow third-party authors to write additional software that 'plugs-in' to ANY-maze and thus extends its abilities.

This section of the ANY-maze reference provides detailed information about ANY-maze plug-ins, including full technical documentation which plug-in authors will require to write a plug-in.

Plug-ins are only available in licensed copies of ANY-maze.

Writing a plug-in

- Introduction to writing a plug-in
- Step-by-step instructions for writing a plug-in
- Writing an Action plug-in

The demo plug-in

- Introduction to the demo plug-in
- Install the ANY-maze demo plug-in
- The demo plug-in code listing

Plug-in reference

- Plug-in API reference
- Plug-in measures

Introduction to writing a plug-in

Introduction

An ANY-maze plug-in is a small Windows DLL which extends the abilities of ANY-maze. For an introduction to plug-ins, refer to Extending ANY-maze using plug-ins.

This topic will describe the basic structure of plug-ins, and how they work at the programming level. If you're thinking of writing a plug-in, this topic is highly recommended reading.

What you need to write a plug-in

To write a plug-in, you need a software development system capable of creating Windows DLLs. The most commonly used development system is Microsoft's Visual Studio, and any version can be used to author an ANY-maze plug-in.

Of course the other requirement is some knowledge of a programming language, and some experience writing software for the Windows environment. However, ANY-maze plug-ins are not complicated and you don't need to be a highly experienced programmer to write one. Furthermore, ANY-maze technical support will be more than happy to help you in any way and will even write a plug-in for you (for free) provided you can supply full details of what you want the plug-in to do. Contact us for more details.

What makes a DLL into a plug-in?

As mentioned above, a plug-in is simply a Windows DLL, but not all DLLs are plug-ins - so what makes a DLL into an ANY-maze plug-in? The answer is that for a DLL to be recognised by ANY-maze as a plug-in, it simply needs to export a routine called `ANYmazeExtension_Main.`

The `ANYmazeExtension_Main` routine

All communication between ANY-maze and the plug-in is through the `ANYmazeExtension_Main` routine, which is passed *messages* by ANY-maze to which it responds. In this respect, the `ANYmazeExtension_Main` routine is very like a Window's `WNDPROC` routine, which handles messages addressed to a window.

Implementing multiple plug-ins in one DLL

A plug-in DLL will always contain just one `ANYmazeExtension_Main` routine but, if desired, the DLL can implement multiple plug-ins. Note that from the user's point of view the plug-ins will appear to be separate entities, and the fact that they happen to all be implemented in the same DLL will be invisible.

Choosing whether to build a 64- or 32-bit DLL

The plug-in DLL must be built for the same 64-bit/32-bit configuration as the version of ANY-maze you plan to use it with. For example, if you are running 64-bit ANY-maze then you must build a 64-bit plug-in. This is because 64-bit applications can only use 64-bit DLLs and 32-bit applications can only use 32-bit DLLs. Of course, if you want your plug-in to be compatible with both 64- and 32-bit versions of ANY-maze then you can simply build two versions of it.

Enumerating the available plug-ins

When ANY-maze starts, it searches for DLLs in the folder where the `ANY-maze.exe` file is located, which by default is `C:\Program Files (x86)\ANY-maze` on a 64-bit operating system, or `C:\Program Files\ANY-maze` on a 32-bit operating system. For each DLL that it finds, it checks whether the DLL exports a routine called `ANYmazeExtension_Main.` If it does, then ANY-maze sends the routine an `AME_ENUMERATE` message. The DLL responds by returning the *ID* of the first plug-in it implements. ANY-maze will then send the DLL another `AME_ENUMERATE` message, and the DLL responds with the ID of the next plug-in it implements. This continues until the DLL has returned the IDs of all of its plug-ins (which will often be just one).

Through this mechanism, ANY-maze discovers which plug-ins are available and what their IDs are. From this, it builds an internal list of available plug-ins. Note that this process is performed whenever ANY-maze starts, which means that to install a new plug-in you simply have to copy the DLL to the `ANY-maze.exe` folder. Also note that this means 'hot' installation of a plug-in is not supported - i.e. if you copy a plug-in DLL into the `ANY-maze.exe` folder while ANY-maze is running, the plug-in won't be recognised until the next time you start ANY-maze.

Plug-in IDs

Plug-in IDs are unique IDs which are used to identify a single plug-in (remember that one DLL may implement multiple plug-ins). Of course different authors will write different plug-ins, and you can't assume that an ID you choose will not be chosen by another author. To address this, ANY-maze plug-in IDs are in fact *Windows Globally Unique Identifiers,* or GUIDs. GUIDs, which can be created using the free tool GUIDGEN.EXE, are guaranteed to be unique.

GUIDGEN.EXE is shipped with the Microsoft Visual Studio system, but can also be downloaded, for free, from the Microsoft web-site.

Plug-in names

Although *internally* ANY-maze will refer to a plug-in by its ID, users will refer to it by a name. Thus a plug-in needs to include some type of descriptive name (usually based on what the plug-in will actually do), and during plug-in enumeration ANY-maze will send an `AME_GETNAME` message to the plug-in to retrieve its name.

Names will not necessarily be unique, but it's quite unlikely that two authors will choose exactly the same name for their plug-ins. If they do, this won't cause any technical problems, although it may confuse the user - so authors should try to choose descriptive names rather than just calling a plug-in something like 'My plug-in'. Names can be up to 48 characters long.

Activating a plug-in

Following enumeration, ANY-maze will know which plug-ins are available and will list them in the *ANY-maze plug-ins* item of the *Analysis* element of the Protocol. Here the user will be able to choose which of the available plug-ins he actually wishes to use in his protocol - see figure 1. By choosing a plug-in, the user is *activating* it. Thus the user can utilise different plug-ins in different protocols, depending on the protocol's needs. When the user activates/deactivates a plug-in, ANY-maze sends the plug-in an `AME_ENABLE` message.

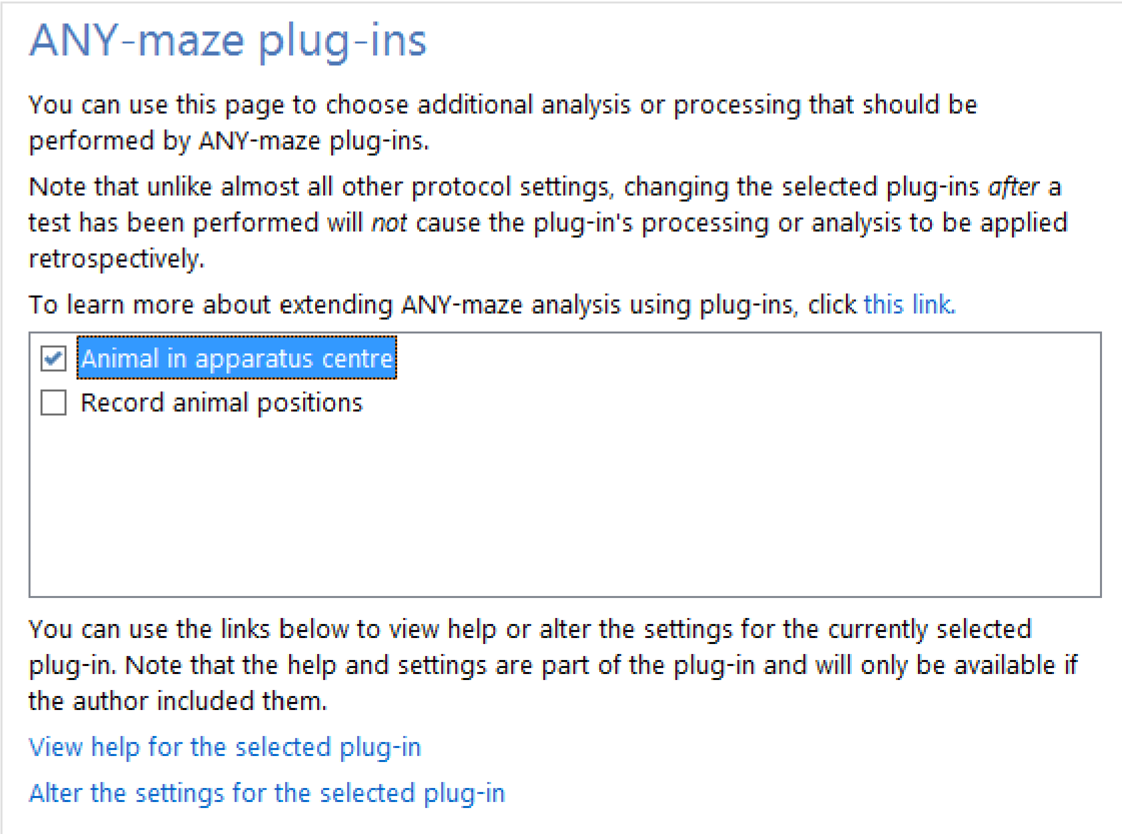


Figure 1. The plug-ins detected during enumeration are listed in the protocol, so the user can choose which he wishes to use.

Plug-ins are only available in licensed copies of ANY-maze.

How a plug-in gets used during a test

When ANY-maze starts a test, it sends an AME_TESTSTART message to all the *active* plug-ins, i.e. those which the user activated in the protocol. This gives each plug-in an opportunity to initialise itself ready for the test.

During the test, ANY-maze sends AME_POSN messages to each activated plug-in, for each detected position of the animal. The plug-in can process this data in any way it wishes, although the processing should not take too long (ideally just a few milliseconds) otherwise it may cause ANY-maze to drop frames from the camera - typically a new frame arrives every 33ms. If the plug-in wishes to, it can return a *result* from this message.

At the end of the test, ANY-maze sends either an AME_TESTEND message or an AME_TESTABORT message to each activated plug-in, to tell them that the test has finished.

Plug-in results

The processing that a plug-in performs when it receives an AME_POSN message will, of course, depend on what the plug-in is actually designed to do - but in many cases, the plug-in will want to return some type of result to ANY-maze. For example, a plug-in that's designed to interface to a piece of equipment that monitors the animal's heart rate will want to return the current heart rate on each AME_POSN message; or a plug-in designed to determine when the animal is feeding will want to return a result to indicate when a bout of feeding starts, and when it ends.

Results are returned in the AME_POSNDATA record, which is one of the parameters of the AME_POSN message, and a plug-in can choose to return one of four different result types:

RT_NONE — An RT_NONE result simply means that the plug-in doesn't return any result at all. This might be the case for a plug-in that simply passes the animal's position data to some other system.

RT_ONOFF — RT_ONOFF results are used when a plug-in detects when something occurs. In this case the result's *data* is 0 (to indicate OFF) or 1 (to indicate ON). ANY-maze will analyse RT_ONOFF results and provide measures such as *Number of activations; Time active; Longest activations;* etc. (See Plug-in measures for a full list.)

RT_VALUE — An RT_VALUE result has a specific value at a certain moment in time - for example, the animal's heart rate would be an RT_VALUE result. ANY-maze will analyse these results and provide measures for the *average value,* the *maximum* and the *minimum*. It's important to remember that ANY-maze will also analyse results for each zone independently, and for test segments. Thus a plug-in that returns the animal's heart rate will actually be making results available for 'Heart rate in the ABC zone', 'Heart rate in the XYZ zone' and 'Heart rate in period 0-30 seconds' (for example). To benefit from this analysis, the plug-in author doesn't need to do anything - it just happens.

RT_CUMULATIVE — RT_CUMULATIVE results are discrete results that are summed during the test. For example, a plug-in which determines how far the animal travels would return an RT_CUMULATIVE result with data of how far the animal travelled from the previous position to the current one. ANY-maze will sum the results and thus report the total distance the animal travelled. Again, it's worth remembering that ANY-maze will also automatically calculate separate results for each zone and for each test segment.

Plug-in settings

A plug-in can optionally include a 'Settings window', which will allow the user to specify options or settings that the plug-in should use during its analysis. For example, the demo plug-in determines when the animal is in the centre of the apparatus and returns an RT_ONOFF result. But what should be the definition of 'In the centre of the apparatus'?

To set this, the plug-in provides a settings window (see figure 2) where the user can specify how far from the centre the animal can be while still being considered to be 'in the centre' - by default the plug-in uses 10cm, but the user can alter it to any value from 1cm to 999cm.

Obviously what actual settings a plug-in uses won't be known to ANY-maze, but it does *store* the settings as part of the protocol and it passes them to the plug-in as part of the AME_TESTSTART message. To do this, ANY-maze simply treats the settings as an opaque buffer of a fixed size.

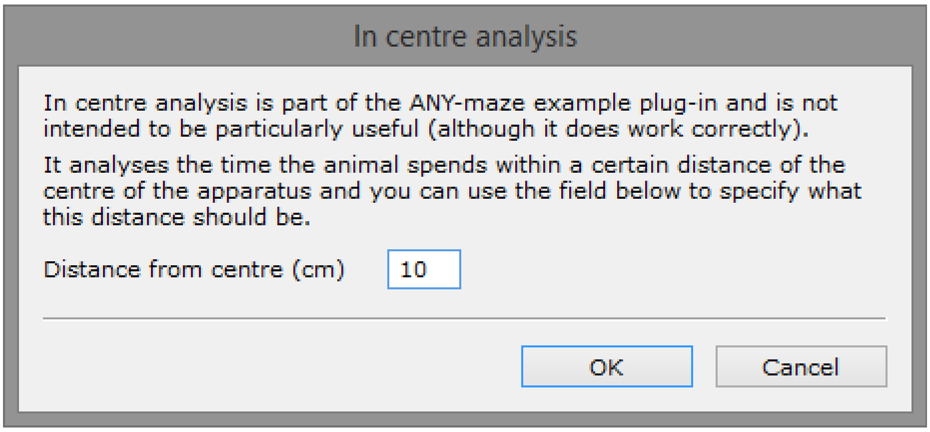


Figure 2. The settings window of the demo plug-in.

Editing settings

When a plug-in is first activated in a protocol, the settings buffer is initialised to all zeros. If the user chooses to edit the settings, ANY-maze will load the current settings from the protocol and pass a pointer to them in an `AME_EDITSETTINGS` message. The plug-in can detect whether the settings are all zero, and if so can complete the settings with default values. It can then use a window to allow the user to edit the settings, and finally it can use the return from the `AME_EDITSETTINGS` message to indicate to ANY-maze whether the settings buffer should now be saved or not.

Action plug-ins

Action plug-ins are a little different to normal plug-ins. Rather than performing analysis of the test data, they can 'do something' during a test.

Action plug-ins are invoked during a test by an ANY-maze procedure firing an action (or, if the protocol is using the older 'Events and Actions' system, by an action directly triggered by an event). An event can detect a wide range of different things that occur during a test - for example, when the animal enters a certain zone. When this occurs, the procedure or event can be set to trigger an action, and the action can invoke an 'Action' plug-in. The plug-in could do virtually anything - perhaps it would send a message to another program, or perhaps it would switch on some piece of equipment.

To make a plug-in into an 'Action plug-in', you simply need to return `RT_ACTION` as the plug-in's result type when sent the `AME_GETRESULTTYPE` message. For full details, see Writing an Action plug-in.

Plug-in Help

Unless a plug-in is only intended to be used by the author, it should include some help to explain to users what it does, how to interpret the results and how to alter the settings (if it includes any).

Plug-in help is displayed within the ANY-maze help page, but the text of the help is provided by the plug-in. For the author, help is very easy to implement - when the user requests help, ANY-maze sends a message to plug-in which returns three strings:

- The help topic title
- Some 'introductory' text
- Some 'details' text

Within the text, the author can use some basic mark-up to alter the text format (using italic, bold, etc.) and to structure the text (using paragraphs and bullet points). Note that the mark-up is NOT HTML - it's the mark-up used by the ANY-maze help page, which is broadly based on RTF (Rich Text Format). For more details, see the help topics for the `AME_GETHELPTITLE`, `AME_GETHELPINTRO` and `AME_GETHELPDETAILS` messages.

It's not as complicated as it sounds - see the demo plug-in

Having read all the above, you may be thinking that writing a plug-in is a complicated business, but actually it's really quite simple - as can be seen in the demo plug-in DLL.

The demo DLL includes two plug-ins, both of which perform useful (albeit simple) functions; furthermore, one of them provides a settings window and both provide help. Excluding comments and blank lines, the total code is just 300 lines, and much of this can be reused as 'boiler-plate' code in your own plug-in projects.

Incidentally, an absolutely minimal plug-in could probably be implemented in about 40 lines of code.

What next?

Having read this topic, you should now have a basic understanding of how a plug-in DLL works. As a next step to writing your own plug-in, I suggest you first install the demo plug-in DLL and review the code. Then, use the Step-by-step instructions to write your own plug-in.

And don't forget, ANY-maze technical support are just an e-mail away if you need any help.

See also:

- Step-by-step instructions for writing a plug-in
- Writing an Action plug-in

- The ANY-maze demo plug-in
- Plug-in API reference

Step-by-step instructions for writing an ANY-maze plug-in

These instructions assume you will be writing a plug-in using C/C++.

Introduction

This topic takes you through the steps required to create an ANY-maze plug-in. To do this, a minimal example plug-in is developed based on the demo plug-in provided with ANY-maze. However, it should be easy to see where you will need to substitute your own code to implement a truly useful plug-in.

1. Create a new DLL project
2. Make sure the project settings are correct
3. Add a new source file to the project
4. Include the Windows.h and PlugIn.h header files
5. Write the DllMain function
6. Write the basic ANYmazeExtension_Main function
7. Add processing of the AME_INITIALISE and AME_TERMINATE messages
8. Create a GUID for your plug-in
9. Add processing of the AME_ENUMERATE message
10. Add processing of the AME_GETNAME message
11. Add processing of the AME_GETRESULTTYPE message
12. Optionally add processing of the AME_GETHELPxxx messages
13. Optionally add processing of the AME_EDITSETTINGS message
14. Add processing of the AME_TESTSTART message
15. Add processing of the AME_POSN message
16. Add processing of the AME_TESTEND and AME_TESTABORT messages
17. You're done!

1. Create a new DLL project

The first step is to create a new DLL project. Most development environments include a 'New project' command, which can be used to create projects of various types. You should create a **Win32 DLL** project. I suggest you avoid any options which will include 'standard' code, as we'll add all the code necessary as we progress through these instructions.

By the way, you can call your project (and thus your DLL) anything you like; the name is not important to ANY-maze.

2. Make sure the project settings are correct

You should use your development environment to ensure that the project settings are correct for the plug-in DLL to run correctly with ANY-maze.

Firstly, you must ensure that the project is being built as 32-bit or 64-bit, depending on the current version of ANY-maze that is running. Usually, if you're using a 64-bit version of the Windows operating system, then ANY-maze will also be the 64-bit version. If you're not sure which version of ANY-maze you're using, you can find out from the Help page under *About ANY-maze -> Program information -> Application version.*

Secondly, you should ensure that your project will be compiled/linked with multi-threaded runtime libraries.

This is important because ANY-maze uses a different thread for each piece of apparatus on which a test is being performed, and it's these threads that will call the plug-in code. Thus, in a protocol with two apparatus, your plug-in will be called from both threads.

You should keep in mind that your code must be thread-safe. In fact this isn't too complex, as ANY-maze will help you with this (as we'll see later), but it does mean that you should avoid using static variables.

3. Add a new source file to the project

With your project created and set to be multi-threaded, you're ready to actually write some code. But first, you will need to add a source file to your project. You can call the file anything you like.

In these instructions we'll use a single source file for the entire plug-in, but of course you could break it into different files if you prefer.

4. Include the 'Windows.h' and 'PlugIn.h' header files

The next step is to include the `Windows.h` and `PlugIn.h` header files. The former defines Windows functions, structures and constants, while the latter defines the structures and constants used to interface a plug-in to ANY-maze.

Of course you may want to include other headers too - for example, if you plan to use some math functions, you'll probably want to include `math.h.`

The path to `windows.h` will almost certainly be defined already within your development environment, whereas the path of `PlugIn.h` won't be. An easy solution is to copy `PlugIn.h` from the demo plug-in folder into your project folder, or you can use a fully qualified path name to the demo plug-in folder, whichever you prefer.

```
#include "windows.h"
#include "PlugIn.h"
```

35. Write the 'DllMain' function

All DLLs must include a `DllMain` function, which Windows will call when the DLL is loaded.

In a plug-in, the `DllMain` has no particular tasks to perform, so a minimal implementation is fine.

```
BOOL APIENTRY DllMain( HMODULE hModule,
                       DWORD   ul_reason_for_call,
                       LPVOID  lpReserved)
{
//BEGIN
  switch (ul_reason_for_call)
  {
    case DLL_PROCESS_ATTACH : //Note the module handle
                              hInstance = hModule;
                              break;
    case DLL_THREAD_ATTACH  :
    case DLL_THREAD_DETACH  :
    case DLL_PROCESS_DETACH : //Nothing to do
                              break;
  }

  //Done
  return TRUE;
}
```

Here the routine does just one thing - it notes the module handle, as this is required to load resources.

Of course, you may want your plug-in to perform additional processing, but do bear in mind that the DLL must be thread-safe and that there are special rules about what can and cannot be done during DLL_PROCESS_ATTACH processing.

6. Write the basic 'ANYmazeExtension_Main' function

The `ANYmazeExtension_Main` function is the only routine you *have* to include in your DLL to make it into an ANY-maze plug-in. This routine acts as the link between ANY-maze and the plug-in, and all communication passes through it.

For now, we're going to add a skeleton implementation of `ANYmazeExtension_Main` which we'll then add to in the following steps.

```
extern "C"
__declspec(dllexport) int ANYmazeExtension_Main (LPGUID guid,
                                                 DWORD  Mesg,
                                                 WPARAM wParam,
                                                 LPARAM lParam)
{
//BEGIN
  //Action depends on the Mesg
  switch (Mesg)
  {
    .... we'll add some case statements in a minute ....

    default :
      //What's this message?
      return AME_NOTIMPLEMENTED;
  }
}
```

There are some important things to note about this routine:

- The function is declared as `__declspec(dllexport)`. This causes the routine to be exported from the DLL - which is required. The exact method of exporting a routine will depend on your development environment; the method used here is Microsoft-specific. Refer to your development environment help files for guidance.
- The `extern "C"` line is included to tell the linker not to apply *name decoration* to the function name. Without this, the function will be exported with a name that has seemingly random numbers and symbols added to it, and it won't be recognised by ANY-maze. This method of avoiding name decoration is something else that's Microsoft-specific - again, refer to your development environment help files for guidance.

As is probably apparent, this routine processes messages, which are passed to it from ANY-maze. Note that these are NOT Windows messages, although the routine does look very like a `WNDPROC` - this similarity is intentional, as it should mean anyone with even a passing familiarity with Windows programming can quickly understand how this routine should work.

The salient point is that ANY-maze will call this routine passing it a *message* (usually with some parameters in wParam and lParam); the routine will process the message and return a code indicating the outcome of the processing. ALL communication between ANY-maze and the plug-in is performed this way.

The GUID parameter is used to identify *which* plug-in a message is being addressed to - remember that a single DLL can implement multiple plug-ins if it wants to. If your DLL implements just a single plug-in, then you can often ignore the GUID - although not always, as we'll see in the next two steps!

7. Add processing of the AME_INITIALISE and AME_TERMINATE messages

The first messages to process are `AME_INITIALISE` and `AME_TERMINATE`. In most cases, you won't actually need to do any processing of these messages - but you must, at least, return `AME_OK`.

ANY-maze will send an `AME_INITIALISE` message to your DLL when it (ANY-maze) starts up. This is your opportunity to perform any initialisation your plug-in requires. For example, if your plug-in needs to create a thread, perhaps to manage communication with some other program, then this is the time to do it.

You should always return `AME_OK` to this message - returning any other value will mean that your plug-in is unloaded by ANY-maze, and it will then be entirely ignored by the system.

As you would expect, the `AME_TERMINATE` message is used to terminate your plug-in. This message is sent when ANY-maze is closing down, and gives you an opportunity to clean-up - for example, if you create a thread to communicate with another program, then this would be the time to tell the other program that you are closing down and to end the thread.

```
case AME_INITIALISE :
case AME_TERMINATE  :
  //Nothing to do
  return AME_OK;
```

8. Create a GUID for your plug-in

Even if your DLL only implements a single plug-in, the plug-in still needs an ID and this should be a Windows Globally Unique Identifier, or GUID.

You can create a GUID using the free GUIDGEN.EXE tool, which comes with Visual Studio. (It can also be downloaded, for free, from Microsoft's web-site.)

GUIDGEN can provide you with the new ID it generates in a variety of formats - choose the `static const...` format.

Having created the GUID, you should paste it into your source file:

```
//GUIDs for our plug-ins. These can be created using guidgen.exe
static const GUID GUID_MYFIRSTPLUGIN  =
{0xf5c18c7b,0x5a39,0x47f8,{0x99,0x2b,0x13,0x37,0xfe,0x0f,0x90,0xd5}};
static const GUID GUID_MYSECONDPLUGIN =
{0xb6d62e2c,0xf888,0x40e8,{0xb8,0x87,0xac,0x1f,0x45,0x14,0xb7,0x70}};
```

You can call your GUID anything you like - the name will only be used within your plug-in.

If your DLL will implement multiple plug-ins, you should generate one GUID for each one.

9. Add processing of the AME_ENUMERATE message

Having created a GUID for the plug-in (or plug-ins), you are ready to process the `AME_ENUMERATE` message.

ANY-maze will send `AME_ENUMERATE` messages to your DLL to ask it what plug-ins it includes. This works as follows:

The first time ANY-maze passes an `AME_ENUMERATE` message, the `guid` parameter (of `ANYmazeExtension_Main`) will be `GUID_NULL.` Your code should detect this and return the GUID of the first plug-in it implements. The next time ANY-maze passes an `AME_ENUMERATE` message, the GUID parameter will be the same as the GUID you just returned; your code should detect this and return the GUID of the next plug-in it implements. You should continue in this way until you have returned the GUIDs of all the plug-ins in your DLL. You should then return `AME_NOTIMPLEMENTED` to indicate to ANY-maze that the enumeration is complete. Here's an example:

```
case AME_ENUMERATE :
  //Simply enumerate the GUIDs of the analysis we can perform,
  //returning the next GUID in the buffer pointed at by lParam.
  //When enumeration starts we are passed a null GUID
  if (*guid == GUID_NULL)
  {
    //Enumeration is starting - return GUID of the first plug-in
    //we implement
    *((LPGUID)lParam) = GUID_MYFIRSTPLUGIN;
    return AME_OK;
  }
  else if (*guid == GUID_MYFIRSTPLUGIN)
  {
    //We've now returned the GUID of the first plug-in, so now
    //return the GUID of the second one
    *((LPGUID)lParam) = GUID_MYSECONDPLUGIN;
    return AME_OK;
  }
  else if (*guid == GUID_MYSECONDPLUGIN)
  {
    //We've now returned the GUID of the second plug-in, so that's
    //all the plug-ins we implement. Return AME_NOTIMPLEMENTED to
    //end enumeration
    return AME_NOTIMPLEMENTED;
  }
  else
  {
    //We should never be passed GUIDs that we don't recognise
    return AME_ERROR;
  }
  break;
```

This code can just be dropped into the `ANYmazeExtension_Main` routine where it says `'....`

`we'll add some case statements in a minute ....'` in the above example.

Here, I've included two plug-ins to show how the enumeration works, but for the rest of these instructions I'll just use a single plug-in. If you want to see more details on how to implement multiple plug-ins in a single DLL, refer to the ANY-maze demo plug-in listing.

10. Add processing of the AME_GETNAME message

Using the `AME_ENUMERATE` message, ANY-maze can learn what plug-ins your DLL implements - but it still needs to know what the plug-ins are *called*, so it can display their names to the user.

To this end, it will send `AME_GETNAME` messages. This message is very simple; you just need to return the name of the plug-in whose GUID is passed (and if you only implement a single plug-in, then you can even ignore the GUID - which is what the code below does).

```
case AME_GETNAME :
  //Return the name of the GUID in the buffer pointed to by lParam. The
  //buffer's size is in wParam
  StringCbCopy (lParam, wParam, "My first plug-in");
  return AME_OK;
```

Note that in this example, the plug-in name is meaningless - you should try to use more descriptive names for your plug-ins.

By the way, a better approach than the example would be to add the name to the DLL's resource file and then load it using `LoadString.` Also note that here I didn't check the passed GUID, because my DLL only implements one plug-in, but it would still be good practice to check the GUID and return `AME_ERROR` if you don't recognise it.

11. Add processing of the AME_GETRESULTTYPE message

For all plug-ins, you need to decide what result type you will return. There are five types, including one for 'None', so you don't actually have to return a result at all.

The other types are `RT_ONOFF`, `RT_VALUE`, `RT_CUMULATIVE` and `RT_ACTION`. For this example, we're going to use a result type of `RT_ONOFF`.

ANY-maze will query your plug-in to find out what type of result it returns, by sending it an `AME_GETRESULTTYPE` message. The processing is very simple:

```
case AME_GETRESULTTYPE :
  //Return the result type in the unsigned char pointed to by lParam
  *((unsigned char*)lParam) = RT_ONOFF;
  return AME_OK;
```

Like for the AME_GETNAME message processing, I've not checked the passed GUID, which it would be better to do.

By the way, we have now performed enough steps for our plug-in to be recognised by ANY-maze and to work - well kind of - of course, it won't actually do anything. But if you paste the above 'cases' into the ANYmazeExtension_Main routine, build your DLL, copy it to the ANY-maze.exe folder, start ANY-maze and look at the plug-ins element of the protocol, you should see your plug-in listed!

12. Optionally add processing of the AME_GETHELPxxx messages

Unless your plug-in is just for your own use, it's strongly recommended that you provide some help (even if it's minimal). This is very easy to do, as you simply have to write some text and pass it back to ANY-maze when it sends AME_GETHELPxxx messages.

There are three AME_GETHELPxxx messages:

- AME_GETHELPTITLE
- AME_GETHELPINTRO
- AME_GETHELPDETAILS

The only difference between the messages is that they're asking for different text: either the Title for the plug-in's help topic, the Introductory text or the Details text. How you choose to divide the text between Introduction and Details is up to you, but be aware that the Introductory text you return will appear in a section of the help topic titled 'Introduction', and the details text will appear in a section titled 'Details'. Also, the introductory text is limited to 1024 characters, whereas the details text can be up to 64K in size.

Like plug-in names, it's a good idea to define your text in the DLL's resource file, rather than embedding strings in the source code, and this is the approach adopted below:

```
case AME_GETHELPTITLE   :
case AME_GETHELPINTRO   :
case AME_GETHELPDETAILS :
  //These all return strings, which we load from the DLL's resources
  if (Mesg == AME_GETHELPTITLE)
    RscID = IDS_HELP_TITLE;
  else if (Mesg == AME_GETHELPINTRO)
    RscID = IDS_HELP_INTRO;
  else
    RscID = IDS_HELP_DETAILS;

  //Load the resource
  return ((LoadString (hInstance, RscID, (LPSTR)lParam, (int)wParam) !=
0) ? AME_OK : AME_ERROR);
```

The actual text of the help message is not necessarily formatted in the way that you would probably expect. Specifically, you need to use special *mark-up* within the text to end paragraphs, this mark-up being \para.

So, for example, returning the string 'Hello\para World' would write the words 'Hello' and 'World' on separate lines. In fact, there is more *mark-up* available to you - here's a full list:

`\para`	Starts a new paragraph
`\i1`	Switches italic on
`\i0`	Switches italic off
`\u1`	Switches underline on
`\u0`	Switches underline off
`\e1`	Switches bold on
`\e0`	Switches bold off
`\bullet`	Start a new bulleted item in a bullet list

Note that the '\' character is the C/C++ *escape* character, so if you type it into a string in C/C++ code, it will be taken to mean that you're *escaping* the following character - so when typing in '`\para`' you should actually enter '`\\para`', as the escaped '\' character is itself.

Also note that CR or LF characters will not cause new paragraphs in the help message - use `\para` to do this.

Finally, if you don't want to include help then you should simply return `AME_NOTIMPLEMENTED` to all of the `AME_GETHELPxxx` messages.

13. Optionally add processing of the AME_EDITSETTINGS message

In some cases, a plug-in might have user definable settings. For example, the demo plug-in detects when the animal is within a certain distance of the centre of the apparatus - but how far that distance is, is something the user can define using the plug-in's settings.

The settings for a plug-in are typically set using a Windows dialogue box, but you can use a different user interface if you wish.

The actual settings are treated by ANY-maze as a simple buffer (i.e. a contiguous block of bytes), which it stores on behalf of the plug-in as part of the experiment's protocol.

When the user clicks the *Alter the settings...* link for a plug-in, ANY-maze sends an `AME_EDITSETTINGS` message to the plug-in, with the lParam pointing at the settings buffer and the wParam set to the buffer's size. The size of the buffer is fixed, but you don't have to use the entire buffer if you don't need to; you just can't use *more* than the size passed in wParam.

The first time ANY-maze passes a settings buffer to your plug-in, the buffer will be filled with zeros - you should detect this and set the buffer to hold some suitable default values.

For example, you might choose to define the settings in your plug-in as:

```
typedef struct {
  bool Initialised;
  int  DistanceFromCentre;
}MYPLUGIN_SETTINGS, *MYPLUGIN_SETTINGS;
```

When ANY-maze first passes you these settings, you could use some code like this to detect that the settings are uninitialised and set the DistanceFromCentre to a default value:

```
LPMYPLUGIN_SETTINGS MyPlugInSettings;

switch (Mesg)
{
  case AME_EDITSETTINGS :
    //Check our settings are not bigger than the buffer
    assert (wParam >= sizeof(MYPLUGIN_SETTINGS);
    if (wParam >= sizeof(MYPLUGIN_SETTINGS) return AME_ERROR;

    //Access the settings
    MyPlugInSettings = (LPMYPLUGIN_SETTINGS)lParam;

    //Check for uninitialised settings
    if (!MyPlugInSettings->Initialised)
    {
      //Set default values
      MyPlugInSettings->DistanceFromCentre = 10;

      //Now the settings are initialised
      MyPlugInSettings->Initialised = true;
    }

    ....continue processing the message

    break;
}
```

The settings for a plug-in are typically set using a Windows dialogue box. I don't intend to go into the details of how to display and process a dialogue box here, but you'll find that the demo plug-in code can act as a good basis for your own implementation. Nevertheless, there are a couple of points worth mentioning: first, you will probably want to use `DialogBoxParam` as this will allow you to pass a pointer to the settings buffer to the `DialogProc;` second, in the `DialogProc` you *can* use a `static` variable to store the pointer, as ANY-maze will only pass `AME_EDITSETTINGS` messages from its main thread, so there are no multi-threaded issues for you to contend with.

To close your settings window, the user will probably have two choices - *OK* and *Cancel*. In the former case, you will want to update the settings buffer with the new settings entered by the user and then have ANY-maze save the settings in the protocol; in the latter case, you will want any new settings to be ignored and you won't want the settings buffer to be saved. ANY-maze will use the value returned from the `AME_EDITSETTINGS` message to determine what it should do - return `AME_OK` if you want the settings to be saved, or AME_CANCELLED if you don't want them saved.

Of course, some plug-ins may not have any settings - in which case you should simply return AME_NOTIMPLEMENTED to the AME_EDITSETTINGS message.

14. Add processing of the AME_TESTSTART message

Now we get to the real work of a plug-in - to actually analyse data during a test. This is handled through four messages: AME_TESTSTART, AME_POSN, AME_TESTEND and AME_TESTABORT. Here, I'll describe the AME_TESTSTART message.

At the moment that a test starts, ANY-maze will send all active plug-ins (i.e. plug-ins the user has selected to use in the protocol) an AME_TESTSTART message. This provides certain test-specific data to the plug-in, and acts as the moment for the plug-in to initialise itself so it's ready to analyse the test.

Specifically, the lParam of the AME_TESTSTART message will point at an AME_TESTDATA record, and the wParam will be set to the size of the AME_TESTDATA record. This record is defined in the PlugIn.h file as follows:

```
typedef struct {
  DWORD    AnimalNum;
  char     AnimalID[80];
  DWORD    TrialNum;
  RECT     ApparatusRect;
  double   Scaling;
  LPVOID   Settings;
  DWORD    SettingsSize;
  LPVOID   Context;
}AME_TESTDATA, *LPAME_TESTDATA;
```

First, a brief note about the wParam. As mentioned above, this specifies the size of the AME_TESTDATA record. Seemingly, this is unnecessary - as the record is defined in PlugIn.h and so you could get the size using sizeof(AME_TESTDATA). However, the reason the size is passed is to allow us (the ANY-maze development team) to add new fields to this record in the future. For example, at some time in the future we might define an AME_TESTDATA2 record with additional fields. In this case, you will be able to tell whether the record being passed is an AME_TESTDATA or AME_TESTDATA2 record by comparing the wParam to the size of these two record types.

The fields of the AME_TESTDATA record are described in the record's help topic (click the record name to view this topic), but I will mention two of them now. The Settings field points at a Settings buffer, which is just the same as the Settings buffer passed in the AME_EDITSETTINGS message (see above). So, if the user has actually changed the settings, these will be provided in this buffer - otherwise the buffer will be all zeros and you should use default values. The other field which deserves attention is the Context field, which you will use to *return* a context value to ANY-maze.

The *Context* can be any value you want, but typically it will be a pointer to a record that you define within your plug-in and which you allocate as part of your AME_TESTSTART processing. The idea is as follows: ANY-maze can run multiple tests simultaneously, so your plug-in may be sent messages about, say, three tests (all of which are being performed at the same time). If you allocate a *Context* record during AME_TESTSTART processing, then you can fill it with useful fields related to the analysis that your plug-in performs and return it to ANY-maze by setting the Context field of the AME_TESTDATA record to point at it. ANY-maze will then pass this value back to your plug-in as part of the AME_POSN message, so you can use the fields it contains to update your analysis. If there are three tests running simultaneously, you will simply end up allocating three records (because you'll be sent three AME_TESTSTART messages) and ANY-maze will pass a reference to the appropriate record with each AME_POSN message. This means you don't need to worry (too much) about the fact that your plug-in may be called by multiple threads, because you can simply keep all analysis variables in your context record which will automatically be thread-specific.

Bearing the above in mind, you may very well want to copy some of the data provided in the AME_TESTDATA record to your Context record, so it's available to you during AME_POSN processing - for example, you'll almost certainly want to do this with any Settings that you have.

In fact, the explanation is more complicated than the code - here's the AME_TESTSTART processing performed by the demo plug-in, which detects when the animal is in the centre of the apparatus:

```
typedef struct {
  double Distance;
  int    AppCentreX;
  int    AppCentreY;
  bool   InCentre;
}INCENTREANALYSISRECORD, *LPINCENTREANALYSISRECORD;

....

case AME_TESTSTART :
  //The lParam points at a AME_TESTDATA record and the wParam is the
  //size of the record. Check the size
  assert (wParam >= sizeof(AME_TESTDATA));
  if (wParam < sizeof(AME_TESTDATA)) return AME_ERROR;

  //Access the AME_TESTDATA record
  AME_TestData = (LPAME_TESTDATA)lParam;

  //Allocate a InCentreAnalysisRecord - if this fails return AME_ERROR
  InCentreAnalysisRec = new INCENTREANALYSISRECORD;
  if (InCentreAnalysisRec == NULL) return AME_ERROR;

  //Calculate and note the centre point of the apparatus
  InCentreAnalysisRec->AppCentreX = (AME_TestData->ApparatusRect.left +
AME_TestData->ApparatusRect.right) / 2;
  InCentreAnalysisRec->AppCentreY = (AME_TestData->ApparatusRect.top +
AME_TestData->ApparatusRect.bottom) / 2;

  //The passed AME_TestData record includes a pointer to the settings
  //for this analysis. The settings define the distance from the centre
  //that we consider to be the "centre" of the apparatus. Convert this
```

```
    //distance to pixels and note it in the InCentreAnalysisRec. If the
    //distance is zero then the user has not specified a distance so
    //use a default of 10cm
    Settings = (LPINCENTREANALYSISSETTINGS)AME_TestData->Settings;
    if (!Settings->Initialised) Settings->Distance = 10;
    d = (double)Settings->Distance;

    //Convert from cm to mm
    d = d * 10;

    //Convert from mm to pixels
    InCentreAnalysisRec->Distance = d * AME_TestData->Scaling;

    //Start by assuming the animal is not in the centre of the apparatus
    InCentreAnalysisRec->InCentre = false;

    //Set Context to the address of the InCentreAnalysisRec
    AME_TestData->Context = (LPVOID)InCentreAnalysisRec;
    return AME_OK;
```

As can be seen in the above example, if an error occurs during AME_TESTSTART processing you can just return AME_ERROR - ANY-maze will then flag your plug-in as inactive (for this test) and won't send any further messages to it.

15. Add processing of the AME_POSN message

The AME_POSN message is the heart of the plug-in system, as this is the message which actually tells you where the animal is. To do this, ANY-maze passes you an AME_POSNDATA record. Specifically, the lParam of the AME_POSN message points at the AME_POSNDATA record, and the wParam tells you the record size.

The size of the record is passed for the same reasons that the AME_TESTSTART message passed the size of the AME_TESTDATA record (see above) - i.e. so we (the ANY-maze development team) can add fields to the record in the future, and plug-in code will be able to tell which version of the record is being passed.

```
typedef struct {
  LPVOID   Context;
  DWORD    Time_xy;
  int      x;
  int      y;
  DWORD    Time_HeadTail;
  int      hx;
  int      hy;
  int      tx;
  int      ty;
  int      cx;
  int      cy;
  bool     Hidden;
```

```
  RECT      AnimalArea;
  int       ResultType;
  int       ResultValue;
}AME_POSNDATA, *LPAME_POSNDATA;
```

The first field of the AME_POSNDATA record is the Context, which is the Context value you passed back to ANY-maze as part of your AME_TESTSTART processing (see above).

The other fields are defined in the AME_POSNDATA topic, so, except for the two 'Result' fields, I won't describe them here.

The ResultType and ResultValue fields are where you will return the result of your analysis. Specifically, you will return an RT_xxx value in ResultType and, unless the ResultType is RT_NONE, the actual value of the result in ResultValue.

The result types are described in their respective topics, but briefly they are:

- RT_NONE, which you should use when you don't have a result to return;
- RT_ONOFF, which is used when your analysis determines when something starts and stops (like the animal being in the apparatus centre, or not);
- RT_VALUE, which is used when your analysis determines a specific instantaneous value, for example, the animal's heart rate;
- RT_CUMULATIVE, which is used when your analysis determines a value for the time between the previous position and this one, such as the distance the animal has travelled (in this case ANY-maze will sum all the values you pass it).

So, here's an example of the processing performed by the demo plug-in which determines when the animal is in the centre of the apparatus:

```
case AME_POSN :
  //The lParam points at a AME_POSNDATA record and the wParam is the
  //size of the record. Check the size
  assert (wParam >= sizeof(AME_POSNDATA));
  if (wParam < sizeof(AME_POSNDATA)) return AME_ERROR;

  //Access the AME_POSNDATA record
  AME_PosnData = (LPAME_POSNDATA)lParam;

  //Access the InCentreAnalysisRec
  if (AME_PosnData->Context == NULL) return AME_ERROR;
  InCentreAnalysisRec = (LPINCENTREANALYSISRECORD)AME_PosnData->Context;

  //Calculate the distance from the animal's posn to the apparatus
centre
  dx = AME_PosnData->x - InCentreAnalysisRec->AppCentreX;
  dy = AME_PosnData->y - InCentreAnalysisRec->AppCentreY;
  d  = sqrt((dx * dx) + (dy * dy));

  //Is the animal within the required distance of the apparatus centre
```

```
  NowInCentre = (d < InCentreAnalysisRec->Distance);

  //Has the animal's "InCentre" state changed
  if (NowInCentre != InCentreAnalysisRec->InCentre)
  {
    //Yes, so return a suitable result
    AME_PosnData->ResultType  = RT_ONOFF;
    AME_PosnData->ResultValue = NowInCentre ? 1 : 0;

    //Update the in centre state
    InCentreAnalysisRec->InCentre = NowInCentre;
  }
  else
  {
    //We're not returning a result
    AME_PosnData->ResultType = RT_NONE;
  }

  //Processed successfully
  return AME_OK;
```

Note that you may not ever return a result. For example, your plug-in might simply pass the coordinates on to some other piece of software (perhaps via a pipe or DDE), in which case you will just return RT_NONE as the ResultType for every AME_POSN message you receive.

16. Add processing of the AME_TESTEND and AME_TESTABORT messages

As you'd probably expect, when a test ends, ANY-maze will send you a message to tell you this. In fact it sends one of two messages, either AME_TESTEND or AME_TESTABORT. As the names imply, the former is sent when a test ends normally, whereas the latter is sent when a test is aborted and its results are not going to be saved.

In either case, the parameters of the message are the same - wParam is unused (set to zero) and lParam points at your Context record for the test. You will typically use this to perform any clean-up operations and to deallocate the Context record (assuming you allocated one in the first place). It's important to note that ANY-maze will not deallocate this record - your plug-in allocates it, so your plug-in must deallocate it.

Here's the code for the demo plug-in's processing of AME_TESTEND and AME_TESTABORT; as you'll see, it's the same in either case:

```
case AME_TESTEND   :
case AME_TESTABORT :
  //The lParam is our context value
  if (lParam == NULL) return AME_OK;
  InCentreAnalysisRec = (LPINCENTREANALYSISRECORD)lParam;

  //Delete the InCentreAnalysisRec
  delete InCentreAnalysisRec;
```

```
  //Done
  return AME_OK;
```

A final point about processing errors. If your plug-in encounters an error during the test (i.e. when processing an AME_POSN message) then you will return AME_ERROR. This *will* ensure that no result is recorded, but it won't stop further AME_POSN messages from being sent. So, if you encounter a serious error which will prevent your analysis from continuing, you will need to flag this in your Context record. Then on each AME_POSN message you can check the flag and return AME_ERROR if it's set.

This is fine, but it leaves one problem - what if you had returned some results to ANY-maze *before* the error occurred? These results will have been saved by ANY-maze, but because of the error they may now be incorrect (for example, if the demo plug-in simply stopped detecting when the animal is in the centre, then all the time from that moment on would be counted as either in-centre or not-in-centre, depending on the state when the error occurred). Clearly this is not satisfactory. To overcome this problem, you can return AME_CANCELLED from your AME_TESTEND processing. This will cause ANY-maze to delete any results it had recorded for your plug-in in the test, so it'll be like the plug-in simply wasn't active for the test.

So, updating the above code to include this:

```
case AME_TESTEND   :
case AME_TESTABORT :
  //The lParam is our context value
  if (lParam == NULL) return AME_OK;
  InCentreAnalysisRec = (LPINCENTREANALYSISRECORD)lParam;

  //Did some serious error occur during analysis? If so
  //we want to return AME_CANCELLED, else we will return AME_OK
  retcode = InCentreAnalysisRec->SeriousErrorDetected ? AME_CANCELLED :
AME_OK;

  //Delete the InCentreAnalysisRec
  delete InCentreAnalysisRec;

  //Done
  return retcode;
```

17. You're done!

If you've read this far, then you should now know enough to get started writing a plug-in for yourself. In fact, I don't recommend you start from scratch - instead you may find it easier to make a copy of the ANY-maze demo plug-in source code, and edit it to suit your needs. You can do this a little at a time, and test things as you go along.

And, of course, ANY-maze technical support are on hand to help you, so do contact us if there's anything you're unsure of. By the way, we're very aware that most people who use ANY-maze are not programmers, so don't worry about contacting us with simple questions - we'll keep our answers as un-technical as possible, and we'll gladly write sections of code for you if you get stuck. Indeed we'll even write entire plug-ins (for free) provided you can supply us with a detailed specification of what you'd like the plug-in to do.

See also:

- The ANY-maze demo plug-in
- Plug-in API reference

Writing an Action plug-in

Introduction

This topic assumes you have already read the step-by-step instructions for writing an ANY-maze plug-in, so if you've not done so yet, I suggest you do this now.

An *Action plug-in* is a plug-in which can be invoked by an ANY-maze procedure firing an action (or, if the protocol is using the older 'Events and Actions' system, by an action directly triggered by an event). This makes Action plug-ins very powerful; they effectively get 'called' when something happens during the test, and what they then do can be anything the author wants.

Making a plug-in into an 'Action' plug-in

An Action plug-in is like a normal plug-in, so it needs a GUID and should respond to most of the same messages. What makes it into an Action plug-in is its response to the `AME_GETRESULTTYPE` message - this should be `RT_ACTION`.

```
case AME_GETRESULTTYPE :
  //Return the result type in the unsigned char pointed to by lParam
  *((unsigned char*)lParam) = RT_ACTION;
  return AME_OK;
```

This return value tells ANY-maze that the plug-in is an Action plug-in, and it will then treat the plug-in a little differently, specifically:

- The plug-in will not be listed in the plug-ins list in the protocol.
- The list of actions available to a procedure will contain actions to trigger each of the Action plug-ins available - these will be listed under the plug-in's name (i.e. whatever the plug-in returns from the `AME_GETNAME` message). If the protocol is using the older 'Events and Actions' system, then an entry will be added to the available 'effects' of an action.
- If the user clicks the plug-in's 'effect' in this list, then the plug-in will receive an `AME_EDITSETTINGS` message, and should display a window where the user can specify

any settings for the action.

- The plug-in will receive AME_TESTSTART, AME_TESTEND and AME_TESTABORT messages, but it won't receive AME_POSN.
- The plug-in will receive an AME_PERFORMACTION message when the action is triggered.

⚠ *Plug-ins are only available in licensed copies of ANY-maze.*

Action plug-in settings

As mentioned above, an Action plug-in is listed as a possible *effect* of an action. What the plug-in does is, of course, defined by the plug-in's author, but in some cases it may be useful to specify some settings for the action. For example, imagine you wrote an Action plug-in which can switch some piece of equipment on or off, and you want the equipment to switch on when the animal enters a zone and off when it exits the zone. In this case, you would include in your procedure an action which would have your plug-in as its *effect*. But how would the plug-in know whether it should switch the equipment on or off? As this action would be linked to an event that is triggered by the animal entering the zone, we would want the equipment to switch on - but you would need to tell the plug-in that this is what it should do.

This is where an Action plug-in's settings come in - you could include in your plug-in a dialogue box where the user could specify what the plug-in should do, perhaps two radio-buttons: 'Switch On' and 'Switch Off'. Now when the user selects the plug-in as the effect of the action, the dialogue box will open and he will need to specify what the plug-in will do - so he'd choose 'Switch On'.

ANY-maze takes care of the settings for an Action plug-in, so it passes them to the plug-in when the user selects the plug-in as the effect of an action, and it passes them back when the plug-in action is to be performed. Physically, the settings are just a block of memory of 48 bytes. You can use as much or as little of this memory to store settings as you wish. It's a good idea to declare a struct for the settings, and just include a field called something like *spare* to pad the size to 48 bytes. For example:

```
typedef struct {
  bool           SwitchOn;
  unsigned char  spare[47];
}MYACTIONSETTINGS;
```

Note that ANY-maze will zero-initialise the settings buffer. This means that you should either arrange things so that your 'default' values are all zero, or use a flag to determine whether the settings have been initialised. For example, you might declare your settings as:

```
typedef struct {
  bool           Initialised;
  bool           SwitchOn;
```

```
  unsigned char  spare[46];
}MYACTIONSETTINGS;
```

Now the initialised flag will default to false, so you will be able to tell whether the settings have been initialised or not. When they have not been initialised, you can set them to some default values before using them - for example:

```
if (!Settings->Initialised) {
  Settings->SwitchOn    = true;
  Settings->Initialised = true;
}
```

When the user selects an Action plug-in as the effect of an action, the plug-in will receive an `AME_EDITSETTINGS` message. This is the same message as is used to edit a non-action plug-in's settings, and it works in the same way. If your Action plug-in doesn't have any settings, then you can just return `AME_NOTIMPLEMENTED`; otherwise you should display a modal dialogue box where the user can edit the settings. Return `AME_OK` to have the updated settings saved by ANY-maze, or `AME_CANCELLED` if the user closes the dialogue box using a cancel button.

It's important to understand that ANY-maze will record a different settings record for your plug-in for each ANY-maze action that has the plug-in as an effect. Returning to our earlier example, when the animal enters the zone we want to switch the equipment on, so we will have an action whose effect is to invoke our plug-in and the settings will have the SwitchOn flag set. When the animal exits the zone, we want to switch the equipment off. In this case we will have a *different* action, and this will also have the effect of invoking our plug-in - just that in this case the plug-in's settings will have the SwitchOn flag set to false.

Processing AME_TESTSTART and AME_TESTEND messages

Like a normal plug-in, an Action plug-in is sent an `AME_TESTSTART` message when a test begins. This is identical to the message sent to a non-action plug-in, and you should process it in the same way - see Step-by-step instructions for writing an ANY-maze plug-in for full details.

Your plug-in will also receive an `AME_TESTEND` or `AME_TESTABORT` message when the test ends; again these are processed in the standard way.

Performing the action

When ANY-maze determines that the action should be performed, it will send your plug-in an `AME_PERFORMACTION` message. The lParam of this message points to an `AME_ACTIONDATA` record, which contains information about the action to perform. The wParam contains the size of this record.

The `AME_ACTIONDATA` record is defined as:

```
typedef struct {
  LPVOID   Context;
  DWORD    EventTime;
  char     TriggeringEventName[80];
  LPVOID   Settings;
  DWORD    SettingsSize;
}AME_ACTIONDATA, *LPAME_ACTIONDATA;
```

Full details can be found in the AME_ACTIONDATA topic, but briefly:

- The Context field holds whatever value you returned as your 'context' during AME_TESTSTART processing (typically, an Action plug-in doesn't use this).
- EventTime is the time the event which triggered this action occurred, and TriggeringEventName is the name of the event. If this plug-in is triggered from a procedure, this will be the name of the procedure.
- The Settings and SettingsSize are the action's settings, as entered by the user when he set up the action in the protocol or procedure (see Action plug-in settings for more details).

When it receives this message, your plug-in should perform whatever action it is designed to do - so it might communicate with some other program, or switch some piece of equipment on or off.

Often, you will need to use the settings to decide what it is you need to do - but, as mentioned above, the settings record may be uninitialised, in which case you will need to decide what to do. You could choose not to perform any action; you could set the settings to some default state and then perform the default action; or you could organise things in such a way that an entirely null settings record is your default (in which case, you won't need to worry about whether the settings have been initialised or not). Of course, your plug-in may not have any settings, in which case you won't care about the settings record's state.

In most cases, you probably won't be interested in the name of the event or procedure that caused the plug-in to be invoked, but it can be useful as it will usually tell you what it was that happened in the test - for example, it might be 'Entered open arm'. If your plug-in communicates with another program, perhaps to tell it when specific things occur in the test, then you may wish to pass this information to the other program.

Example Action plug-in

Here's the code for a very minimal Action plug-in. When invoked, this plug-in will make the computer go 'beep'. Note that for clarity, this implementation does not check the GUID passed to messages such as AME_GETNAME, as the DLL only implements one plug-in - this will work correctly, but it is good practice to always check the GUID.

```
extern "C"
__declspec(dllexport) int ANYmazeExtension_Main (LPGUID guid,
                                                 DWORD  Mesg,
                                                 WPARAM wParam,
                                                 LPARAM lParam)
{
//Const
  static const GUID GUID_BEEPERPLUGIN =
{0x7f32b71d,0x8985,0x4e71,{0xb1,0xcc,0x2c,0x83,0xa,0x43,0xb5,0xb1}};

//BEGIN
  //Action depends on the Mesg
  switch (Mesg)
  {
    case AME_INITIALISE :
    case AME_TERMINATE  :
      //We must return AME_OK, even if we do nothing
      return AME_OK;

    case AME_ENUMERATE :
      //We only implement a single plug-in
      if (*guid == GUID_NULL)
        *((LPGUID)lParam) = GUID_BEEPERPLUGIN;
      else
        return AME_NOTIMPLEMENTED;
      return AME_OK;

    case AME_GETNAME :
      //Set the name
      StringCbCopy ((LPSTR)lParam, (int)wParam, "Make the computer
beep");
      return AME_OK;

    case AME_GETRESULTTYPE :
      //Return our result type which is RT_ACTION
       *((unsigned char*)lParam) = RT_ACTION;
      return AME_OK;

    case AME_TESTSTART :
    case AME_TESTEND   :
      //We must return something otherwise the plug-in will not work.
We have
      //nothing to do when a test starts or ends so we just return
AME_OK
      return AME_OK;

    case AME_TESTPAUSE :
    case AME_TESTABORT :
      //We don't care about test pause or abort, so return AME_OK
      return AME_OK;

    case AME_PERFORMACTION :
      //Make the speaker beep
      MessageBeep (0);
      return AME_OK;
```

```
    default :
      //We don't need to process any other messages
      return AME_NOTIMPLEMENTED;
  }
}
```

The ANY-maze demo plug-in

Installing the demo plug-in

The ANY-maze demo plug-in is a standard part of ANY-maze, but it is not installed by default.

Click this link to install it now: Install ANY-maze demo plug-in

The plug-in code will be copied to a sub-folder of the `ANY-maze.exe` folder called `Demo plugin source files`, and the plug-in DLL will be copied to the `ANY-maze.exe` folder itself.

After restarting ANY-maze, you will then find that the *Analysis* element of the Protocol (under *Analysis options*) will include a sub-entry for *ANY-maze plug-ins*, and that two plug-ins will be available:

- Animal in apparatus centre
- Record animal positions

Both the plug-ins include help, so you can use this to learn what they actually do.

Plug-in demo files

The `Demo plugin source files` folder created by installing the demo plug-in (see above) includes the following files:

`PlugIn.h`	This is the header file which defines the interface between ANY-maze and a plug-in. All plug-ins, not just the demo, should include this header file. A listing of this file can be found in the topic titled The PlugIn.h header file.
`DemoPlugIn.cpp`	This file contains the demo plug-in source code. Although it's a C++ file, the code is essentially just C.
`DemoPlugIn.rc`	This file contains the resources for the demo plug-in. A plug-in does not *have* to include resources, although if you implement a Settings window then you will need them. It's also good practice to store strings in resources rather than embedding them in the source.
`resource.h`	The resource header file, which contains the IDs of the resources used in the demo plug-in. This file is usually generated automatically by resource editors.
`DemoPlugIn.dsp`	The Visual Studio project file for the demo plug-in. This file was generated with Visual Studio 6 and should be recognised by all subsequent versions.

If you are using Visual Studio 6 or later, then you can simply open the `DemoPlugIn.dsp` file and build the plug-in. (On more recent versions of Visual Studio, the file will be converted to a newer format automatically). Otherwise, you will need to create your own project - but this is usually quite easy to do. You should specify that the project will be a **Win32 DLL,** and then add the source files to it. Note that you should ensure that the project is set to build 32-bit or 64-bit code, depending on the version of ANY-maze that you are using (see *Help* -> *About ANY-maze* -> *Program information* -> *Application version* if you're unsure), and you should configure the project to use multi-threaded run-time libraries, because ANY-maze may call the plug-in from different threads.

As an alternative to installing the demo plug-in, you'll find a full listing of the demo plug-in source code in the topic titled The DemoPlugIn.cpp source code file.

The DemoPlugIn.cpp source code file

The `DemoPlugIn.cpp` file contains the source code of the demo ANY-maze plug-in. Although this is a C++ file, the code is essentially just C.

The listing

```
/*---------------------------------------------------------------------------
--------

  Name        : DemoPlugIn

  Description : Example of an ANY-maze plug in (also know as an analysis
extension
                DLL).
                This plug-in implements the following plug-ins:

                Animal in apparatus centre
                --------------------------
                This returns an "Animal in centre" ON/OFF result to
ANY-maze
                whenever the animal's centre point is within the centre of
the
                apparatus. The definition of "in the centre of the
apparatus"
                is that the animal is within a certain distance of the
point
                at the centre of a rectangle that exactly encloses the
apparatus.
                This distance defaults to 10cm, but can be edited by the
user
                via the plug-in's settings dialog.


                Record animal position
                ----------------------
                This plug in dumps the animal's centre x,y coordinates
with a
                time value, to a csv file named for the animal and trial
being
                performed. Thus this plug-in does not return any result.
Nor
                does this plug-in does not a have a settings dialog.


  History
  -------
  7/2/2008   CPL   Created first version of this example code

---------------------------------------------------------------------------
--------*/

#include <windows.h>
```

```
#include <assert.h>
#include <math.h>
#include "plugin.h"
#include "resource.h"


//GUIDs for our analysis. These can be created using guidgen.exe
static const GUID GUID_INCENTREANALYSIS =
{0xf5c18c7b,0x5a39,0x47f8,{0x99,0x2b,0x13,0x37,0xfe,0x0f,0x90,0xd5}};
static const GUID GUID_POSNEXPORT       =
{0xbf44eea9,0x4a72,0x4b1d,{0xb4,0xaf,0x67,0xdd,0x80,0x5a,0x16,0xd2}};


//Type used to hold settings for in centre analysis
typedef struct {
  bool Initialised;
  int  Distance;
}INCENTREANALYSISSETTINGS, *LPINCENTREANALYSISSETTINGS;



//Globals
static HMODULE hInstance;

/*---------------------------------------------------------------------------
---------

  Name    CentreWindow

  Desc    Centres a top level window on the screen. This routine can't
centre child
          windows

  Params  hWnd   Handle of the window to centre

  Return  None

----------------------------------------------------------------------------
--------*/

void CentreWindow (HWND hWnd)
{
//Vars
  RECT Rect, MaxSize;
  int  x, y;

//BEGIN
  //Get the window's rectangle
  GetWindowRect (hWnd, &Rect);

  //Get screen area not obscured by the task bar
  SystemParametersInfo (SPI_GETWORKAREA, 0, &MaxSize, 0);

  //Adjust the window rectangle to the centre of the screen
  x = MaxSize.left +
(((MaxSize.right-MaxSize.left)-(Rect.right-Rect.left))/2);
  y = MaxSize.top +
(((MaxSize.bottom-MaxSize.top)-(Rect.bottom-Rect.top))/2);

  //Centre the window
  SetWindowPos (hWnd, NULL, x, y, 0, 0, SWP_NOZORDER | SWP_NOSIZE);
}
```

```
/****************************** Posn export routines
******************************

  Name    PosnExport

  Desc    Part of the example analysis: This routine exports the posn of
the
          animal's centre point to a CSV file

  Params  Mesg     The type of message ANY-maze is sending to the plug in.
This code
                   handles the "Test" messages: AME_TESTSTART, AME_POSN,
AME_TESTEND
                   and AME_TESTABORT
          wParam   wParam sent with the message - depends on the message
          lParam   lParam sent with the message - depends on the message

  Return  And AME_xxx return code

----------------------------------------------------------------------------
--------*/

int PosnExport (DWORD   Mesg,
                WPARAM  wParam,
                LPARAM  lParam)
{
//Types
  typedef struct {
    char   FileName[MAX_PATH];
    HANDLE hFile;
  }POSNEXPORTRECORD, *LPPOSNEXPORTRECORD;

//Vars
  LPAME_TESTDATA     AME_TestData;
  LPAME_POSNDATA     AME_PosnData;
  LPPOSNEXPORTRECORD PosnExportRec;
  char               str[32];
  DWORD              BytesWritten;

//BEGIN
  //Action depends on the message
  switch (Mesg)
  {
    case AME_TESTSTART :
      //The lParam points at a AME_TESTDATA record and the wParam is the
      //size of the record. Check the size
      assert (wParam >= sizeof(AME_TESTDATA));
      if (wParam < sizeof(AME_TESTDATA)) return AME_ERROR;

      //Access the AME_TESTDATA record
      AME_TestData = (LPAME_TESTDATA)lParam;

      //Allocate a PosnExportRecord - if this fails return Context set to
NULL.
      //We'll use this in subsequent calls to determine that we can't
export
      PosnExportRec = new POSNEXPORTRECORD;
      if (PosnExportRec == NULL) {
        AME_TestData->Context = NULL;
        return AME_ERROR;
      }
```

```
      //Build the file name for the file we'll export to
      wsprintf (PosnExportRec->FileName,
                "c:\\Animal %u Trial %u.csv",
                AME_TestData->AnimalNum,
                AME_TestData->TrialNum);

      //Create the file, replacing any file that already exists
      PosnExportRec->hFile = CreateFile (PosnExportRec->FileName,
                                         GENERIC_WRITE,
                                         0,
                                         NULL,
                                         CREATE_ALWAYS,
                                         FILE_ATTRIBUTE_NORMAL,
                                         NULL);

      //If failed junk the PosnExportRec and return Context set to NULL
      if (PosnExportRec->hFile == INVALID_HANDLE_VALUE) {
        delete PosnExportRec;
        AME_TestData->Context = NULL;
        return AME_ERROR;
      }

      //Success. Set Context to the address of the PosnExportRec
      AME_TestData->Context = (LPVOID)PosnExportRec;
      return AME_OK;


    case AME_POSN :
      //The lParam points at a AME_POSNDATA record and the wParam is the
      //size of the record. Check the size
      assert (wParam >= sizeof(AME_POSNDATA));
      if (wParam < sizeof(AME_POSNDATA)) return AME_ERROR;

      //Access the AME_POSNDATA record
      AME_PosnData = (LPAME_POSNDATA)lParam;

      //Access the PosnExportRec
      if (AME_PosnData->Context == NULL) return AME_ERROR;
      PosnExportRec = (LPPOSNEXPORTRECORD)AME_PosnData->Context;

      //Build an entry that describes this posn. This is of the form
"time,x,y"
      wsprintf (str,
                "%u,%d,%d\r\n",
                AME_PosnData->Time_xy,
                AME_PosnData->x,
                AME_PosnData->y);

      //Write the entry to the file
      WriteFile (PosnExportRec->hFile, str, (DWORD)strlen(str),
&BytesWritten, NULL);

      //We don't return any result
      AME_PosnData->ResultType = RT_NONE;

      //Done
      return AME_OK;


    case AME_TESTEND   :
    case AME_TESTABORT :
```

```
      //The lParam is our context value
      if (lParam == NULL) return AME_ERROR;
      PosnExportRec = (LPPOSNEXPORTRECORD)lParam;

      //Close the file
      CloseHandle (PosnExportRec->hFile);

      //If Type is Abort then delete the file
      if (Mesg == AME_TESTABORT) DeleteFile (PosnExportRec->FileName);

      //Delete the PosnExportRec
      delete PosnExportRec;
      return AME_OK;


    default :
      //What's this
      assert (false);
      return AME_NOTIMPLEMENTED;

  }
}

/***************************  In centre analysis routines
**************************

  Name    InCentreAnalysisSettingsDlgProc

  Desc    Dialog proc for the in centre analysis settings dialog - this
allows the
          user to set the distance from the centre that the analysis uses

  Params  hDlg         Std
          Message      Std
          wParam       Std
          lParam       Std

  Return  Std

---------------------------------------------------------------------------
--------*/

BOOL CALLBACK InCentreAnalysisSettingsDlgProc (HWND   hDlg,
                                               UINT   Mesg,
                                               WPARAM wParam,
                                               LPARAM lParam)
{
//Vars
  static LPINCENTREANALYSISSETTINGS Settings;

//BEGIN
  //Action depends on the mesg
  switch (Mesg)
  {
    case WM_INITDIALOG :
      //lParam points at the settings we're to edit. Note the pointer
      Settings = (LPINCENTREANALYSISSETTINGS)lParam;

      //Centre the dialog
      CentreWindow (hDlg);

      //Check for uninitialised settings (in which case all fields are
```

```
zero) and
      //set them to some sensible defaults
      if (!Settings->Initialised) {
        Settings->Distance    = 10;
        Settings->Initialised = true;
      }

      //Set the edit control to the current distance and limit the control
to
      //3 characters, i.e. 999cm
      SetDlgItemInt (hDlg, IDC_DISTANCE, Settings->Distance, false);
      SendDlgItemMessage (hDlg, IDC_DISTANCE, EM_LIMITTEXT, 3, 0);
      break;


    case WM_COMMAND :
      //Action depends on the control
      switch (LOWORD(wParam))
      {
        case IDOK :
          //Save the user's entry in the Settings
          Settings->Distance = max (1, GetDlgItemInt (hDlg, IDC_DISTANCE,
NULL, false));

          //Close the dialog, returning IDOK to signal that the settings
need to be
          //saved
          EndDialog (hDlg, IDOK);
          break;


        case IDCANCEL :
          //Close the dialog returning IDCANCEL to signal that the
settings need not be
          //saved
          EndDialog (hDlg, IDCANCEL);
          break;
      }
      break;

    default:
      //We didn't process the message
      return 0;
  }
  //We did process the message
  return 1;
}


/*--------------------------------------------------------------------------
----------

  Name    EditInCentreAnalysisSettings

  Desc    Displays a dialog box where the user can set the distance from
the centre
          for the analysis

  Params  Settings  Pointer to the InCentreAnalysis settings record that
the user
                    will be editing
```

```
  Return  True if the user edited the settings and they should now be saved

---------------------------------------------------------------------------
--------*/

bool EditInCentreAnalysisSettings (LPINCENTREANALYSISSETTINGS Settings)
{
//BEGIN
  //Display the in centre analysis settings dialog, passing the settings
  //pointer as lParam. The dialog will edit the settings in place
  return (DialogBoxParam (hInstance,
                          MAKEINTRESOURCE(IDD_INCENTREANALYSISSETTINGS),
                          NULL,
                          InCentreAnalysisSettingsDlgProc,
                          (LPARAM)Settings) == IDOK);
}

/*-------------------------------------------------------------------------
---------

  Name    InCentreAnalysis

  Desc    Part of the example analysis: This routine determines when the
animal is
          in the centre of the apparatus. "Centre" is defined as within a
certain
          distance of the centre which the user can set using the analysis
settings
          (the default is 10cm)

  Params  Mesg     The type of message ANY-maze is sending to the plug in.
This code
                   handles the "Test" messages: AME_TESTSTART, AME_POSN,
AME_TESTEND
                   and AME_TESTABORT
          wParam   wParam sent with the message - depends on the message
          lParam   lParam sent with the message - depends on the message

  Return  And AME_xxx return code

---------------------------------------------------------------------------
--------*/

int InCentreAnalysis (DWORD   Mesg,
                      WPARAM  wParam,
                      LPARAM  lParam)
{
//Types
  typedef struct {
    double Distance;
    int    AppCentreX;
    int    AppCentreY;
    bool   InCentre;
}INCENTREANALYSISRECORD, *LPINCENTREANALYSISRECORD;

//Vars
  LPAME_TESTDATA             AME_TestData;
  LPAME_POSNDATA             AME_PosnData;
  LPINCENTREANALYSISRECORD   InCentreAnalysisRec;
  LPINCENTREANALYSISSETTINGS Settings;
  double                     d, dx, dy;
  bool                       NowInCentre;
```

```
//BEGIN
  //Action depends on the type
  switch (Mesg)
  {
    case AME_TESTSTART :
      //The lParam points at a AME_TESTDATA record and the wParam is the
      //size of the record. Check the size
      assert (wParam >= sizeof(AME_TESTDATA));
      if (wParam < sizeof(AME_TESTDATA)) return AME_ERROR;

      //Access the AME_TESTDATA record
      AME_TestData = (LPAME_TESTDATA)lParam;

      //Allocate a InCentreAnalysisRecord - if this fails return AME_ERROR
      InCentreAnalysisRec = new INCENTREANALYSISRECORD;
      if (InCentreAnalysisRec == NULL) return AME_ERROR;

      //Calculate and note the centre point of the apparatus
      InCentreAnalysisRec->AppCentreX = (AME_TestData->ApparatusRect.left
+ AME_TestData->ApparatusRect.right) / 2;
      InCentreAnalysisRec->AppCentreY = (AME_TestData->ApparatusRect.top +
AME_TestData->ApparatusRect.bottom) / 2;

      //The passed AME_TestData record includes a pointer to the settings
      //for this analysis. The settings define the distance from the centre
      //that we consider to be the "centre" of the apparatus. Convert this
      //distance to pixels and note it in the InCentreAnalysisRec. If the
      //distance is zero then the user has not specified a distance so
      //use a default of 10cm
      Settings = (LPINCENTREANALYSISSETTINGS)AME_TestData->Settings;
      if (!Settings->Initialised) Settings->Distance = 10;
      d = (double)Settings->Distance;

      //Convert from cm to mm
      d = d * 10;

      //Convert from mm to pixels
      InCentreAnalysisRec->Distance = d * AME_TestData->Scaling;

      //Start by assuming the animal is not in the centre of the apparatus
      InCentreAnalysisRec->InCentre = false;

      //Set Context to the address of the InCentreAnalysisRec
      AME_TestData->Context = (LPVOID)InCentreAnalysisRec;
      return AME_OK;


    case AME_POSN :
      //The lParam points at a AME_POSNDATA record and the wParam is the
      //size of the record. Check the size
      assert (wParam >= sizeof(AME_POSNDATA));
      if (wParam < sizeof(AME_POSNDATA)) return AME_ERROR;

      //Access the AME_POSNDATA record
      AME_PosnData = (LPAME_POSNDATA)lParam;

      //Access the InCentreAnalysisRec
      if (AME_PosnData->Context == NULL) return false;
      InCentreAnalysisRec =
(LPINCENTREANALYSISRECORD)AME_PosnData->Context;
```

```
      //Calculate the distance from the animal's posn to the apparatus
centre
      dx = AME_PosnData->x - InCentreAnalysisRec->AppCentreX;
      dy = AME_PosnData->y - InCentreAnalysisRec->AppCentreY;
      d  = sqrt((dx * dx) + (dy * dy));

      //Is the animal within the required distance of the apparatus centre
      NowInCentre = (d < InCentreAnalysisRec->Distance);

      //Has the animal's "InCentre" state changed
      if (NowInCentre != InCentreAnalysisRec->InCentre)
      {
        //Yes, so return a suitable result
        AME_PosnData->ResultType  = RT_ONOFF;
        AME_PosnData->ResultValue = NowInCentre ? 1 : 0;

        //Update the in centre state
        InCentreAnalysisRec->InCentre = NowInCentre;
      }
      else
      {
        //We're not returning a result
        AME_PosnData->ResultType = RT_NONE;
      }

      //Processed successfully
      return AME_OK;


    case AME_TESTEND   :
    case AME_TESTABORT :
      //The lParam is our context value
      if (lParam == NULL) return false;
      InCentreAnalysisRec = (LPINCENTREANALYSISRECORD)lParam;

      //Delete the InCentreAnalysisRec
      delete InCentreAnalysisRec;

      //Done
      return AME_OK;


    default :
      //What's this?
      assert (false);
      return AME_NOTIMPLEMENTED;
  }
}

/*---------------------------------------------------------------------------
---------

  Name    ANYmazeExtension_Main

  Desc    When starting up ANY-maze will search the same directory as the
executable
          file for any DLLs. For each one it finds it will call
GetProcAddress for
          a routine with the name "ANYmazeExtension_Main".
          DLLs which export this routine will be considered to be ANY-maze
plug-ins.
```

```
          This routine is the interface between ANY-maze and the plug-in.
ANY-maze
          will call this routine passing it a Mesg and a w- and l- param
(much like
          windows message processing).
          This routine should process the obligatory messages and can
implement as
          many of the options messages as it wishes. For details of each
message
          and its parameters refer to the ANY-maze help.

  Params  guid        Pointer to a GUID that identifies the analysis to
which the
                      message is addressed. When the message is
AME_ENUMERATE this
                      param will hold the guid if the analysis returned by
the
                      prior call, or it will be GUID_NULL when the
enumeration is
                      starting.
          Mesg        One of the AME_xxx message constants

          wParam      Depends on the message, but typically used for
integer
                      parameters.
          lParam      Depends on the message, but typically a pointer to a
buffer

  Return  One of the AME_xxx return constants. For any messages not
implmented for
          a certain analysis the return should be AME_NOTIMPLEMENTED

---------------------------------------------------------------------------
--------*/

extern "C"
__declspec(dllexport) int ANYmazeExtension_Main (LPGUID guid,
                                                 DWORD  Mesg,
                                                 WPARAM wParam,
                                                 LPARAM lParam)
{
//Vars
  int RscID;

//BEGIN
  //Action depends on the Mesg
  switch (Mesg)
  {
    case AME_ENUMERATE :
      //Simply enumerate the GUIDs of the analysis we can perform,
returning the
      //next GUID in the buffer pointed at by lParam. When enumeration
starts we
      //are passed a null GUID
      if (*guid == GUID_NULL)
        *((LPGUID)lParam) = GUID_INCENTREANALYSIS;
      else if (*guid == GUID_INCENTREANALYSIS)
        *((LPGUID)lParam) = GUID_POSNEXPORT;
      else
        return AME_NOTIMPLEMENTED;
      return AME_OK;
```

```
    case AME_GETNAME        :
    case AME_GETHELPTITLE   :
    case AME_GETHELPINTRO   :
    case AME_GETHELPDETAILS :
      //These all return strings, which we load from the DLL's resources.
The
      //strings could be hard coded here instead
      if ((Mesg == AME_GETNAME) && (*guid == GUID_INCENTREANALYSIS))
        RscID = IDS_INCENTREANALYSIS_HELP_NAME;
      else if ((Mesg == AME_GETNAME) && (*guid == GUID_POSNEXPORT))
        RscID = IDS_POSNEXPORT_HELP_NAME;
      else if ((Mesg == AME_GETHELPTITLE) && (*guid ==
GUID_INCENTREANALYSIS))
        RscID = IDS_INCENTREANALYSIS_HELP_TITLE;
      else if ((Mesg == AME_GETHELPTITLE) && (*guid == GUID_POSNEXPORT))
        RscID = IDS_POSNEXPORT_HELP_TITLE;
      else if ((Mesg == AME_GETHELPINTRO) && (*guid ==
GUID_INCENTREANALYSIS))
        RscID = IDS_INCENTREANALYSIS_HELP_INTRO;
      else if ((Mesg == AME_GETHELPINTRO) && (*guid == GUID_POSNEXPORT))
        RscID = IDS_POSNEXPORT_HELP_INTRO;
      else if ((Mesg == AME_GETHELPDETAILS) && (*guid ==
GUID_INCENTREANALYSIS))
        RscID = IDS_INCENTREANALYSIS_HELP_DETAILS;
      else if ((Mesg == AME_GETHELPDETAILS) && (*guid == GUID_POSNEXPORT))
        RscID = IDS_POSNEXPORT_HELP_DETAILS;
      else
        return AME_ERROR;

      //Load the resource
      return ((LoadString (hInstance, RscID, (LPSTR)lParam, (int)wParam)
!= 0) ? AME_OK : AME_ERROR);


    case AME_GETRESULTTYPE :
      //Return the result type in the unsigned char pointed to by lParam
      if (*guid == GUID_INCENTREANALYSIS)
        *((unsigned char*)lParam) = RT_ONOFF;
      else if (*guid == GUID_POSNEXPORT)
        *((unsigned char*)lParam) = RT_NONE;
      else
        return AME_ERROR;
      return AME_OK;


    case AME_EDITSETTINGS :
      //Edit the settings of the analysis whose GUID is passed. wParam is
size of a buffer
      //pointed to by lParam which holds the current settings. If return
is false
      //which the title should be returned
      if (*guid == GUID_INCENTREANALYSIS)
      {
        //Check the passed buffer is no larger than the settings we
maintain
        //for in centre analysis
        assert (wParam >= sizeof (INCENTREANALYSISSETTINGS));
        return (EditInCentreAnalysisSettings
((LPINCENTREANALYSISSETTINGS)lParam) ? AME_OK : AME_CANCELLED);
      }
      else if (*guid == GUID_POSNEXPORT)
```

```
      {
        //Posn export has no settings
        return AME_NOTIMPLEMENTED;
      }
      else
      {
        //What's this?
        return AME_ERROR;
      }
      break;


    case AME_TESTSTART :
    case AME_POSN      :
    case AME_TESTEND   :
    case AME_TESTABORT :
      //Pass to analysis specific routine for processing
      if (*guid == GUID_INCENTREANALYSIS)
        return InCentreAnalysis (Mesg, wParam, lParam);
      else if (*guid == GUID_POSNEXPORT)
        return PosnExport (Mesg, wParam, lParam);
      else
        return AME_ERROR;
      break;


    default :
      //What's this message?
      return AME_NOTIMPLEMENTED;
  }
}

/*-------------------------------------------------------------------------
---------

  Name    DllMain

  Desc    Main entry point for the DLL. Just notes the instance handle

  Params  Std

  Return  Std

---------------------------------------------------------------------------
--------*/

BOOL APIENTRY DllMain( HMODULE hModule,
                       DWORD   ul_reason_for_call,
                       LPVOID  lpReserved)
{
//BEGIN
  switch (ul_reason_for_call)
  {
    case DLL_PROCESS_ATTACH : //Note the module handle
                              hInstance = hModule;
                              break;
    case DLL_THREAD_ATTACH  :
    case DLL_THREAD_DETACH  :
    case DLL_PROCESS_DETACH : //Nothing to do
                              break;
  }
```

```
  //Done
  return TRUE;
}

//===========================================================================
==========
```

The PlugIn.h header file

The `PlugIn.h` header file defines the link between ANY-maze and a plug-in. Although this file is a C/C++ header, it is very short and doubtless authors will find it easy to convert to other languages if required.

The listing

```
/*----------------------------------------------------------------------------
--------

  Name        : PlugIn.h

  Description : Header file for the ANY-maze Plug-in API which defines
                how ANY-maze plug-ins interface to ANY-maze.

  History
  -------
  7/2/2008  CPL   Created file

----------------------------------------------------------------------------
--------*/

//Messages
#define AME_ENUMERATE       1
#define AME_GETNAME         2
#define AME_GETHELPTITLE    3
#define AME_GETHELPINTRO    4
#define AME_GETHELPDETAILS  5
#define AME_GETRESULTTYPE   6
#define AME_EDITSETTINGS    7
#define AME_TESTSTART       8
#define AME_POSN            9
#define AME_TESTEND        10
#define AME_TESTABORT      11


//Return codes
#define AME_OK              0
#define AME_NOTIMPLEMENTED  1
#define AME_ERROR           2
#define AME_CANCELLED       3


//Result types
#define RT_NONE            -1
#define RT_ONOFF            0
#define RT_VALUE            1
#define RT_CUMULATIVE       2


//Structure used to pass test data for a AME_TESTSTART message
typedef struct {
  DWORD     AnimalNum;
```

```
  char      AnimalID[80];
  DWORD     TrialNum;
  RECT      ApparatusRect;
  double    Scaling;
  LPVOID    Settings;
  DWORD     SettingsSize;
  LPVOID    Context;
}AME_TESTDATA, *LPAME_TESTDATA;

//Structure used to pass position data for a AME_POSN message
typedef struct {
  LPVOID    Context;
  DWORD     Time_xy;
  int       x;
  int       y;
  DWORD     Time_HeadTail;
  int       hx;
  int       hy;
  int       tx;
  int       ty;
  int       cx;
  int       cy;
  bool      Hidden;
  RECT      AnimalArea;
  int       ResultType;
  int       ResultValue;
}AME_POSNDATA, *LPAME_POSNDATA;
```

Plug-in API reference

Functions

- ANYmazeExtension_Main

Messages

- AME_INITIALISE
- AME_TERMINATE
- AME_ENUMERATE
- AME_ENABLE
- AME_GETNAME
- AME_GETHELPTITLE
- AME_GETHELPINTRO
- AME_GETHELPDETAILS
- AME_GETRESULTTYPE
- AME_EDITSETTINGS
- AME_APPLYSETTINGS
- AME_TESTSTART
- AME_POSN
- AME_TESTPAUSE
- AME_TESTEND
- AME_TESTABORT
- AME_PERFORMACTION

Structures

- AME_TESTDATA
- AME_POSNDATA
- AME_ACTIONDATA

Constants

Return codes

- AME_OK
- AME_NOTIMPLEMENTED
- AME_ERROR
- AME_CANCELLED

Result types

- RT_NONE
- RT_ONOFF
- RT_VALUE
- RT_CUMULATIVE
- RT_ACTION

ANYmazeExtension_Main

`ANYmazeExtension_Main` is a routine defined in a plug-in which is used as the link between ANY-maze and a plug-in. All plug-ins must include this routine.

Syntax

```
__declspec(dllexport) int ANYmazeExtension_Main (LPGUID guid,
                                                 DWORD  Mesg,
                                                 WPARAM wParam,
                                                 LPARAM lParam)
```

Parameters

guid

Pointer to a GUID that identifies the plug-in to which the message is addressed, except when the message is AME_ENUMERATE.

Mesg

One of the AME_xxx message constants.

wParam

Depends on the message, but typically used for integer parameters.

lParam

Depends on the message, but typically a pointer to a buffer.

Return

One of the AME_xxx return codes:

AME_OK	The message was processed successfully
AME_NOTIMPLEMENTED	The plug-in does not implement the functionality of the passed message. For example, a plug-in which has no settings should return

	AME_NOTIMPLEMENTED to an AME_EDITSETTINGS message. This return code is also used to end enumeration (see remarks).
AME_ERROR	An error occurred during the processing of the message. ANY-maze will log the error in the ANYmazeLog.txt file.
AME_CANCELLED	The message processing was cancelled by the user. For example, if the user closes a settings window by clicking a *Cancel* button, the plug-in should return AME_CANCELLED. This return code is also used as a response to AME_TESTEND, to signal that any results from this plug-in recorded by ANY-maze for the test that has just ended should be deleted.

Remarks

This routine should be exported from your plug-in DLL. How you ensure that the routine is exported will depend on your linker and your preference (many linkers provide different ways to export routines). In Visual C++, the routine can be exported by declaring it as __declspec(dllexport).

Care should be taken with the naming of this routine, as linkers will normally *decorate* the name. You should refer to your linker's help file for details on how to prevent name decoration, but for Visual C++, you can declare the function for C linkage by using extern "C" as in the following example:

```
extern "C"
__declspec(dllexport) int ANYmazeExtension_Main (LPGUID guid,
                                                  DWORD  Mesg,
                                                  WPARAM wParam,
                                                  LPARAM lParam)
{
.... your code here...
}
```

An easy way to check that your DLL is exporting this routine correctly is to use the DEPENDS.EXE utility from Microsoft and check the exports list; the routine should be shown with the name ANYmazeExtension_Main i.e. without any name decoration.

When the message passed to this routine is AME_ENUMERATE, the guid parameter will initially point at the value GUID_NULL; the DLL should use this to determine that enumeration is starting and should return the GUID of the first plug-in it implements. ANY-maze will then send a second AME_ENUMERATE message, with the guid parameter pointing at the GUID that the DLL just returned; the DLL should detect this and return the GUID of the *next* plug-in it implements; and so on until the DLL has returned the GUIDs of all the plug-ins it implements. At this point, it should return AME_NOTIMPLEMENTED to end the enumeration. See Step-by-step instructions for more details, and an example implementation of AME_ENUMERATE processing.

ANY-maze help topic H0925

AME_INITIALISE message

The `AME_INITIALISE` message is used to initialise a plug-in implemented by a plug-in DLL. Note that only one `AME_INITIALISE` message will be sent to your DLL, irrespective of the number of plug-ins that it implements.

Parameters

guid

Not used - always NULL.

wParam

This is the version of ANY-maze that is being used. Versions are coded as a DWORD value, with the bytes (high to low) as, for example, 0,4,7,7 for version 4.77. If your plug-in will only work with a certain version of ANY-maze, then you should check this value and return `AME_ERROR` if the version is inappropriate for your plug-in.

Note that this test can also be performed for each plug-in in your DLL individually, using the wParam of `AME_ENUMERATE`.

lParam

Handle of the ANY-maze application window.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`
- `AME_NOTIMPLEMENTED` (see below)

If initialisation is performed successfully, or you don't need to perform any initialisation, return `AME_OK`.

If an error occurs, return `AME_ERROR`. Note that returning `AME_ERROR` will cause ANY-maze to unload the plug-in DLL and ignore the plug-in entirely (it will be like the DLL doesn't exist).

For compatibility reasons, it is also valid to return `AME_NOTIMPLEMENTED`, but this is not recommended - even if you perform no initialisation, you should still return `AME_OK`.

Remarks

In most cases, you can just return `AME_OK` to this message and do nothing else. However, you may want to take the opportunity to do things like create a thread for your plug-in. Note that you should perform initialisation when you receive this message, and NOT as part of DLL_PROCESS_ATTACH. This is because Windows restricts the operations which can be performed during DLL_PROCESS_ATTACH - see Microsoft's Best Practices for Creating DLLs for full details.

Example code for a DLL which creates a thread as part of initialisation.

```
case AME_INITIALISE:
  //The wParam holds the handle of the main ANY-maze window; note it
  hMainWnd = (HWND)wParam;

  //Create our thread. If this fails we can't run so return
  //AME_ERROR - the plug-in will then be unloaded by ANY-maze
  hThread = (HANDLE)_beginthreadex (NULL, 0, MyThreadProc, NULL, 0,
NULL);
  if (hThread == NULL) {
    return AME_ERROR;
  }

  //We initialised successfully
  return AME_OK;
```

AME_TERMINATE message

The `AME_TERMINATE` message is used to terminate a plug-in. It is sent to all plug-in DLLs during ANY-maze shutdown, just before the DLL is unloaded. Note that only one `AME_TERMINATE` message will be sent to your DLL, irrespective of the number of plug-ins that it implements.

Parameters

guid

Not used - always NULL.

wParam

Not used - always 0.

lParam

Not used - always 0.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`

It doesn't actually matter what you return, but it is nevertheless good practice to return `AME_OK` if your termination was performed successfully and `AME_ERROR` otherwise.

Remarks

In most cases you can just return `AME_OK` to this message and do nothing else. However, you may need to perform some clean-up, especially if you processed `AME_INITIALISE`. For example, if your `AME_INITIALISE` processing created a thread, then this would be the time to terminate it.

AME_ENUMERATE message

The AME_ENUMERATE message is used to enumerate the plug-ins implemented by a plug-in DLL. *Remember, a single DLL can implement multiple plug-ins.*

Parameters

guid

Pointer to GUID_NULL when enumeration is starting. Pointer to the previous GUID returned when enumeration is continuing. See Remarks section, below.

wParam

This is the version of ANY-maze that is being used. Versions are coded as a DWORD value with the bytes (high to low) as, for example, 0,4,7,7 for version 4.77. If your plug-in will only work with a certain version of ANY-maze, then you should check this value and return AME_NOTIMPLEMENTED if the version is inappropriate for your plug-in.

lParam

Pointer to a GUID. The DLL code should set the GUID pointed to by this parameter to the GUID of the next plug-in it implements.

```
*((LPGUID)lParam) = GUID_MYPLUGIN;
```

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR
- AME_NOTIMPLEMENTED

If the code has set the GUID pointed at by lParam, then return AME_OK.

If the code has returned all the GUIDs implemented by the DLL, then return AME_NOTIMPLEMENTED.

If an error occurs, return AME_ERROR.

Remarks

When an enumeration is starting, the `guid` parameter will point to GUID_NULL. The DLL code should detect this, set the GUID pointed at by the lParam to the GUID of the *first* plug-in it implements, and return `AME_OK`.

ANY-maze will then send this message to the plug-in again, this time with the `guid` parameter set to the GUID returned by the previous message. The DLL code should detect this, set the GUID pointed at by the lParam to the GUID of the *next* plug-in it implements, and return `AME_OK`.

When the DLL code has returned the GUID of all the plug-ins it implements, it should return `AME_NOTIMPLEMENTED` to end the enumerations.

Example code for a DLL which implements a single plug-in.

```
if (*guid == GUID_NULL)
{
  //Enumeration is starting - return our GUID
  *((LPGUID)lParam) = GUID_MYPLUGIN;
  return AME_OK;
}
else
{
  //We only implement a single plug-in, so we're done
  return AME_NOTIMPLEMENTED;
}
```

Example code for a DLL which implements three plug-ins.

```
if (*guid == GUID_NULL)
  *((LPGUID)lParam) = GUID_1ST_PLUGIN;
else if (*guid == GUID_1ST_PLUGIN)
  *((LPGUID)lParam) = GUID_2ND_PLUGIN;
else if (*guid == GUID_2ND_PLUGIN)
  *((LPGUID)lParam) = GUID_3RD_PLUGIN;
else
  return AME_NOTIMPLEMENTED;
return AME_OK;
```

AME_ENABLE message

The AME_ENABLE message is sent to a plug-in when the user enables or disables the plug-in in Protocol > Analysis and Results > Analysis Options > ANY-maze plug-ins. In most cases, you will not need to do anything and can simply return AME_NOTIMPLEMENTED. However, there may be circumstances where you wish to processes this message - for example, if your plug-in is communicating with another program, you might need to tell the other program that the plug-in has been disabled within ANY-maze.

Parameters

guid

Pointer to the GUID of the plug-in which has been enabled/disabled.

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Enabled flag. TRUE if the plug-in has been enabled, FALSE if the plug-in has been disabled.

lParam

Not used; always 0.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR
- AME_NOTIMPLEMENTED

If the message is processed successfully, then return AME_OK.

If an error occurs, return AME_ERROR.

If your plug-in does not process this message, you can return AME_NOTIMPLEMENTED.

Remarks

This message is also sent to a plug-in when an ANY-maze experiment is opened, to tell it whether it is enabled or disabled in the experiment's protocol. It is also sent to a plug-in when an experiment is closed - in this case the enabled flag will always be FALSE.

There is no obligation for a plug-in to 'track' its enabled state; for example a plug-in could consider itself to be enabled all the time, as ANY-maze won't send messages such as `AME_TESTSTART`, `AME_POSN` or `AME_TESTEND` to a disabled plug-in.

AME_GETNAME message

The AME_GETNAME message is used to get a user-readable name for the plug-in. This is the name that will be displayed to the user, so it should be descriptive of what the plug-in does; however, you should try to keep the name concise as it is limited to 47 characters.

Parameters

guid

Pointer to the GUID of the plug-in whose name is being requested. If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

The size of the buffer pointed to by lParam, in bytes.

lParam

Pointer to a character buffer in which the name should be returned as a NULL-terminated ASCII string.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR

If the code has written the name to the buffer, then return AME_OK.

If an error occurs, return AME_ERROR.

Remarks

In the following code, the DLL implements a single plug-in, whose name is held in the DLL's resource file.

```
//Check the GUID - not strictly necessary, but good practice
if (*guid != GUID_MYPLUGIN) return AME_ERROR;

//Load the name from the resource file
```

```
if (LoadString (hInstance, IDS_MYPLUGINNAME, (LPSTR)lParam,
(int)wParam) == FALSE)
  return AME_ERROR;
else
  return AME_OK;
```

AME_GETHELPTITLE message

The `AME_GETHELPTITLE` message is used to get the title of the help message for a plug-in.

Help titles are limited to 255 characters.

Parameters

guid

Pointer to the GUID of the plug-in whose help title is being requested. If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

The size of the buffer pointed to by lParam, in bytes.

lParam

Pointer to a character buffer in which the help title should be returned as a NULL-terminated ASCII string.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`
- `AME_NOTIMPLEMENTED`

If the code has written a help title to the buffer, then return `AME_OK`.

If the plug-in does not have a help message, then return `AME_NOTIMPLEMENTED`.

If an error occurs, return `AME_ERROR`.

Remarks

Implementing a help message for your plug-in is optional, but highly recommended.

The help title is shown at the top of the help message, and is usually the same as the name of the plug-in.

AME_GETHELPINTRO message

The `AME_GETHELPINTRO` message is used to get the introductory text of the help message for a plug-in.

Introductory text is limited to 1023 characters, and can include *mark-up* - see the Remarks section below.

Parameters

guid

Pointer to the GUID of the plug-in whose introductory text is being requested. If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

The size of the buffer pointed to by lParam, in bytes.

lParam

Pointer to a character buffer, in which the introductory text should be returned as a NULL-terminated ASCII string.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`
- `AME_NOTIMPLEMENTED`

If the code has written text to the buffer, then return `AME_OK`.

If the plug-in does not have a help message, then return `AME_NOTIMPLEMENTED`.

If an error occurs, return `AME_ERROR`.

Remarks

Implementing a help message for your plug-in is optional, but highly recommended.

The introductory text is shown under a title of 'Introduction'. We suggest that for this text, you write a brief description of your plug-in and its purpose.

This text can include the following *mark-up*:

`\para`	Starts a new paragraph
`\i1`	Switches italic on
`\i0`	Switches italic off
`\u1`	Switches underline on
`\u0`	Switches underline off
`\e1`	Switches bold on
`\e0`	Switches bold off
`\bullet`	Start a new bulleted item in a bullet list

Note that the '\' character is the C/C++ *escape* character, so if you type it into a string in C/C++ code, it will be taken to mean that you're *escaping* the following character - so when typing in '`\para`' you should actually enter '`\\para`', as the escaped '\' character is itself.

Also note that CR or LF characters will not cause new paragraphs in the help message - use `\para` to do this.

AME_GETHELPDETAILS message

The `AME_GETHELPDETAILS` message is used to get the details text of the help message for a plug-in.

Details text is limited to 64K (less 1) characters and can include *mark-up* - see the Remarks section below.

Parameters

guid

Pointer to the GUID of the plug-in whose details text is being requested. If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

The size of the buffer pointed to by lParam, in bytes.

lParam

Pointer to a character buffer in which the details text should be returned as a NULL-terminated ASCII string.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`
- `AME_NOTIMPLEMENTED`

If the code has written text to the buffer, then return `AME_OK`.

If the plug-in does not have a help message, then return `AME_NOTIMPLEMENTED`.

If an error occurs, return `AME_ERROR`.

Remarks

Implementing a help message for your plug-in is optional, but highly recommended.

The details text is shown under a title of 'Details'. Here you should include full details of your plug-in - what it does, how it does it, what results it returns, what settings it has (if any), etc.

This text can include the following *mark-up*:

`\para`	Starts a new paragraph
`\i1`	Switches italic on
`\i0`	Switches italic off
`\u1`	Switches underline on
`\u0`	Switches underline off
`\e1`	Switches bold on
`\e0`	Switches bold off
`\bullet`	Start a new bulleted item in a bullet list

Note that the '\' character is the C/C++ *escape* character, so if you type it into a string in C/C++ code, it will be taken to mean that you're *escaping* the following character - so when typing in '`\para`' you should actually enter '`\\para`', as the escaped '\' character is itself.

Also note that CR or LF characters will not cause new paragraphs in the help message - use `\para` to do this.

AME_GETRESULTTYPE message

The AME_GETRESULTTYPE message is used to get the type of result a plug-in provides. This is one of the RT_xxx values (see Remarks section below). If your plug-in does not provide any results, then you should return RT_NONE.

Parameters

guid

Pointer to the GUID of the plug-in whose result type is being requested. If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Not used - always 0.

lParam

Pointer to an unsigned char (i.e. a single byte) which should be set to the result type the plug-in provides. See the Remarks section for possible result types.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR

If an error occurs, return AME_ERROR, otherwise return AME_OK.

Note that you must return a result TYPE. If your plug-in does not provide any results, then set the result type to RT_NONE and return AME_OK.

Remarks

The result types are:

RT_NONE	The plug-in does not provide any result.
RT_ONOFF	The plug-in provides a binary state result, either on or off.
RT_VALUE	The plug-in provides a result which is a specific value at a specific moment in time.

RT_CUMULATIVE The plug-in provides a result which is cumulative, the sum of all the results being the result for the overall test.

RT_ACTION The plug-in doesn't provide a result as such; rather it can perform some type of action. See RT_ACTION for details.

Example

```
case AME_GETRESULTTYPE :
  //Return the result type in the unsigned char pointed to by lParam
  *((unsigned char*)lParam) = RT_ONOFF;
  return AME_OK;
```

AME_EDITSETTINGS message

The `AME_EDITSETTINGS` message is sent to a plug-in as a request for it to display some type of interface for the user to edit the plug-in's settings. Typically, this is a dialogue box.

Plug-ins do not have to implement any settings, and so you can simply return `AME_NOTIMPLEMENTED` to this message if desired.

Parameters

guid

Pointer to the GUID of the plug-in whose settings are to be edited. If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Size of the settings buffer pointed to by lParam, in bytes.

lParam

Pointer to a buffer which holds the plug-in's settings. When first passed to a plug-in, this buffer will be filled with zeros. See Remarks section.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_CANCELLED`
- `AME_ERROR`
- `AME_NOTIMPLEMENTED`

If the settings have been edited and should now be saved by ANY-maze, then return `AME_OK`.

If the user cancelled the editing of the settings (for example by clicking a *Cancel* button), return `AME_CANCELLED`.

If the plug-in does not have any settings, then return `AME_NOTIMPLEMENTED`.

If an error occurs, return `AME_ERROR`.

Remarks

When first passed to a plug-in, the settings buffer will be filled with zeros. The plug-in should detect this and set the settings to default values. An easy way to detect uninitialised settings is to include a boolean 'Initialised' value as the first field of your settings record. Thus if Settings->Initialised is false, you know the settings are uninitialised - so you can set them to default values and then set the Settings->Initialised flag to true.

The size of the Settings buffer is 48 bytes (which will be the value passed in wParam); you don't need to use all of this for your settings, although you can't use more. It is a good idea to check the size of your settings record against the wParam, to avoid a buffer overflow.

ANY-maze saves the settings for a plug-in as part of the protocol. Thus the settings in different protocols can be different.

It is not a good idea to save your settings in the Windows Registry, as doing so will mean that the settings will be the same for all experiments - unless, of course, this is exactly what you want. If you do need such 'global' settings, then a good place to store them would be in a sub-key of `HKEY_LOCAL_MACHINE\SOFTWARE\ANY-maze`.

AME_APPLYSETTINGS message

The `AME_APPLYSETTINGS` message is sent to a plug-in whenever the plug-in's settings (if it has any) are altered by the user. This gives the plug-in a chance to update its state based on the new settings. However, in many cases this is unnecessary as the settings are sent to the plug-in as part of the `AME_TESTSTART` message, and the plug-in may find it easier to process settings at this point.

Parameters

guid

Pointer to the GUID of the plug-in whose settings are to be applied.

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

The size of the settings buffer pointed to by lParam, in bytes.

lParam

Pointer to a buffer holding the plug-in's settings.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`
- `AME_NOTIMPLEMENTED`

If the message is processed successfully, then return `AME_OK`.

If an error occurs, return `AME_ERROR`.

If your plug-in does not process this message, you can return `AME_NOTIMPLEMENTED`.

Remarks

This message is also sent to a plug-in when an ANY-maze experiment is opened, assuming the plug-in is enabled in the experiment.

This message is most useful for processing settings which it is inappropriate to process during

`AME_TESTSTART`. For example, a plug-in which communicates with another program may include a setting for the name of the computer where the other program is running. To open communication across the network with this program may be quite slow (and could even mean a log-on window opens) - performing such processing when a test starts would clearly be a bad idea.

ANY-maze help topic H0936

AME_TESTSTART message

The processing of this message is time critical. Try to avoid any processing which will take more than a few milliseconds to perform.

The `AME_TESTSTART` message is sent to a plug-in when a test is starting.

Parameters

guid

Pointer to the GUID of the plug-in which is being notified of the test start. All plug-ins that the user has activated (in the protocol) will be sent this message.

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Size of the `AME_TESTDATA` record pointed to by lParam, in bytes.

lParam

Pointer to an `AME_TESTDATA` record. See Remarks section for details.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`

If the message is processed successfully, then return `AME_OK`.

If an error occurs, return `AME_ERROR`. Note that doing so will mean that the plug-in is excluded from the test, i.e. no `AME_POSN` messages will be sent to it, nor will it receive an `AME_TESTEND` or `AME_TESTABORT` message at the end of the test. Thus on failure, you should perform all necessary cleaning-up (for example freeing memory) before returning.

Remarks

A plug-in should use this message to initialise itself ready for a test. The `AME_TESTDATA` record includes a pointer to a Settings buffer, which will hold the settings for the test. These may be uninitialised (filled with zeros), in which case the plug-in should detect this and use default values.

As part of the processing of this message, the plug-in should allocate a record in which it will store all the data it needs in order to process the test - this will be the plug-in's 'Context record'.

If there is any data passed in the AME_TESTDATA record (such as settings) which the plug-in will need during the processing of the test, then it should copy this data to its context record.

If the plug-in will require any 'static' variables for its processing, then it should include these in its context record too. For example, if the plug-in will be determining the state of something, it will probably want to record the 'Current' state in the context record. Don't use static variables in the processing of AME_TESTSTART, AME_POSN, AME_TESTEND or AME_TESTABORT messages, as this processing must be thread-safe.

A pointer to the context record should be passed back to ANY-maze in the AME_TESTDATA->Context field.

Remember, ANY-maze is multi-threaded and your plug-in may be used to analyse data from multiple tests all running simultaneously. By recording everything about the analysis of a test in a context record, you avoid having to worry (very much) about multi-threading issues, as ANY-maze will simply pass the correct context for a test to your plug-in in each AME_POSN and AME_TESTEND or AME_TESTABORT message.

Below is an example of the processing for this message:

```
case AME_TESTSTART:
  //The lParam points at a AME_TESTDATA record and the wParam is the
  //size of the record. Check the size
  assert (wParam >= sizeof(AME_TESTDATA));
  if (wParam < sizeof(AME_TESTDATA)) return AME_ERROR;

  //Access the AME_TESTDATA record
  AME_TestData = (LPAME_TESTDATA)lParam;

  //Allocate our context record - if this fails return AME_ERROR
  MyPlugInContextRec = new MYPLUGINCONTEXTREC;
  if (MyPlugInContextRec == NULL) return AME_ERROR;

  //The passed AME_TestData record includes a pointer to the settings
  //for this analysis. Check whether they are initialised and if not
  //set defaults
  Settings = (LPMYPLUGINSETTINGS)AME_TestData->Settings;
  if (!Settings->Initialised) SetDefaultSettings (Settings);

  //Note the settings in the context record
  MyPlugInContextRec->Settings = *Settings;

  //Initialise other fields of the context record ready for analysis
  //.... exactly what you'll do here depends on your implementation.....

  //Return the address of MyPlugInContextRec so that ANY-maze can pass
it
  //back to us in AME_POSN messages
```

```
    AME_TestData->Context = (LPVOID)MyPlugInContextRec;
    return AME_OK;
```

AME_POSN message

The processing of this message is time-critical. See Remarks section for details.

The AME_POSN message is sent to a plug-in during a test. One message is sent to each active plug-in for each detected position of the animal.

Parameters

guid

Pointer to the GUID of the plug-in which is being notified of the position. All plug-ins that the user has activated (in the protocol) will be sent this message (except for any plug-in which returned AME_ERROR from its AME_TESTSTART processing.)

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Size of the AME_POSNDATA record pointed to by lParam, in bytes.

lParam

Pointer to an AME_POSNDATA record. See Remarks section for details.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR

If the message is processed successfully, then return AME_OK.

If an error occurs, return AME_ERROR. Note that your plug-in will continue to receive AME_POSN messages during the rest of the test, and will also receive an AME_TESTEND or AME_TESTABORT message at the end of the test. Therefore, if an error occurs which means the results of your analysis are no longer valid, you should flag this in your context record and simply return immediately (with a return value of AME_ERROR) on all subsequent AME_POSN messages. At the test end, you should return AME_CANCELLED to force any results that you returned prior to the error to be deleted.

Remarks

The AME_POSNDATA record includes a Context field. This points at the context value you returned during AME_TESTSTART processing.

If your analysis returns a result, you should write the result value to the ResultValue field of the AME_POSNDATA record and the result type to the ResultType field. Here the ResultType should agree with the result type you return from AME_GETRESULTTYPE messages.

If your plug-in does not return a result, then you should set the ResultType field to RT_NONE and return AME_OK.

The processing of this message is time critical. ANY-maze can perform simultaneous analysis of up to 40 video images, and can be receiving images at 30 frames per second. Thus, in an extreme case, it may be required to analyse 1200 frames per second. This gives under a millisecond for each frame. Realistically, in this case, ANY-maze will probably start to *drop frames* and only actually process 15 frames per second from each image. Nevertheless, there's little time for the analysis of each frame, so you should try to ensure that the analysis you perform is as fast as possible. Of course, you may know that no one will use your plug-in with more than (for example) 2 pieces of apparatus, and therefore ANY-maze will need to analyse, at most, 60 frames per second, giving 16ms per frame; or you may choose to specify that your analysis should only be used on a single piece of apparatus, thus giving you time to perform complex analysis. However, whatever limits may apply, you should always bear in mind that the analysis performed by a plug-in occurs in real-time and should be as efficient as possible.

AME_TESTPAUSE message

The AME_TESTPAUSE message is sent to each active plug-in when a test is paused or unpaused.

Parameters

guid

Pointer to the GUID of the plug-in which is being notified of the pause. All plug-ins that the user has activated (in the protocol) will be sent this message (except for any plug-in which returned AME_ERROR from its AME_TESTSTART processing.)

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

TRUE if the test has been paused, FALSE if the test has been unpaused.

lParam

The 'Context' value returned in the AME_TESTDATA Context field by the AME_TESTSTART processing.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR

If the message is processed successfully, then return AME_OK.

If an error occurs, you should return AME_ERROR. However, this won't have any effect on the testing, and ANY-maze will continue to send messages to your plug-in as the test proceeds.

Remarks

Most plug-ins can ignore this message. It is mainly useful for plug-ins which communicate with some other program, as it allows them to notify the other program that the test has been paused.

AME_TESTEND message

The AME_TESTEND message is sent to each active plug-in when a test ends successfully. See AME_TESTABORT for the message sent when a test is aborted.

Parameters

guid

Pointer to the GUID of the plug-in which is being notified of the test end. All plug-ins that the user has activated (in the protocol) will be sent this message (except for any plug-in which returned AME_ERROR from its AME_TESTSTART processing.)

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Not used; always 0.

lParam

The 'Context' value returned in the AME_TESTDATA Context field by the AME_TESTSTART processing.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR
- AME_CANCELLED

If the message is processed successfully, then return AME_OK.

If your plug-in detected some type of error during the test, and wants any results recorded for it to be deleted, return AME_CANCELLED. Note that this will only cause the results of *this* plug-in to be deleted. Other test results, including any from other plug-ins, will be retained.

If an error occurs, return AME_ERROR. In this case, ANY-maze will *not* delete any results recorded for the plug-in during the test.

Remarks

A plug-in should use this message to free any resources it is using, and to delete its context record (assuming it allocated one).

AME_TESTABORT message

The AME_TESTABORT message is sent to each active plug-in when a test is aborted - usually by the user ending the test and choosing *not* to save the test results.

Parameters

guid

Pointer to the GUID of the plug-in which is being notified of the test end. All plug-ins that the user has activated (in the protocol) will be sent this message (except for any plug-in which returned AME_ERROR from its AME_TESTSTART processing.)

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Not used; always 0.

lParam

The 'Context' value returned in the AME_TESTDATA Context field by the AME_TESTSTART processing.

Return value

The only valid return codes from this message are:

- AME_OK
- AME_ERROR

If the message is processed successfully, then return AME_OK.

If an error occurs, return AME_ERROR.

Remarks

A plug-in should use this message to free any resources it is using, and to delete its context record (assuming it allocated one). Note that all results of the test will be deleted (because the test is being aborted), so (unlike AME_TESTEND processing) there is no need to return AME_CANCELLED if your plug-in encountered some type of error during the test.

AME_PERFORMACTION message

The processing of this message is time critical. See Remarks section for details.

The `AME_PERFORMACTION` message is sent to an Action plug-in to tell it to perform its action.

Parameters

guid

Pointer to the GUID of the plug-in whose action is being invoked. You should check this value to ensure that this message is for your plug-in.

If a DLL only implements a single plug-in, it can ignore this parameter - although it is good practice to always check it.

wParam

Holds the size of the `AME_ACTIONDATA` record pointed at by lParam.

lParam

Points at an `AME_ACTIONDATA` record.

Return value

The only valid return codes from this message are:

- `AME_OK`
- `AME_ERROR`

If the message is processed successfully, then return `AME_OK`.

If an error occurs, return `AME_ERROR`.

Note that this message will only be sent to plug-ins which return `RT_ACTION` as their result type, in response to an `AME_GETRESULTTYPE` message. Therefore there is no need to ever return `AME_NOTIMPLEMENTED`.

Remarks

This message is sent to a plug-in during a test. For this reason, processing should be performed as quickly as possible. If processing could take more than a few milliseconds, then you should consider creating a separate thread for your plug-in and have the thread perform whatever processing is required. This will ensure that the tracking is not adversely affected by your plug-in.

AME_TESTDATA structure

Contains information about a test that is about to start. This record is pointed at by the lParam of an `AME_TESTSTART` message.

Syntax

```
typedef struct {
  DWORD    AnimalNum;
  char     AnimalID[80];
  DWORD    TrialNum;
  RECT     ApparatusRect;
  double   Scaling;
  LPVOID   Settings;
  DWORD    SettingsSize;
  LPVOID   Context;
}AME_TESTDATA, *LPAME_TESTDATA;
```

Members

AnimalNum

The number of the animal about to be tested.

AnimalID

This is the *ID* of the animal about to be tested. If the protocol does not specify that it is using animal IDs, then this will simply be a string representation of `AnimalNum,` otherwise it will be the ID the user has specified for this animal.

Note: If the protocol *does* use animal IDs, but the user has not entered an ID for this animal, then the `AnimalID` is the `AnimalNum` preceded by a # sign.

TrialNum

This is the number of this trial for this animal. Trial numbers start at 1.

ApparatusRect

This is a Windows `RECT` structure which holds the coordinates of a rectangle that exactly encloses the apparatus.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

Scaling

This value allows you to scale from pixels to millimetres (or vice versa).

Given a value in pixels, its value in millimetres is `pixels / scaling.`

Given a value in millimetres, its value in pixels is `mm * scaling.`

Settings

This points at a buffer which holds the plug-in's settings. If the settings have never been set, the buffer pointed to by this field will be filled with zeros. See `AME_EDITSETTINGS` for more details about settings.

SettingsSize

This is the size of the buffer pointed to by the `Settings` field. The size is in bytes.

Context

This is a return value.

You can set this value to anything you want - it will be passed back to you in `AME_POSN`, `AME_TESTEND` and `AME_TESTABORT` messages. Typically, this value is the address of a record structure which your plug-in allocates during `AME_TESTSTART` processing, and which holds details and context data for the analysis the plug-in will perform.

Note that you don't *have* to use this field. If your plug-in has no need for a context record, then you can simply ignore this field.

AME_POSNDATA structure

Contains information about the position of the animal during a test. This record is pointed at by the lParam of an `AME_POSN` message.

Syntax

```
typedef struct {
  LPVOID   Context;
  DWORD    Time_xy;
  int      x;
  int      y;
  DWORD    Time_HeadTail;
  int      hx;
  int      hy;
  int      tx;
  int      ty;
  int      cx,
  int      cy,
  bool     Hidden;
  RECT     AnimalArea;
  int      ResultType;
  int      ResultValue;
}AME_POSNDATA, *LPAME_POSNDATA;
```

Members

Context

The value returned in the `Context` field of the `AME_TESTDATA` record passed to the plug-in during `AME_TESTSTART` processing. This is typically a pointer to a context record structure allocated by the plug-in, but can be any value you want to record (or you can ignore it).

Time_xy

The time, in hundredths of a second since the test start, at which the position of the animal's centre point (as reported in `x,y`) was captured.

x

The x-coordinate of the animal's centre position. This is the position typically marked by an orange dot while tracking in ANY-maze.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

y

The y-coordinate of the animal's centre position. This is the position typically marked by an orange dot while tracking in ANY-maze.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

Time_HeadTail

The time, in hundredths of a second since the test start, at which the position of the animal's head and tail (as reported in `hx,hy` and `tx,ty`) was captured.

hx

The x-coordinate of the animal's head position. Note that this value will be passed whether or not the protocol is set to track the animal's head.

This value will be -1 until ANY-maze identifies the animal's head for the first time.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

See notes about head/tail tracking in the Remarks section below.

hy

The y-coordinate of the animal's head position. Note that this value will be passed whether or not the protocol is set to track the animal's head.

This value will be -1 until ANY-maze identifies the animal's head for the first time.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

See notes about head/tail tracking in the Remarks section below.

tx

The x-coordinate of the animal's tail position. Note that tail positions tend to be less stable than head and centre positions (that's to say they are subject to more noise).

This value will be -1 until ANY-maze identifies the animal's tail for the first time.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

See notes about head/tail tracking in the Remarks section below.

ty

The y-coordinate of the animal's tail position. Note that tail positions tend to be less stable than head and centre positions (that's to say they are subject to more noise).

This value will be -1 until ANY-maze identifies the animal's tail for the first time.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

See notes about head/tail tracking in the Remarks section below.

cx

The x-coordinate of the animal's centre position that corresponds to the head and tail coordinates in `hx,hy` and `tx,ty.` See notes about head/tail tracking in the Remarks section below for further details about this value.

This value will be -1 until ANY-maze identifies the animal's head/tail for the first time.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

cy

The y coordinate of the animal's centre position that corresponds to the head and tail coordinates in `hx,hy` and `tx,ty.` See notes about head/tail tracking in the Remarks section below for further details about this value.

This value will be -1 until ANY-maze identifies the animal's head/tail for the first time.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

Hidden

True if the animal is in a hidden zone. In this case none of the coordinate fields are valid (they will all be 0).

AnimalArea

A Windows `RECT` structure which exactly encloses the area of the animal.

The coordinate system has its origin in the **top, left** of the video source showing the apparatus.

ResultType

This is a return value.

You should set this to an `RT_xxx` value to indicate the type of value your plug-in is returning. If you don't return any value, set this to `RT_NONE`.

ResultValue

⚠ *IMPORTANT NOTE: This is a signed 24-bit value, meaning that the range of values available is -8,388,608 to 8,388,607. If you return more than 24-bits, the high bits will be lost.*

This is a return value.

Unless you return a `ResultType` of `RT_NONE`, you should set this value to the result of your analysis.

If the result type is `RT_ONOFF`, then you should return a value of 1 for 'on' and 0 for 'off'.

Remarks

ANY-maze attempts to track the animal's centre position in all the frames it receives from the video source. If it fails to identify the animal's centre position, then the frame is ignored (from the point of view of results) and it won't be passed to any plug-in.

If the centre is tracked successfully, then ANY-maze goes on to try to identify the animal's head and tail positions.

If the head and tail are identified successfully, this record will hold data for the centre, head and tail position for the current frame. However, if the head and tail cannot be identified, then this record will hold the coordinates of the animal's centre in this frame and the coordinates of the animal's head and tail in the last frame in which the head and tail *were* successfully tracked (the time of this frame will be in `Time_HeadTail`). In this case, you may wish to also know the position of the animal's centre which corresponds to the head and tail positions being passed (i.e. the centre position at time `Time_HeadTail`) and this is passed in `cx,cy.` Note that when the head and tail *are* tracked in a frame, then `Time_HeadTail` will equal `Time_xy` and `cx,cy` will equal `x,y`).

AME_ACTIONDATA structure

Contains information about the action to be performed during a test. This record is pointed at by the lParam of an `AME_PERFORMACTION` message.

Syntax

```
typedef struct {
  LPVOID   Context;
  DWORD    EventTime;
  char     TriggeringEventName[80];
  LPVOID   Settings;
  DWORD    SettingsSize;
}AME_ACTIONDATA, *LPAME_ACTIONDATA;
```

Members

Context

The value returned in the `Context` field of the `AME_TESTDATA` record passed to the plug-in during `AME_TESTSTART` processing. This is typically a pointer to a context record structure allocated by the plug-in, but can be any value you want to record (or you can ignore it).

EventTime

The time, in hundredths of a second since the test start, at which the event that triggered this action occurred.

TriggeringEventName

The name of the procedure or event that triggered the action.

Settings

This points at a buffer which holds the Action plug-in's settings. If the settings have never been set, the buffer pointed to by this field will be filled with zeros. See `AME_EDITSETTINGS` for more details about settings.

SettingsSize

This is the size of the buffer pointed to by the `Settings` field. The size is in bytes.

AME_OK return code

AME_OK is used as the return code when a message has been processed successfully.

Value

The ordinal value of this code is 0.

AME_NOTIMPLEMENTED return code

`AME_NOTIMPLEMENTED` is used as the return code when a plug-in does not implement the feature related to the passed message. This is a valid return from the following messages:

- `AME_ENUMERATE`
- `AME_ENABLE`
- `AME_GETHELPTITLE`
- `AME_GETHELPINTRO`
- `AME_GETHELPDETAILS`
- `AME_EDITSETTINGS`
- `AME_APPLYSETTINGS`

In the case of `AME_ENUMERATE`, this return code is used to indicate the end of the enumeration, i.e. the DLL has returned the GUIDs of all the plug-ins it implements.

Value

The ordinal value of this code is 1.

AME_ERROR return code

AME_ERROR is used as the return code when a plug-in fails to process a message.

When a plug-in returns this code, ANY-maze will log the error in its log file (which is called ANYmazeLog.txt and is located in the same folder as the ANY-maze.exe file).

You should not return AME_ERROR when processing an AME_POSN message in which you have no result to return. In this case, you should set the ResultType to RT_NONE (unless, of course, you really have encountered an error).

Value

The ordinal value of this code is 2.

AME_CANCELLED return code

AME_CANCELLED is used as the return code when the processing of a message is being cancelled. This is a valid return from:

- AME_EDITSETTINGS
- AME_TESTEND

In the case of AME_EDITSETTINGS, this implies that the user cancelled the editing of the settings and that ANY-maze should not save the settings (even if they have changed).

In the case of AME_TESTEND, this implies that an error occurred during the processing of the test, and any results that had been returned to ANY-maze for the test should be cancelled.

Value

The ordinal value of this code is 3.

RT_NONE result type

`RT_NONE` is the result type used when a plug-in doesn't require ANY-maze to record any result.

This may be because the results are recorded elsewhere, for example in a file created by the plug-in, or because the plug-in doesn't actually generate *results* - for example, the plug-in might just pass the animal's position on to another piece of software.

Value

The ordinal value of this result type is -1.

RT_ONOFF result type

The `RT_ONOFF` result type is used for a plug-in which detects a simple on/off state. For example, the demo plug-in determines when the animal is in the centre of the apparatus; so the animal either *is* in the centre (on) or it isn't (off).

ANY-maze will report the following measures for `RT_ONOFF` results:

- Number of activations
- Time active
- Latency to first activation
- Latency to first deactivation
- Longest activation
- Shortest activation
- Average activation
- Activation frequency

These measures are automatically made available for the apparatus as a whole, and also for each zone (except the *Average activation* which is not available for zones).

Furthermore, the non-latency measures can also be reported across time.

Value

The ordinal value of this result type is 0.

See also:

- RT_ONOFF result type measures

RT_VALUE result type

The RT_VALUE result type is used for a plug-in which detects a simple value which is valid at a moment in time. For example, a plug-in which detects the animal's heart rate would return an RT_VALUE result.

ANY-maze will report the following measures for RT_VALUE results:

- Average
- Maximum
- Minimum

These measures are automatically made available for the apparatus as a whole, and also for each zone. They can also be analysed across time.

Value

The ordinal value of this result type is 1.

See also:

- RT_VALUE result type measures

RT_CUMULATIVE result type

The `RT_CUMULATIVE` result type is used for a plug-in which detects the magnitude of something between successive positions during the test. Thus the sum of the results gives an overall result for the entire test. The distance travelled by the animal is a good example.

ANY-maze reports just the value itself for `RT_CUMULATIVE` results. This measure is made available for the apparatus as a whole as well as for each zone, and can also be analysed across time.

Value

The ordinal value of this result type is 2.

RT_ACTION result type

The `RT_ACTION` result type is used by a plug-in which doesn't return a result, but rather performs some type of action. This is valid as the result type returned from an `AME_GETRESULTTYPE` message. It is not valid as the result type returned in an `AME_POSNDATA` record.

Value

The ordinal value of this result type is 3.

Remarks

For more details about *Action plug-ins*, see Writing an Action plug-in.

Plug-in measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

The measures available for a plug-in depend on the type of result the measure returns:

- RT_ONOFF result type measures
- RT_VALUE result type measures
- RT_CUMULATIVE result type measures

RT_ONOFF result type measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a plug-in which has a result type of `RT_ONOFF`:

- Number of activations
- Time active
- Latency to first activation
- Latency to first deactivation
- Longest activation
- Shortest activation
- Average activation duration
- Activation frequency

Each of the above measures (except for *Average activation duration*) is available for the apparatus as whole (i.e. irrespective of where the animal was when the activation occurred) and also for each defined zone.

RT_ONOFF: Number of activations

Description	Reports the number of times the plug-in was activated (returned a result value of 'On').
Calculation method	Counts the number of activations.
Analysis in zones	Counts the number of activations that occurred when the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the number of activations during the time period.
Units	None
Notes	None

RT_ONOFF: Time active

Description	Reports the total amount of time the plug-in was active.

Calculation method	Sums the amount of time for which the plug-in was active.
Analysis in zones	Sums the amount of time for which the plug-in was active when the animal was in the zone. For a particular zone, it's possible for the *Time active* to be non-zero while the *Number of activations* is zero. This can occur if the animal enters the zone when the plug-in is active. In this case the time the plug-in is active will be registered, but the activation itself won't be.
Analysis across time	This measure can be analysed across time. For any time period, the result is the amount of time that the plug-in was active during the period. For a particular time period, it's possible for the *Time active* during the period to be non-zero while the *Number of activations* for the period is zero. This can occur if the plug-in is already active at the start of the period. In this case, the time the plug-in is active will be registered, but the activation itself won't be.
Units	Seconds
Notes	None

RT_ONOFF: Latency to first activation

Description	Reports the amount of time that elapsed in the test before the plug-in was active for the first time, i.e. returned a result of 'On'.
Calculation method	The value of the test clock at the first activation.
Analysis in zones	The value of the test clock at the first activation that occurred within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

RT_ONOFF: Latency to first deactivation

Description	Reports the amount of time that elapsed in the test before the plug-in was deactivated for the first time, i.e. returned a result of 'Off'.
Calculation method	The value of the test clock at the first deactivation.
Analysis in zones	The value of the test clock at the first deactivation that occurred within the zone.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

RT_ONOFF: Longest activation

Description	Reports the duration of the longest period for which the plug-in was continuously active.
Calculation method	The duration of each activation is calculated, and the largest value is found.
Analysis in zones	The longest period for which the plug-in was continuously active in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the longest period for which the plug-in was continuously active during the period.
Units	Seconds
Notes	None

RT_ONOFF: Shortest activation

Description	Reports the duration of the shortest period for which the plug-in was continuously active.
Calculation method	The duration of each activation is calculated, and the smallest value is found.
Analysis in zones	The shortest period for which the plug-in was continuously active in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the shortest period for which the plug-in was continuously active during the period.
Units	Seconds
Notes	None

RT_ONOFF: Average activation duration

Description	Reports the average duration for which the plug-in was active.
Calculation method	Calculated by dividing the *Time active* by the *Number of activations.*
Analysis in zones	This measure cannot be analysed in zones.
Analysis across time	This measure cannot be analysed across time.
Units	Seconds
Notes	None

RT_ONOFF: Activation frequency

Description	Reports the frequency with which the plug-in was activated.
Calculation method	Calculated by dividing the *Number of activations* by the *Test duration.*
Analysis in zones	Calculated by dividing the *Number of activations* in the zone by the *Total time in the zone.*
Analysis across time	This measure can be analysed across time. For any time period, the result is the *Number of activations* which occurred during the period divided by the period's duration.
Units	Hertz
Notes	None

ANY-maze help topic H0956

RT_VALUE result type measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze will score the following measures for a plug-in which has a result type of RT_VALUE:

- Average
- Maximum
- Minimum

Each of the above measures is available for the apparatus as whole (i.e. irrespective of where the animal was when the plug-in detected a value) and also for each defined zone. Thus, for example, you can compare the average value between zones.

RT_VALUE: Average

Description	Reports the average value returned by the plug-in during the test.
Calculation method	Calculated by averaging all the values returned during the test.
Analysis in zones	Calculated by averaging the values returned while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the average of the values returned during the period.
Units	Plug-in specific - not known to ANY-maze.
Notes	None

RT_VALUE: Maximum

Description	Reports the maximum value returned by the plug-in during the test.
Calculation method	All the values returned during the test are compared, and the maximum is found.
Analysis in zones	The maximum value returned while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the maximum value returned during the period.
Units	Plug-in specific - not known to ANY-maze.
Notes	None

RT_VALUE: Minimum

Description	Reports the minimum value returned by the plug-in during the test.
Calculation method	All the values returned during the test are compared, and the minimum is found.
Analysis in zones	The minimum value returned while the animal was in the zone.
Analysis across time	This measure can be analysed across time. For any time period, the result is the minimum value returned during the period.
Units	Plug-in specific - not known to ANY-maze.
Notes	None

RT_CUMULATIVE result type measures

ANY-maze has been designed to be extended, and we'll be delighted to add any new measures you might find useful, for free! Just contact ANY-maze technical support.

ANY-maze scores a single measure for a plug-in which has a result type of `RT_CUMULATIVE`:

- Cumulative value

This measure is available for the apparatus as whole (i.e. irrespective of where the animal was when the plug-in detected a value) and also for each defined zone. Thus, for example, you can compare the value between zones.

RT_CUMULATIVE: Cumulative value

Description	Reports the sum of all the values returned by the plug-in during the test.
Calculation method	All the values returned during the test are summed.
Analysis in zones	The sum of all the values returned while the animal was in the zone. This will necessarily yield a small inaccuracy, as the plug-in may report a value at, say, time 10; the animal may then enter a zone at time 20, and the plug-in may then report another value at time 30. Thus the value reported at time 30 represents the result for the period 10-30, but the animal was only in the zone for half of this period (i.e. from 20-30); nevertheless the entire value is attributed to the zone. The inverse occurs when an animal leaves a zone, as the value is always attributed to the zone the animal is entering - not the zone it is leaving. These inaccuracies will generally be very small, as the time between values will usually be in the order of 30-60 milliseconds.
Analysis across time	This measure can be analysed across time. For any time period, the result is the sum of all the values returned during the period. Similarly to zones (see above), small inaccuracies can occur on the division between periods - here a value is always attributed to the period that is *starting*.
Units	Plug-in specific - not known to ANY-maze.
Notes	None

The Spike2 plug-in

Introduction

The Spike2 plug-in provides features to integrate ANY-maze and the Spike2 program from CED. Although installing this plug-in has no adverse effects, it is only recommended if you actually use Spike2.

Full details of the plug-in's features are provided in the plug-in itself. After installing, go to *Protocol > Analysis and Results > Analysis Options > ANY-maze plug-ins*, select one of the Spike2 plug-ins (there are two) and then click the link to *View help for the selected plug-in.*

If you want to know more about this plug-in, then please contact ANY-maze technical support who will be happy to help you.

Install

Click to install the Spike2 plug-in now

Glossary

Index : 0-9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0-9

32-bit	32-bit means that the computer is capable of transferring 32 bits of information at a time. If you're running Windows XP, it's likely to be a 32-bit version.
64-bit	64-bit means that the computer is capable of transferring 64 bits of information at a time. Most version of Windows from Vista onwards (Windows 7, 8, 8.1, and 10) are now 64-bit.
#N/A	An undefined value in a procedure is represented by the value #N/A (not applicable). A value can be undefined if there is an error in a mathematical function, for example.

A

ANY-maze example video	ANY-maze includes some example videos of animals in different apparatus, which are intended to be used for training purposes. These videos can be selected into a video source, allowing you to set up and even run an experiment, to show you all the features of ANY-maze without actually needing to be connected to a camera.
ANY-maze video file	An ANY-maze video file is a file which contains a digital video recorded in ANY-maze's own digital video format. This format typically compresses video by between 90% and 98% without losing any important image information.
Apparatus map	An apparatus map is the drawing of the apparatus which defines its edges, and which divides the apparatus up into different areas.
Area	An area of the apparatus is one of a number of shapes drawn on the apparatus image. One or more areas can then be selected to form a zone.
Auto-complete	Auto-complete is a feature of fields in ANY-maze which only accept a limited selection of entries. As soon as what you've typed uniquely matches one of the entries, the field will automatically

complete. For example, if the valid entries were 'Blue' and 'Brown', and you typed 'B', the field would not auto-complete as both possibilities start with a 'B'. However, if you then typed 'r', the field would auto-complete with the word 'Brown', as no other entry starts with the letters 'Br'.

Auto-start	A feature of ANY-maze which automatically starts tests when the system sees the experimenter leave the camera's field of view. This has the advantage of starting the test without the user having to press a key on the computer's keyboard. Auto-start is specified as part of the protocol's 'Automatic starting of tests'.

B

No glossary entries

C

Chapter	A DVD is divided into titles, and each title is further divided into chapters. A chapter will often represent a single scene in a movie recorded on a DVD. ANY-maze allows you to navigate to any chapter in any title on a DVD using the DVD menu.
Control panel	The Windows 'Control Panel' is the part of the Windows user interface which allows users to view and manipulate basic system settings, such as adding or editing hardware or software options, controlling user accounts or changing accessibility options. On Windows versions up to Windows 7, it can be accessed from the Start menu. On Windows 8 and above, access it by right clicking on the Start button in the bottom left-hand corner of the screen.
Crop	To crop an image is to 'cut off' parts from the edges, making the overall image smaller. This is usually done to remove areas of the image that are irrelevant.

D

Debounce interval	When switches close, the contacts tend to bounce apart rapidly several times before coming to rest. This can make it appear that the switch has been repeatedly turned on and off. To avoid this, a debounce delay can be used such that the apparent opening of the

	switch contacts during the delay are not considered to be the switch turning off. The duration of this delay is called the debounce interval.
Digital camera	In ANY-maze terms, a digital camera is one which doesn't require a digitiser, i.e. the signal from the camera is already in a digital form. A digital camera connects to a USB or FireWire port on your computer.
Digitiser	A digitiser is a device which converts an analogue video signal into digital data which a computer can process. They're sometimes called 'frame grabbers' or 'capture cards'.
Double click	Clicking the left-hand mouse button twice in rapid succession.
DVD title	A DVD is divided into a number of titles, with each one typically representing one movie recorded on the DVD. ANY-maze allows you to navigate to any title on a DVD using the DVD menu.
DVD chapter	A DVD is divided into titles, and each title is further divided into chapters. A chapter will often represent a single scene in a movie recorded on a DVD. ANY-maze allows you to navigate to any chapter in any title on a DVD using the DVD menu.

E

No glossary entries

F

File extension	A string of characters attached to a filename, usually preceded by a '.' and indicating the format of the file. For example, .doc and .szd are file extensions. When looking at files in Windows Explorer, you may not see any file extensions, if the option to show them has been turned off.
FireWire	FireWire, which is also called IEEE 1394 or i-Link, is a standard for connecting high bandwidth devices (such as CD players and video cameras) to computers. The standard is slowly becoming obsolete (most modern computers no longer have FireWire ports), although there are still some cameras around that use the FireWire interface.
Frame	A frame is the name given to a single image from a video camera.
Frame rate	Frame rate is the term used to describe the number of frames (i.e. images) transmitted by a video camera in a second. For most

	cameras, this is either 30 or 25 frames per second (usually abbreviated to fps).
Frame size	Frame size is the term used to describe the size of the frames (i.e. images) transmitted by a video camera. Frame size is given in units of pixels (i.e. the dots which make up the video picture) and some typical sizes include 640 x 480 and 320 x 240.

G

GUID	A GUID (Globally Unique IDentifier) is a 128-bit number that is used as a unique identifier in computer software. GUIDs are not GUARANTEED to be unique, since they are usually generated randomly, although the probability of the same number being generated twice is negligible. They can also be called UUIDS (or Universally Unique IDs).

H

Heat map	A heat map (or 'occupancy plot') is a graphical representation of the amount of time the animal spent in different parts of the apparatus. It uses a range of colours to indicate the time spent in an area, with blue as the shortest time and red as the longest.

I

Icon	An icon is a symbol or graphical representation on the computer screen of a program, option or window.
IP address	An IP address (Internet Protocol address) is a unique numerical label assigned to each device (e.g. computer, printer) connected to a computer network that uses the Internet Protocol for communication.

J

No glossary entries

K

No glossary entries

L

No glossary entries

M

Measure	A measure is a single 'value' that ANY-maze can score. For example 'Test duration', 'First zone entered' and 'Number of entries into the X zone' are all measures. ANY-maze includes over 300 different types of measure, although the overall number of individual measures available will usually be much greater, as many of these are scored separately for each zone, key, input switch, sequence, etc.
Montage video source	A montage video source joins together the images from two normal video sources to create a single image. They're used when more than one camera is required to fully view a piece of apparatus.

N

#N/A	An undefined value in a procedure is represented by the value #N/A (not applicable). A value can be undefined if there is an error in a mathematical function, for example.

O

No glossary entries

P

Pan	To pan an image is to move or drag the image, so that parts which are off the sides become visible.

Pane	In ANY-maze, the Protocol, Experiment and Tests pages are divided into different sections - these sections are called panes.
Personal ANY-maze folders	A personal ANY-maze folder is a specific folder which ANY-maze uses as the default location for an individual user's ANY-maze files.
Pixel	Pixels, which stands for 'Picture elements', are the individual dots which make up a picture. They're often used as the unit of picture size. For example a picture of 640 x 480 pixels consists of 640 columns of dots horizontally and 480 rows of dots vertically, in other words the picture contains a total of 307,200 dots.
Protocol	The Protocol is the part of an experiment which contains details about how the experiment will be performed. It defines everything that ANY-maze needs to know in order to run an experiment, such as the apparatus that will be used, the duration of tests, etc.
Protocol list	The protocol list is the list of protocol elements shown on the left-hand side of the Protocol page.
Protocol mode	The protocol mode is set using the Protocol element at the top of the protocol list. It controls which other elements are available in the protocol. For example, when an I/O protocol mode is selected, then the protocol list will be expanded to include a wide range of I/O-related elements.

Q

No glossary entries

R

Raster graphics format	A raster graphics format is a way of storing images as a series of dots. Examples of raster graphics formats include Bitmap and JPEG.
Raw data	ANY-maze uses the term 'raw data' to refer to the result of an individual measure in an individual test. In other words, values which represent the lowest level of results within the system.
Region of interest	The region of interest of a video source is the part of the image provided by a camera or digitiser that the video source is actually using.
Resampled	Digital signals (such as sounds) are 'sampled' at a certain frequency - for example, the sound stored on a CD is sampled at 44100Hz. In some situations, a device cannot process a signal because it doesn't

support the frequency it was sampled at, and so it will 'resample' the signal to alter the frequency to one that it does support. One example of this is the ANY-maze interface (AMi), which resamples CD audio to 11025Hz, which is the frequency it uses for its speakers.

Retired — An animal which is retired from an experiment ceases to be tested in any future tests, but any tests it has already completed are maintained. This is in contrast to an animal which is deleted, where completed tests are removed.

Ribbon Bar — The section at the top of all ANY-maze pages, containing options for use with the current page.

Right click — Clicking the right-hand mouse button.

S

Segment — ANY-maze can divide tests into equal sized segments across time, and can then analyse most measures within each test segment. You can specify the length of the segments used as part of the Protocol's 'Analysis and Results' options. When referring to a segment, ANY-maze uses the notation x-y meaning the segment from time x to time y, for example 30-60 would be the segment from time 30 seconds to time 60 seconds (actually to time 59.9999 seconds).

Shared ANY-maze folder — The shared ANY-maze folder is a specific folder where shared ANY-maze files can be stored. It's particularly useful for storing Protocol files, as these are often shared between different users of the system.

Stage — A stage is a logical part of an experiment which consists of one or more identical trials. In a water-maze experiment, for example, there might be two stages - Acquisition and Retention.

Stage end rule — A stage end rule defines a condition which can cause an animal to stop being tested in a stage before it has had the maximum number of trials. Stage end rules are often used when animals are being trained to achieve some goal, for example finding the island in a water-maze within 30 seconds in three consecutive trials.

T

Test control key — Each piece of apparatus in an experiment can optionally be controlled by one of the 'F' keys found at the top of the computer's

	keyboard. The key used for a piece of apparatus is defined in the protocol. The key's action depends on the apparatus's current status - for example, when the apparatus is 'Ready...', pressing the key will start a test.
Test control switch	Each piece of apparatus in an experiment can optionally be controlled by an input switch connected to an I/O device. The switch used for a piece of apparatus is defined in the protocol. The switch's action depends on the apparatus's current status - for example, when the apparatus is 'Ready...', closing the switch will start a test. Using a test control switch can be very useful if your apparatus is located at a distance from your computer, and you're unable to use auto-start - in this case you can connect the switch to a long cable and mount it close to, or even on, the apparatus.
Title	A DVD is divided into a number of titles, with each one typically representing one movie recorded on the DVD. ANY-maze allows you to navigate to any title on a DVD using the DVD menu.
ToolTip	A ToolTip is a small pop-up window that appears when the mouse hovers over a specific item in the software (e.g. a button) to give you more information about that item.
Track plot	A track plot is a diagram showing the path, or track, that an animal followed during a test.
Trial	A trial is a single test performed on a specific animal in a specific stage. For example, ANY-maze might refer to 'Animal 1's first acquisition trial' - that's to say the first test of animal 1 in the acquisition stage of the experiment.
TTL	TTL stands for 'Transistor-Transistor Logic' and describes a type of binary electrical signal commonly used in computers. A TTL signal is defined as 'low' when between 0V and 0.8V with respect to ground, and 'high' when between 2V and 5V.

U

Unassigned	An animal is said to be unassigned when it's not part of any treatment group - this is the case when there's no treatment code recorded against the animal.
UNC	A UNC (Universal Naming Convention) file name is a way to identify a shared file on a computer without having to know exactly which drive it's on. On Windows operating systems, a UNC name has the format: \\servername\sharename\path\filename.

Undefined	A result is undefined if it has no meaningful value. For example, the latency to enter a zone which the animal never entered. Division by zero also yields an undefined result. You can use the Protocol's 'Analysis and Results' element to control how ANY-maze processes some undefined results.
Unzip	Unzipping a file is the process of decompressing a zipped file into its constituent parts.
USB	USB, which stands for Universal Serial Bus, is the modern standard for connecting peripheral devices such as printers, mice, cameras etc. to computers. Almost all modern computers include USB ports as standard, and to connect a device you simply plug it into one of them - Windows will automatically configure the device ready for use.
USB port	A USB port is the physical connector on your computer where you plug in a USB device. Many modern computers include 4 or even 6 USB ports as standard.

V

Vector graphics format	A vector graphics format is a way of storing images that allows them to be easily scaled without losing their resolution. Parts of the image are stored as lines or shapes, which can be easily scaled. Examples of vector graphics formats include PNG and Windows metafiles.
Video codec	A video codec is piece of software that compresses and decompresses video pictures. There are many different codecs available, for example MPG4 codecs, Indeo codecs and WMV9 codecs. Each codec generally follows a single standard, and each standard has advantages and disadvantages - perhaps compressing the video greatly but at the loss of quality, or compressing greatly with good quality but requiring a lot of processing power to do it. ANY-maze supports most codecs designed to work with Windows (excluding DMO codecs). Windows is provided with a reasonable selection of codecs as standard, and you can download others from the internet.
Video quad	A Video quad is a device which merges pictures from up to four cameras into one image, with the picture from each camera appearing in one quadrant of the final output.

W

Wizard	Wizards are used to simplify a complicated task in software by breaking it into a number of steps. They often take the form of a series of questions, where your response to one question will determine what the wizard needs to know/do next.

X

XML	XML stands for eXtensible Markup Language. It is a text-based file format that allows a file to be easily read by both computers and human beings.

Y

No glossary entries

Z

Zip	Zipping a file is the process of compressing data into a file, without losing any information. The size of the resulting file is usually much smaller than the original size of the data.

ANY-maze Legacy Policy

Introduction

With the release of version 6, ANY-maze is available as both a 32-bit and a 64-bit program. Some older image capture and input/output devices that ANY-maze currently supports are incompatible with the 64-bit version, and we are therefore designating them as 'legacy' devices. This means that only the 32-bit version of ANY-maze will support them.

We are also designating Windows XP and Windows Vista as legacy operating systems, and they will also only be supported by the 32-bit version of ANY-maze.

We will continue to produce 32-bit versions of ANY-maze until we release version 7, probably in mid-2019. From version 7 onwards, ANY-maze will only be available as a 64-bit program.

The rest of this document describes this legacy policy in more detail and explains how it may affect you.

What does “32-bit” and “64-bit” actually mean?

32-bit and 64-bit simply define the amount of information that can be processed by a computer in one go. More bits means that data can be processed in larger chunks, and that a program can access more of the computer’s memory. In general: the more bits, the better the performance of your computer and its software.

Most modern computers are now 64-bit, and recent versions of Microsoft Windows have all been 64-bit operating systems. A 32-bit program can run on a 64-bit version of Windows, but it’s not as efficient as it could be.

What’s this got to do with ANY-maze?

We have changed ANY-maze so it can now run as a 64-bit program. This should mean long-term compatibility with Windows, better performance, and the ability to use more of your computer’s memory.

However, ANY-maze relies on third-party software (called ‘drivers’) to talk to a number of the image capture and input/output devices that it uses. If a device is particularly old, the manufacturer may not have produced a 64-bit driver, and so the 64-bit version of ANY-maze won’t be able to communicate with it.

If you’re using any of these older, ‘legacy’ devices with ANY-maze, then you won’t be able to use the 64-bit version of ANY-maze and you’ll need to stick with the 32-bit version.

Other legacy devices

We are also classifying some older image capture and input/output devices as legacy even though they do have 64-bit drivers available. This applies to devices that have been discontinued by their manufacturers, are simply out-of-date, or are little used by ANY-maze customers. These devices will not be supported by the 64-bit version of ANY-maze. You'll find a full list of affected devices here.

Legacy versions of Windows

Microsoft ended support for Windows XP in 2014 and for Windows Vista in 2017. Despite this, we will ensure that the 32-bit version of ANY-maze will continue to work with these operating systems. However, the 64-bit version will only work with Windows 7 and above.

End of 32-bit development

Development of the 32-bit version of ANY-maze will cease with the release of ANY-maze version 7 (probably in mid-2019), which will be the first 64-bit only version of the program.

This means that if you do use a legacy device or a legacy version of Windows, you will need to exchange it for a more modern alternative in order to use version 7 or above of ANY-maze.

Which devices are affected?

The information below tells you about the various devices that won't be supported by the 64-bit version of ANY-maze, together with recommendations for devices you can use in their place.

National Instruments IMAQ digitisers

IMAQ digitisers are used to connect analogue cameras to a computer. Currently ANY-maze supports the following devices:

- PCI-1405 (discontinued model)
- PCI-1407 (discontinued model)
- PCI-1408 (discontinued model)
- PCI-1409 (discontinued model)
- PCI-1410 (discontinued model)
- PCI-1411 (discontinued model)

All of these are classified as obsolete by National Instruments, and they are all now considered to be legacy devices by ANY-maze.

Recommended replacement: RTV-24 digitiser

Data Translation digitisers

Data Translation digitisers are used to connect analogue cameras to a computer. Currently ANY-maze supports the following devices:

- DT3120 (discontinued model)
- DT3153 (discontinued model)
- DT3131 (discontinued model)
- DT3132 (discontinued model)
- DT3133 (discontinued model)
- DT3155 (discontinued model)

None of these devices are available any more, and they are all now considered to be legacy devices by ANY-maze.

Recommended replacement: RTV-24 digitiser

Euresys digitisers

Euresys digitiser boards are used to connect analogue cameras to a computer. Currently ANY-maze works with a wide range of these devices, and it will continue to support many of them. However, the following devices will now be considered to be legacy devices by ANY-maze:

- Picolo (serial numbers below 6321)
- Picolo Pro (discontinued model)
- Picolo Pro 2 (serial numbers below 7145)
- Picolo Pro 3I (discontinued model)
- Picolo Pro 3E (discontinued model)
- Picolo Tetra-X (discontinued model)
- Picolo Tetra-X-RC (discontinued model)
- Picolo Jet X (discontinued model)
- Picolo Diligent (discontinued model)

Recommended replacement: RTV-24 digitiser

For the avoidance of doubt, the following Euresys digitiser boards **will continue** to be supported in the 64-bit version of ANY-maze:

- Picolo (serial numbers above 6321)
- Picolo Pro2 (serial numbers above 7145)
- Picolo Tetra
- Picolo Junior4
- Picolo Alert
- Picolo Alert PCIe

FireWire cameras

All firewire cameras are now considered to be legacy devices by ANY-maze. Models affected:

- Unibrain Fire-i

Recommended replacement: ANY-maze USB camera

Axis network cameras

The Axis network cameras are now considered to be legacy devices by ANY-maze. Models affected:

- Axis 210 camera (discontinued model)

Recommended replacement: ANY-maze USB camera

The Switch & Sense 8/8

The Switch and Sense 8/8 is an I/O device with 8 opto-isolated inputs and 8 relay outputs, which connects to a USB port on your computer. Models affected:

- Switch and Sense 8/8

Recommended replacement: The AMi-2 Digital interface and/or the AMi-2 Relay interface.

The computer's Parallel port

ANY-maze can use a parallel port as a simple I/O device - providing 4 TTL level inputs and 8 TTL level outputs. However, very few computers include a parallel (printer) port nowadays, and this is now considered to be a legacy device in ANY-maze.

Models affected:

- Parallel (printer) ports built into PCs
- Parallel (printer) ports on expansion cards

Recommended replacement: The AMi-2 Digital interface.

Input/output ports on Euresys Picolo digitiser cards

Some Euresys digitiser boards include general purpose input/output lines. Currently ANY-maze works with a wide range of these devices, and it will continue to support some of them. However, the following will now be considered to be legacy devices by ANY-maze.

- Picolo (serial numbers below 6321)
- Picolo Pro (discontinued model)
- Picolo Pro 2 (serial numbers below 7145)
- Picolo Pro 3I (discontinued model)
- Picolo Pro 3E (discontinued model)
- Picolo Tetra-X (discontinued model)
- Picolo Tetra-X-RC (discontinued model)
- Picolo Jet X (discontinued model)
- Picolo Diligent (discontinued model)

Recommended replacement: The AMi-2 Digital interface.

Input/output ports on Axis network cameras

Some Axis network cameras include general purpose input/output lines. These cameras are now considered to be legacy devices by ANY-maze. Models affected:

- Axis 210 camera (discontinued model)

Recommended replacement: The AMi-2 Digital interface.

Summary

To recap, the legacy devices listed above are not supported in the 64-bit version of ANY-maze, but they are supported in the 32-bit version and will continue to be supported until the 32-bit version is discontinued, which will be at the release of version 7 of ANY-maze, probably in mid-2019.

Note that you will still be able to use the 32-bit version of ANY-maze after it has been discontinued (and we'll still provide help on how to use it), but you won't be able to take advantage of improvements and new features that are included from version 7 onwards.

More information

If you have any questions or concerns regarding this policy, then please contact ANY-maze technical support.

ANY-maze help topic H0961