

# Behavioral Observation Research Interactive Software (BORIS) user guide



**BORIS** is an easy-to-use event logging software for video/audio coding and live observations. **BORIS** is a free and open-source software available for GNU/Linux, Windows and MacOS.

**version 8.5**

[www.boris.unito.it](http://www.boris.unito.it)

# Contents

<b>Follow BORIS</b>	<b>3</b>
<b>Legal</b>	<b>4</b>
<b>Install BORIS</b>	<b>4</b>
Linux	4
Microsoft-Windows	4
MacOS	4
<b>Launch BORIS</b>	<b>4</b>
<b>Create a new project with BORIS</b>	<b>5</b>
Set an ethogram	6
Define the subjects	16
Define the Independent variables	17
Observations' tab	18
Converters' tab	18
<b>Open an existing project with BORIS</b>	<b>19</b>
<b>Import a project</b>	<b>20</b>
<b>Observations</b>	<b>20</b>
Create a new observation	20
Observations list	29
Delete observations	32
Import observations from a project file	32
<b>Coding your media</b>	<b>33</b>
Media based coding	33
Live observations	33
<b>Ethogram</b> widget in the main window	34
<b>Subjects</b> widget in the main window	35
The <b>media player</b> widgets	36
The <b>Events</b> widget	37
Record an event	37
Fix unpaired state events	42
Explore project	42
Frame-by-frame mode	43
<b>Exporting events data</b>	<b>44</b>
Export events	44
Export aggregated events	44
Export events for analysis with JWatcher	45
Export events as behavioral strings	45
Export events as Praat TextGrid	46
Export events as Behavioral Binary Table	46
Extract sequences from media files corresponding to coded events	48
Extract frames corresponding to coded events	48
Export transitions matrix	48

<b>Playback menu</b>	<b>49</b>
Jump	49
Zoom	49
<b>Tools</b>	<b>49</b>
Lock the dockwidgets	49
Focus a video area	50
Geometric measurements	51
Coding pad	53
Subjects pad	54
Image overlay on video	54
Converters for external data values	55
Transitions flow diagram	56
Re-encoding and resizing a video file	57
Rotating a video file	58
Project server	58
<b>Coding map</b>	<b>58</b>
The Behaviors coding map	58
The Modifiers coding map	62
<b>Analysis</b>	<b>63</b>
Time budget analysis	63
Synthetic time budget	65
Plot events	66
Plot time budget	67
Inter-rater reliability	70
Advanced event filtering	70
Latency	73
<b>Preferences</b>	<b>74</b>
General preferences	74
Results	75
FFmpeg framework	75
Spectrogram	76
Plot colors	76
<b>Various</b>	<b>77</b>
Removing path of media files	77
Citing BORIS	77
Docking	77
Configuration files	79
Bug reports and features request	80
<b>Acknowledgement</b>	<b>80</b>
<b>Valid keys for triggering behavior</b>	<b>80</b>

## Follow BORIS

BORIS web site: [www.boris.unito.it](http://www.boris.unito.it)

[BORIS GitHub repository](#)

[BORIS twitter feed](#)

[BORIS YouTube channel](#)

## Legal

Copyright 2012-2022 Olivier Friard - Marco Gamba

**BORIS** is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or any later version.

**BORIS** is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the [GNU General Public License](#) for more details.

## Install BORIS

BORIS can be installed following instructions on the BORIS site at [download page](#)

All previous versions of BORIS are available in the **Releases** section of the [GitHub repository](#)

### Linux

BORIS can run on various Linux distributions including Ubuntu, Debian, [Raspberry Pi OS](#), Chromebook ...).

See the [BORIS for Linux](#) page to install BORIS for Linux.

### Microsoft-Windows

See the [Download BORIS for Microsoft-Windows](#) page to install BORIS for Windows.

Two versions are available: **BORIS Portable** and **BORIS Setup**

### MacOS

BORIS v.8 is not available for MacOS.

## Launch BORIS

Start BORIS by clicking on its icon. For launching BORIS from sources see the BORIS web site. When launching BORIS for the first time it may take some time to show up. Please be patient!

The main window of BORIS will show up. The toolbar with media player commands is disabled for now.



## Warning

On **Microsoft-Windows**, launching can be stopped by a *Security warning: The publisher can not be verified. Are you sure you want to run this software?* Click the *Run* button to launch BORIS.

## Note

The **first launch** of BORIS can be long, so be patient. A terminal/command prompt window will show up, it is normal. Do NOT close it!

At the first launch, BORIS will ask you to allow the automatic check for new version. Internet access is required for this feature. If you choose **Yes** BORIS will check for new version about every 15 days. This option can be changed on the **Preferences** window (See [general preferences](#))

You can launch BORIS from command line by specifying the project file to open:

```
start_boris -p MY_PROJECT.boris
```

## Create a new project with BORIS

The BORIS project file is the container for all information related to the project. It contains the **ethogram**, the **independent variables**, the **subjects definition**, the **behaviors coding maps**, the **converters** and all **observations** data. The save menu option (**File > Save project** or **Save project as ...**) will save the project in a path in your local file system. You can also activate the [automatic backup](#) feature (see **Preferences**).

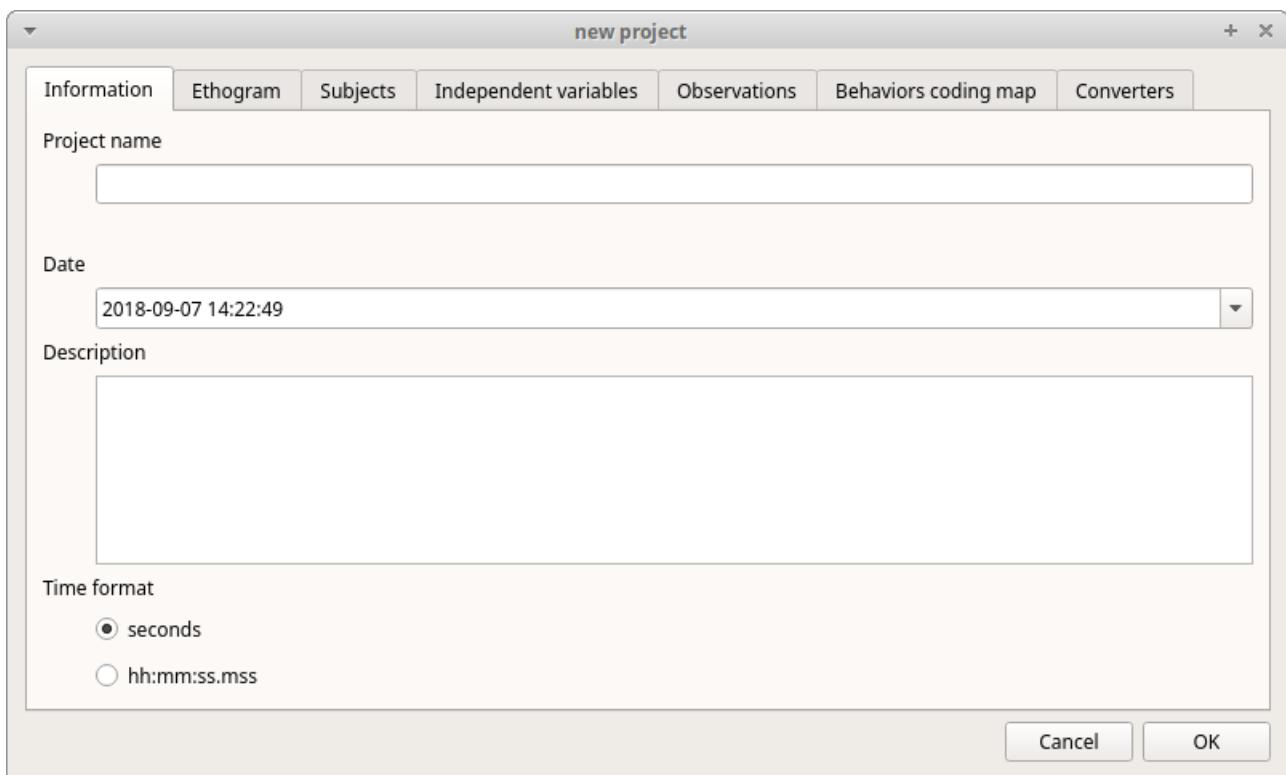
## Important

It is EXTREMELY IMPORTANT to do regular backups of your project files to prevent the lost of data. Software can be reinstalled but your data could quite possibly be lost for ever. Use an external drive or/and a cloud service.

BORIS allows creating an unlimited number of projects but only one project can be opened at a time.

A video tutorial about creating a project is available at <https://www.youtube.com/watch?v=I97Dny5hFOE>

To create a new project, under the menu **File** , select **New project**. You can determine your project name by writing in the **Project name** field in the **Information** tab. Once the project will be saved, the **Project file path** will show the full path to your project file. **Date** will automatically set on the current date and time, but you can alternatively set this info on your media date and time, or whatever you prefer. **Description** can host all the relevant information about your project, can be also left empty. **Time format** can be alternatively set to **seconds** or to **hh:mm:ss.mss**. This choice can be changed at anytime under **File > Preferences** (for MacOs users, **BORIS > Preferences**) > **Default project time format**.

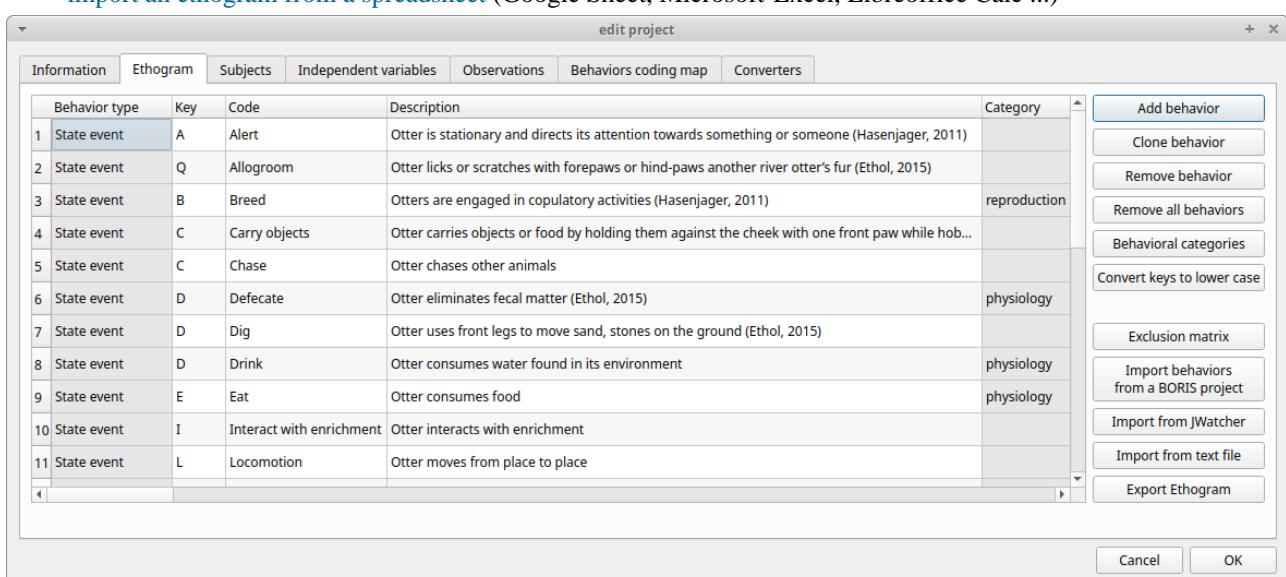


## Set an ethogram

See <https://en.wikipedia.org/wiki/Ethogram> for ethogram definition.

Switching to the **Ethogram** tab, you can alternatively:

- set your ethogram from scratch;
- import an existing ethogram from another BORIS project;
- import an ethogram from a JWatcher global definition file (.gdf).
- [import an ethogram from a plain text file](#)
- [import an ethogram from a spreadsheet \(Google Sheet, Microsoft-Excel, Libreoffice Calc ...\)](#)



## Set your ethogram from scratch

Clicking on the **Add behavior** button you can add a new row in the **Ethogram** table and behavior type will be automatically set to **Point event**.

The cells with gray background can not be directly edited. You must double-click on them and then select a value.

### Behavior types

2 types of behaviors can be defined. Double-click on the cell and select the type of behavior:

- **Point event behavior when the behavior has no duration.**

The behavior will be coded by pressing the defined keyboard key (see below) or by double-clicking to the corresponding row in the Ethogram table.

- **State event behavior when the behavior has a duration.**

The behavior start and stop will be coded by pressing the defined keyboard key (see below) or by double-clicking to the corresponding row in the Ethogram table. These behaviors **must** have a start event and a stop event otherwise an **UNPAIRED events** warning will be reported when you will close the observation or during an analysis

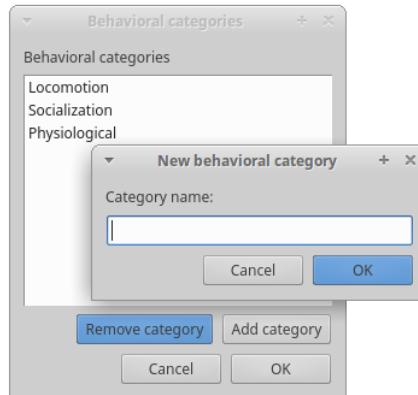
You can switch between **State event** and **Point event** at your convenience with a double-click on the **Behavior type** cell. You can also add a **Coding map** to either a **State event (State event with coding map)** or a **Point event (Point event with coding map)**; see the "Coding map" section for details).

An existing behavior can be duplicated using the **Clone behavior** button. Its code have then to be changed. On a selected behavior, click on the **Remove behavior** button to remove. The **Remove all behaviors** button will clear the **Ethogram** table. Both the above-mentioned operations must be confirmed when prompted.

Behavioral codes (**Code** column) can be sorted alphabetically by checking the **Alphabetical order** checkbox. Alternatively they can be sorted manually by using the **Move up** and **Move down** buttons.

### Categories of behaviors

Defining categories of behaviors can be useful for the analysis of coded events (for example the time budget analysis). Click the **Behavioral categories** button and add a the categories of behaviors. Behaviors can then be included or not in a defined category.



### Set keys and codes

For each behavior you have to set a keyboard key (**Key** column) that will be then used to code the behavioral events. You can choose whether you want to set a unique key for each behavior or use the same key for more than one behavior. In the case you set the same key for more than a behavior, BORIS will pause your coding and ask which of the behavior you want to record. From version 7 the keys are **case-sensitive**.

If your project was created with a previous version of BORIS (< v.7) you can use the **Convert keys to lower case** to convert all keys to lower case otherwise you will have to code your observation using upper case key.

### Important

If you open a project file created with a version older than v.7 BORIS will ask you to convert the upper case behavior and subject keys to lower case.

### Important

**Do not use the / and \* keys! They are reserved for the frame-by-frame mode.**

In the **Code** column, you have to add a unique code for each behavior. Duplicated codes are not accepted and BORIS will warn in red about duplicates on the bottom left of the *Ethogram* tab. The code can be an alphanumeric string (which must not include the pipe character | ).

The **Category** column allow you to include the behavior to a predefined category.

The **Description** of your behavior is optional. The **Description** column can be useful to add information about a specific behavior, its characteristics (e.g. to standardise observation between different users) or to refer to external information (e.g. reference to a previous ethogram).

The following three columns (**Modifiers**, **Exclusion**, **Coding map**) cannot be edited from the **Ethogram** table.

Set the modifiers

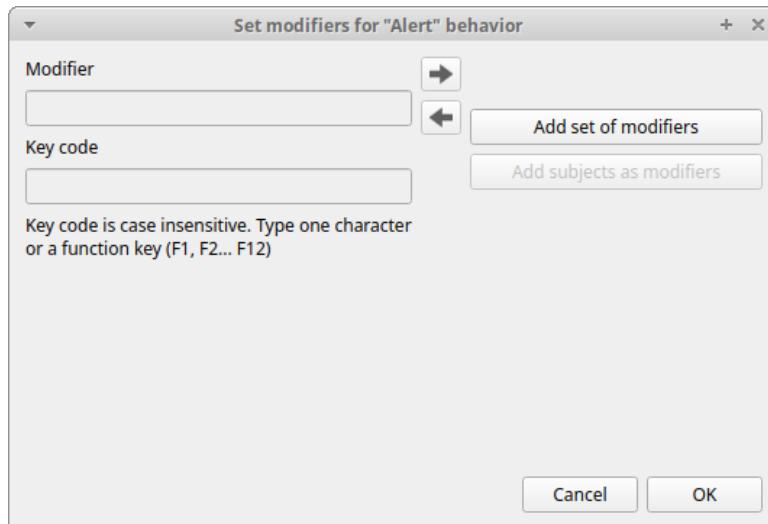
Modifiers can be used to add attributes to a behavior. A single behavior can have two or more modifiers attached (e.g. "play" may have "solitary" or "social" as modifiers). The use of modifiers can be convenient to significantly reduce the number of keys and simplify the behavioral coding.

**4 types of modifiers** are available: **Single selection**, **Multiple selection**, **Numeric** and **Value from external data file**:

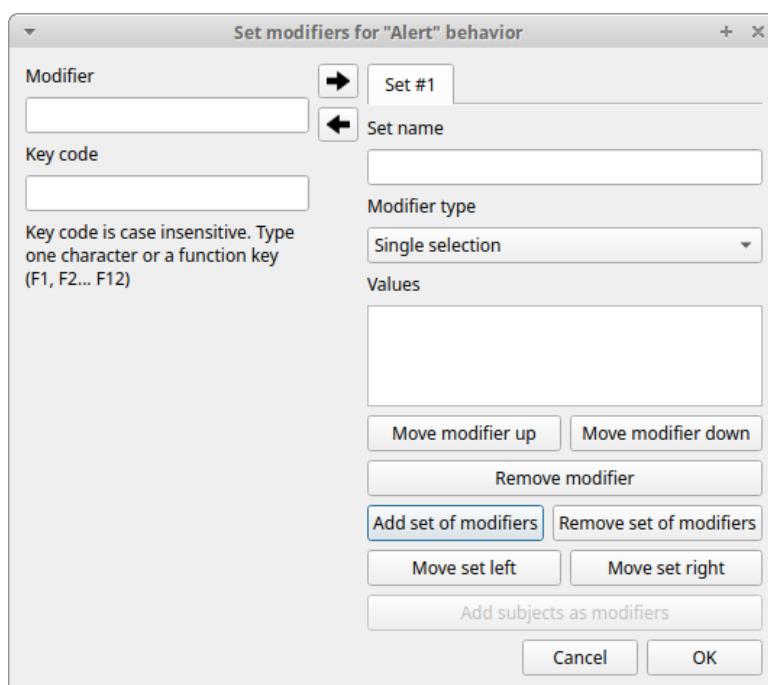
- the **Single selection** type will allow the observer to select only **one** modifier for the current behavior.
- the **Multiple selection** type will allow the observer to select one or more modifiers for the current behavior.
- the **Numeric** type will allow the observer to input a number. For example a distance of interaction.
- the **Value from external data file** type will save the value of a variable from an external data file.

In BORIS modifiers can also be added in different modifier sets [e.g. "play, social" may have a modifier set (#1) for "brothers" and another (#2) for "sisters"]. In the case of using sets of modifiers, you can select one/more modifier for each set.

To add modifiers to a behavior, you need to double-click the **Modifiers** cell corresponding to the behavior you want to add the modifiers to. The following window will show up:



Click the **Add a set of modifiers** button:

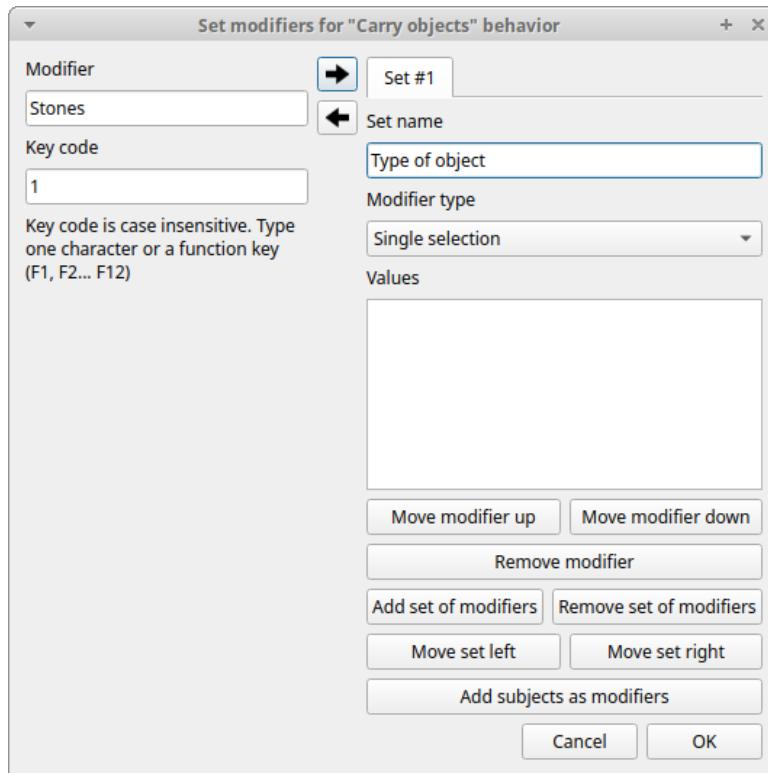


Select the modifier type using the **Modifier type** combo box. You have to choose between **Single selection**, **Multiple selection** **Numeric** and **Value from external data file**.

#### **Single selection** and **Multiple selection** modifiers

Set a name for the new modifiers set by typing it in the **Set name** edit box. Setting a modifiers' set name is not mandatory.

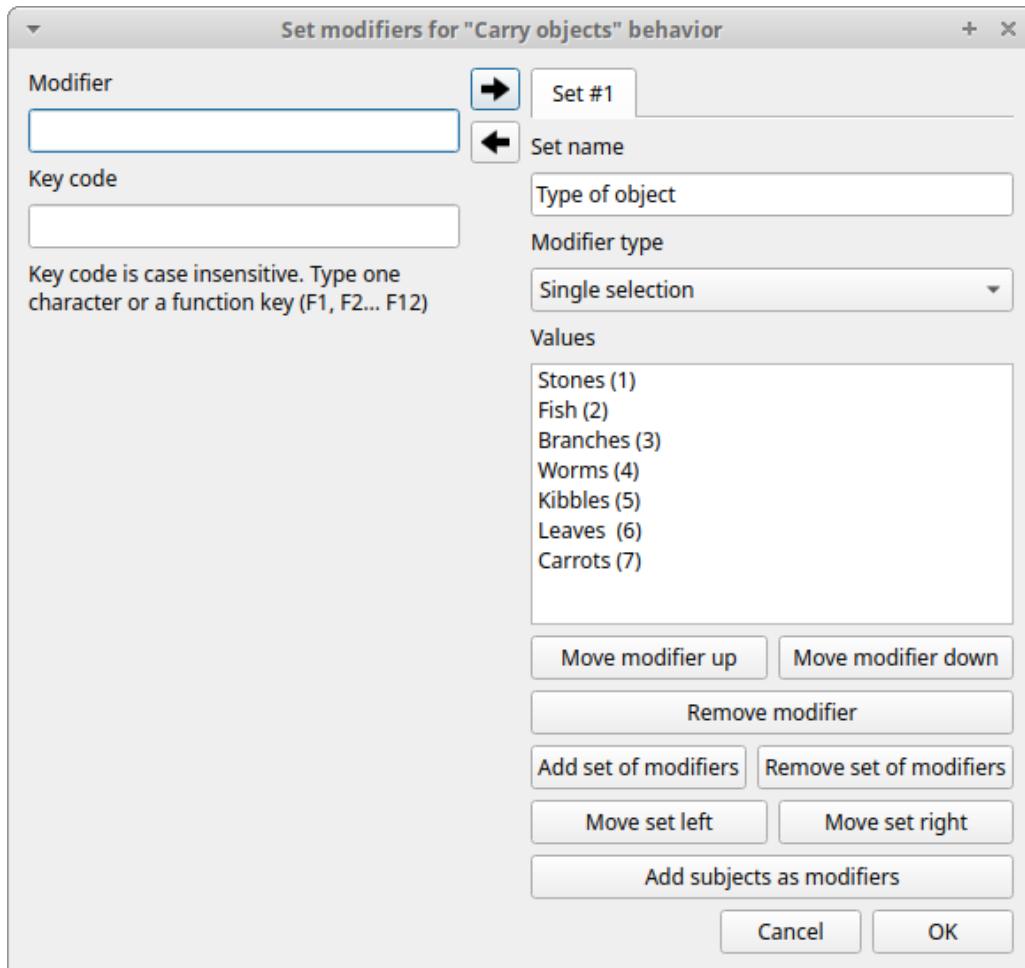
Within a set of modifiers, you can add a modifier by writing the modifier in the **Modifier** edit box. You can choose a shortcut (one character - case sensitive) to this modifier (optional). Then press the **right-arrow** button to add the new modifiers to the set.



To modify a modifier, select it and press the **left-arrow** button, edit the modifier and press the **right-arrow** button.

A modifier can be removed by pressing the **Remove modifier** button.

After adding all modifiers the window will appear like this:



All defined subjects can be added as modifiers using the **Add subjects as modifiers** button. This can help in case of coding the interactions between subjects for example.

The modifiers can be loaded from a plain text file Use the **Load modifiers from file** button.

The modifier position into the modifiers' set can be manually set using the **Move modifier up** and **Move modifier down** buttons. The modifiers can be sorted alphabetically (use the **Sort modifiers** button).

You can add and/or remove sets using the buttons **Add set of modifiers** and **Remove set of modifiers**.

The position of a modifiers' set can be customized (using the **Move set left** and **Move set right** buttons)

Modifiers can not contain the following characters: ()`~!

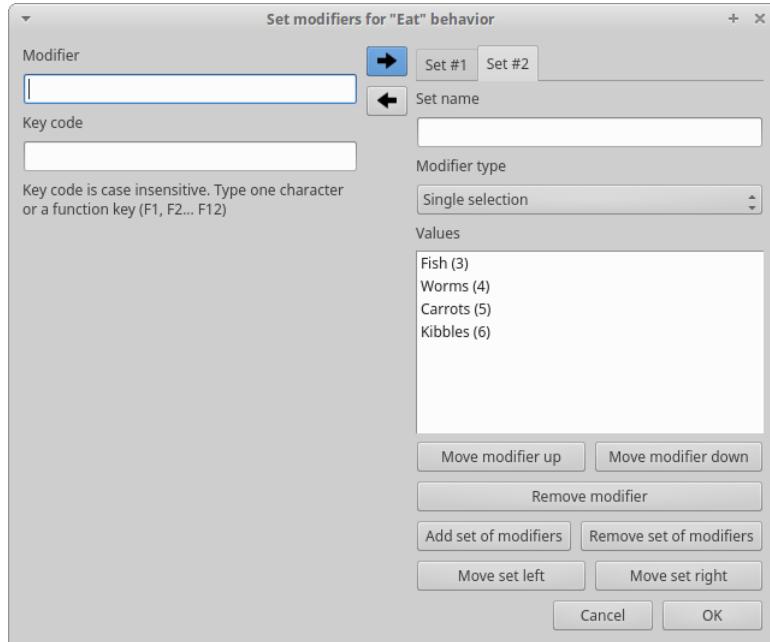
Example of a **multiple selection** modifiers set:



Many values can be selected together.

Example of 2 sets of modifiers:





### Numeric modifier

Set a name for the new set by typing it in the **Set name** edit box. Setting a modifiers' set name is not mandatory.

When a **Numeric** modifier will trigger, BORIS will ask the observer for a numeric value.

### Value from external data file modifier

This modifier can be used to record the value of a variable coming from an external data file (defined during the creation of the observation).

You have to define the variable name in the **Variable name** edit box. This is mandatory and the name of the variable **must** be the same than the variable defined in the observation.

See [External data files](#)



Click **OK** to save modifiers in the **Ethogram** table.

Set the exclusion matrix

The occurrence of an event (State or Point) can exclude the occurrence of a state event. This can be set using the **Behaviors exclusion matrix** window, which can be opened clicking on the **Exclusion matrix** button. BORIS will ask for including **Point events** or not and a new **Exclusion matrix** window will open.

Exclusive behavior may be selected by checking on the corresponding checkbox in the automatically-generated matrix. We suggest to work on the **Exclusion matrix** when all the behaviors have been added to your ethogram.

All behaviors can be excluded by a particular behavior by selecting the corresponding entire row (click on the row header of the behavior) and by clicking on the **Check selected** button. You can also uncheck all behaviors by selecting the **Uncheck selected** button.

Example of an **exclusion matrix**:

Behaviors exclusion matrix

Check behaviors excluded by		Alert	Allogroom	Breed	Carry objects	Chase	Defecate	Dig	Drink	Eat	Interact with enrichment	Locomotion
Alert			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Allogroom	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Breed	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
Carry objects	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓
Chase	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓
Defecate	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
Dig	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
Drink	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
Eat	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
Interact with enrichment	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Locomotion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Look for food	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Manipulate	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Play in the water	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Play on the ground	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rest	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

For example in the previous figure, the **Alert** behavior will exclude the following behaviors: **Allogroom, Breed, Carry objects, Chase ...**

During the observation, the excluding event will stop all the current excluded state events one millisecond before the occurrence of the event.

### Import an ethogram from an existing project

Behaviors within an ethogram can be imported from an existing BORIS project (.boris) using the **Import behaviours from a BORIS project** button. BORIS will ask to select a BORIS project file and whether imported behaviors should replace or be appended to the **Ethogram** table. Imported behaviors will retain all the previously defined behavior parameters (namely Behavior type, Key, Code, Description, Modifiers and Exclusion information).

### Import an ethogram from a JWatcher global definition file (.gdf)

Behaviors can be imported from a JWatcher global definition file (.gdf) using the **Import from JWatcher** button. BORIS will ask to select a JWatcher file (.gdf) and whether imported behaviors should replace or be appended to the **Ethogram** table. Behavior type and exclusion information for the behaviours imported from JWatcher have to be redefined.

### Import an ethogram from a plain text file

Behaviors can be imported from a plain text file using the **Import from text file** button. The fields must be separated by TAB, comma (,) or semicolon (;). All rows must contain the same number of fields.

The fields will be interpreted as:

- 1st column: Behavior type **State event** or **Point event** (mandatory)
- 2nd column: Key (one character - case insensitive)
- 3rd column: Behavior code (must be unique)
- 4th column: behavior category (empty if no category)
- 5th column: Description of behavior(optional)

All fields after the 5th will be ignored.

BORIS will ask to select a plain text file (by default: \*.txt \*.csv \*.tsv) and whether imported behaviors should replace or be appended to the **Ethogram** table. The missing information for the behaviours imported from text file have to be redefined.

## Import an ethogram from a spreadsheet (Google Sheet, Microsoft-Excel or LibreOffice Calc...)

The ethogram can be imported from a spreadsheet. The spreadsheet must contain one behavior by row and have to be organized as above:

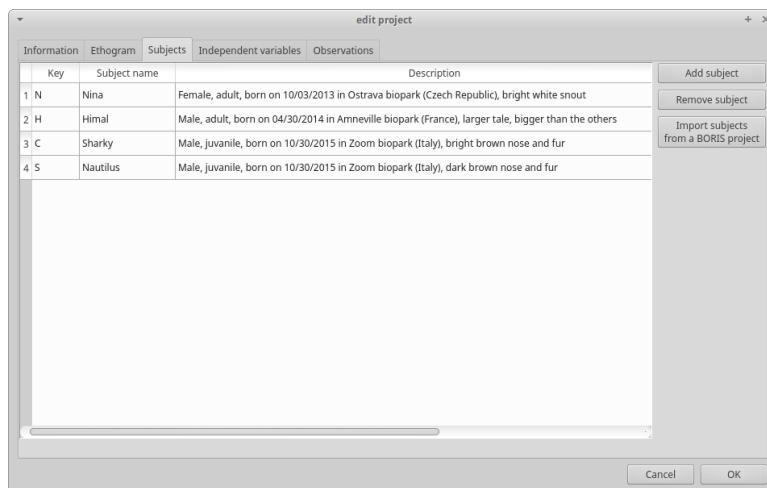
- 1st column: Behavior type: **State event** or **Point event** (mandatory)
- 2nd column: Key (One character - Case sensitive - Optional)
- 3rd column: Behavior code (mandatory - must be unique)
- 4th column: Description of behavior (optional)
- 5th column: Behavior category (optional)

Select all cells of your spreadsheet (CTRL + A), copy to clipboard (CTRL + C). Click the **Import from clipboard** button.

## Export the ethogram

The entire ethogram can be exported in various formats (TSV, CSV, XLS, ODS, HTML). See **File > Edit project > Ethogram tab > Export ethogram**

## Define the subjects



BORIS allow coding behaviors for different subjects within a single observation. The **Subject** table allows specifying subjects using a **Key** (e.g. the **k** on your keyboard), **Subject name** (e.g. "Kanzi"), **Description** (e.g. male, born October 28 - 1980).

In this case, pressing **n** will set "Nina" as the focal subject of the behavioural coding. Pressing **n** again will deselect "Nina" and set to "no focal subject".

The definition of one or more subjects is not mandatory. Addition, removal and sorting of the subjects follows the same logic of the *Ethogram* table (see [Set your ethogram from scratch](#) for info).

From version 7 the keys are **case-sensitive**.

If your project was created with a previous version of BORIS (< v.7) you can use the **Convert keys to lower case** to convert all keys to lower case otherwise you will have to code your observation using upper case key.

The subjects can also be imported from an existing BORIS project: use the **Import Subjects from a BORIS project** button.

## Import subject from a spreadsheet

The subjects can be imported from a spreadsheet (Google spreadsheet, Microsoft-Excel, LibreOffice Calc).

The spreadsheet must contain one subject by row and have to be organized as above:

- 1st column: Subject key (One character - Case sensitive - Optional)

- 2nd column: Subject name (mandatory)
- 3rd column: Description of subject (optional)

Select all cells of your spreadsheet (CTRL + A), copy to clipboard (CTRL + C). Click the **Import from clipboard** button.

### Important

If you open a project file created with a version older than v.7 BORIS will ask you to convert the upper case behavior and subject keys to lower case.

## Define the Independent variables

Label	Description	Type	Predefined value	Set of values
1 Location	Location where observations where made	text	44° 55' 59" N – 7° 25' 18" E	
2 Temperature	Average temperature of the day (°C)	numeric		
3 Visitors	Visitors per day	numeric		
4 Weather	Meteorological conditions	value from set	sun	sun,rain,cloud

Label: Location  
 Description: Location where observations where made  
 Type: text  
 Predefined value: 44° 55' 59" N – 7° 25' 18" E

**Add variable**    **Remove variable**    **Import variables from a BORIS project**

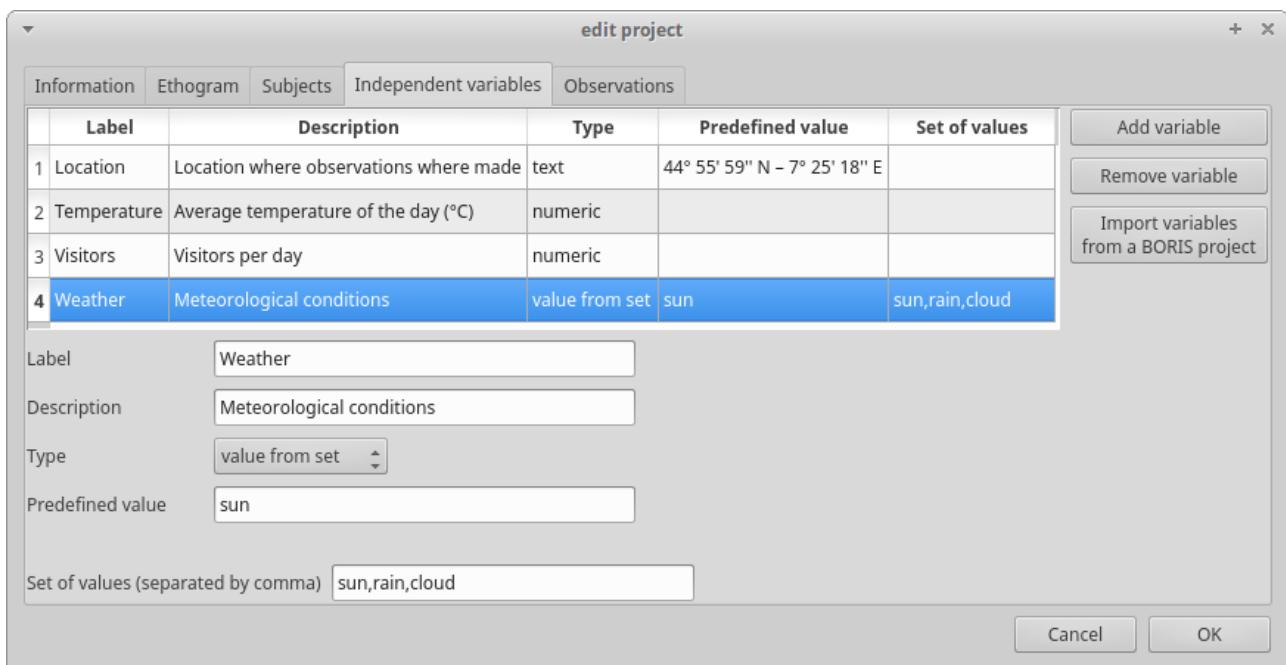
**Cancel**    **OK**

BORIS allows adding information about the observation using **Independent variables**. This can be used to specify factors that may influence the behaviors (e.g. group composition, temperature, weather conditions) but will not change during a single observation within a project. Each independent variable can be defined by a **Label** (e.g. weather), a **Description** (e.g. weather conditions), a **Type** (*text, numeric, value from set or timestamp*).

The values of a set are defined in the **Set of values** column separating the available values with a comma (,). Please note that the first value of the set will be selected by default. It should be useful to define a NA value as first value of every set.

The values for the independent variables will be asked when creating a new observation. Addition, removal and sorting of the independent variables follows the same logic of the **Ethogram** table (see **Set your ethogram from scratch** for info). The independent variables can also be imported from an existing BORIS project using the **Import Variables from a BORIS project**.

Example of independent variable defined as "set of values"



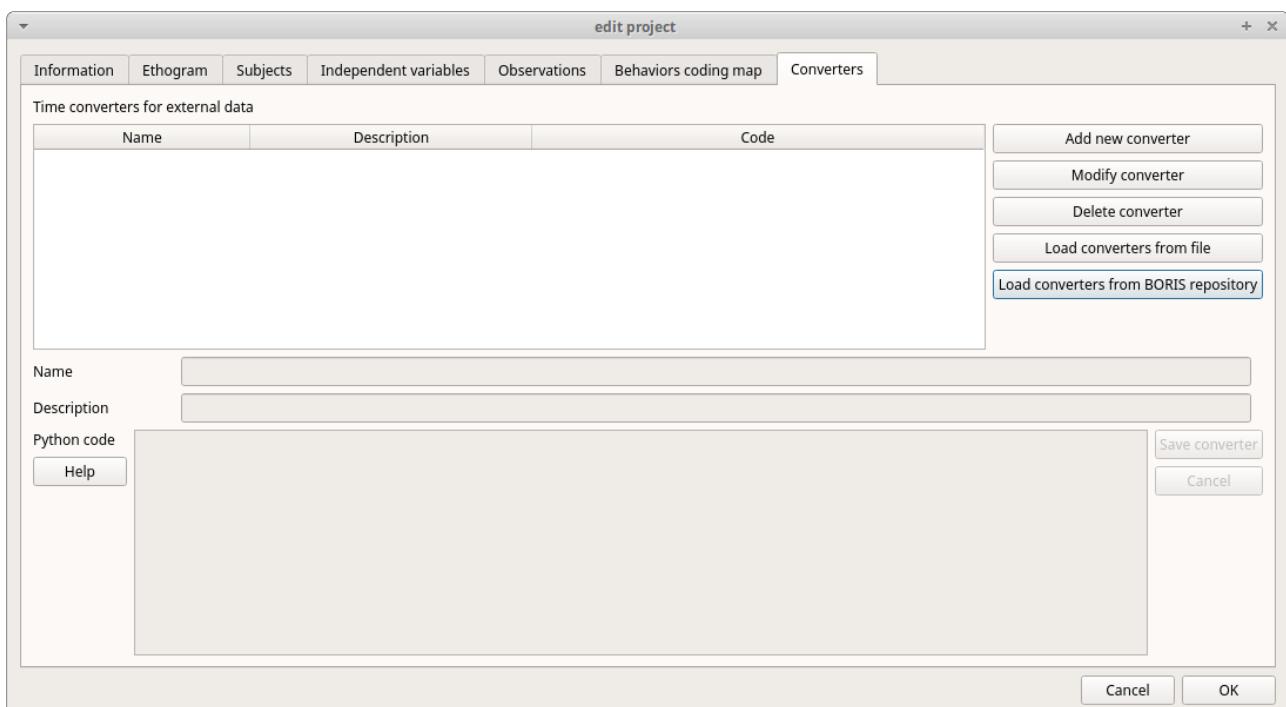
The predefined value must be contained in the set of value.

## Observations' tab

The **Observations** table in BORIS shows information about all the previous observations within a project. A selected "Observation" can be removed using the "Remove observation" button (you will be prompted for confirmation). This operation cannot be undone and deleted observations cannot be recovered once the project is saved. The **Observations** table shows four columns **id Date Description Media**.

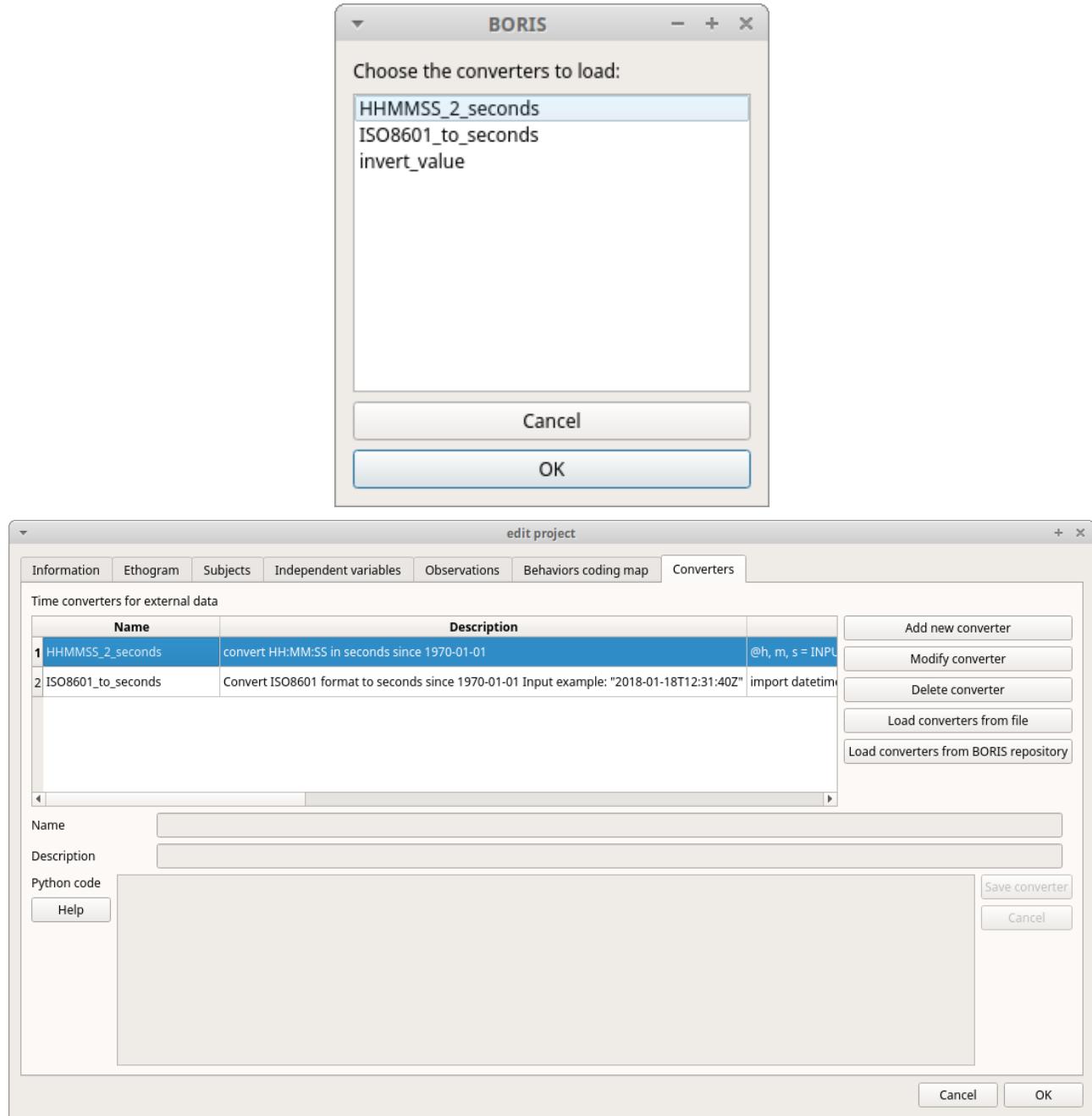
## Converters' tab

Converters are used for plotting external data when the timestamp values are not expressed in seconds. Converters can be written by the user, loaded from file or loaded from the repository of the BORIS web site (<http://www.boris.unito.it/static/converter.json>).



Load converters from BORIS web site

Click **Load converters from BORIS repository** and select the converters to be added to your project.



## Writing a converter

See "[Converters for external data values](#)"

The converters loaded in your project can be then selected for converting timestamp (or other values) in external data file

See [Converters](#)

## Open an existing project with BORIS

### Important

**Due to changes in the project format all the projects created with v. 7+ will not be suitable for previous version of BORIS. The v. 7 will open projects created with previous version and convert them. A copy of your old project will be kept.**

### Important

If you open a project file created with a version older than v.7 BORIS will convert your project file and keep an original version of your project file. BORIS will ask you to convert the upper case behavior and subject keys to lower case.

To open an existing BORIS project, select the menu **File > Open project**. A BORIS project file is a container for all information related to a set of observations as the ethogram, the independent variables, and the subjects. BORIS allows creating an unlimited number of projects but only one project can be opened at a time.

## Import a project

You can import a project from a **Noldus The Observer XT Coding Scheme Exchange** file (.otx or .otb extensions). The Observer XT software allows you to save a project as a template (see the reference manual). This template can be imported into BORIS and then saved as a BORIS project (File > Import a project from ... > Noldus The Observer XT template)

The following information will be imported from the template file:

- Ethogram (coding scheme)
- Modifiers
- Behavioral categories
- Subjects
- Independent variables

The Observer is a registered trademark of Noldus Information Technology b.v. See the [Noldus web site for details](#)

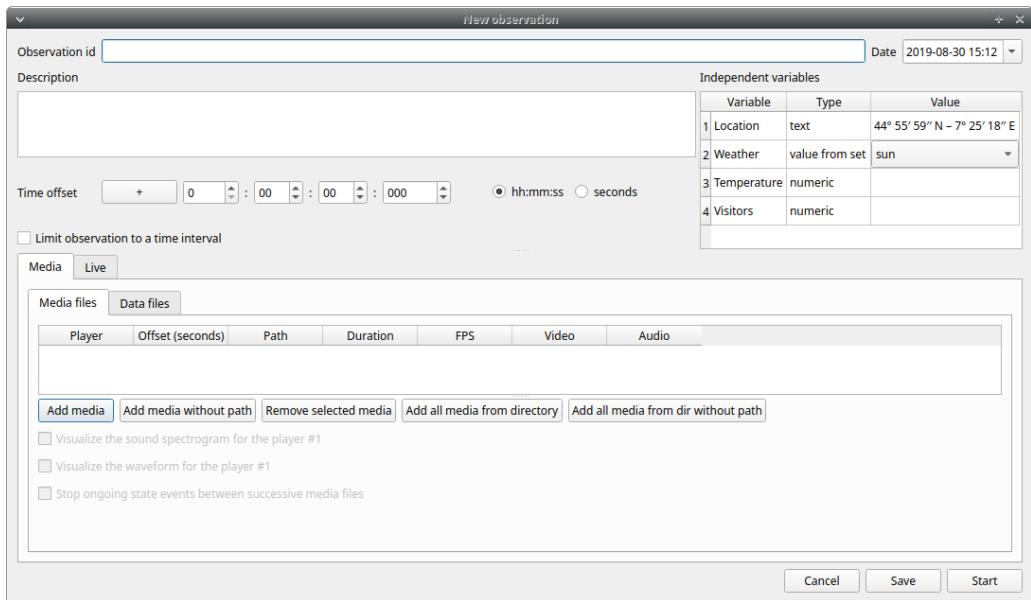
## Observations

### Create a new observation

A video tutorial about making an observation is available at <https://www.youtube.com/watch?v=CKeBITCICXc>

To create a new observation you must first [Create a new project with BORIS](#) or [Open an existing project with BORIS](#).

Clicking on **Observations > New observation** will show the **New observation** window.



This window allow adding various observation data:

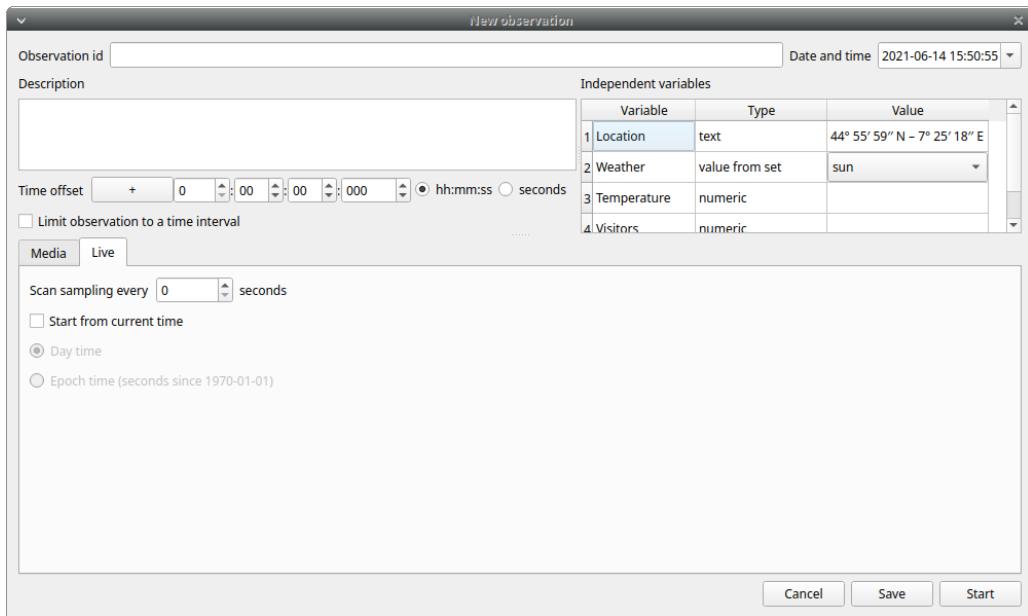
- a mandatory **Observation id** (must be unique across all observations in the open project);
- **Date**, which will be automatically set on the current date and time, but you can alternatively set this info on your media date and time, or whatever you prefer.
- **Description**, which can host all the relevant information about your observation, but can be also left empty.
- **Independent variables** (e.g. to specify factors that may influence the behaviors but will not change during the observation within a project). See the [independent variables](#) section for details.
- **Time offset**. BORIS allow specifying a time offset that can be added or subtracted from the media timecode.
- The **Limit observation to a time interval** option can be used to limit the observation to an arbitrary time interval.

You must then indicate if you want to make an observation based on pre-recorded media (audio / video) or a live observation.

## Live observation

During the live observation BORIS will show you a timer that will be used for recording time for coded events.

Choose the **Live tab** to make a live observation.

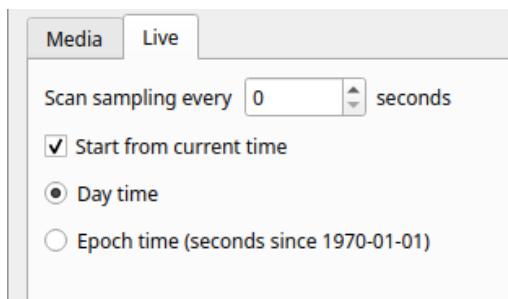


### Scan sampling

In the above tab you can select a time for **Scan sampling** observation. In this case the timer will stop at every time offset you indicated and all the coded events will have the same time value.

#### Start from current time

If you want that the time starts from the current time you can check the **Start from current time** checkbox.



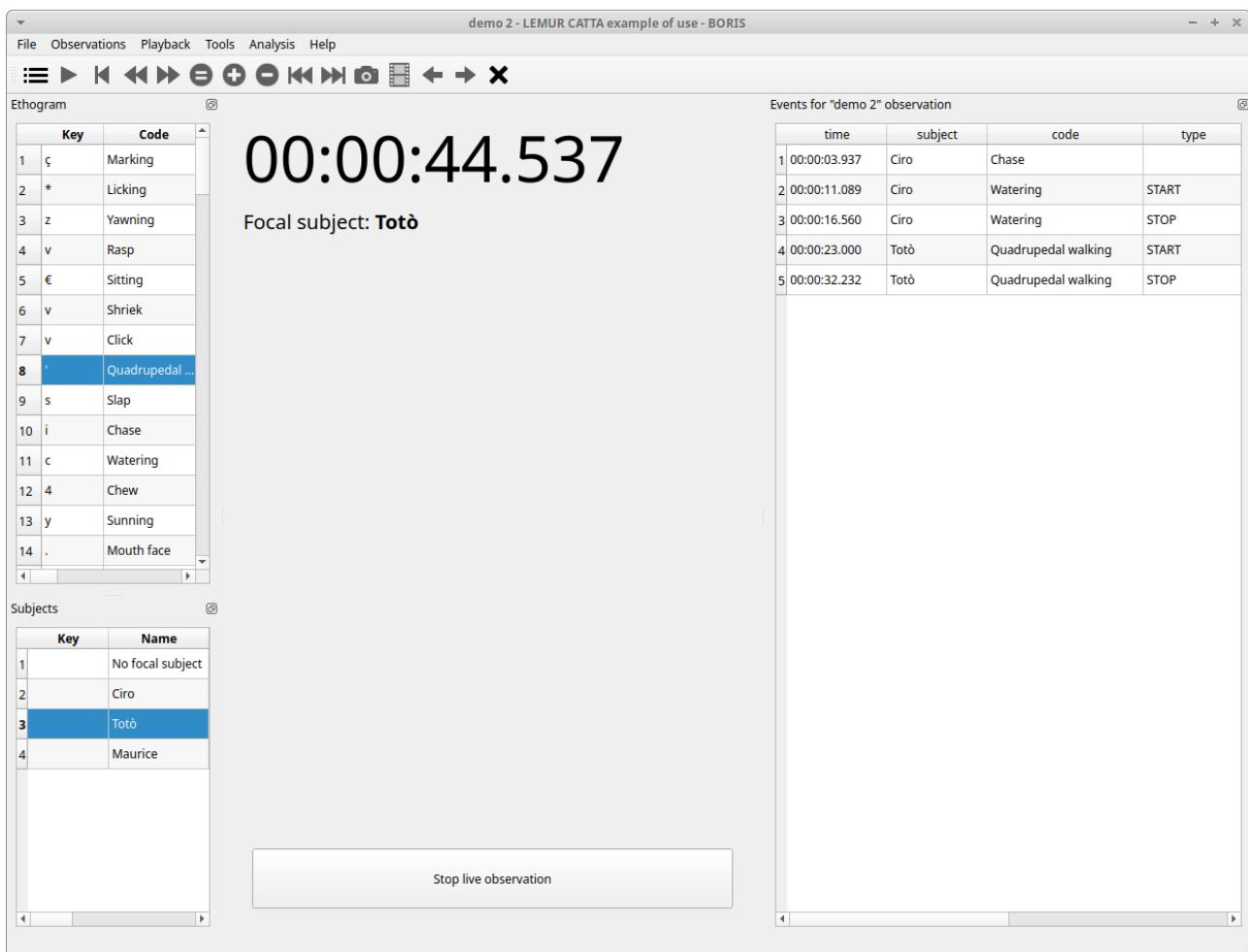
If the **Day time** option is checked the start time will be the computer current time when you will press the **Start** button.

If the **Epoch time** is checked the start time will be the number of seconds since the Jan 1st, 1970 (1970-01-01). See [Unix time](#) for details. This option is useful for long observations (few days) or observations that start before midnight and end after.

#### Start the observation

Click the **Start** button to begin the live observation or **Save** to save it in the [Observations list](#).

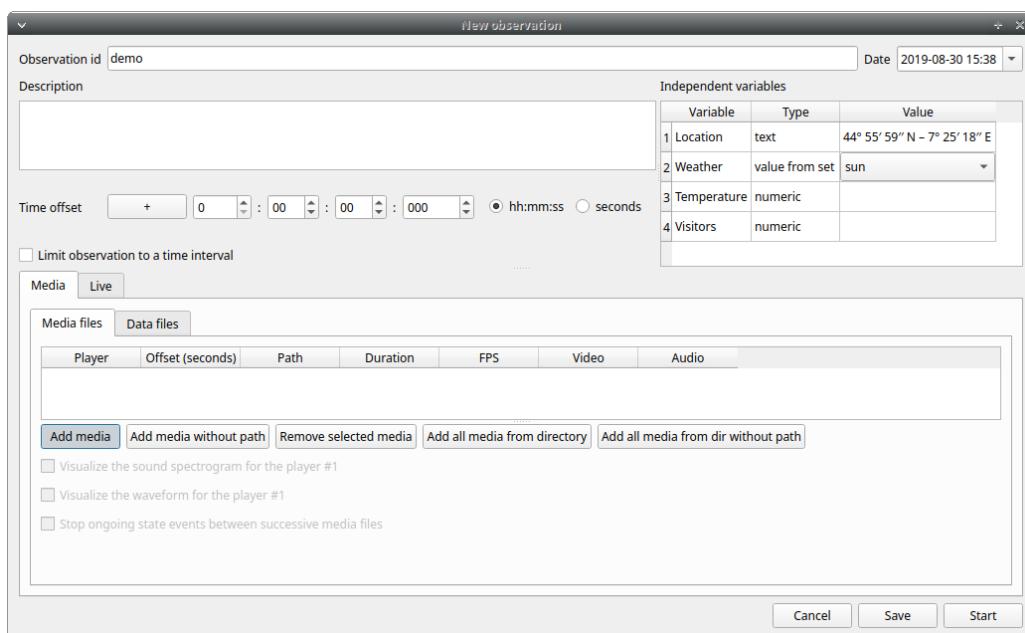
The main window during a live observation will look like this:



See the [Live observations](#) section to start coding.

## Media based observation

Click on the **Media** tab to make an observation based on one or more media files.



The **Media** tab contains 2 tabs: **Media files** and **Data files**.

Click the **Media files** tab and add one or more media files using the **Add media** or **Add media without path** button. The **Add media without path** button allow you to not store the path of your media file(s) in the BORIS project file. In this case the media file(s) must be in the same directory than the BORIS project file to be found. This option is useful if you are working with various computers on the same project file(s).

If you want to remove the media file paths of an entire project see [Removing path of media files](#).

The **Add all media from directory** and **Add all media from dir without path** allow to add all media files found in a directory to the playlist.

Information about the selected media file will be extracted and displayed in the media list: media file path, media duration, number of frames by second (FPS), the presence of a video stream, the presence of an audio stream .

The screenshot shows the BORIS interface with the 'Media' tab selected. A single media file, 'video1.mp4', is listed in the 'Media files' section. The table columns are: Player, Offset (seconds), Path, Duration, FPS, Video, and Audio. The values are: Player 1, Offset 0, Path 'video1.mp4', Duration '00:02:59.960', FPS 25, Video True, Audio False. Below the table are several buttons: 'Add media', 'Add media without path', 'Remove selected media', 'Add all media from directory', and 'Add all media from dir without path'. There are also two checkboxes: 'Visualize spectrogram for the player #1' and 'Stop ongoing state events between successive media files'.

The dropdown list in the first column allow you to choose a player (for a maximum of 8). If you want to observe more media files simultaneously you must use consecutive players (starting from 1). See example below:

The screenshot shows the BORIS interface with the 'Media' tab selected. Three media files are listed in the 'Media files' section, each assigned to a different player. The table columns are: Player, Offset (seconds), Path, Duration, FPS, Video, and Audio. The entries are: Player 1, Offset 0, Path 'video1.mp4'; Player 2, Offset 0, Path 'video2.mp4'; Player 3, Offset 0, Path 'video3.mp4'. All three entries have Duration '00:02:59.960', FPS 25, Video True, and Audio False. The 'Offset (seconds)' column shows the offset for each player: 0, 0, and 0 respectively.

If you have to synchronize 2 (or more) videos you can use the **Offset column** to indicate when the 2nd player should start. For example if the video loaded in the second player starts 15 seconds after the first video you have to input **15** in the **Offset** cell. If the second video starts before the first player you can set a negative value in the **Offset** cell

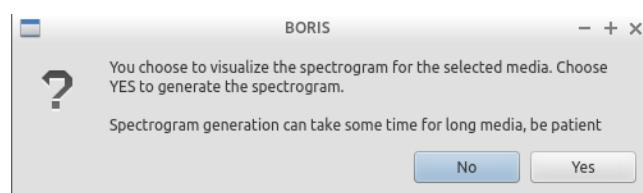
If you have to play sequentially many videos you have to select the same player (#1) for all video you have loaded. This means that an event occurring at time  $t_x$  in the media file queued as second (e.g. second\_video.mp4) in the playlist will be scored as happening at time  $t_1 + t_x$  (where  $t_1$  is the duration of the first media file, e.g. first\_video.mp4).

The **Remove media** button can be used to remove all the selected media files.

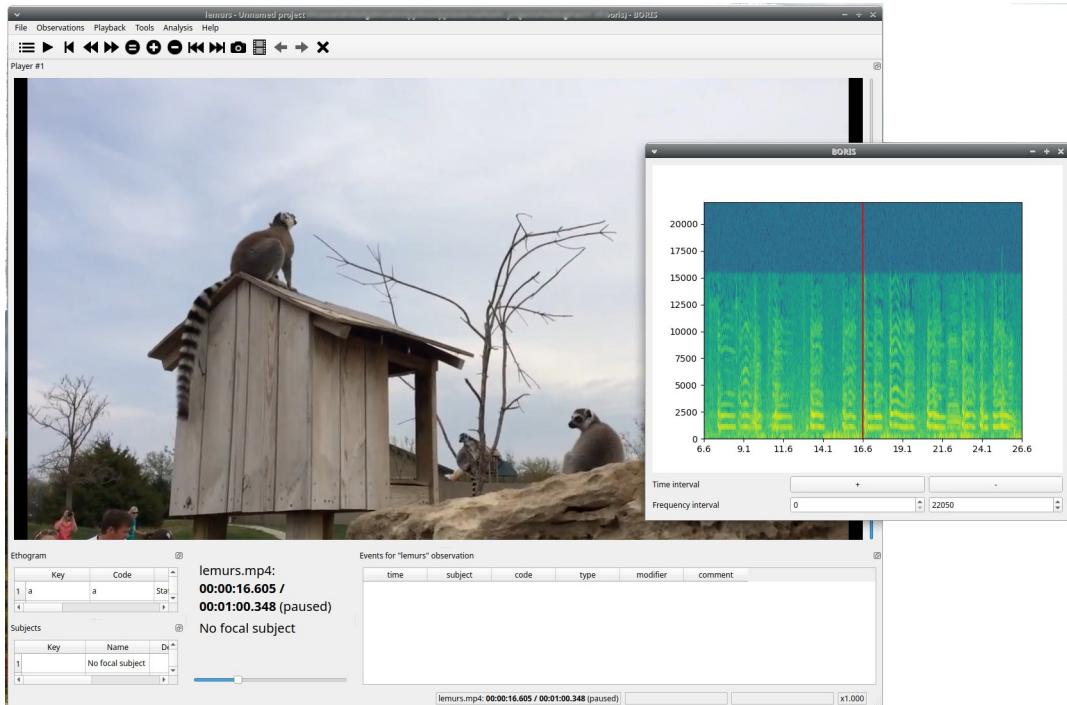
All the media types reported at <http://www.videolan.org/vlc/features.html> can be played in BORIS.

#### Spectrogram visualization

BORIS allow you to visualize the sound spectrogram during the media observation. Activate the **Visualize spectrogram** check box. BORIS will ask you to generate the spectrograms for all media files loaded in the first player.



The spectrogram visualization will be synchronized to the media position during the observation.



Close current behavior between videos

This option is disabled for now.

External data files

### Warning

**At this time only 2 external data can be plotted with your media file**

You can select one or more external data files to be plotted synchronously with your media. Click the **Data files** tab and use the **Add data file** button to select a data file.

Media files	Data files							
Data files to plot								
Path	Columns to plot	Plot title	Variable name	Converters	Time interval (s)	Start position (s)	Subtract first value	Color
<input type="button" value="Add data file"/> <input type="button" value="Add data file without path"/> <input type="button" value="View first rows"/> <input type="button" value="Show plot"/> <input type="button" value="Remove selected data file"/>								

The data files must be plain text files with at least **2 columns** separated by comma or TAB. One column must contain a timestamp that will be used to synchronize the plot with the media. The sampling rate can be variable.

Example of a plain text data file with 5 columns separated by comma (,):

```
Display,X Pos,Y Pos,Start Time (secs),Pupil Diameter
1,864,509,549.233,0.00295773451216519
1,863,505,549.25,0.00281810853630304
1,863,503,549.266,0.00287826382555068
```

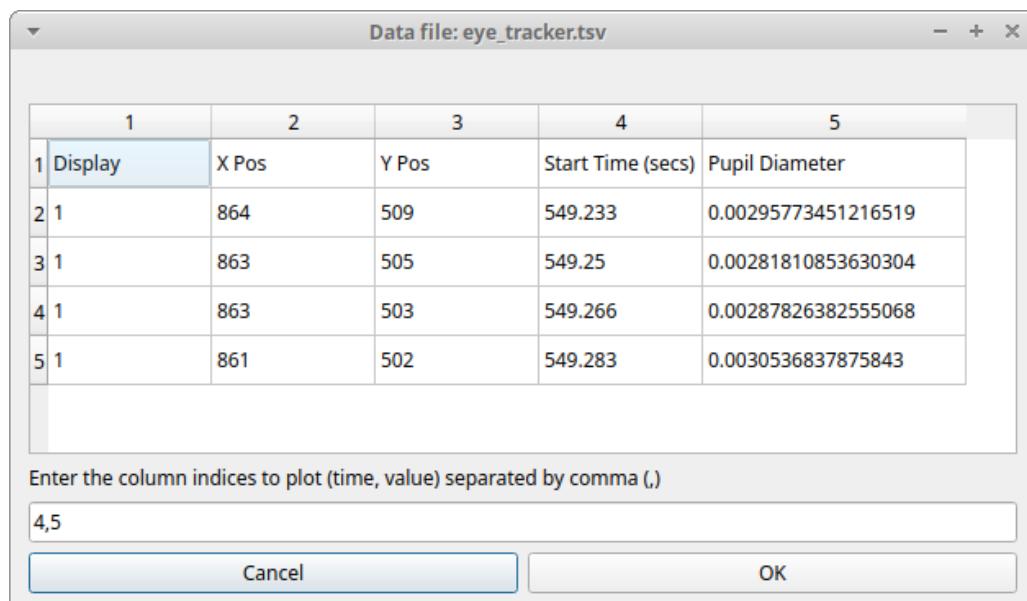
```

1,861,502,549.283,0.0030536837875843
1,858,501,549.3,0.00308083021081984
1,856,499,549.316,0.00306266942061484
1,854,499,549.333,0.00305776367895305
[...]

```

In the above example the 4th column contains the timestamp and the 5th the value to be plotted.

Input the index of the column containing the timestamp and the index of the column containing the value to be plotted. The two indices must be separated by a comma (,). Click **OK** to close the window.



A new row will be added in the data files table.

Path	Columns to plot	Plot title	Variable name	Converters	Time interval (s)	Start position (s)	Subtract first value	Color

Add data file    Add data file without path    View first rows    Show plot    Remove selected data file

You can modify/complete the following parameters by directly typing in the table cells:

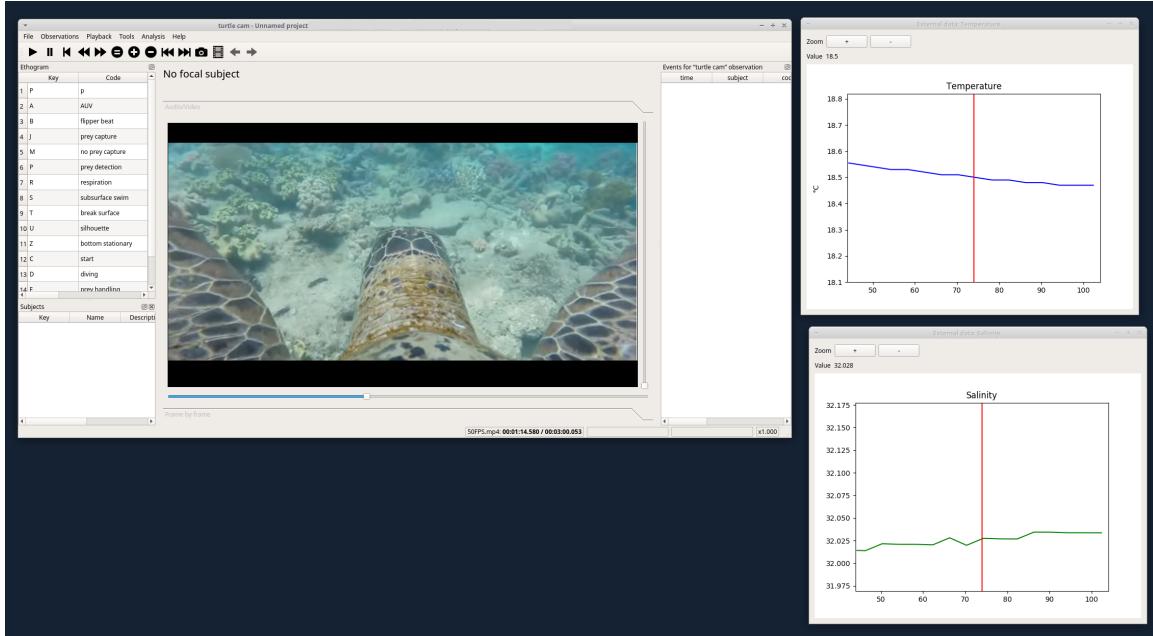
- Columns to plot
- Plot title: the title of the plot
- Variable name
- Converters: Used if the timestamp is not expressed in seconds (see below for details)
- Time interval: The time interval that will be plotted (in seconds)
- Start position: the start position of data for synchronisation with the media (in seconds)
- Subtract first value: if the timestamp does not start with a 0 value you can choose to subtract the first value to all timestamp values.
- Color: the color of the color

**NOTE :** if you want to record the value of the plotted variable in a modifier of a behavior (see [Value from external data file modifier](#)) the modifier must have the same **variable name**.

You can check if the data from file can be correctly plotted by using the **Show plot** button. If the data are compatible you will see a plot otherwise you will obtain a message with an explanation.

For now only 2 values can be plotted synchronously with your media file. The values can come from the same file or from two different files.

During the observation the values you have selected in external data files will be plotted synchronously with your media file.



## Converters

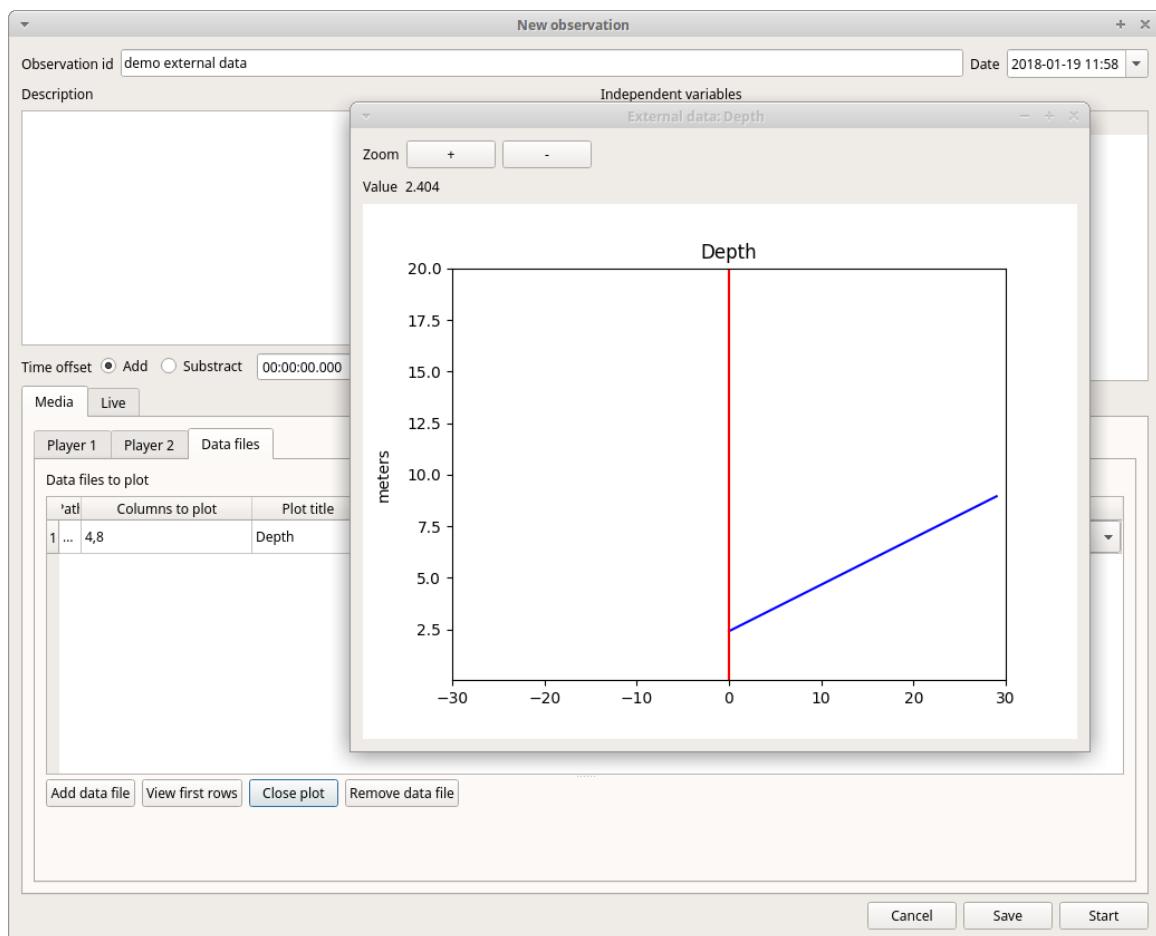
If the values in the timestamp column are not expressed in seconds (like 12.45) but in another format (HH:MM:SS, MM:SS, ISO8601 2018-01-18T12:31:40Z ...) you must use a converter that will convert the current format in seconds.

See the Converters' tab in the project configuration: [converters tab](#)

A **double-click** on the converters cell will allow you to select a converter for each column to be plotted

The screenshot shows the 'Converters' configuration dialog. It displays a dropdown menu for 'Column #4' with options: 'None', 'HHMMSS\_2\_seconds' (which is selected and highlighted in blue), 'ISO8601\_to\_seconds', and 'invert\_value'. Below the dropdown, there is a 'Data files to plot' table with one row. The table has columns: Path, Columns to plot, Plot title, Variable name, Converters, Time interval (s), and Start position (s). The row contains: '/home/user/external\_data2.csv', '4,8', 'Depth', 'meters', '{4: 'HHMMSS\_2\_seconds'}', '60', and '0'.

Use the **Show plot** button to verify if your external data can be plotted without problem. The **Close plot** button will close the plot window.



Converters can also be used to convert values that are not time value.

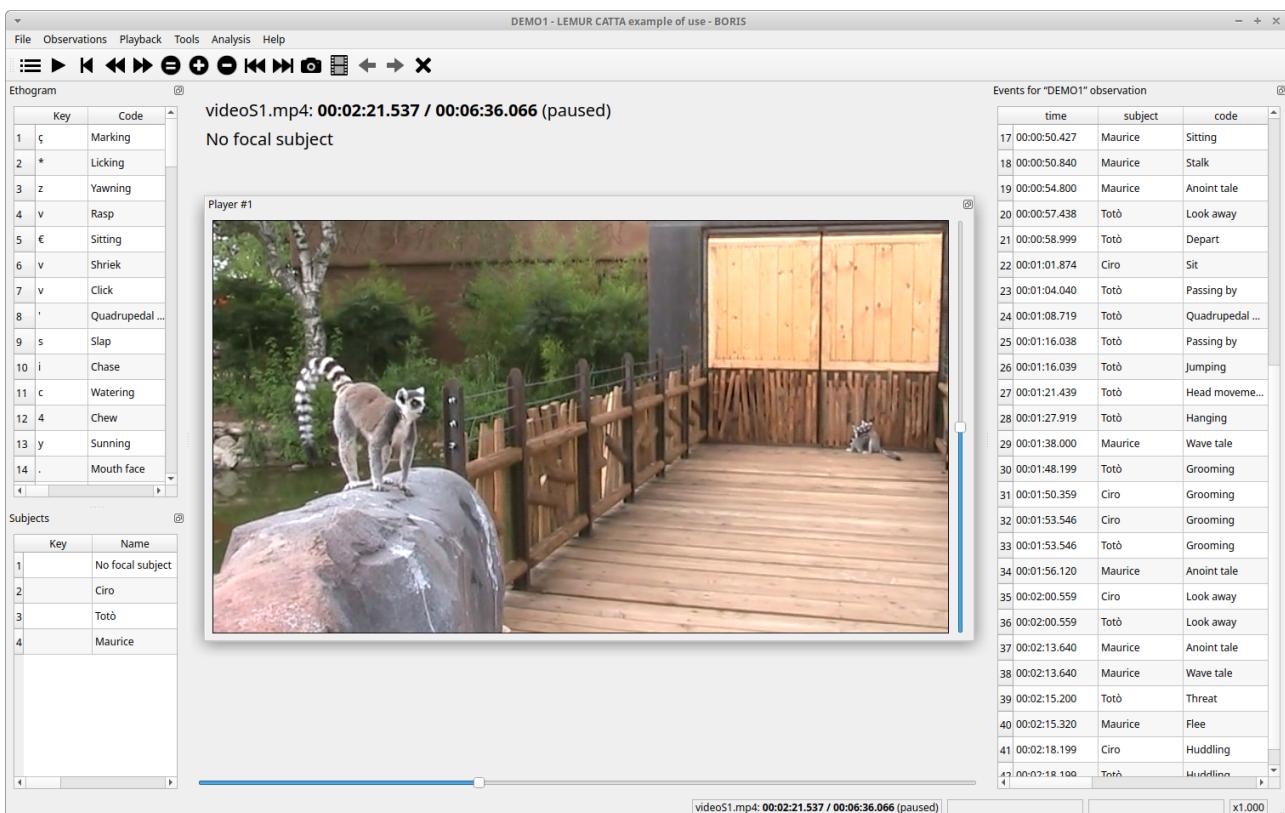
Example of a converter for inverting value:

```
OUTPUT = - float(INPUT)
```

Start the observation

Click the **Start** button to start coding. The **Observation** window will be closed and you'll be transferred to the main **BORIS** window. If you do not want to start the observation click the **Save** button. The observation will be saved in the **observations list**.

The main window during the observation of a single media file will look like this:



See the [media coding](#) section to start coding.

### Limit observation to a time interval

This option can be used to limit the observation to a time interval for live or media based observations.



### Media based observation

When the observation will start the media will be automatically positioned to the **Start time** value and the player will stop when the video time will reach the **Stop time** value.

### Live observation

The observation will start at time 0 (or the current time if the **Start from current time** option is enabled). The **Start time** of the time interval will not be applied. The observation will stop when the **Stop time** will be reached.

## Observations list

The **Observations > Observations list** will show you all the observations contained in the current BORIS project.

The following values are displayed: the observation id, the description of observation, the coded subjects, the observation duration (as the difference between the last recorded event and the first one), the percent of exhaustivity of the coding (as the sum of the length of the coded events divided by the observation duration), the media file(s) or LIVE in case of live observation, the values of the independent variables.

Observations list - BORIS												
1442 observations												
	id	date	contains	description	subjects	observation duration	exhaustivity %	media	Location	Weather	Temperature	Visitors
1	0001_a	2016-05-17 00:00:31		Vegetation	Himal, Nautilus	32.400	100.0	#1: 20160517162539.m2ts 44° 55' 59" N - 7° 25' 18" E sun		25	1046	
2	0001_b	2016-05-17 00:00:31		Vegetation	Himal, Nautilus	32.400	98.5	#1: 20160517162539.m2ts 44° 55' 59" N - 7° 25' 18" E sun		26	1046	
3	0002	2016-05-17 00:00:24		Vegetation	Sharky, Himal, Nautilus	34.470	89.6	#1: 20160517162540.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
4	0003	2016-05-17 00:00:05		Vegetation	Nina, Sharky, Himal, Nautilus	22.500	87.1	#1: 20160517162641.m2ts 44° 55' 59" N - 7° 25' 18" E sun		25	1046	
5	0004	2016-05-17 00:00:59		Central trunks	Nina, Sharky, Himal, Nautilus	50.697	72.9	#1: 20160517162692.m2ts 44° 55' 59" N - 7° 25' 18" E sun		26	1046	
6	0005	2016-05-17 00:00:49		In the pool	Sharky, Nautilus	40.770	100.0	#1: 20160517163131.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
7	0006	2016-05-17 00:00:42		In the pool	Sharky, Himal, Nautilus	61.830	73.3	#1: 20160517163231.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
8	0007	2016-05-17 00:00:13		In the pool	Nina, Sharky, Himal	124.986	41.3	#1: 20160517163347.m2ts 44° 55' 59" N - 7° 25' 18" E sun		25	1046	
9	0008	2016-05-17 00:00:17		In the pool	Sharky, Himal, Nautilus	64.800	85.1	#1: 20160517163743.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
10	0009	2016-05-17 00:00:10		In the pool	Nina, Sharky, Himal	46.277	84.6	#1: 20160517163927.m2ts 44° 55' 59" N - 7° 25' 18" E sun		25	1046	
11	0010	2016-05-17 00:00:57		Area near the glass window	Sharky, Himal, Nautilus	16.779	81.1	#1: 20160517164021.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
12	0011	2016-05-17 00:00:50		Area near the glass window	Nina, Sharky, Himal, Nautilus	25.101	49.1	#1: 20160517164206.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
13	0012	2016-05-17 00:00:45		Area near the glass window	Nina, Sharky, Nautilus	78.210	62.4	#1: 20160517164204.m2ts 44° 55' 59" N - 7° 25' 18" E sun		26	1046	
14	0013	2016-05-17 00:00:25		Central trunks	Nina, Sharky, Himal, Nautilus	63.631	66.2	#1: 20160517164715.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
15	0014	2016-05-17 00:00:52		In the pool	Nina, Sharky, Himal, Nautilus	242.596	53.5	#1: 20160517164729.m2ts 44° 55' 59" N - 7° 25' 18" E sun		24	1046	
16	0015	2016-05-17 00:00:18		Central trunks								

The observations can be sorted by clicking in the desired column header (alphabetic order ascending or descending).

## Checking the observations

The status of observation is displayed in the first column (**Observation id**). If the background of this column is red the observations has one or more UNPAIRED state events. These UNPAIRED observations will not be analyzed. See [Fix unpaired state events](#) for details.

Observations list - BORIS												
1443 observations												
	id	date	contains	description	subjects	observation duration	exhaustivity %	media	Location	Weather	Temperature	Visitors
1	0001_a	2016-05-17 00:00:31		Vegetation	Himal, Nautilus	32.400	100.0					
2	0001_b	2016-05-17 00:00:31		Vegetation	Himal, Nautilus	32.400	98.5					
3	0002	2016-05-17 00:00:24		Vegetation								
4	0003	2016-05-17 00:00:05		Vegetation	Nautilus, Himal, Sharky	34.470	89.6					
5	0004	2016-05-17 00:00:59		Central trunks	Sharky, Nina, Himal, Nautilus	22.500	87.1					
6	0005	2016-05-17 00:00:49		In the pool	Sharky, Nina, Himal, Nautilus	50.697	72.9					
7	0006	2016-05-17 00:00:42		In the pool	Nautilus, Sharky	40.770	100.0					
8	0007	2016-05-17 00:00:13		In the pool	Nina, Himal, Sharky	124.986	43.5					
9	0008	2016-05-17 00:00:17		In the pool	Nautilus, Himal, Sharky	64.800	85.1					
10	0009	2016-05-17 00:00:10		In the pool	Nina, Himal, Sharky	46.277	84.6					
11	0010	2016-05-17 00:00:57		Area near the glass window	Nautilus, Himal, Sharky	16.779	81.1					
12	0011	2016-05-17 00:00:50		Area near the glass window	Nina, Nautilus, Himal, Sharky	25.101	49.1					
13	0012	2016-05-17 00:00:45		Area near the glass window	Nina, Nautilus, Sharky	78.210	62.4					
14	0013	2016-05-17 00:00:25		Central trunks	Sharky, Nina, Himal, Nautilus	63.631	66.2					
15	0014	2016-05-17 00:00:52		In the pool	Nina, Nautilus, Himal, Sharky	242.596	53.5					
16	0015	2016-05-17 00:00:18		Central trunks	Sharky, Nina, Himal, Nautilus	111.302	49.7					

## Filtering the observations

The observations list can be filtered selecting a field and a condition in the drop-list boxes.

In the following example observations are filtered: only observations with **description** containing the **In the pool** subject are shown:

Observations list - BORIS

93 observations

description	contains	in the pool
1 0005	2016-05-17 00:00:49	In the pool
2 0006	2016-05-17 00:00:42	In the pool
3 0007	2016-05-17 00:00:13	In the pool
4 0008	2016-05-17 00:00:17	In the pool
5 0009	2016-05-17 00:00:10	In the pool
6 0014	2016-05-17 00:00:52	In the pool
7 0185	2016-05-28 00:00:23	In the pool
8 0212	2016-05-31 00:00:53	In the pool
9 0217	2016-05-31 00:00:49	In the pool
10 0220	2016-05-31 00:00:36	In the pool
11 0359	2016-06-06 00:00:54	In the pool
12 0411	2016-06-07 00:00:53	In the pool
13 0412	2016-06-07 00:00:21	In the pool

Cancel Start View Edit

Observations can be filtered with **Independent variables** values.

The following example displays only the observations that do not contain "Sunny" in the **Weather** independent variable :

Observations list - BORIS

1430 observations

Weather	does not contain	sun
1 0008	2016-05-17 00:00:17	In the pool
2 0010	2016-05-17 00:00:57	Area near the glass window
3 0011	2016-05-17 00:00:50	Area near the glass window
4 0013	2016-05-17 00:00:25	Central trunks
5 0014	2016-05-17 00:00:52	In the pool
6 0015	2016-05-17 00:00:18	Central trunks
7 0016	2016-05-17 00:00:58	Central trunks
8 0017	2016-05-17 00:00:37	Suspended trunks
9 0018	2016-05-17 00:00:18	Central trunks
10 0019	2016-05-17 00:00:23	Central trunks
11 0020	2016-05-25 00:00:24	Indoor entrance
12 0021	2016-05-25 00:00:47	Indoor entrance
13 0022	2016-05-25 00:00:30	Indoor entrance

Cancel Start View Edit

Observations with a value of **Temperature** independent variable between 18 and 22:

Observations list - BORIS

10 observations

Temperature between (use and to separate terms) 24 and 26

id	date	description	subjects	media	Location	Weather	Temperature
1 0001_a	2016-05-17 00:00:31	Vegetation	Nautilus, Himal	#1: 20160517162539.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	25
2 0001_b	2016-05-17 00:00:31	Vegetation	Nautilus, Himal	#1: 20160517162539.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	26
3 0002	2016-05-17 00:00:24	Vegetation	Sharky, Nautilus, Himal	#1: 20160517162540.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	24
4 0003	2016-05-17 00:00:05	Vegetation	Nina, Sharky, Nautilus, Himal	#1: 20160517162641.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	25
5 0004	2016-05-17 00:00:59	Central trunks	Sharky, Nina, Nautilus, Himal	#1: 20160517162952.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	26
6 0005	2016-05-17 00:00:49	In the pool	Sharky, Nautilus	#1: 20160517163131.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	24
7 0006	2016-05-17 00:00:42	In the pool	Sharky, Nautilus, Himal	#1: 20160517163231.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	24
8 0007	2016-05-17 00:00:13	In the pool	Nina, Sharky, Himal	#1: 20160517163347.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	25
9 0009	2016-05-17 00:00:10	In the pool	Sharky, Nina, Himal	#1: 20160517163927.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	25
10 0012	2016-05-17 00:00:45	Area near the glass window	Sharky, Nina, Nautilus	#1: 20160517164204.m2ts	44° 55' 59"N - 7° 25' 18"E	sun	26

Cancel Start View Edit

Observations with a value of **Visitors** independent variable greater than 1000:

Observations list - BORIS

825 observations

Visitors > 1000

id	date	description	subjects	medi	Location	Weather	Temperature	Visitors
4 0003	2016-05-17 00:00:05	Vegetation	Nina, Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	25	1046	
5 0004	2016-05-17 00:00:59	Central trunks	Sharky, Nina, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	26	1046	
6 0005	2016-05-17 00:00:49	In the pool	Sharky, Nautilus	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	24	1046	
7 0006	2016-05-17 00:00:42	In the pool	Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	24	1046	
8 0007	2016-05-17 00:00:13	In the pool	Nina, Sharky, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	25	1046	
9 0008	2016-05-17 00:00:17	In the pool	Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	
10 0009	2016-05-17 00:00:10	In the pool	Sharky, Nina, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	25	1046	
11 0010	2016-05-17 00:00:57	Area near the glass window	Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	
12 0011	2016-05-17 00:00:50	Area near the glass window	Sharky, Nina, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	
13 0012	2016-05-17 00:00:45	Area near the glass window	Sharky, Nina, Nautilus	#1: ... 44° 55' 59"N - 7° 25' 18"E	sun	26	1046	
14 0013	2016-05-17 00:00:25	Central trunks	Nina, Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	
15 0014	2016-05-17 00:00:52	In the pool	Nina, Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	
16 0015	2016-05-17 00:00:18	Central trunks	Nina, Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	
17 0016	2016-05-17 00:00:58	Central trunks	Nina, Sharky, Nautilus, Himal	#1: ... 44° 55' 59"N - 7° 25' 18"E			1046	

Cancel Start View Edit

## Delete observations

The observations can be deleted from the project using the following procedure: **File > Edit project > Observation's tab**

Select all observations to remove

Click the **Remove selected observations** button and confirm the deletion.

Please note that the deletion is irreversible.

## Import observations from a project file

The **Observations > Import observations** option allows to import observations from a BORIS project file into the current project. Choose the project file and then the observations to import. BORIS will check if observations with same id are already existing in the current project. BORIS will also check if behaviors and/or subjects used in the imported observations are not defined in the current project.

# Coding your media

When looking at the BORIS main window, the window title bar shows the **Observation id - Project name - BORIS**. The media (the first in the queue) will be loaded in the media player and paused.

## Media based coding

Media controls are available in the toolbar



Key to the symbols:

- **List of observations**
- **Play** (become **Pause** when media is played)
- **Rewind** reset your media at the beginning
- **Fast backward** jumps for n seconds backward in your media (See [general preferences](#) to set n)
- **Fast forward** jumps for n seconds forward in your media (See [general preferences](#) to set n)
- **Set the playback speed to 1x**
- **Increase the playback speed** (See [general preferences](#) to set the step value)
- **Decrease the playback speed** (See [general preferences](#) to set the step value)
- **Jump to the previous media file**
- **Jump to the next media file**
- **Take a snapshot** of current video or frame
- **Move on frame back**
- **Move one frame forward**
- **Close current observation**
- **Time budget of the current observation**
- **Plot events of the current observation**
- **Find in events**

The media can be controlled by special keyboard keys:

- **Page Up** key: switch to the next media
- **Page Down** key: switch to the previous media
- **Up arrow** key: jump forward in the current media
- **Down arrow** key: jump backward in the current media
- **Home** key: Increase the playback speed (See [general preferences](#) to set the step value)
- **End** key: Decrease the playback speed (See [general preferences](#) to set the step value)
- **Backspace**: Set the playback speed to 1x
- **Left arrow** key: go to the previous frame
- **Right arrow** key: go to the next frame

## Live observations

During a live observation the media control toolbar is disabled.

Press the **Start live observation** button to start your observation. If some events are already coded BORIS will ask you for deleting them.

A timer will be displayed. The events will be recorded in the events widget.

See the following screenshot.

The screenshot shows the BORIS software interface with the following components:

- Ethogram:** A table of behavioral keys and their corresponding codes. Key 8 ('Quadrupedal ...') is currently selected.
- Subjects:** A table listing subjects with their names. Subject 3 ('Totò') is selected.
- Events:** A table showing recorded events for "demo 2" observation. The events are:
 

time	subject	code	type
1 00:00:03.937	Ciro	Chase	
2 00:00:11.089	Ciro	Watering	START
3 00:00:16.560	Ciro	Watering	STOP
4 00:00:23.000	Totò	Quadrupedal walking	START
5 00:00:32.232	Totò	Quadrupedal walking	STOP
- Timer:** A large digital clock displaying **00:00:44.537**.
- Buttons:** A toolbar at the top with various playback and control icons.
- Stop button:** A button labeled "Stop live observation" located below the subjects list.

## Ethogram widget in the main window

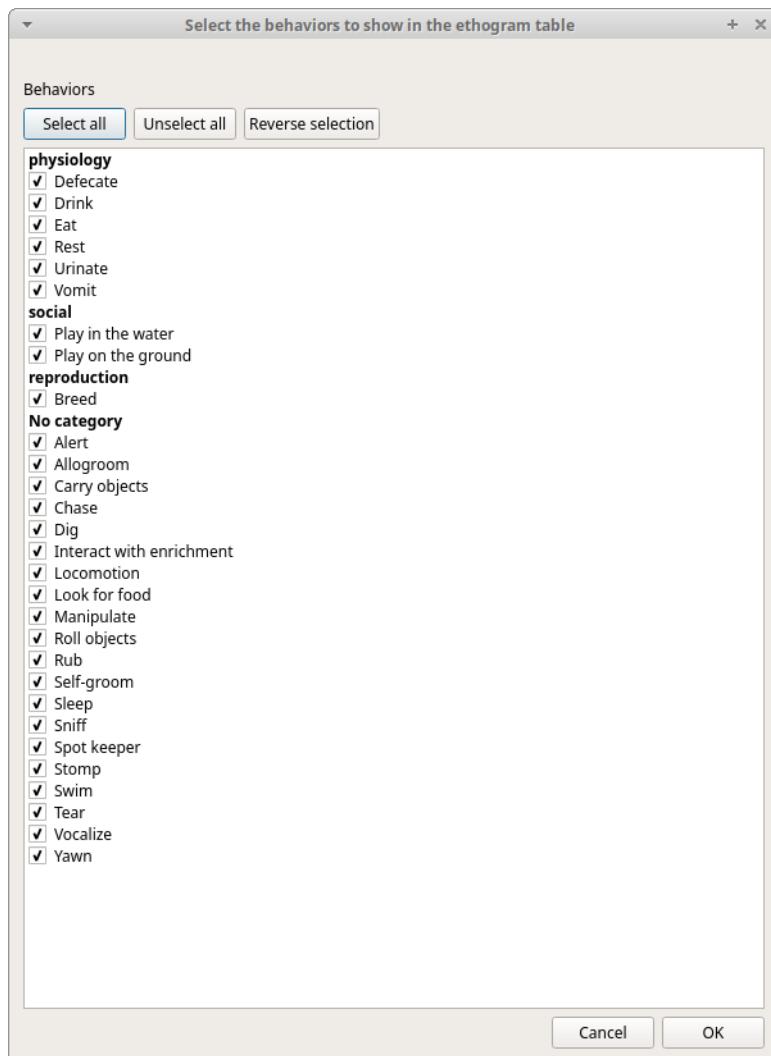
Ethogram							
Key	Code	Type	Description	Category	Modifiers	Excluded	▲
1 A	Alert	State event	Otter is stationary and directs its attention to...		{}	Allogroom,Breed,Carry ob...	
2 Q	Allogroom	State event	Otter licks or scratches with forepaws or hind...		{}	Alert,Breed,Carry objects,	
3 B	Breed	State event	Otters are engaged in copulatory activities (H...	reproduction	{}	Alert,Allogroom,Carry obj...	
4 C	Carry objects	State event	Otter carries objects or food by holding them...		{"0": {"name": "...", ...}}	Alert,Allogroom,Breed,Ch...	
5 C	Chase	State event	Otter chases other animals		{"0": {"name": "...", ...}}	Alert,Allogroom,Breed,Ca...	
6 D	Defecate	State event	Otter eliminates fecal matter (Ethol, 2015)	physiology	{}	Alert,Allogroom,Breed,Ca...	
7 D	Dig	State event	Otter uses front legs to move sand, stones on...		{}	Alert,Allogroom,Breed,Ca...	
8 D	Drink	State event	Otter consumes water found in its environme...	physiology	{}	Alert,Allogroom,Breed,Ca...	
9 E	Eat	State event	Otter consumes food	physiology	{"0": {"name": "...", ...}}	Alert,Allogroom,Breed,Ca...	
10 I	Interact with e...	State event	Otter interacts with enrichment		{"0": {"name": "...", ...}}	Alert,Allogroom,Breed,Ca...	
11 L	Locomotion	State event	Otter moves from place to place		{"0": {"name": "...", ...}}	Alert,Allogroom,Breed,Ca...	
12 F	Look for food	State event	Otter looks for food in the environment		{}	Alert,Allogroom,Breed,Ca...	

The **Ethogram** widget provide the user with the list of behaviors defined in the **Ethogram**. It can be used to record an event by double clicking on the corresponding row. The **Key** column indicates the keyboard key assigned to each behavior (if any). Pressing a key will record the corresponding behavior (that will appear in the *Events* widget).

The behaviors shown in the ethogram widget can be filtered:

#### Right-click on ethogram widget > Filter behaviors

Check/Uncheck single behaviors or double-click on the behavioral category



## Subjects widget in the main window

Subjects			
Key	Name	Description	Current state(s)
1	No focal subject		
2 N	Nina	Female, adult, born on 10/03/2013 i...	
3 H	Himal	Male, adult, born on 04/30/2014 in ...	
4 C	Sharky	Male, juvenile, born on 10/30/2015 i...	
5 S	Nutilus	Male, juvenile, born on 10/30/2015 i...	

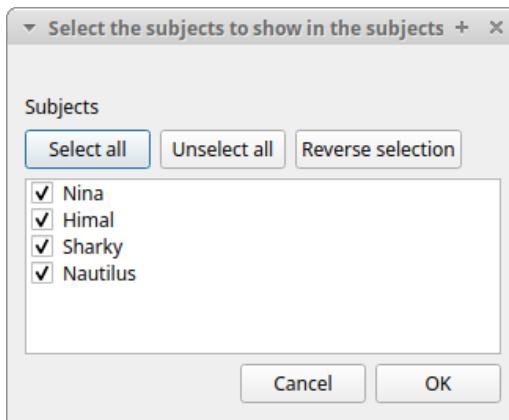
The **Subjects** widget provide the user with the list of subjects defined in the **Subject** tab in the **Project** window. It can be used to add information about the focal subject on the recorded behaviors by double clicking on the corresponding

row. When a subject is selected his/her name appears above the media player. The **Key** column indicates the keyboard key assigned to each subject (if any).

The subjects shown in the subjects widget can be filtered:

**Right-click** on subjects widget > **Filter subjects**

Check/Uncheck single subject



## The media player widgets

The screenshot shows the BORIS software interface with three media players (Player #1, Player #2, Player #3) displaying video frames. Each player has its own audio volume control (vertical slider bar on the right). Below the players are several control and monitoring widgets:

- Ethogram:** A table showing events with columns for Key, Code, and Type. Data:
 

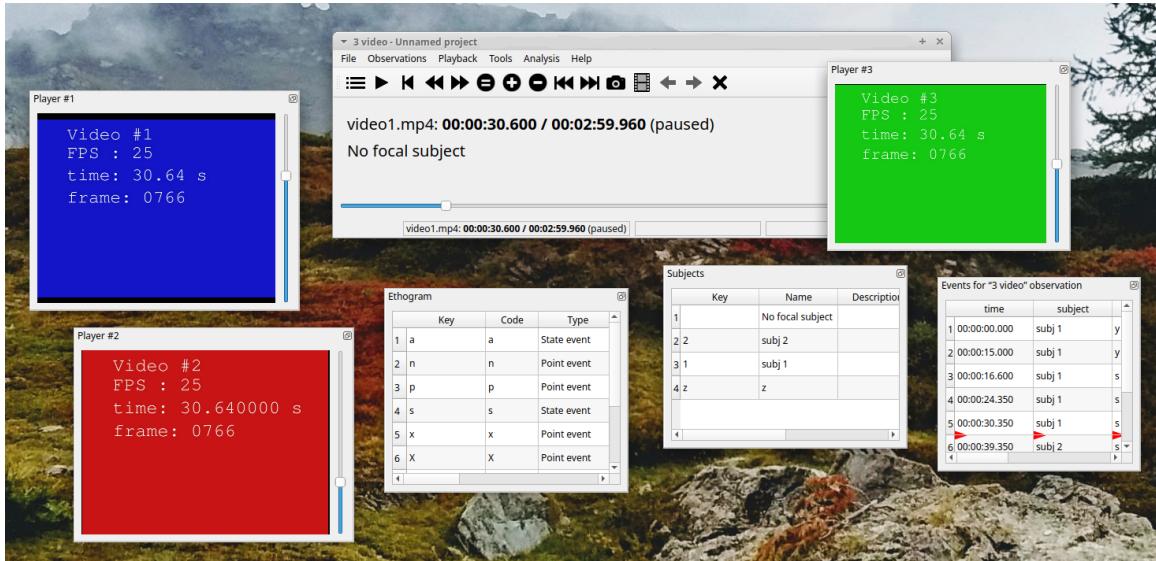
Key	Code	Type
1 a	a	State event
2 n	n	Point event
- Subjects:** A table showing subjects with columns for Key, Name, and Description. Data:
 

Key	Name	Description
1	No focal subject	
2	subj 2	
- Events for "3 video" observation:** A table showing events with columns for time, subject, and code. Data:
 

time	subject	code
1 00:00:00.000	subj 1	y
2 00:00:15.000	subj 1	y
3 00:00:16.600	subj 1	s
4 00:00:24.350	subj 1	s
5 00:00:30.350	subj 1	s
6 00:00:39.350	subj 2	s
7 00:00:50.600	subj 2	s

The media position can be set using the horizontal slide bar. Each media player has its own audio volume control (the vertical slide bar on the right side of the player).

The organization of the various widget can be customized:



## The Events widget

Events for "0001_a" observation					
time	subject	code	type	modifier	comment
1 0.000	Himal	Tear	START	Branches	
2 0.000	Nautilus	Tear	START	Branches	
3 30.199	Himal	Tear	STOP	Branches	
4 30.200	Himal	Locomotion	START	Walk	
5 32.400	Himal	Locomotion	STOP	Walk	
6 32.400	Nautilus	Tear	STOP	Branches	

The **Events** widget shows all the recorded behaviors (events) with the following parameters (organized in columns):

- **time**, the time at which the event occurred;
- **subject**, the focal subject (if any);
- **code**, the behavior code;
- **type**, in case of a state event indicates whether the time corresponds to the start or to the stop.
- **modifier**, indicates the modifier(s) that was(ere) selected (if any);
- **comment**, is an open field where the user can add notes.

A tracking cursor (red triangle) will show the current event. This cursor can be positioned above the current event, see [tracking cursor position](#) option in Preferences window.

A double-click on a row will reposition the media player to the moment of the corresponding event. See [Time offset for media reposition](#) in Preferences window to customize the time offset for media repositioning.

## Record an event

Once ready to begin your coding, you can start the media player using the **Play** button (or the **Space bar**). The behaviors can be recorded using the keyboard with the predefined keys, by double-clicking the corresponding row in the **Ethogram** table or by using the **Coding pad** (See [coding pad](#)).

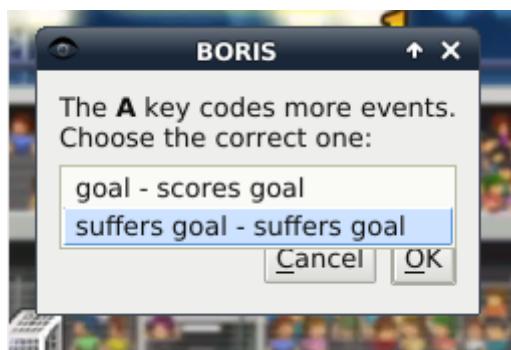
The screenshot shows the BORIS software interface. At the top is a menu bar with File, Observations, Playback, Tools, Analyze, and Help. Below the menu is a toolbar with various icons. The main area is divided into two tables: 'Ethogram' and 'Subjects'. The 'Ethogram' table has columns for Key, Code, Type, Description, Modifiers, and Excluded. It contains six rows with the following data:

Key	Code	Type	Description	Modifiers	Excluded
1 T	moves towards	State event			moves backwards
2 B	moves backwards	State event			moves towards
3 J	jumps	Point event	jumps		
4 H	hits	Point event	hits ball or oppo...	ball, opponent	
5 A	goal	Point event	scores goal		
6 A	suffers goal	Point event	suffers goal		

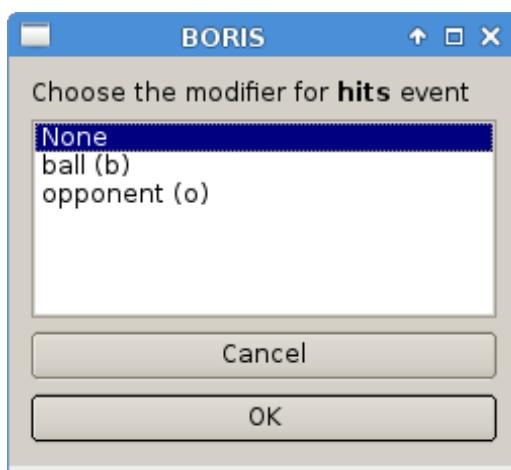
The 'Subjects' table has columns for Key, Name, Description, and Current state(s). It contains two rows:

Key	Name	Description	Current state(s)
1 K	korea		
2 G	ghana		

If the pressed key defines a single event, the corresponding event will be recorded in the **Events** table. In the case you have specified the same key for two (or more) events (e.g. key A in the figure below), BORIS will prompt you for the desired behavior.



In the case you have specified modifiers (one or more sets), BORIS will prompt you for the desired modifier(s) if any (e.g. **ball** or **opponent** in the figure below). You can select the correct one using the mouse or the keyboard (**b** key or **o** key)



If no keys are defined for the modifier selection, you can type the first character of the modifier and use the **Up arrow** and **Down arrow** keyboard keys to select the correct modifier.

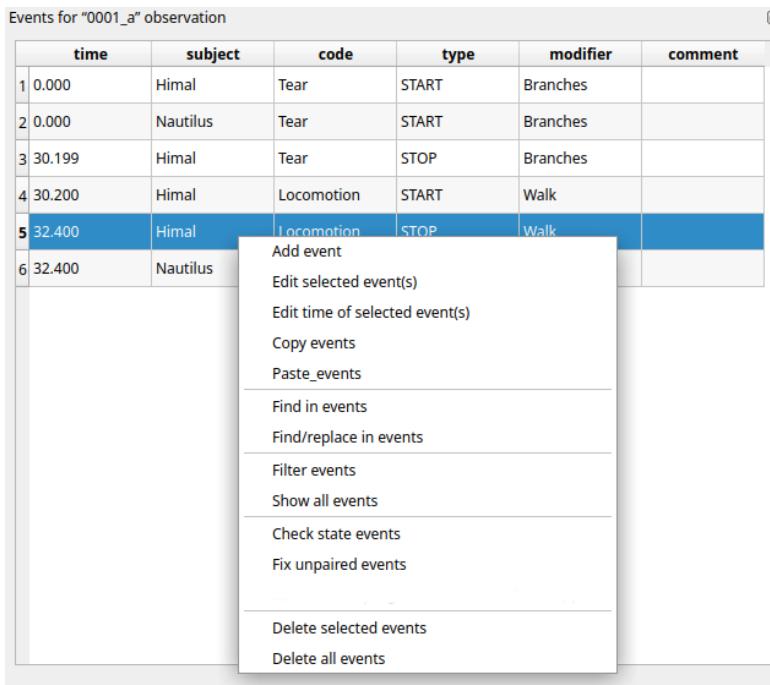
In the case your behavior type is a **Point event with coding map** or a **State event with coding map**, BORIS will show the **Coding map** window and will allow selecting the desired area(s). In case you click a part of the map in which two (or more) areas overlap, the corresponding codes will be recorded.

A recorded event can be edited (once selected) using the **Observations > Edit event** menu option. The resulting *Edit event parameters* allows modifying every parameter (e.g. time, subject, code, modifiers, and comment).

The **Observations > Add event** menu option allows adding a new event by specifying its time and the other parameters.

## The Events widget context menu

Some functions are available in the Events widget context menu. Right-click on the Events widget.



### Undo an even recording

A wrong event can be removed from the events list using the Undo function (CTRL + Z). You can go back till 25 events recorded events.

### Add event

This option allows adding a new event by specifying its time and the other parameters.

### Edit selected event(s)

This option allows to edit the selected event(s). When many events are selected you have to choose the field to edit between **Subject**, **Behavior** and **Comment**. In this case the new value will apply to all selected events.

time	subject	code	type	modifier	comment
1 0.000	Himal	Tear	START	Branches	
2 0.000	Nautilus	Tear	START	Branches	
3 30.199	Himal	Tear	STOP	Branches	
4 30.200	Himal	Locomotion	START	Walk	
5 32.400	Himal	Locomotion	STOP	Walk	
6 32.400	Nautilus				

Edit selected events

Subject  
 Behavior  
 New value  
 Nina  
 Himal  
**Sharky**  
 Nautilus

Comment  
 New comment

Edit time of selected event(s)

This option allows to add or subtract a time value (in seconds) to all selected events. For subtracting a value use a negative value.

time	subject	code	type	modifier	comment
1 0.000	Himal	Tear	START	Branches	
2 0.000	Nautilus	Tear	START	Branches	
3 30.199	Himal	Tear	STOP	Branches	
4 30.200	Himal	Locomotion	START	Walk	
5 32.400	Himal	Locomotion	STOP	Walk	
6 32.400	Nautilus	Tear	STOP	Branches	

Time value

Value to add or subtract (use negative value):  
 -10.200

Copy events

This option allows to copy the selected events in the clipboard. The clipboard will contain the values of the selected events (except the **type** field) separated by a <TAB> character.

Example of clipboard content:

```
0.0 Himal  Tear  Branches
0.0 Nautilus  Tear  Branches
30.199  Himal  Tear  Branches
30.2  Himal  Locomotion  Walk
32.4  Himal  Locomotion  Walk
32.4  Nautilus  Tear  Branches
```

Paste events

This option allows to paste the clipboard content into the events widget. The clipboard must respect the format described in the previous section: 5 columns separated by <TAB> character.

### Find in events

This option allows to search for a string in the various field of events. Select the fields to be searched. The find/replace operation can be restricted to the selected events.

time	subject	code	type	modifier	comment
1 0.000	Himal	Tear	START	Branches	
2 0.000	Nautilus	Tear	START	Branches	
3 30.199	Himal	Tear	STOP	Branches	
4 30.200	Himal	Locomotion	START		
5 32.400	Himal	Locomotion	STOP		
6 32.400	Nautilus	Tear	STOP		

Find in events

Subject  
 Behavior  
 Modifiers  
 Comment

Find

Locom

Find in selected events

Close Find

### Find/Replace in events

This option allows to search for a string and replace it by a new value in the various field of events. Select the fields to be searched. The find operation can be restricted to the selected events.

time	subject	code	type	modifier	comment
1 0.000	Himal	Tear	START	Branches	
2 0.000	Nautilus	Tear	START	Branches	
3 30.199	Himal	Tear	STOP	Branches	
4 30.200	Himal	Locomotion	START	Walk	
5 32.400	Himal				
6 32.400	Nautilus				

Find/Replace events

Subject  
 Behavior  
 Modifiers  
 Comment

Find

Himal

Replace

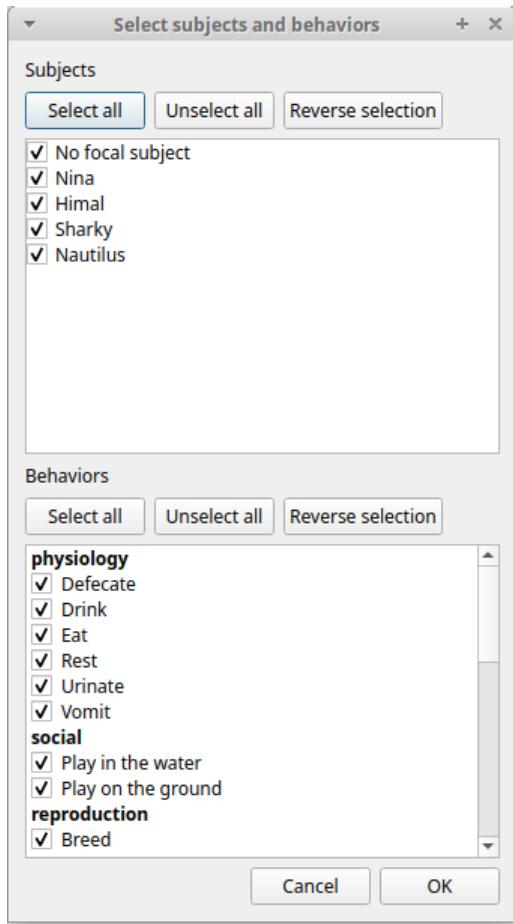
Coco

Find/Replace in selected events

Cancel Find and replace Find and replace all

### Filter events

This option allows to filter the events by field value.



## Show all events

This option reverts the previous one and allows to visualize all coded events

## Check state events

This option allows to check if the **state events** are **PAIRED**, if they have a **START** and a **STOP** occurrences.

## Delete selected events

This option allows to delete the selected events. This operation is irreversible!

## Delete all events

This option allows to delete all then events in the current observation. This operation is irreversible!

## Fix unpaired state events

You can use the "Fix unpaired events" function to fix the **state events** without a **STOP** event.

**Observations > Fix unpaired events** (keyboard shortcut: CTRL+U)

The program will ask for a time at which insert the **STOP** events for all unpaired **state events**

This function can be run on a set of selected observations (when no observation is open). In this case the **STOP** events will be inserted at the end of observation.

## Explore project

You can search information in various fields in all observations in the current project (Observations > Explore project).

The searchable fields are: **subject**, **behavior**, **modifier** and **comment**.

If more than one field is searched a logic AND will apply.



The events that were found are listed in a table. By double-clicking on the row the corresponding observation will be opened and the visualization will be scrolled to the row corresponding to the event.

The screenshot shows the BOVIS software interface. On the left, there's an 'Ethogram' window listing various behavioral codes like Alert, Breed, Chase, etc., with their corresponding keys and types. Below it is a 'Subjects' list with entries for No focal subject, Nina, and Himal. In the center, a modal dialog titled 'BORIS' displays '148 results' for 'Observation id' 0004. This dialog has columns for 'Observation id' and 'row index'. To the right of the dialog is a large table titled 'Events for "0004" observation' with columns for time, subject, code, and type. The table lists numerous events, such as Alert, Locomotion, Rest, and Drink, for subjects like Himal, Nautillus, and Nina at specific times. The last visible row in the table is highlighted in blue.

## Frame-by-frame mode

You can switch between the media player and the frame-by-frame mode using the **arrow buttons** in the toolbar:

In frame-by-frame mode the video will stop playing and the user will visualize the video frame by frame.

### Warning

Please note that MTS video files should be re-encoded to be used in frame-by-frame mode. Otherwise the extracted frames are not reliable.

You can move between frames by using the arrow keys in the toolbar (on the right) or by using keyboard special keys:

- **Left arrow key:** go to the previous frame
- **Right arrow key:** go to the next frame
- **Page Up key:** switch to the next media
- **Page Down key:** switch to the previous media
- **Up arrow key:** jump forward in the current media
- **Down arrow key:** jump backward in the current media

If you have a numeric keypad you can use the following keys in alternative:

- The key / will allow you to view the previous frame
- The key \* will allow you to view the next frame

To return in the media player mode press the **Play** button in the toolbar.

See **File > Preferences > frame-by-frame mode**

# Exporting events data

The coded events can be exported in various formats:

## Export events

### Observations > Export events

This function will export the events of selected observations in TSV, CSV, ODS, XLSX, XLS or HTML formats. If many observations are selected BORIS will ask for a directory to save the various files. For the XLS and XLSX formats the events can be exported on various worksheet in a single workbook. These formats are suitable for further analysis.

A	B	C	D	E	F
1 Observation id	demo #1				
2					
3 Media file(s)					
4					
5 Player #1	/home/olivier/crop.avi				
6 Player #1	/home/olivier/crop2.avi				
7					
8 Observation date	2015-Nov-30 10:39:18				
9					
10 Description					
11					
12 Time offset (s)	0				
13					
14 independent variables					
15 variable	value				
16 group size	7				
17 weather conditions	rain				
18					
19 time	subject	code	Modifier	comment	status
20	4.3 Subject #1	eat	salad	test	START
21	10 Subject #1	eat	salad		STOP
22	26.6 Subject #2	eat	meat		START
23	113.988 Subject #2	eat	meat		STOP
24	116.588 Subject #1	jump			POINT
25	118.988 Subject #1	jump			POINT
26	120.863 Subject #2	jump			POINT
27	122.438 Subject #2	jump			POINT
28					

### Warning

Please note that for some formats (XLS - Excel 97) the name of the sheet will be based on a modified **observation id** in order to not contain forbidden characters (: / ? \* [ or ]) and shortened to 31 characters.

## Export aggregated events

### Observations > Export aggregated events

This function will export the events of the selected observations in the following formats:

- **tabular format** (TSV, CSV, XLSX, XLS, ODS, HTML)
- **SQL** format for populating a SQL database
- **SDIS** format for analysis with the GSEQ program available at <http://www2.gsu.edu/~psyrab/gseq>

If many observations are selected you can choose to group all results in one file. If you do not want to group results BORIS will ask for a directory to save the various files.

The **State events** are paired and the duration is available.

An arbitrary time interval can be selected (check the **Limit to time interval** option). In this case the ongoing events will be started at start time and stopped at end time in the export file.

Example of tabular export

A	B	C	D	E	F	G	H	I	J	
	Observation id	Observation date	Subject	Behavior	Modifiers	Behavior type	Start	Stop	Comment start	Comment stop
1	demo #1	2015-Nov-30 10:39:18	Subject #1	jump		POINT	116.588	0		
2						POINT	118.988	0		
3	demo #1	2015-Nov-30 10:39:18	Subject #1	jump	salad	STATE	4.3	10	test	
4						POINT	120.863	0		
5	demo #1	2015-Nov-30 10:39:18	Subject #2	jump		POINT	122.438	0		
6					meat	STATE	26.6	113.988		
7	demo #1	2015-Nov-30 10:39:18	Subject #2	eat						
8										
9										
10										

Example of SQL export:

```
CREATE TABLE events (id INTEGER PRIMARY KEY ASC, observation TEXT,
date DATE, subject TEXT, behavior TEXT,
modifiers TEXT, event_type TEXT, start FLOAT,
stop FLOAT, comment_start TEXT,
comment_stop TEXT);

INSERT INTO events (observation, date, subject, behavior, modifiers,
event_type, start, stop, comment_start, comment_stop ) VALUES
("demo #1", "2015-11-30 10:39:18", "Subj #1", "jump", "", "POINT", 116.588, 0, "", ""),
("demo #1", "2015-11-30 10:39:18", "Subj #1", "jump", "", "POINT", 118.988, 0, "", ""),
("demo #1", "2015-11-30 10:39:18", "Subj #1", "eat", "salad", "STATE", 4.3, 10.0, "vvv", ""),
("demo #1", "2015-11-30 10:39:18", "Subj #2", "jump", "", "POINT", 120.863, 0, "", ""),
("demo #1", "2015-11-30 10:39:18", "Subj #2", "jump", "", "POINT", 122.438, 0, "", ""),
("demo #1", "2015-11-30 10:39:18", "Subj #2", "eat", "meat", "STATE", 26.6, 113.988, "", "");
```

## Warning

Please note that for some formats (XLS - Excel 97) the name of the sheet will be based a modified **observation id** in order to not contain forbidden characters (: / ? \* [ or ]) and shortened to 31 characters.

## Export events for analysis with JWatcher

[JWatcher](#) is a powerful tool for the quantitative analysis of behavior.

The events coded with BORIS can be exported to be analyzed with JWatcher.

Click **Observations > Export events > for analysis with JWatcher** to export the coded events.

BORIS will ask for selecting a directory. After this, for each combination of selected observation and selected subject the following files will be created:

- the Focal Data File (.dat)
- the Focal Analysis Master File (.faf)
- the Focal Master File (.fmf)

These files can be used to analyze your observations with JWatcher.

## Export events as behavioral strings

Behavioral strings can be used with the **Behatrix** program: **Behatrix (formerly BSA)**

Example:

```
# observation id: demo#1
# observation description:
# Media file name: video1.mp4, video2.mp4
```

```

Subject #1:
eat|jump|eat|jump

Subject #2:
eat|rest|jump|eat|jump

```

## Export events as Praat TextGrid

Example:

```

File type = "ooTextFile"
Object class = "TextGrid"

xmin = 4.3
xmax = 113.988
tiers? <exists>
size = 2
item []:
    item [1]:
        class = "IntervalTier"
        name = "Subject #1"
        xmin = 4.3
        xmax = 10.0
        intervals: size = 1
        intervals [1]:
            xmin = 4.3
            xmax = 10.0
            text = "eat"
    item [2]:
        class = "IntervalTier"
        name = "Subject #2"
        xmin = 26.6
        xmax = 113.988
        intervals: size = 1
        intervals [1]:
            xmin = 26.6
            xmax = 113.988
            text = "eat"

```

## Export events as Behavioral Binary Table

### Warning

This function is still experimental in v. 7.8.5

A time interval will be asked to the user (in seconds). The observation will be checked every n seconds and the presence (1, absence: 0) of the selected behaviors will be exported in a table for each selected subjects.

Example for a time interval of 1 second:

time	Alert	Drink	Locomotion	Swim
0.0	0	1	0	0

1.0	0	1	0	0
2.0	0	1	0	0
3.0	0	1	0	0
4.0	0	1	0	0
5.0	0	1	0	0
6.0	0	1	0	0
7.0	0	1	0	0
8.0	0	1	0	0
9.0	1	0	0	0
10.0	1	0	0	0
11.0	0	0	1	0
12.0	1	0	0	0
13.0	1	0	0	0
14.0	1	0	0	0
15.0	1	0	0	0
16.0	1	0	0	0
17.0	1	0	0	0
18.0	0	0	1	0
19.0	0	0	1	0
20.0	0	0	1	0
21.0	0	0	1	0
22.0	1	0	0	0
23.0	0	0	0	0
24.0	0	0	0	0
25.0	0	0	0	0
26.0	0	0	0	0
27.0	0	0	0	0
28.0	0	0	0	0
29.0	0	0	0	0
30.0	0	0	0	0
31.0	0	0	0	0
32.0	0	0	0	0
33.0	0	0	0	1
34.0	0	0	0	1
35.0	0	0	0	1
36.0	0	0	0	1
37.0	0	0	0	1
38.0	0	0	0	1
39.0	0	0	0	1
40.0	0	0	0	1
41.0	0	0	0	1
42.0	0	0	0	1
43.0	0	0	0	1
44.0	0	0	0	1
45.0	0	0	0	1
46.0	0	0	0	1
47.0	0	0	0	1
48.0	0	0	0	1
49.0	0	0	0	1
50.0	0	0	0	1
51.0	0	0	0	1
52.0	0	0	0	1
53.0	0	0	0	0
54.0	0	0	0	0
55.0	0	0	0	0
56.0	0	0	0	0
57.0	0	0	0	1

58.0	0	0	0	1
59.0	0	0	0	1
60.0	0	0	0	1
61.0	0	0	0	1
62.0	0	0	0	1
63.0	0	0	0	1

## Extract sequences from media files corresponding to coded events

Sequences of media file corresponding to coded events can be extracted from media files:

- 1 . Click on **Observations > Extract events from media files** option.
- 2 . Choose the observation(s).
- 3 . Select the events to be extracted.
- 4 . Select a destination directory that will contain the extracted sequences.
- 5 . Select a time offset (in seconds, the default value is 0).

The time offset will be subtracted from the starting time of event and added to the stopping time. All the extracted sequences will be saved in the selected directory followind the file name format:

{observation id}\_{player}\_{subject}\_{behavior}\_{start time}-{stop time}

## Extract frames corresponding to coded events

The frames corresponding to coded events can be extracted and saved as images.

- 1 . Click on **Observations > Extract frames from media files** option.
- 2 . Choose the observation(s).
- 3 . Select the events to be extracted.
- 4 . Select a destination directory that will contain the extracted sequences.
- 5 . Select a time offset (in seconds, the default value is 0).

## Export transitions matrix

3 transitions matrix outputs are available: The matrix of frequencies of transitions, the matrix of frequencies of transition after each behavior and the matrix of number of transitions.

### Matrix of frequencies of transitions

This matrix contains the frequencies of total transitions. The sum of all frequencies must be 1.

Example of frequencies of transitions matrix:

	eat	sleep	walk
eat	0.0	0.286	0.143
sleep	0.143	0.0	0.143
walk	0.286	0.0	0.0

In this matrix you can see that the **eat** behavior precedes the **sleep** behavior with a frequency of **0.286** of the total number of transitions.

### Matrix of frequencies of transitions after behavior

This matrix contains the frequencies of transitions after each behavior. The sum of each row must be 1.

Example:

	eat	sleep	walk
eat	0.0	0.667	0.333
sleep	0.5	0.0	0.5
walk	1.0	0.0	0.0

In this example you can see that **sleep** follows **eat** with a frequency of **0.667** and **walk** follows with a frequency of **0.333**.

## Matrix of number of transitions

This matrix contains the number of transitions after each behavior.

Example:

	eat	sleep	walk
eat	0	2	1
sleep	1	0	1
walk	2	0	0

## Playback menu

### Jump

#### Jump forward

Allow to jump forward in the current media file. See **File > Preferences** for setting the jump value.

#### Jump backward

Allow to jump backward in the current media file. See **File > Preferences** for setting the jump value.

#### Jump to specific time

Allow to go to a specific time in the current media file.

### Zoom

Allow to zoom into the current video file. The available zoom values are: 1:2, 1:4, 2:1 and 1:1. **Fit to window** adapts the zoom value to the current window size. In case of simultaneous playing you can set different values of zoom in player #1 and player #2.

#### Warning

**The Zoom function is not working on Mac**

## Tools

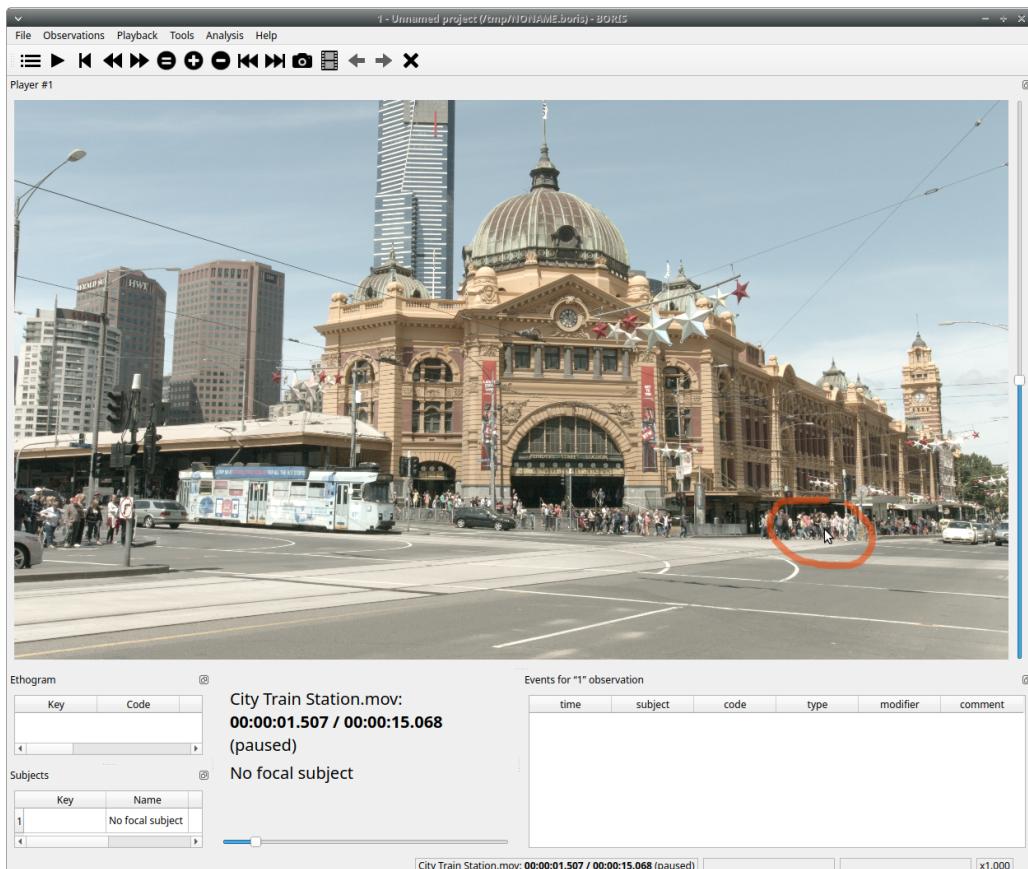
### Lock the dockwidgets

The dockwidgets (except the player dockwidgets) can be locked on the main window (See **Tools > Lock dockwidgets**).

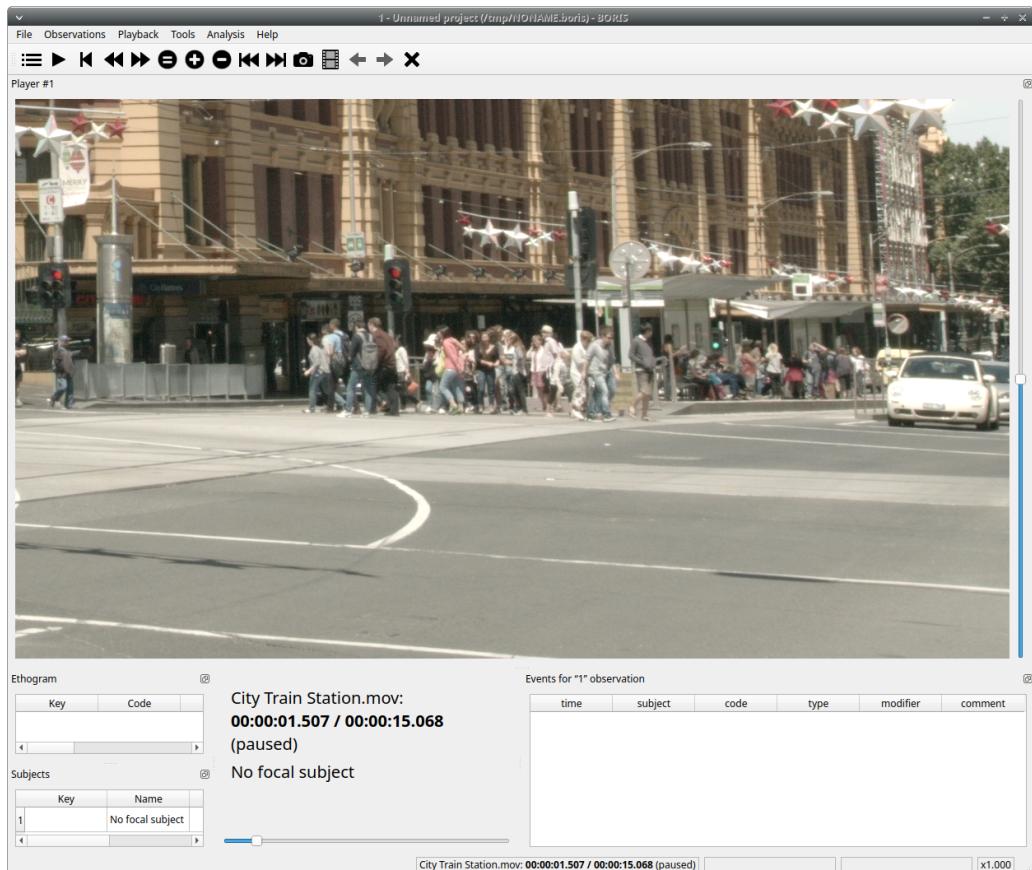
## Focus a video area

It is possible to focus an area of the video. Click on the center of the area you want to focus. The video will be displayed at its native resolution (not shrunk) and centered on the clicked point. Another click on the video will restore the previous visualization.

The focused area can also be zoomed (only on the first player for now).



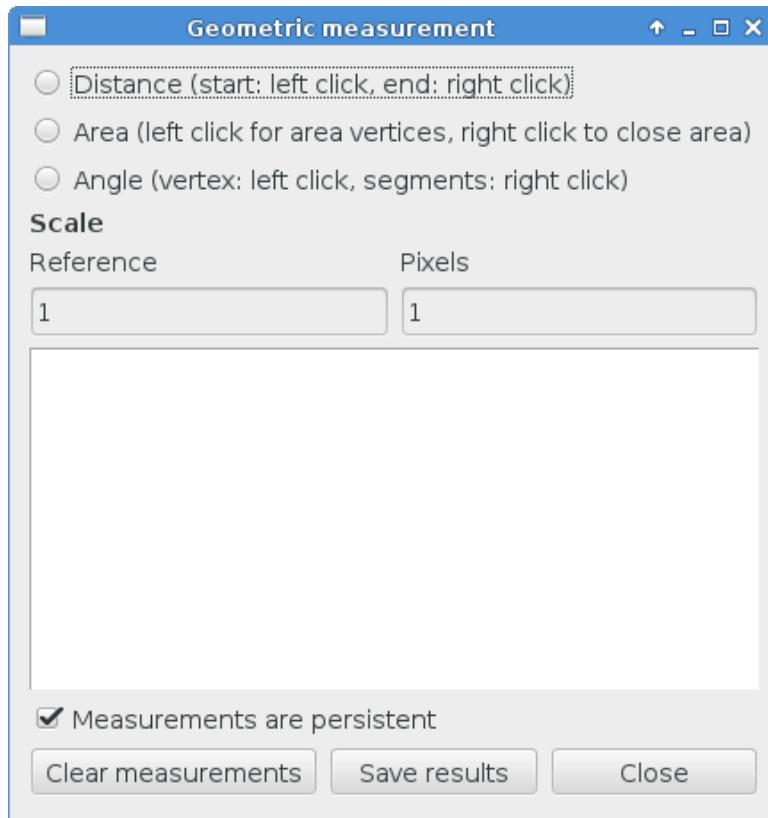
The visualization can be focused on a particular region of the video by clicking on the video (with the left button). See the mouse cursor on the screenshot above.



Now the video is displayed with its real resolution allowing to see details. Another click on the video will restore the previous visualization without focus.

## Geometric measurements

Geometric measurements can only be made in frame-by-frame mode. Distances, areas and angles can be measured. Click on **Tools > Geometric measurements** to activate the measurements. The **Measurements window** will be shown:



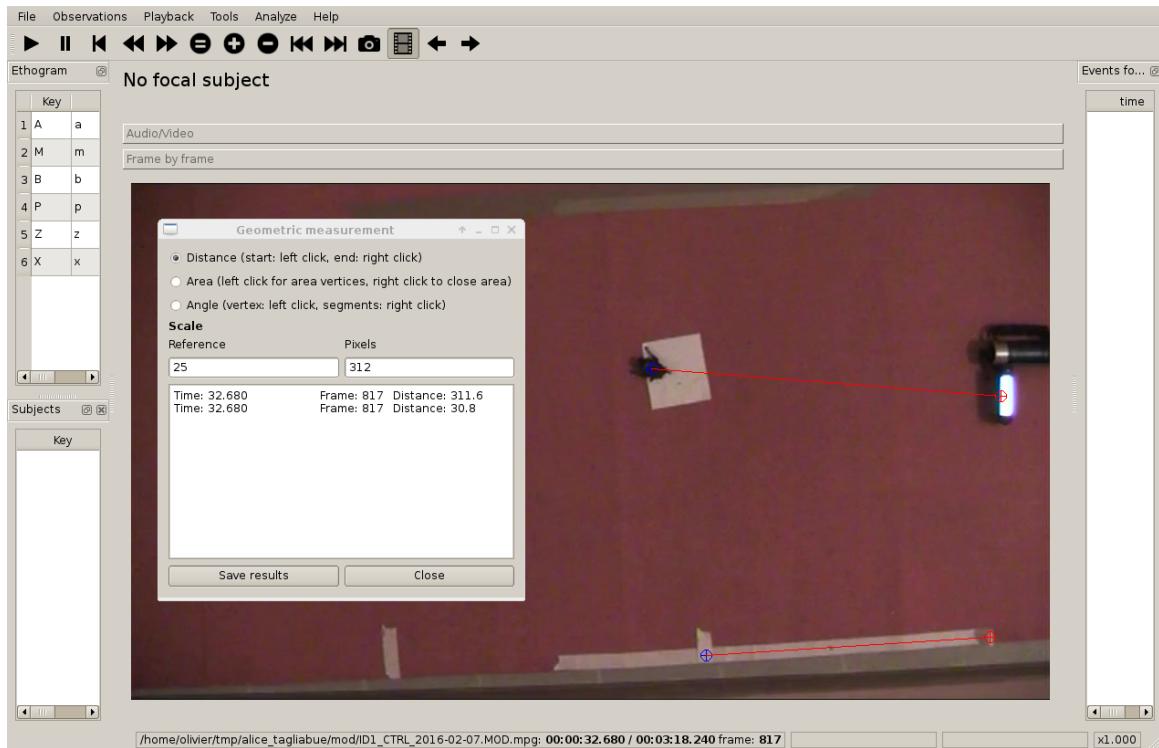
## Setting the scale

For distance and area measurements you can set a scale in order to have results of measurements in a real unit (like centimeters, meters etc).

- 1 . measure a reference object (that have a known size) on the frame (with the distance tool. See next chapter for details) and set the pixel distance in the **Pixel** text box.
- 2 . Set the real size of the reference object in the **Reference** text box (must be a number without unit).

## Distance measurements

Select the **Distance** radio button. Click the left mouse button on the frame bitmap to set the start of the segment that will be measured. A blue circle with a cross will be drawn. Click the right mouse button to set the end. A red circle with a cross will be drawn. The distance between the two selected points will be available in the text area of the **Measurements window**.



## Area measurements

Select the **Area** radio button. Click the left mouse button on the frame bitmap to set the area vertices. Blue circles with a cross will be drawn. Click the right mouse button to close the area. The area of the drawn polygon will be available in the text area of the **Measurements window**.

## Angle measurements

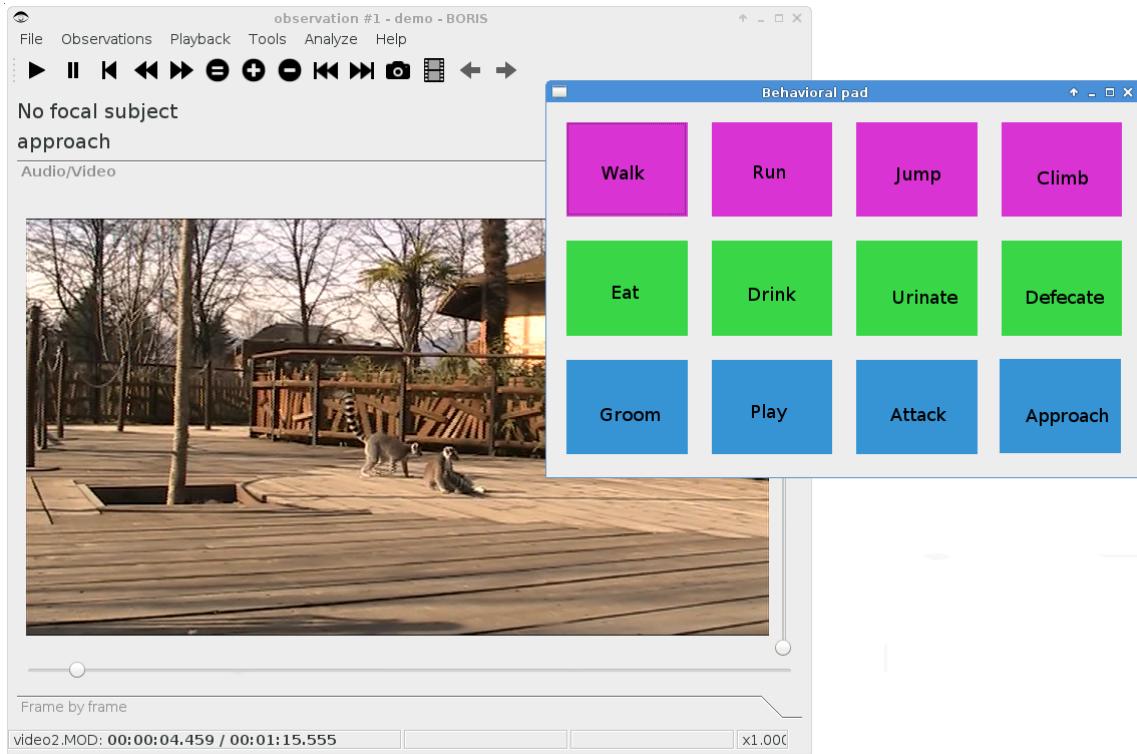
Select the **Angle** radio button. Click the left mouse button on the frame bitmap to set the angle vertex. A red circle with a cross will be drawn. Click the right mouse button to set the two segments. Blue circles with a cross will be drawn. The angle between the two drawn segments will be available in the text area of the **Measurements window**.

## Persistent measurements

If the **Measurements are persistent** checkbox is checked the measurement schemes will be available on all frames otherwise they will be deleted between frames.

## Coding pad

During observation a coding pad with the available behaviors can be displayed (**Tools > Coding pad**). This **Coding pad** allows the user to code using a touch-screen or by clicking on the buttons. When the **Coding pad** is displayed you can continue to code using the keyboard or the ethogram.



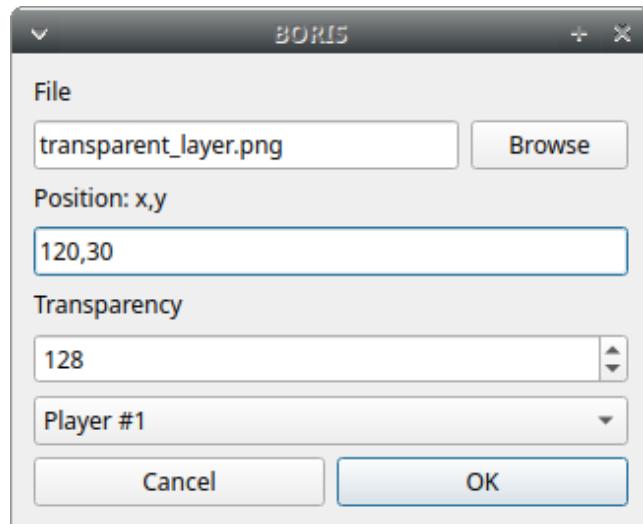
## Subjects pad

A pad with all defined subjects (or filtered subjects) can be displayed during the observation (**Tools > Subjects pad**). This **Subjects pad** allows the user to select the focal subject using a touch-screen or by clicking on the buttons. When the **Subjects pad** is displayed you can continue to select the focal subject using the keyboard or the subjects list.



## Image overlay on video

An image can be overlayed on the video (**Tools > Image overlay on video > Add an image overlay**).



The image must be in PNG format, if the image is smaller than the video resolution the image position can be set from the top-left corner. The transparency of image can be set from 0 (full transparency) to 255 (no transparency).

To remove the image overlay click **Tools > Image overlay on video > Remove**

## Converters for external data values

Converters can be written using the Python 3 programming language.

The **INPUT** variable will be loaded with the original value of the external data file (for example 01:22:32).

The **OUTPUT** variable must contain the converted value in seconds (the dot must be used for decimal separator).

Example of a converter for converting HH.MM:SS format in seconds:

```
h, m, s = INPUT.split(':')
OUTPUT = int(h) * 3600 + int(m) * 60 + int(s)
```

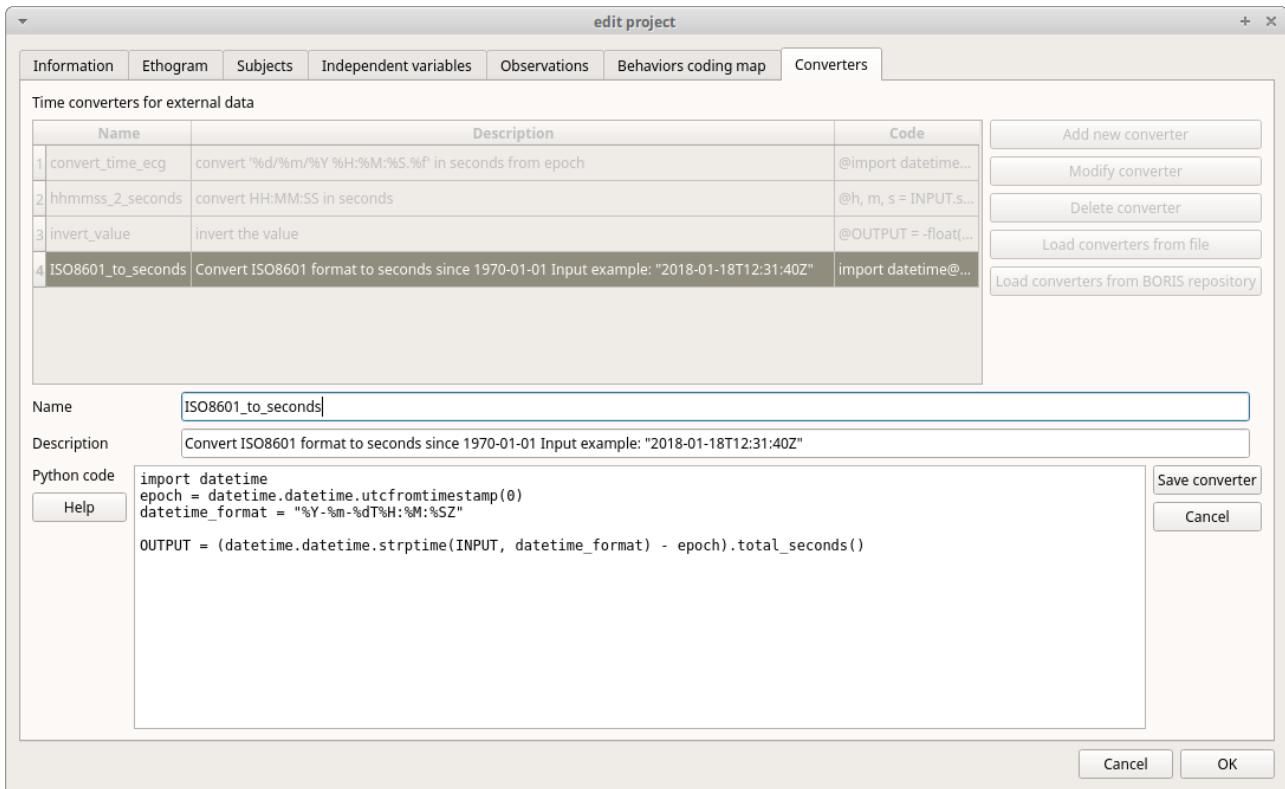
The Python function **strptime()** from the **datetime** module can be useful for converting time values:  
<https://docs.python.org/3/library/datetime.html#strftime-strptime-behavior>

Example of a converter for converting ISO8601 format in seconds using the strptime() function:

```
import datetime
epoch = datetime.datetime.utcnow().timestamp(0)
datetime_format = "%Y-%m-%dT%H:%M:%S"

OUTPUT = (datetime.datetime.strptime(INPUT, datetime_format) - epoch).total_seconds()
```

**File > Edit project > Converters tab**



## Transitions flow diagram

BORIS can generate DOT scripts and flow diagrams from the transitions matrices (See Observations > Create transition matrix for obtaining the transitions matrices).

### DOT script (Graphviz language)

#### Tools > Transitions flow diagram > Create transitions DOT script

Choose one ore more transitions matrix files and BORIS will create the relative DOT script file(s).

The DOT script files can then be used with [Graphviz](#) (Graph Visualization Software) or [WebGraphviz](#) (Graphviz in the Browser) to generate flow diagram of transitions.

See [DOT \(graph description language\)](#) for details.

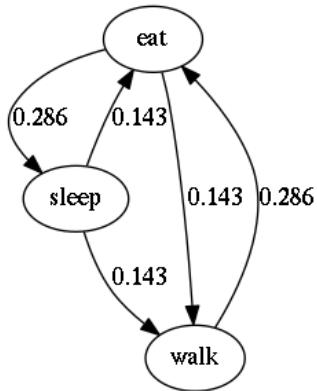
### Flow diagram

If [Graphviz](#) (Graph Visualization Software) is installed on your system (and the **dot** program available in the path) BORIS can generate flow diagram (PNG format) from a transitions matrix file.

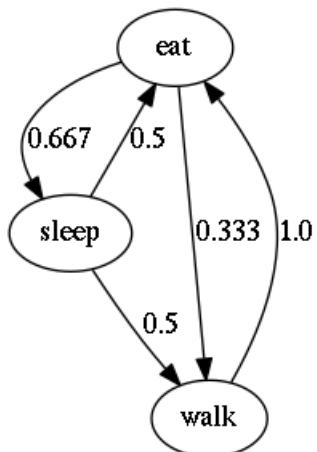
#### Tools > Transitions flow diagram > Create transitions flow diagram

Choose one ore more transition matrix files and BORIS will create the relative flow diagram.

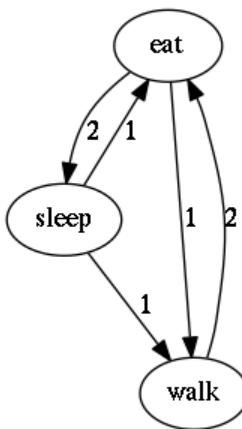
Flow diagram of frequencies of transitions



Flow diagram of frequencies of transitions after behavior



Flow diagram of number of transitions



## Re-encoding and resizing a video file

BORIS can re-encode and resize your video files in order to reduce the size of the files and have a smooth coding (specially with two video files playing together). The re-encoding and resizing operations are done with the embedded ffmpeg program with high quality parameters (bitrate 2000k).

Select the files you want re-encode and resize and select the horizontal resolution in pixels (the default is 1024). The aspect ratio will be maintained.

You can continue to use BORIS during the re-encoding/resizing operation.

The re-encoded/resized video files are renamed by adding the re-encoded.avi extension to the original files.

### Warning

Please note that MTS video files should be re-encoded to be used in frame-by-frame mode. Otherwise the extracted frames are not reliable.

## Rotating a video file

BORIS can rotate your video files in order to code them using the right view. The rotating operation is done with the embedded ffmpeg program using the same quality parameters than the original video.

Select the files you want rotate and select the rotation between: **Rotate 90 clockwise**, **Rotate 90 counter clockwise** and **Rotate 180**.

The aspect ratio will be maintained.

You can continue to use BORIS during the rotation operation.

The rotated video files are renamed by adding the **rotated<ANGLE>** to the original file name.

## Project server

This function allow you to communicate with the [BORIS App](#) by sending project and receiving observations made with the [BORIS App](#). For details read the [BORIS App user guide](#)

### Starting the project server

- Open your project
- Select **File > Project server**

A window will open and show you the URL of the server that should be used on the [BORIS App](#). The project server will serve the project for 30 minutes.

### Stopping the project server

- Select **File > Stop serving project**

## Coding map

A coding map is a bitmap image with user-defined clickable areas that will help to code for behaviors or modifiers for a behavior.

2 types of coding maps are available:

- Behaviors coding map
- Modifiers coding map

### The Behaviors coding map

BORIS allows creating a **Behaviors coding map** using the **Map creator** tool (**Tools > Create a coding map > for behaviors**).

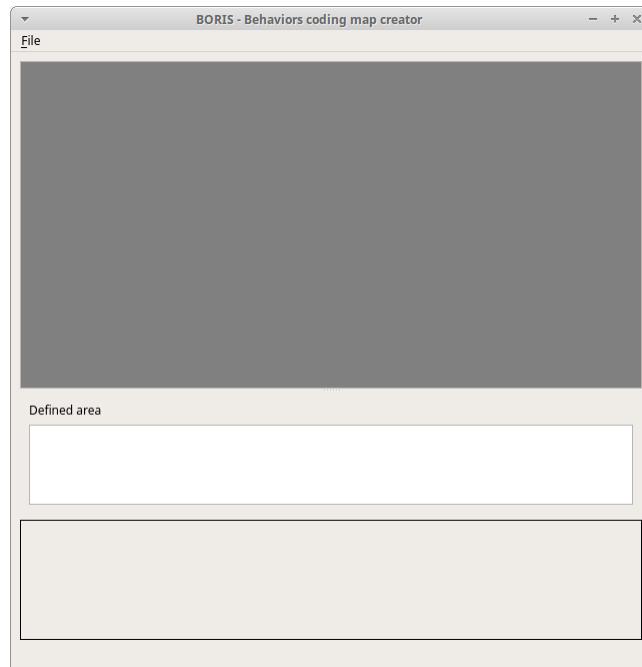
A **Behaviors coding map** can be created only if you have defined behaviors in your ethogram.

## Creating a Behaviors coding map

To create a new **Behaviors coding map** launch the **Behaviors coding map creator**

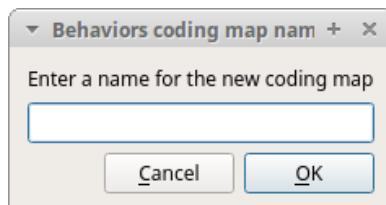
**Tools > Create a coding map > for behaviors).**

A new window will open



**File > New behaviors coding map**

Enter a name for the new **Behaviors coding map**

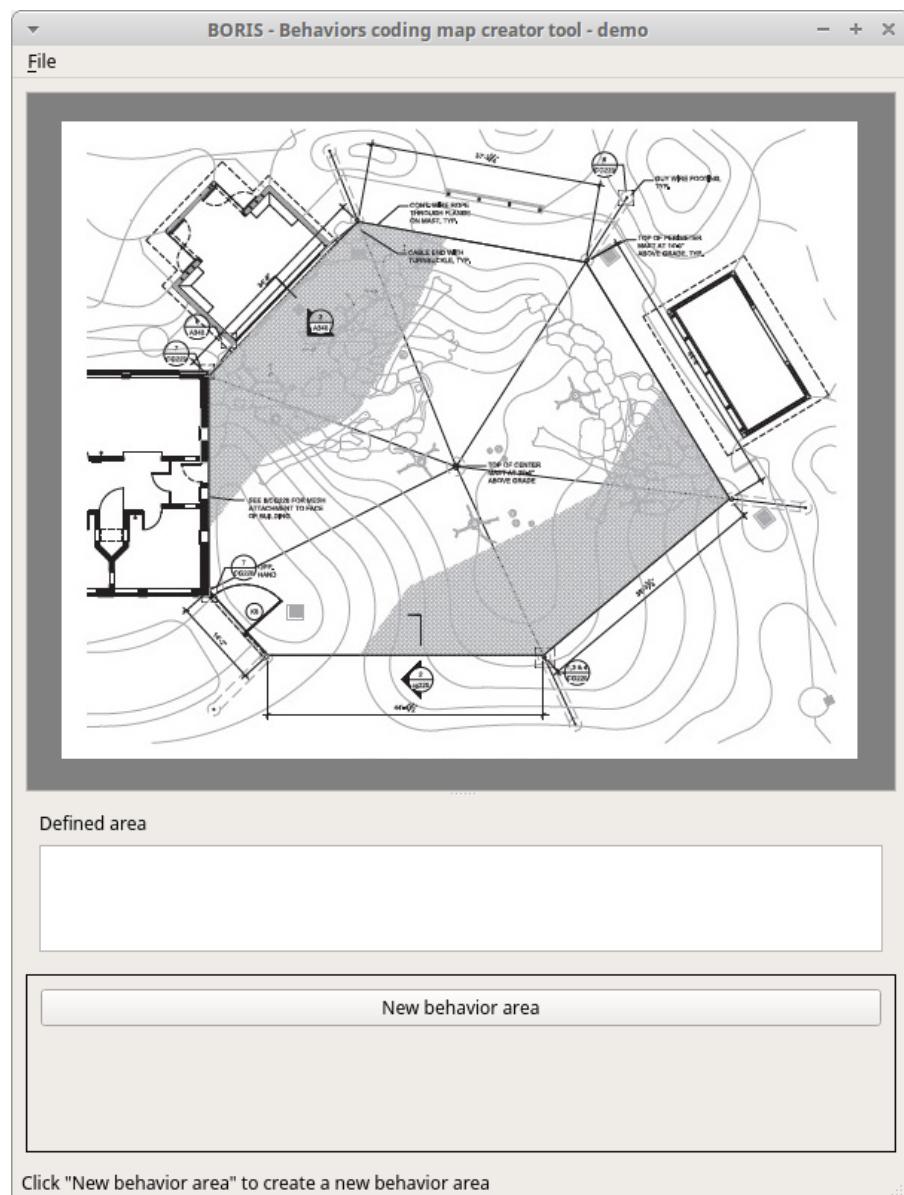


Loading a bitmap for a behaviors coding map

Click the **Load bitmap** button in the bottom of the window and select a bitmap image (PNG and JPEG formats are accepted).

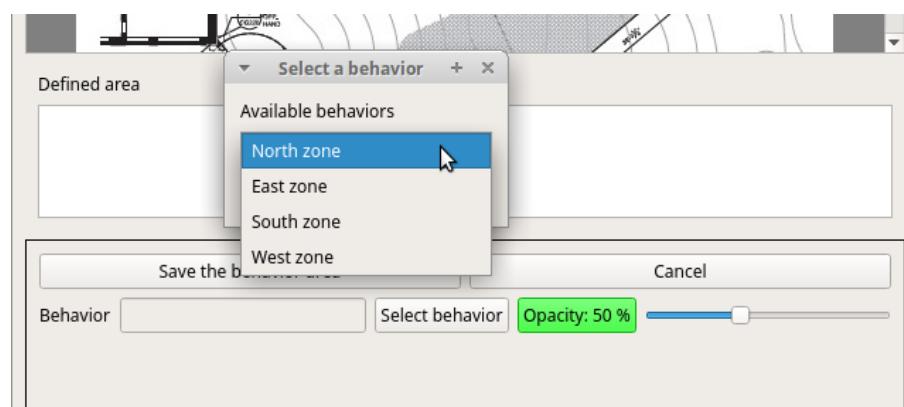
If the size of your bitmap image is bigger than 640 x 640 pixels BORIS will resize it to 640 x 640 pixels keeping the aspect ratio and store the resized version in the coding map file.

The bitmap will be displayed



Adding areas corresponding the behaviors

Click the **New behavior area** button in the bottom of the window and select a behavior by clicking on the **Select behavior** button.



The available behaviors are taken from the ethogram of the current project.

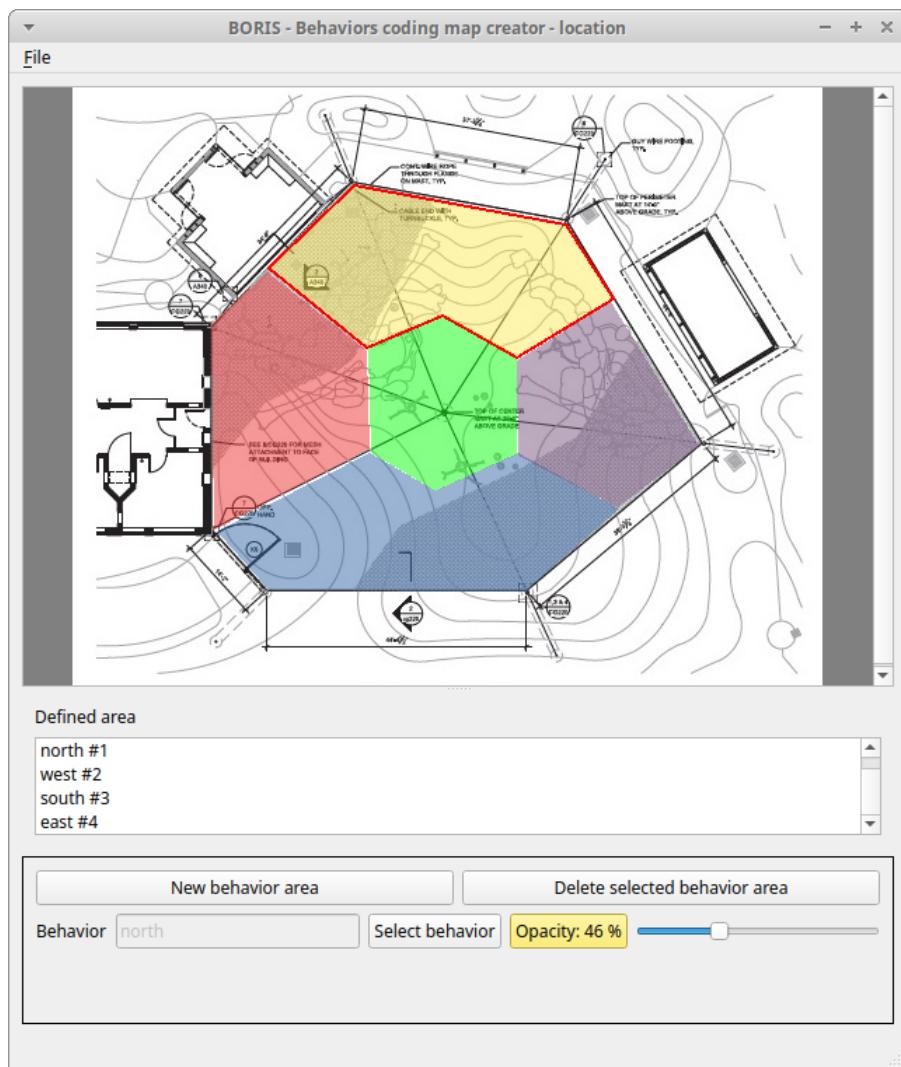
Click on the bitmap to define the vertex on the area that will code the selected behavior. Close the area by clicking again on the first point.

The color of the new area can be changed using the **Opacity** button. The opacity can be changed (from 0 to 100%) using the slider.

Save the behavior area by clicking on the **Save the behavior area** button

The area will be added to the **Defined area** list

You can add more area and also add more than one area for a same behavior. Two or more areas can overlap. In this case all corresponding behaviors will be triggered.



Add the Behaviors coding map to the current project

**File > Add coding map to project**

The coding map will be added to the current project

You can add a **Behaviors coding map** to the current project from a file containing the coding map:

(**File > Edit project > Behaviors coding map > Add a behaviors coding map**)

Saving the Behaviors coding map

Saving the **Behaviors coding map** will create a file containing the **Behaviors coding map** including the bitmap image.

**File > Save the current Behaviors coding map**

The file containing the **Behaviors coding map** can be then reloaded in the **Behaviors coding map creator** or added to a BORIS project (**File > Edit project > Behaviors coding map > Add a behaviors coding map** )

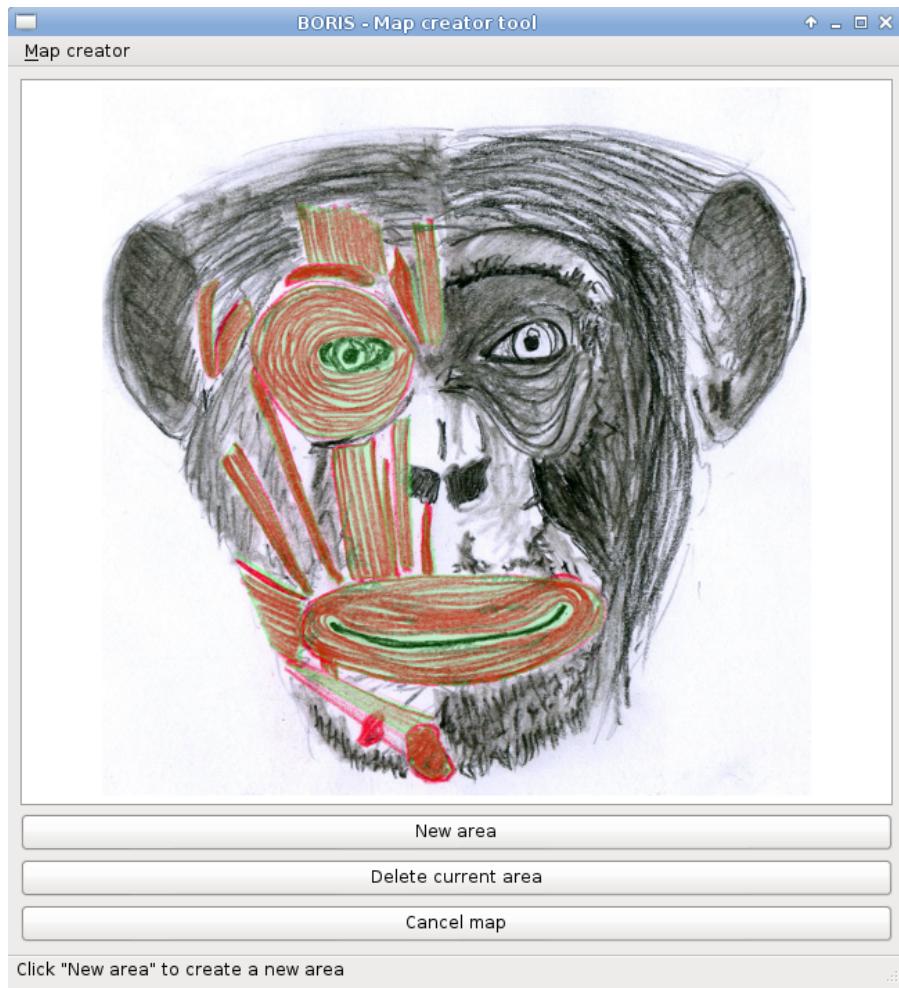
## The Modifiers coding map

BORIS allows creating a modifiers coding map using the **Modifiers Map creator** tool (**Tools > Create a coding map > for modifiers**.) Clickable areas may correspond to specific modifiers that can be meaningful for the behavioral coding. Facial expression is the case we thought to when developing this function.

### Creating a modifiers coding map

Loading a bitmap for a modifiers coding map

To create a new **Modifiers coding map**, launch the **Modifiers Map creator** tool (**Tools > Create a coding map > for modifiers**). The BORIS main window will be replaced by the **Modifiers Map creator** window. Click on **Modifiers Map creator > New Modifiers map** and enter a name for the new map in the edit box. You have to load a bitmap image (JPEG or PNG) using the **Load bitmap** button. The loaded image will be displayed.



If the size of your bitmap image is bigger than 640 x 640 pixels BORIS will resize it to 640 x 640 pixels keeping the aspect ratio and store the resized version in the coding map file.

## Adding areas corresponding to the modifiers

To create clickable areas on a coding map, you have to click on the **New area** button and enter an **Area code** in the edit box. The new area can now be defined by clicking on the image. The drawing tool allows defining an irregular polygon (a plane shape with straight sides, which does not have all sides equal and all angles equal) by clicking to determine subsequent vertices. It can be convex or concave. Straight sides must not cross each other. Once selected an area can be deleted using the **Delete area** button. When an area is closed and its name has been defined in the **Area code** field, it can be saved by using the **Save area** button. The areas can partially overlap each other. See the **Using a Coding map** section for more details. Once all areas are added the entire map can be saved using the **Save map** option menu (**Map creator > Save map**). The map is now saved in its own file (.boris\_map) which is NOT part of the BORIS project. A map can be edited at anytime by opening the map file from the **Open map** menu option (**Map creator > Open map**).

## Adding a modifiers coding map to your project

Creating a Coding map is not automatically adding the map to your project. The Coding map have to be added to your project by selecting the corresponding **Behavior type (Point event with coding map, State event with coding map)**. BORIS will ask to select the file name containing the coding map (.boris\_map) and load the coding map in the project. The coding map name will appear in the **Coding map** column and will be saved in the BORIS project file.

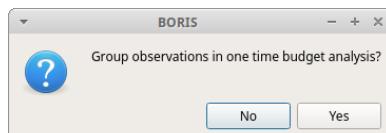
NOTE: If you later modify your coding map you must reload the new version in your BORIS project.

# Analysis

## Time budget analysis

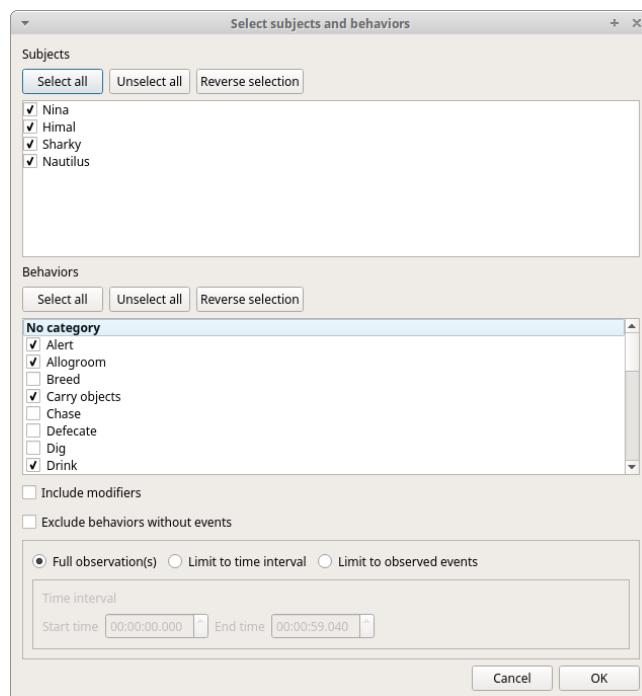
The **Time budget analysis** can be done by behavior (including or not the modifiers) or by category of behaviors. Choose the option from the **Analysis** menu.

The **Time budget analysis** can be done on one or more observations. If you select more than one observation you must then choose for a global time budget analysis that will contain all selected observations or a time budget analysis for every single observation.



Choose **Yes** to group all observations in one time budget analysis

The **Analysis > Time budget** menu option shows the time budget for the events of the selected observations. Select the subjects and behaviors you want to include in the time budget analysis:



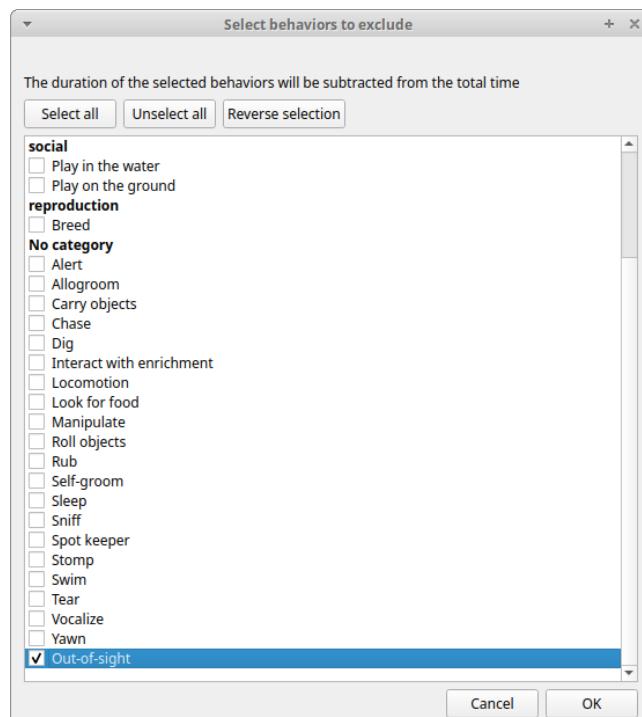
All behaviors can be selected or unselected by clicking on the Category (bold).

You can choose to include or not the behavior modifiers in the Time budget analysis and to exclude behaviors without coded events.

The Time budget analysis can be restricted to a part of the observation:

- Full observation(s): the analysis will be made on the full observation length.
- Limit to time interval: use the **Start time** and **End time** boxes to select starting time and ending time.
- Limit to observed events: the analysis will be made from the first observed event to the last observed event.

The last dialog window will allow you to subtract the duration of one or more behaviors from the total duration of the observation. This can help you if you have defined a "out-of-sight" code in your ethogram for example:



The results contain for each subject and behavior the **total number of occurrences**, the **total duration** (for the behaviors defined as state events), The **duration mean** (for the behaviors defined as state events), the **standard deviation of duration**, the **inter-events intervals duration mean**, th **standard deviation of the inter-events intervals duration** and the **percent of total duration of observation(s)**. All duration times are expressed in seconds (s).

Time budget									
Selected observations									
Total observation length: 00:07:25.680									
Subject	Behavior	Modifiers	Total number	Total duration (s)	Duration mean (s)	Duration std dev	inter-event intervals mean (s)	inter-event intervals std dev	% of total length
1 Nina	Alert	-	1	1.882	1.882	NA	NA	NA	0.4
2 Nina	Locomotion	-	2	2.913	1.457	0.824	NA	NA	0.7
3 Nina	Manipulate	-	2	65.048	32.524	15.414	NA	NA	14.6
4 Nina	Rest	-	1	25.549	25.549	NA	NA	NA	5.7
5 Himal	Alert	-	4	18.131	4.533	2.542	10.95	NA	4.1
6 Himal	Allgroom	-	1	3.603	3.603	NA	NA	NA	0.8
7 Himal	Drink	-	1	1.133	1.133	NA	NA	NA	0.3
8 Himal	Locomotion	-	10	36.633	3.663	5.777	6.124	5.88	8.2
9 Himal	Manipulate	-	3	66.702	22.234	17.18	4.723	NA	15.0
10 Himal	Play on the ground	-	1	16.46	16.46	NA	NA	NA	3.7
11 Himal	Rub	-	1	11.031	11.031	NA	NA	NA	2.5
12 Himal	Self-groom	-	1	3.571	3.571	NA	NA	NA	0.8

Save results

Close

The time budget results can be saved in various formats for further analysis: \* Plain text files: TSV, CSV or HTML \* Spreadsheets: Open Document (ODS), Microsoft Excel (XLSX) and Legacy Microsoft (XLS)

### Note

If a STATE behavior has an odd number of coded events, BORIS will report "UNPAIRED" instead of results"

## Synthetic time budget

The synthetic time budget is similar to time budget but with fewer parameters and a different organization of results. Results of all selected observations are organized in columns on a single page. Two parameters are provided for now: **number of occurrences** and **total duration** (for the behaviors defined as state events)

	A	B	C	D	E	F	G	H	I	J	K
1		Total length (s)	Nina Alert	Nina Alert	Nina Allogroom	Nina Allogroom	Nina Breed	Nina Breed	Nina Carry objects	Nina Carry objects	Nina Chas
2											
3											
40	0037	32.160									
41	0038	86.880	1.675	1							
42	0039	96.960	13.471	2							
43	0040	335.520	86.615	12	15.84	2					
44	0041	130.560	9.069	2	5.85	1					
45	0042	36.960	24.298	3							
46	0043	107.040	42.928	5							
47	0044	109.920	11.769	4	12.471	2					
48	0045	102.240	72.857	6							
49	0046	84.960	44.459	6							
50	0047	77.280	68.929	2							
51	0048	23.520	3.1	1	10.797	1					
52	0049	34.080	9.897	2	8.431	1					
53	0050	23.040									
54	0051	44.160	1.079	1							
55	0052	18.240									
56	0053	39.840									
57	0054	35.040									
58	0055	256.800	61.321	15	19.73	1					
59	0056	26.400	15.241	5							
60	0057	45.120	6.887	3	10.984	2					
61	0058	45.120	2.647	1							
62	0059	41.760									
63	0060	292.800	1.543	2							
64	0061	25.920									

All duration times are expressed in seconds (s).

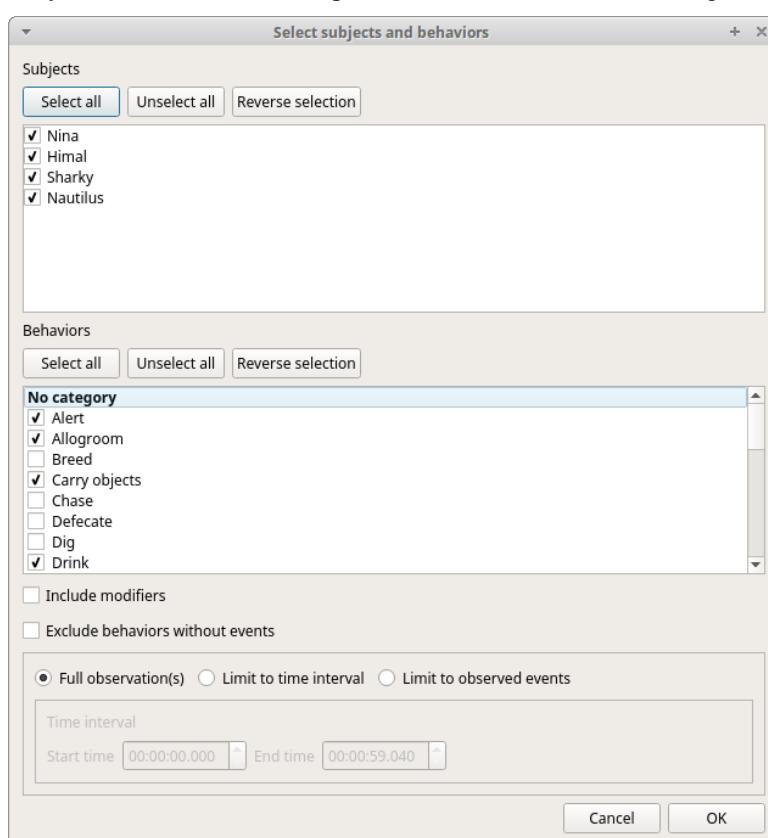
The time budget results can be saved in various formats for further analysis:  
 \* Plain text files: TSV, CSV or HTML  
 \* Spreadsheets: Open Document (ODS), Microsoft Excel (XLSX) and Legacy Microsoft (XLS)

## Plot events

The coded events from many observations can be plotted along a time axis.

### Analysis > Plot > Plot events

The subjects and behaviors you want to include in the plot can be selected in the following window:



You can choose to include or not the behavior modifiers (if any) and to exclude behaviors without coded events.

The time interval can be selected (See time budget)

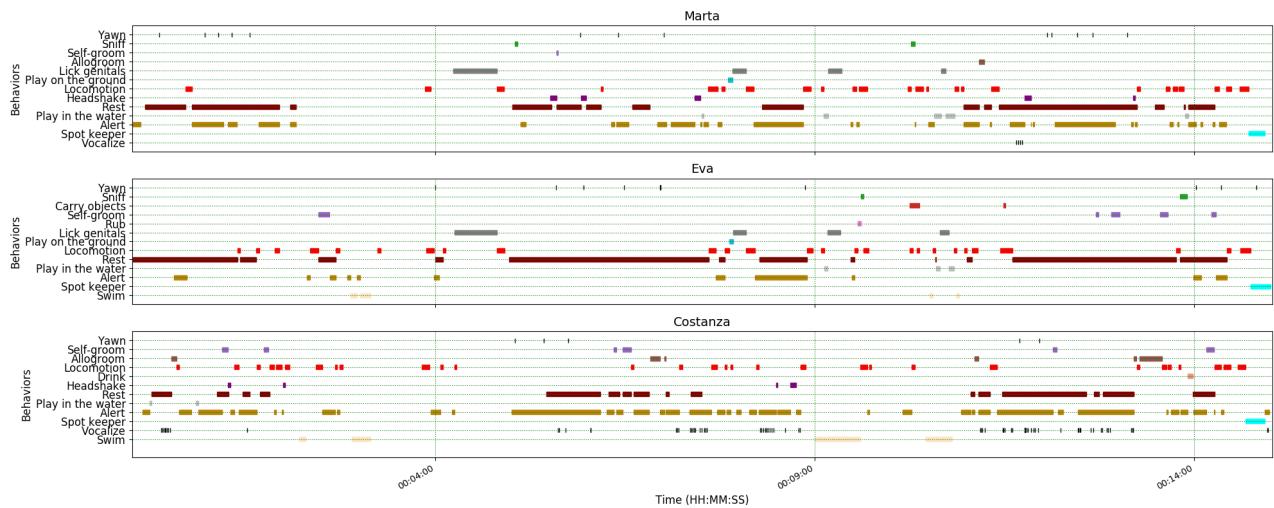
The plot can be exported in various formats like bitmap (PNG, JPG, TIFF) or vectorial graphic (SVG, PDF, EPS, PS). The SVG format can be further edited with the [Inkscape vector graphics editor](#).

### Note

If a STATE behavior has an odd number of coded events, you will see this error message: "The STATE behavior XXX is not paired"

This function creates one plot by subject on one figure.

The color of behaviors can be customized. See [plot colors](#)

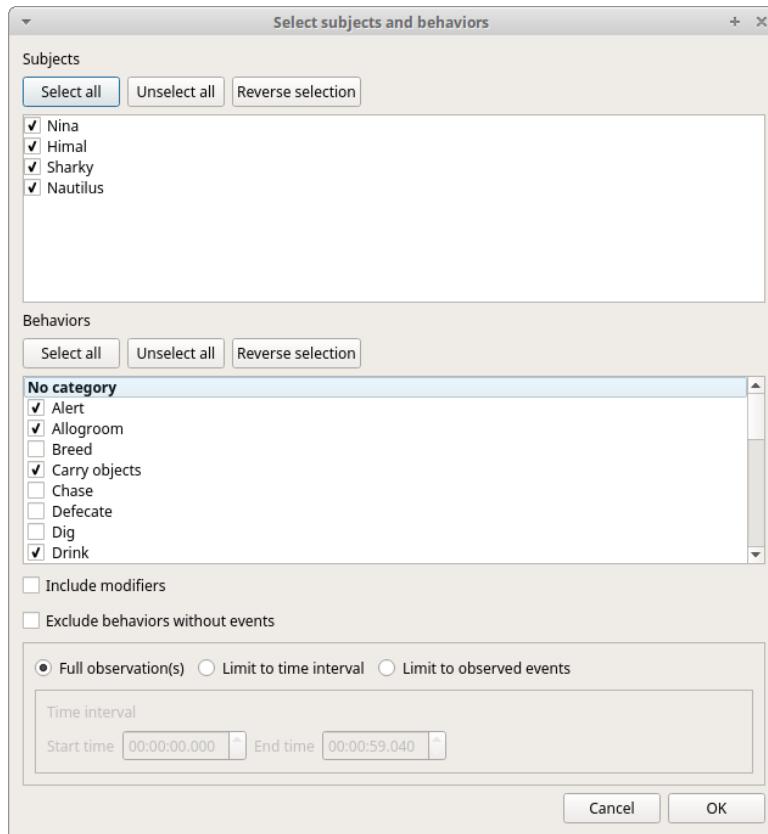


### Plot time budget

The duration and number of occurrences can be plotted for each subject and behavior.

**Analysis > Plot > Plot time budget**

The subjects and behaviors you want to include in the plot can be selected in the following window:



The behavior modifiers can not be included in the plot for now.

The time interval can be selected (See time budget)

The plot can be exported in various formats like bitmap (PNG, JPG, TIFF) or vectorial graphic (SVG, PDF, EPS, PS).  
The SVG format can be further edited with the [Inkscape vector graphics editor](#).

#### Note

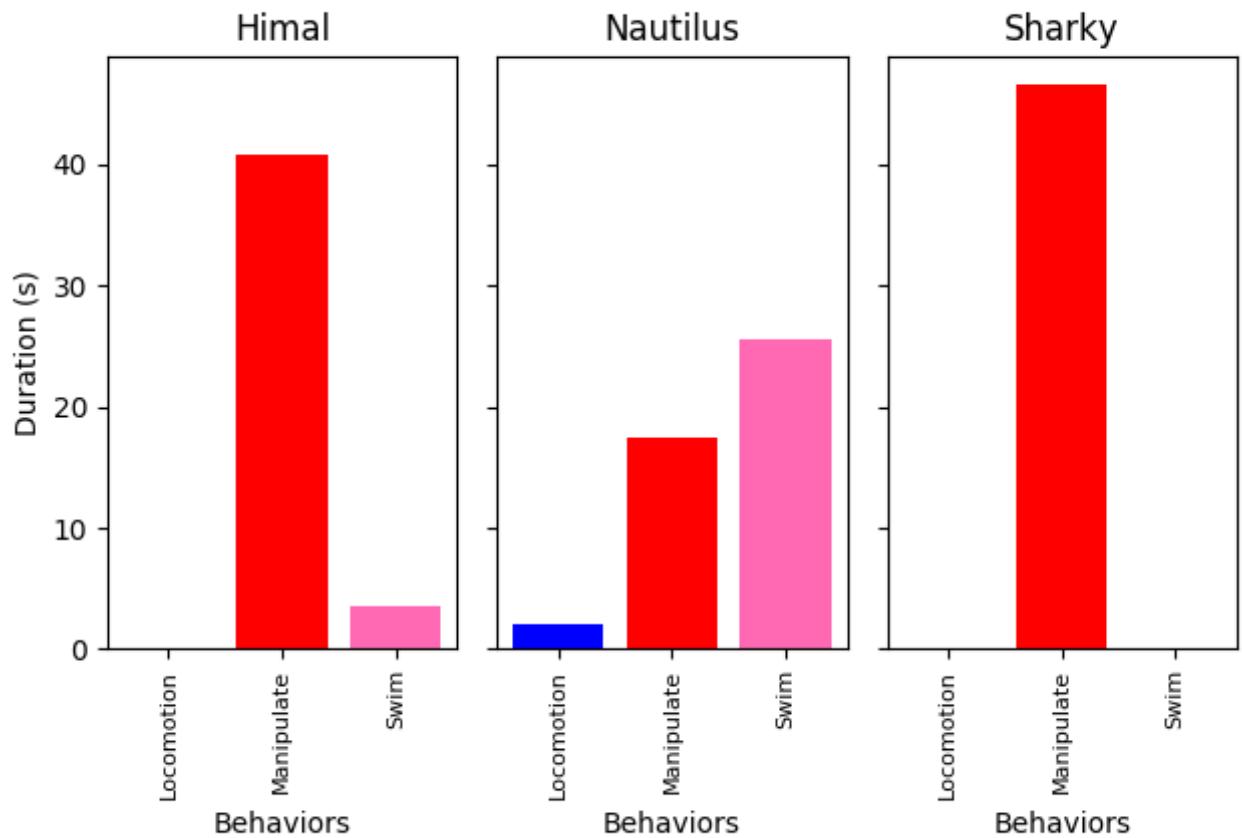
If a STATE behavior has an odd number of coded events, you will see this error message: "The STATE behavior XXX is not paired"

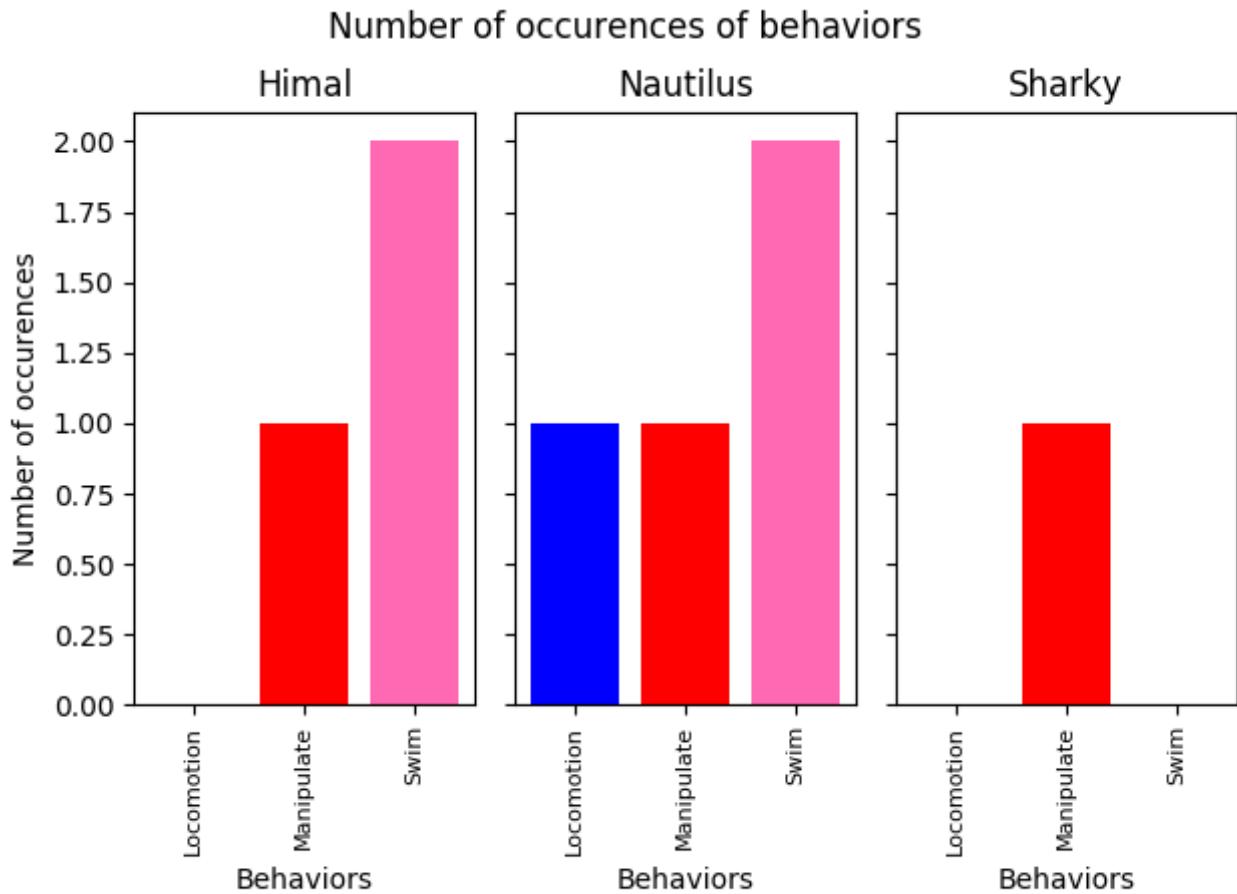
This function creates 2 plots with all subjects for each observation:

- a plot of the behavior durations for the behaviors defined as STATE event.
- a plot of the number of occurrences for all the behaviors.

The color of behaviors can be customized. See [plot colors](#)

### Durations of behaviors



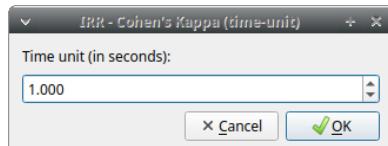


## Inter-rater reliability

The Cohen's kappa coefficient can be calculated (**Analysis > Inter-rater reliability > Cohen's kappa**).

[Cohen's kappa on Wikipedia](#)

After selecting 2 observations and a time window (in seconds) for the analysis (the default value is 10 seconds) the Cohen's kappa will be displayed in the results window.



## Implementation of IRR Cohen's Kappa in BORIS

If a time window of n seconds is set the 2 selected observations will be checked every n seconds for agreement/disagreement from the first event to the last event of the 2 observations . In case of a point event the presence of a corresponding event in the other observation will be verified using a time window of n seconds centered on the point event.

A IRR Cohen's Kappa analysis is available in the GSEQ program (<http://www2.gsu.edu/~psyrb/gseq>). For this the coded events can be exported as aggregated events in SDIS format. See [export aggregated events](#)

## Advanced event filtering

## Warning

This function is still experimental in v. 7.8.5

This function allows to filter events from one or more observations by selecting subjects, behaviors and logical operators.

To use the filter, select a subject, select a behavior and click on the button with the green arrow on the side of the behaviors list. The tuple subject/behavior will be added in the **filter text edit**. A complex filter query can be constructed by adding parenthesis and logical operator (**AND & / OR |**) for combining various subjects and behaviors.

Example of a summarized output showing the occurrences of Himal resting while Nautilus in alert:

Observation id	Number of occurrences	Total duration	Mean	Std Dev
1 0015	1	3.351	3.351	NA
2 0019	2	8.581	4.29	2.461
3 0028	1	3.905	3.905	NA
4 0030	1	8.231	8.231	NA
5 0032	1	14.245	14.245	NA
6 0043	1	9.219	9.219	NA
7 0048	1	7.016	7.016	NA

Example of a detailed output showing the overlapping intervals while Himal rests and Nautilus is in alert:

**Advanced event filtering**

**Filter**  
"Himal|Rest" & "Nautilus|Alert"

Summary       Details

**Subjects**  
Himal  
Nautilus  
Nina  
Sharky

**Behaviors**  
Alert  
Allogroom  
Breed  
Carry objects  
Chase  
Defecate  
Dig  
Drink  
Eat  
Interact with enrichment  
Locomotion  
Look for food  
Manipulate  
Play in the water

**Logical operators**  
AND  
OR

**Results (146 events)**

	Observation id	Comment	Start time	Stop time	Duration
1	0019		3.559	6.088	2.529
2	0019		14.907	17.457	2.550
3	0028		0.0	3.905	3.905
4	0048		18.333	18.627	0.294
5	0054		28.77	30.019	1.249
6	0138		174.796	178.699	3.903
7	0138		180.77	181.107	0.327

**Save results**    **Close**

The same subject can be used many times in the query with OR or AND (in case of non exclusive behaviors):

**Advanced event filtering**

**Filter**  
"Himal|Drink" | "Himal|Eat"

Summary       Details

**Subjects**  
Himal  
Nautilus  
Nina  
Sharky

**Behaviors**  
Carry objects  
Chase  
Defecate  
Dig  
Drink  
Eat  
Interact with enrichment  
Locomotion  
Look for food  
Manipulate  
Play in the water

**Logical operators**  
AND  
OR

**Results (188 events)**

	Observation id	Comment	Start time	Stop time	Duration
1	0004		49.275	50.408	1.133
2	0009		5.579	10.48	4.901
3	0009		18.683	40.19	21.507
4	0013		24.324	30.869	6.545
5	0055		3.65	5.326	1.676
6	0055		21.362	29.117	7.755
7	0056		22.885	26.37	3.485

**Save results**    **Close**

An unlimited number of conditions can be used:

**Advanced event filtering**

**Filter**: "Himal|Rest" & "Nautilus|Rest" & "Nina|Rest" & "Sharky|Rest"

Summary       Details

**Subjects**: Himal, Nautilus, Nina, **Sharky**

**Behaviors**: Rest, Roll objects, Rub, Self-groom, Sleep, Sniff, Spot keeper, Stomp, Swim, Tear, Urinate

**Logical operators**: AND, OR

**Results (2 observations)**

Observation id	Number of occurrences	Total duration	Mean	Std Dev
1 0999	1	0.403	0.403	NA
2 1000	2	2.769	1.384	1.592

**Save results**    **Close**

Parenthesis can be used to group logical conditions into block(s):

**Advanced event filtering**

**Filter**: "Himal|Alert" & ("Nautilus|Eat" | "Nina|Eat" & "Sharky|Eat")

Summary       Details

**Subjects**: Himal, Nautilus, Nina, **Sharky**

**Behaviors**: Carry objects, Chase, Defecate, Dig, Drink, Eat, Interact with enrichment, Locomotion, Look for food, Manipulate, Play in the water

**Logical operators**: AND, OR

**Results (30 events)**

Observation id	Comment	Start time	Stop time	Duration
1 0055		8.689	12.435	3.746
2 0055		19.778	21.361	1.583
3 0055		29.118	30.692	1.574
4 0055		66.079	69.591	3.512
5 0055		199.744	200.81	1.066
6 0239		149.63	152.719	3.089
7 0230		163.807	164.850	1.052

**Save results**    **Close**

The results can be saved in a Tab Separated Values (TSV) file using the **Save results** button. Other formats will be added in future.

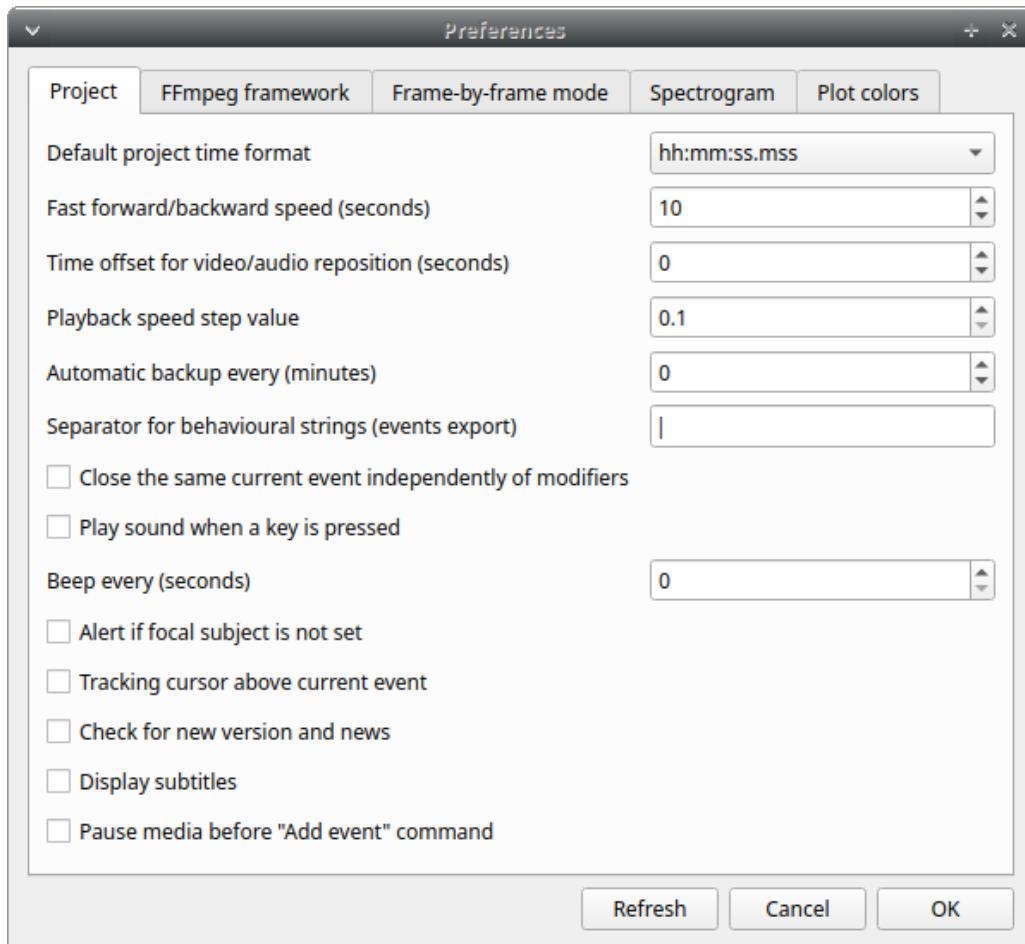
## Latency

The latency will analyze the time between one or more markers (arbitrary behaviors(s)) and other behaviors.

# Preferences

You can customize BORIS using the Preferences window (**File > Preferences**)

## General preferences



### Refresh button

Option to reinitialize the configuration to default. BORIS will be closed.

### Default project time format

This option allows the user to choose the format for displaying time in the project. Please note that time is internally always saved in seconds with a precision of 3 decimal digits

### Fast forward/backward speed (seconds)

This option allows the user to customize the amount of time for "jumping" forward or backward in media.

### Time offset for media reposition (seconds)

This value indicates the time offset for repositioning the media after double-click on a row event of the *Events* table. 'for example -4 seconds indicates that after a double-click the media will be repositioned 4 seconds before the recorded event.'

### Playback speed step value

This value indicate how much the speed will be increased or decreased after pressing the *change playback speed* buttons.

### Auto-save project every (minutes)

if set BORIS will save your project automatically every n minutes. 0 indicate no automatic backup. The project will be saved if the project is already saved and an observation is open.

### Separator for behavioural strings

Character (or string) used to separate behaviors when exporting events as behavioural strings. See also Behatrix

#### **Close the same current event independently of modifiers**

Option used to STOP the current behavior without regarding the modifiers

#### **Play sound when a key is pressed**

Activate a sound signal after every keypress event

#### **Alert if focal subject is not set**

If this option is activated BORIS will show an alert box if no focal subject is selected

#### **Tracking cursor above current event**

Check this box to position the tracking cursor above the current event in events list table.

#### **Check for new version**

Check for new version on BORIS web site every 15 days (internet access required)

#### **Display subtitles**

Option to display or hide the visualization of subtitles. In case of separate file, the file containing subtitles must have the same base name than the video files with a .srt extension.

#### **Pause media before "Add event" command**

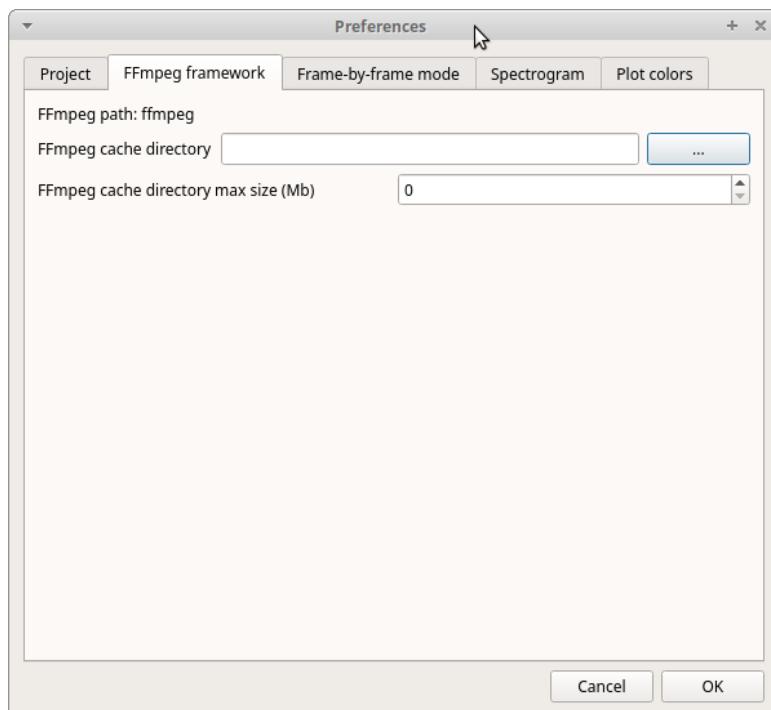
Option to pause the media before manually adding an event.

## Results

#### **Compact time budget**

This option allow to produce a compact time budget with the same number of columns for each row.

## FFmpeg framework



The path for the ffmpeg executable program is displayed. The FFmpeg executable is now included with BORIS in Windows and Mac OS versions.

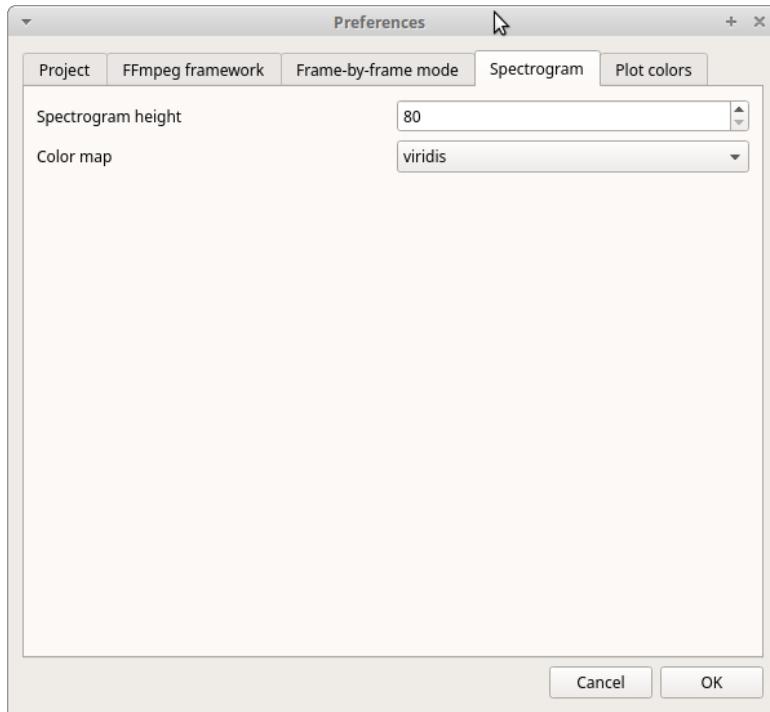
#### **FFmpeg cache directory**

This indicates the directory that will be used as image cache for frame-by-frame mode and spectrogram visualization. If you do not specify a path, BORIS will use the default temporary directory of your system.

#### **FFmpeg cache directory max size**

Indicate a size limit (in Mb) for the image caching. 0 indicates no limit

## Spectrogram



### Spectrogram height

Select the height of generated spectrogram (in pixels). You will need to restart the current observation to apply changes.

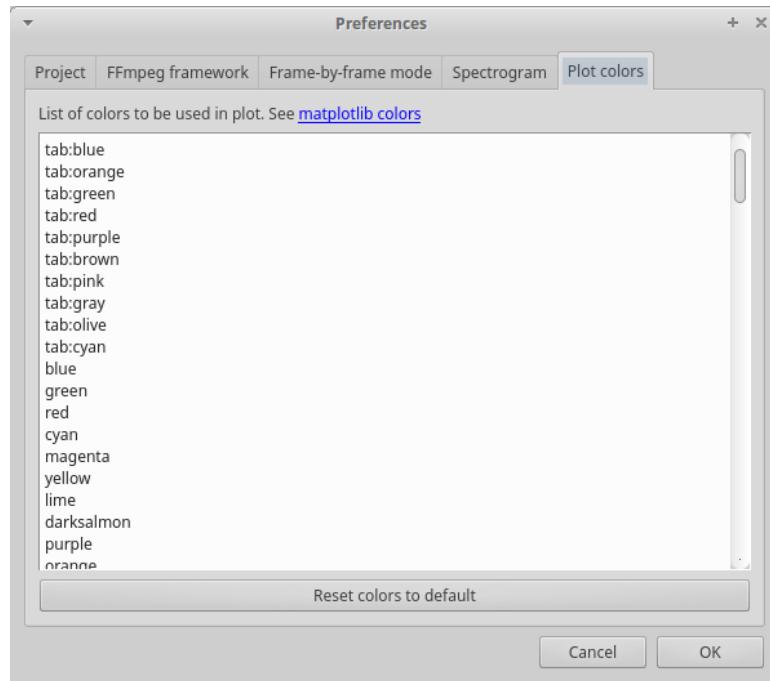
### Color map

Select the color map for displaying the generated spectrogram. See [Matplotlib colormaps](#) for details.

### Plot colors

The color of behaviors in the plot events functions can be customized. The first color will be associated to the first behavior in your ethogram, the second color to the second behavior and so on. Various color formats can be used to specify a color: **named color** or **hex RGB** (like #0F0F0F). See [https://matplotlib.org/api/colors\\_api.html](https://matplotlib.org/api/colors_api.html) and [https://matplotlib.org/examples/color/named\\_colors.html](https://matplotlib.org/examples/color/named_colors.html) for details

The **reset colors to default** button will reload the default colors.



## Various

### Removing path of media files

Using BORIS you can choose to store the full path of the media files into the file project (for example: /home/user/Video/video\_n1.mp4 or c:\Users\user\Documents\video1.avi).

If you want to move your project on a different computer or if you want to move your media files you may want to do not store the full path. For this you can choose to add media files without the full path (See **Add media files** section). You can also remove the full path of your media files from the all observations of the current project (**File > Remove path from media files**). Please note that this operation is irreversible. After remotion the full path of your media will be lost and will not be recoverable.

**If you choose to do not store the full path of media files the media files must located in the same directory than your BORIS project file.**

### Citing BORIS

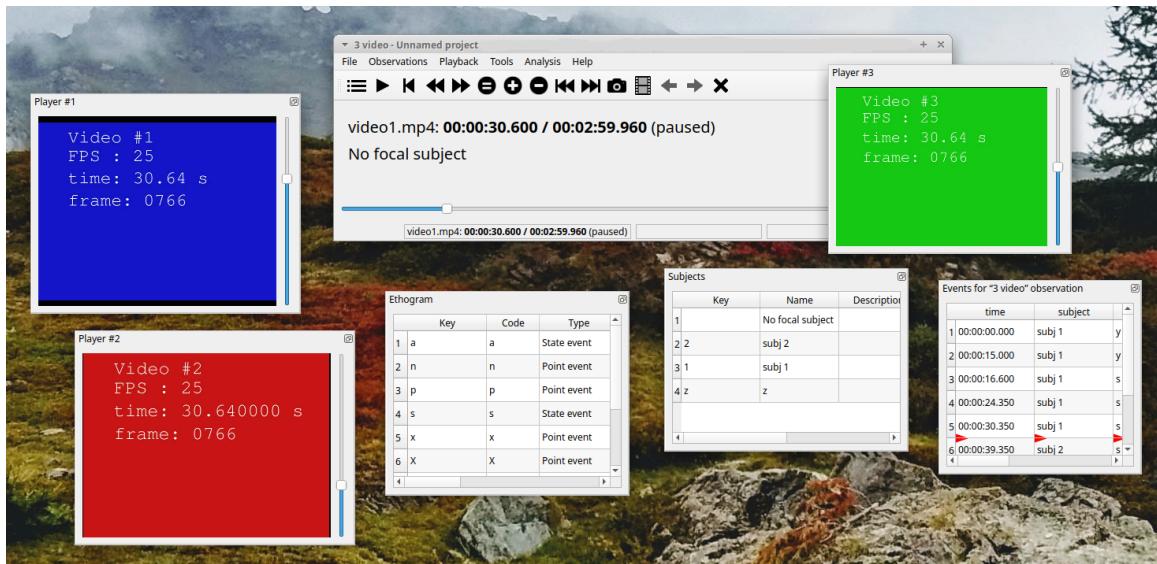
If you have used BORIS for publications, please cite:

Olivier Friard and Marco Gamba. (2016) BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology and Evolution*, 7(11), 1324-1330  
DOI: 10.1111/2041-210X.12584

## Docking

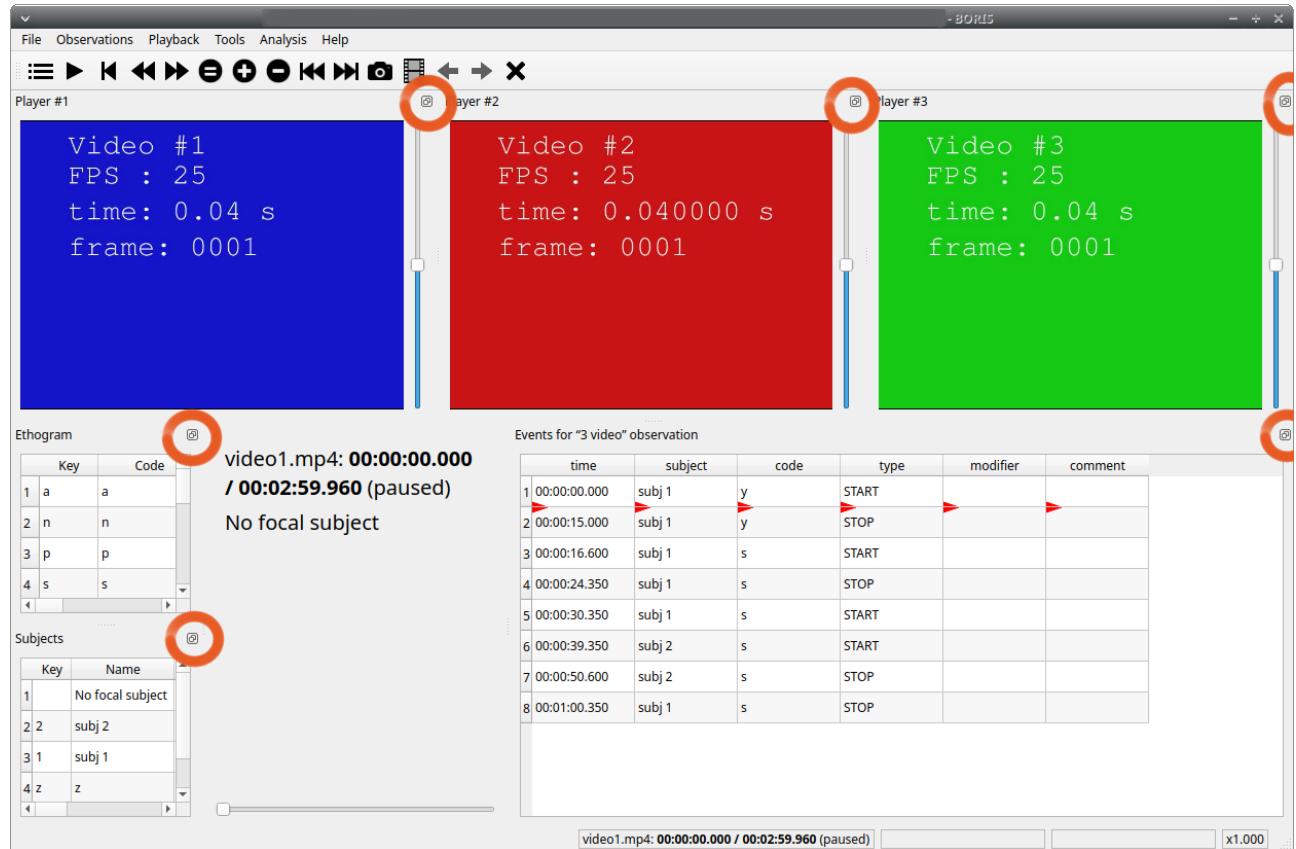
All elements, including all the media players can be undocked from the main window and positioned where you prefer (e.g. they can be on the same desktop over one or many screens).

The position of the various widgets is saved in the [configuration file](#) at the end of the work session.

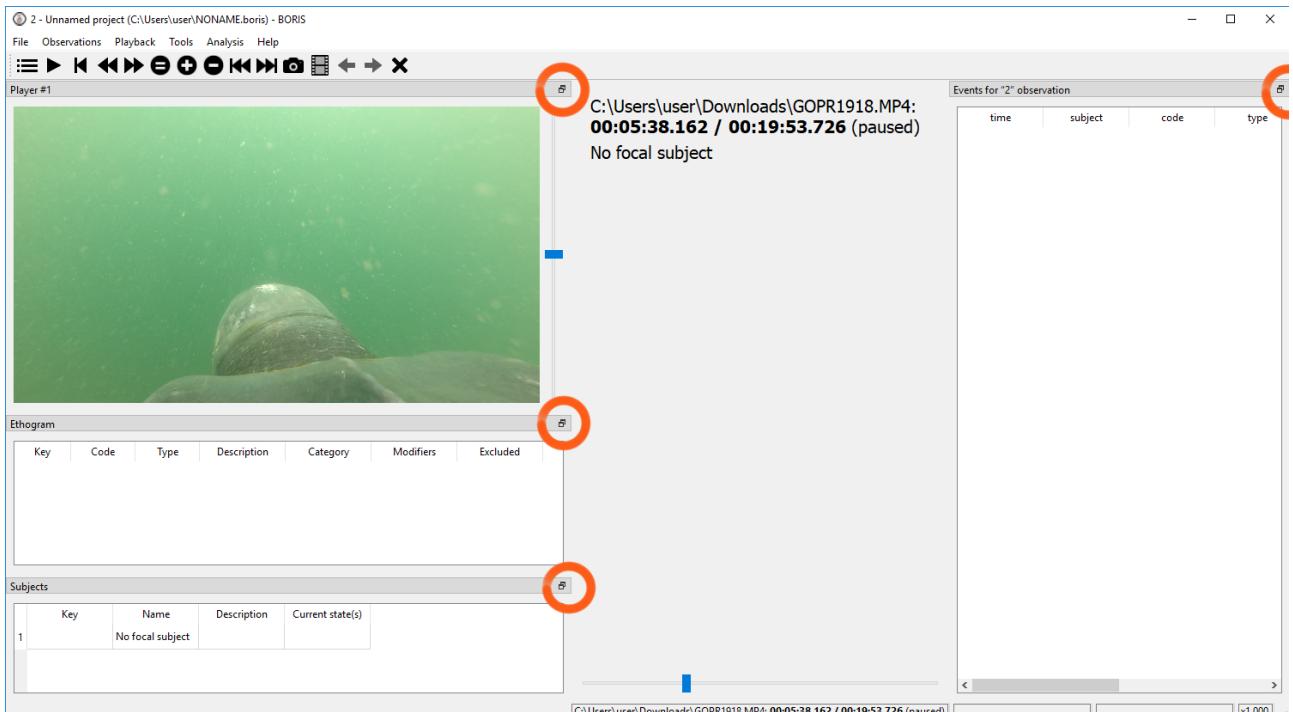


Click the icon present at the top-right corner of the widget (for MacOS the icon is located at the left-top corner) will undock the widgets that can be repositioned on another docking area or moved out of the main window. A double-click on the top bar of the widget will reposition it on the main window.

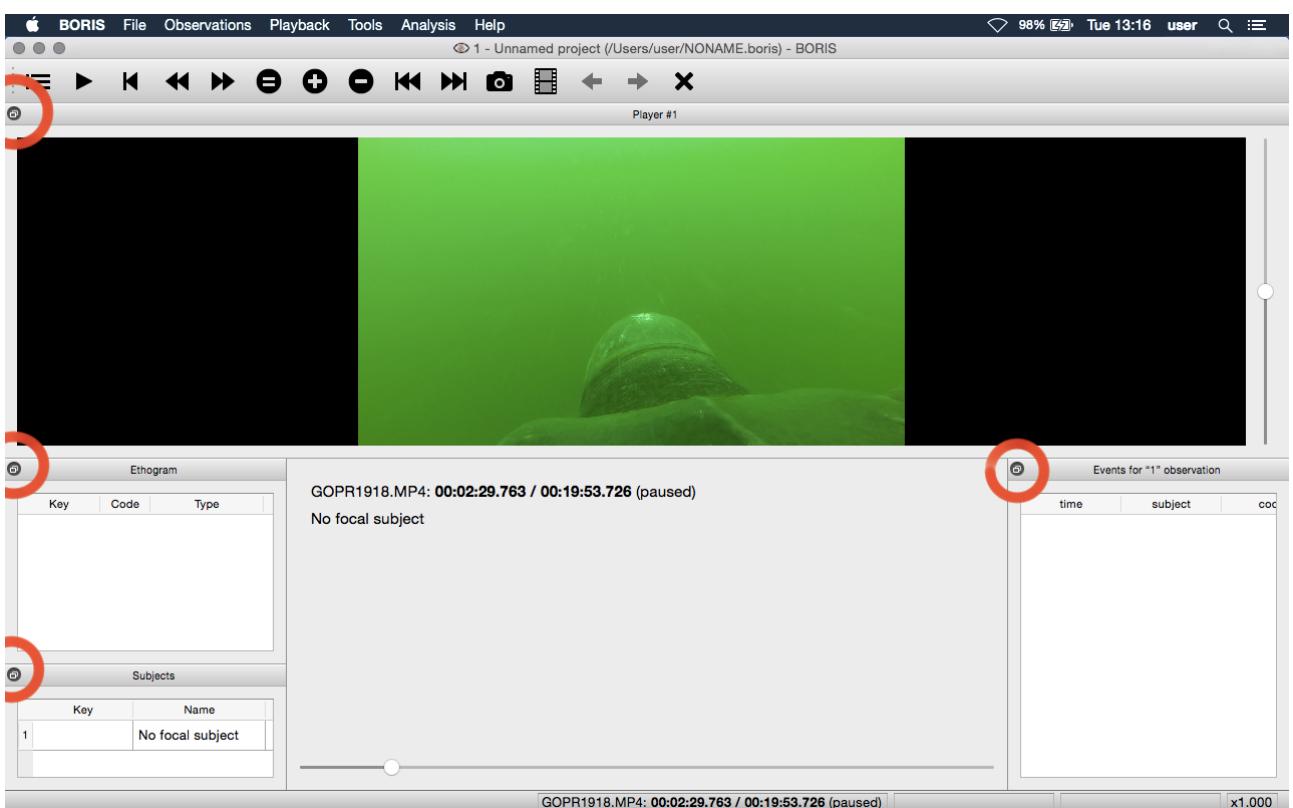
For Linux:



For Microsoft-Windows:



For Mac OS:



If you feel uncomfortable with the dockwidgets you can block them on the main window by checking the **Lock dockwidgets** option (see **Tools > Lock dockwidgets**). All the dockwidgets will be docked on the main window and locked on it except the player dockwidgets.

## Configuration files

BORIS saves the configuration (user preferences, windows position, widgets position) in a configuration file. This file is named **.boris** and it is saved on the home directory of the current user:

```
for Linux:  
/home/USERNAME/.boris  
  
for Microsoft-Windows:  
C:\Users\USERNAME\.boris  
  
for MacOS:  
/Users/USERNAME/.boris
```

If you have some trouble using BORIS try to close the program, delete this file and relaunch BORIS.

The **recent projects list** is saved on the **.boris\_recent\_projects** file in the home directory of the current user.

## Bug reports and features request

Please send bug reports and features request using the BORIS GitHub repository (<https://github.com/olivierfriard/BORIS>) or by e-mail in english, french or italian (see web site <http://www.boris.unito.it/pages/about.html>).

In case of bug report please verify that you are using the **lastest version of BORIS** and indicate your operating system, its version and the CPU architecture (32/64 bits). You may also include the BORIS project that gave you an error. Any information you will provide will not be disclosed to any third party.

A debugging version (for Windows 64-bit) is available if you want to collaborate to BORIS.

## Acknowledgement

The authors would like to acknowledge Valentina Matteucci, Laura Ozella, Giovanni Di Panfilo and all others users for their precious help.

## Valid keys for triggering behavior

BORIS makes difference between lower case and upper case characters

- keys from a to z
- keys from A to Z
- keys from 0 to 9
- function keys from F1 to F12
- à é è ù ì ç
- ! " £ \$ % & / ( ) = ? ^ [ ] { } @ | § ° #