Content

What is it?

Reports4Unity is the easy solution to create unlimited reports about the assets and game objects in your unity projects. These reports are generated as HTML, CSV or plain text files for you to store, review and process in other applications. Get all information about the assets, scenes and game objects in your projects with just a few mouse clicks.

Why using Reports4Unity?

Did you ever ask yourself one of the following questions?

- How many textures are in my game and how much space do they use on a specific platform?
- What are the most space consuming assets in my game?
- Did I use the correct platform import settings for my assets in order to get the best performance for my game?
- What scaling and compression was used for my textures when Unity imported them?
- Are all my texture sizes a power of two?
- Which assets should I optimize to gain a better performance or download size?
- Are there missing scripts, meshes or other references in one of my scenes that will crash my game on runtime?

- Which assets depend on which other assets? Is something missing there?
- What assets have been changed since the last build, on purpose or maybe accidently?
- What new assets are in my game and which assets have been removed since the last build or iteration?

To answer some of these and many other questions regarding your game assets and levels you usually have to spend much time searching and clicking in the Unity editor or the file system. Using *Reports4Unity* you now have all these information right at your fingertips within seconds. And you can get them over and over again without any effort. Just select the information you are interested in, set some filter and sorting criteria and get a complete report in HTML format for reviewing or CSV for further processing in Excel.

There are a lot of report formats included in the package, but you can create your own in minutes by adding new templates.

What's included?

There are 5 report types you can generate with an unlimited number of different formats. You can always select any subsets of information and filter the result in many ways to get only the specific data you are interested in. Please find a detailed description of all included reports, templates, filter options, template placeholders and fields to use in the *Documentation* section of this manual.

Asset Report

This report scans the asset folder in your project and collects all information about the asset folder hierarchy and the asset files. It reports all asset file information, texture sizes and some statistics by file type.

Asset Importer Report

The *Asset Importer Report* prints out all importer settings for textures, audio files, models, movies and fonts in your project.

Asset Change Report

Every time an *Asset Report* is created the system will track the current state of your assets in order to compare it to a future state. This report shows you which assets have been changed, added or deleted since the date you select in the report window.

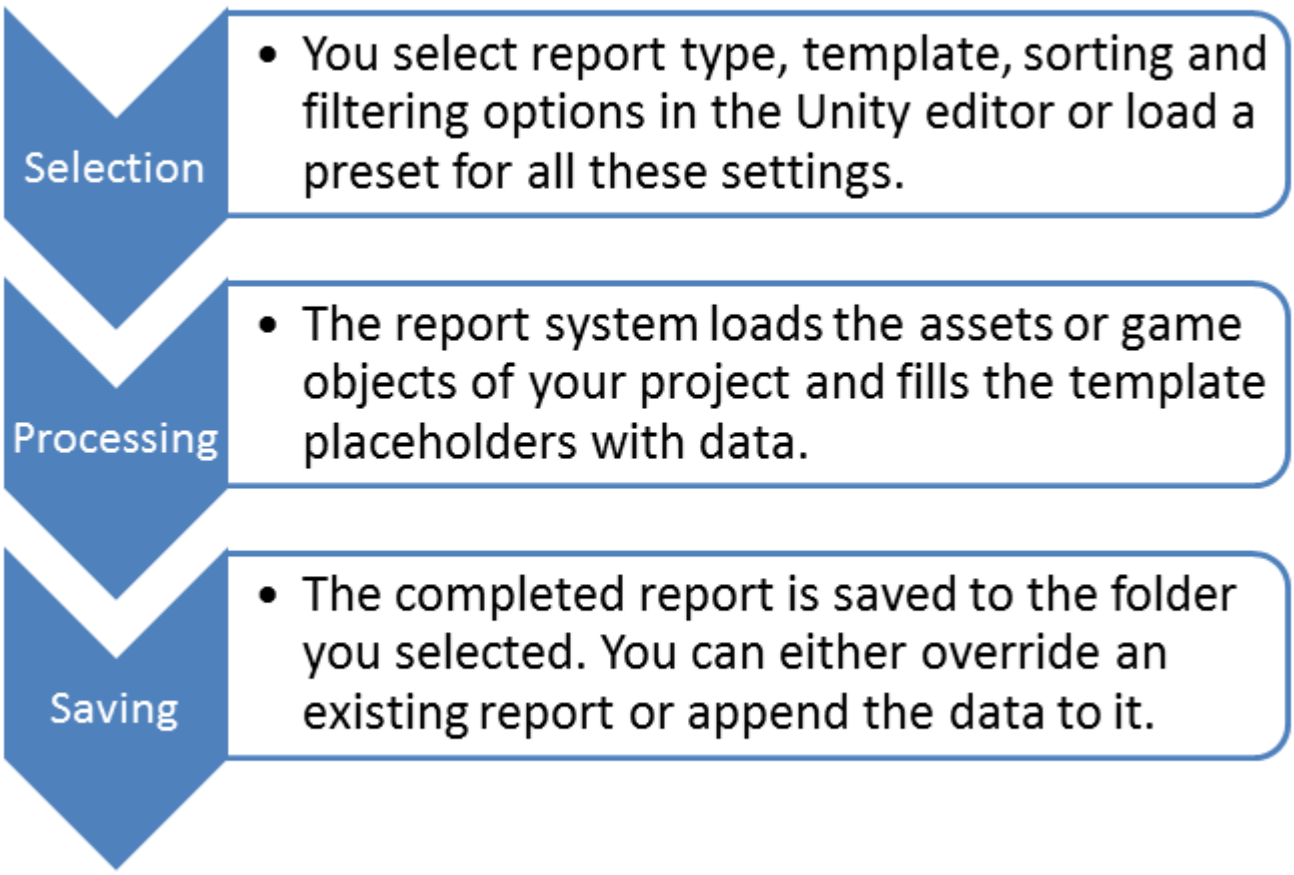
Asset Dependencies Report

This report shows you dependencies and reverse dependencies of all assets in your project or a subset of them. It also collects and reports missing dependencies for you to investigate and fix.

Scene Report

The *Scene Report* gives you a complete overview of all your scenes or just the scenes you select. It reports all game objects with its components and their public fields.

How does it work?



To create a report you have to follow these steps: To generate the same report with custom information subset and options multiple times you can save your settings as a preset and then load it to create the same report without the need to configure it again.

The report system then collects all the information you selected. It uses the template file, fills in this information and saves the report as a HTML or CSV/text file. By choosing a different template you can display the data in a different format. A template is a text file (i.e. in HTML format) with placeholders in it for the report data. The placeholders in the example below are enclosed by “[--“ and “--]” so the system recognizes them as placeholders for particular data.

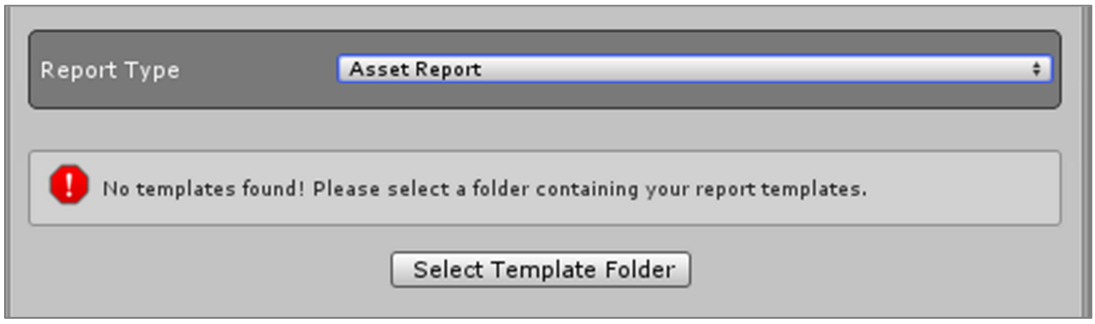
```
<h2>Statistics</h2>
<table cellpadding="2" cellspacing="1" style="width:350px;">
<tr>
  <td>Report Date:</td>
  <td>[--ReportDate--], [--ReportTime--]</td>
</tr>
<tr>
  <td>Build Target:</td>
  <td>[--BuildTarget--]</td>
</tr>
<tr>
  <td>Asset Folder Count:</td>
  <td>[--AssetFolderCount--]</td>
</tr>
<tr>
  <td>Asset Count Total:</td>
  <td>[--AssetCount--]</td>
</tr>
<tr>
  <td>Asset Filesize Total:</td>
  <td>[--AssetTotalFileSizeAuto--]</td>
</tr>
</table>
```

Getting started

In this walkthrough we will create an *Asset Report* to identify the textures in the project with the biggest file size. With this information we can try to reduce the download size of our game if there's a chance to.

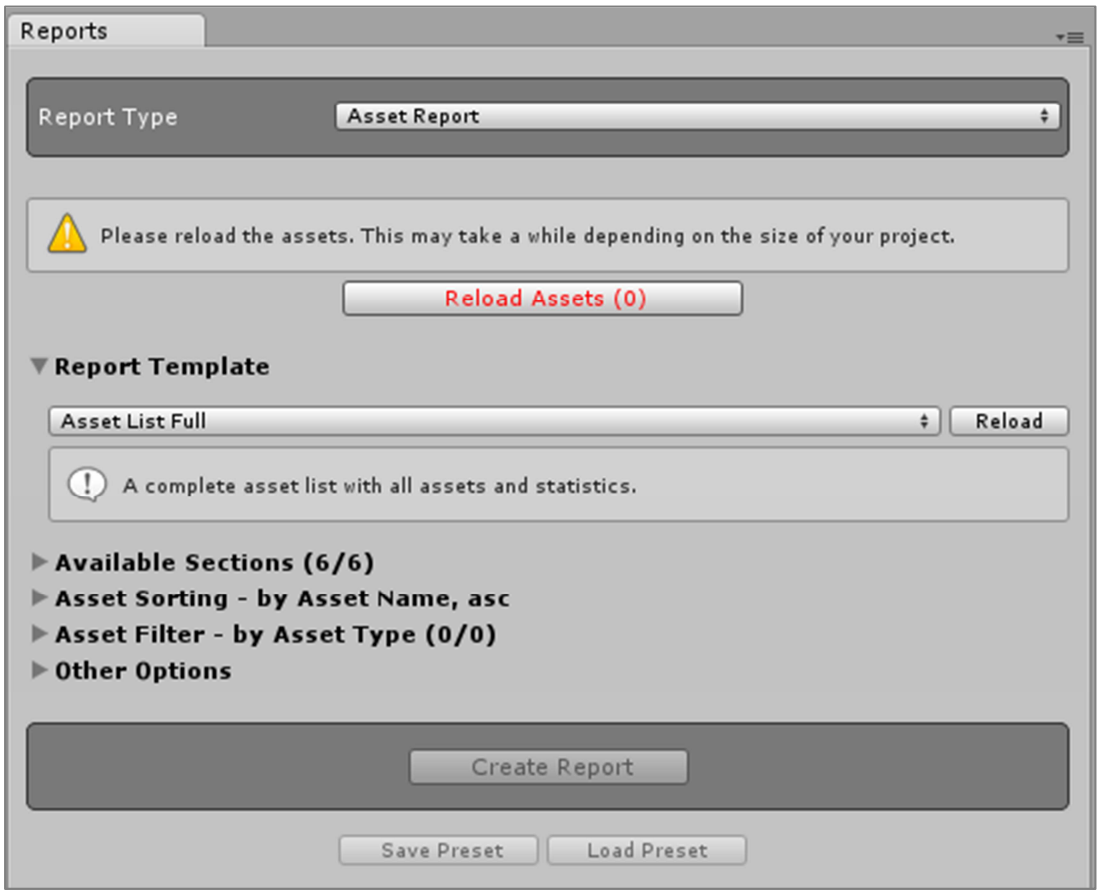
Open the Asset Report window by selecting the menu item *Window/Reports4Unity/Asset Report* in the Unity editor.

If the following message shows up, you have to select the folder where your templates for the selected report type are stored. There are some example templates in the installation folder of *Reports4Unity* for you to start with.



Please note that the location of your report templates is stored in the editor preferences, so they are the same for all your projects. If you want to share report templates between projects it's a good idea to move the template folder to a location outside of a specific project on your computer or network. After moving it you will be prompted by the message above to select the new location.

Asset loading

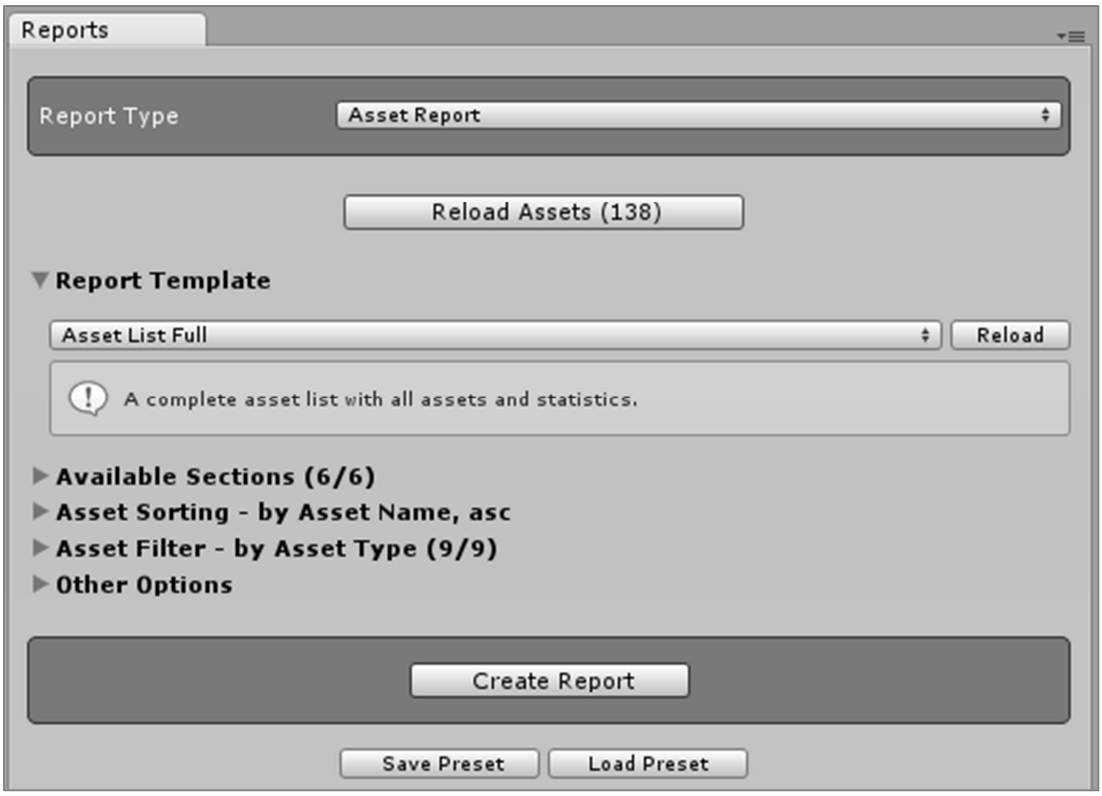


When we first open this window you need to reload the assets of your project. This is not done automatically because it may take some seconds depending on the size of the project. So click the button *Reload Assets*.

Note: This message occurs after every change in your asset folders (new files, changed files).

Selecting report type and template

Now let's start by selecting the report type *Asset Report* from the dropdown at the top of the window. The Report Template foldout opens to show you the list of available templates for this type of reports.

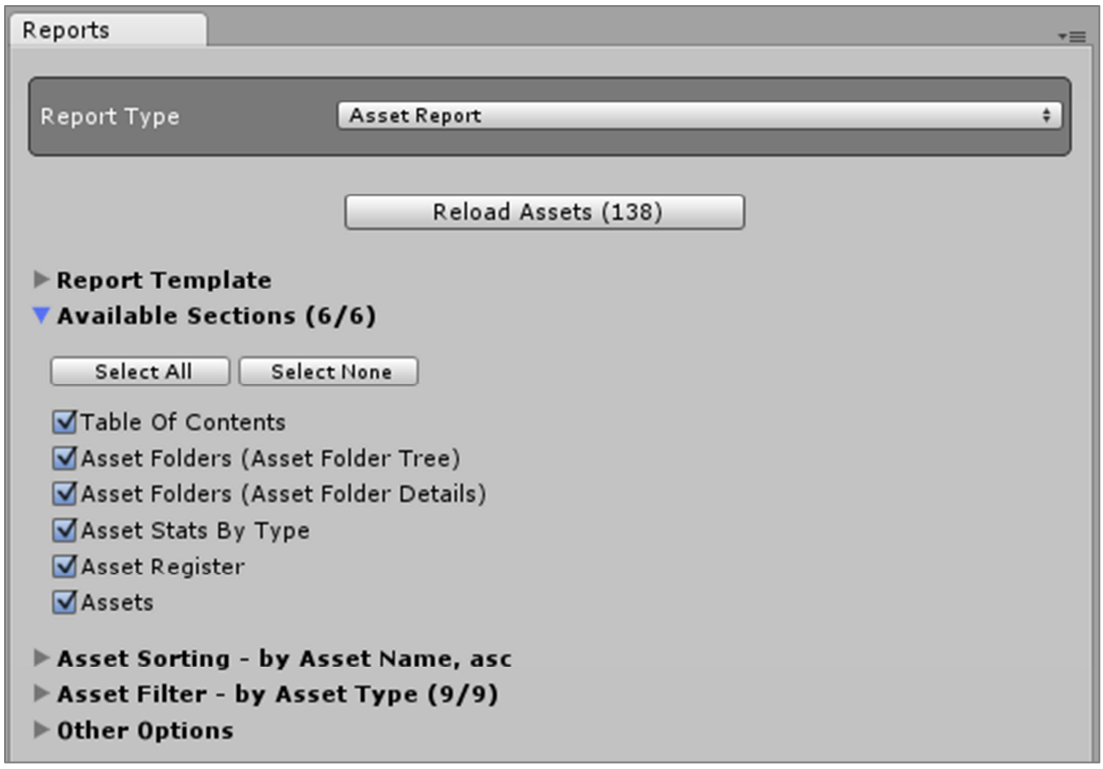


The templates are loaded from a folder in your file system containing the necessary files. After you downloaded and imported *Reports4Unity* from the Asset Store a folder with example report templates is installed as a starting point. If you developed additional templates and want to store them outside of your project you can set a different template folder in the *Reports4Unity* settings (menu item *Window/Reports4Unity/Preferences*).

If you added or deleted any assets you can reload the assets in all asset folders by clicking the button *Reload Assets*. The numeric value on the button is the number of assets found in your asset folder. Excluded files and folders are ignored when counting the assets.

Selecting the data

Depending on the template you can now select the data to generate in the report. Each template contains a number of sections for different kind of data. Which of these sections are available is defined in the template itself. For the *Asset Report* for example you can define up to 6 sections to choose from as you can see in the screen shot below.

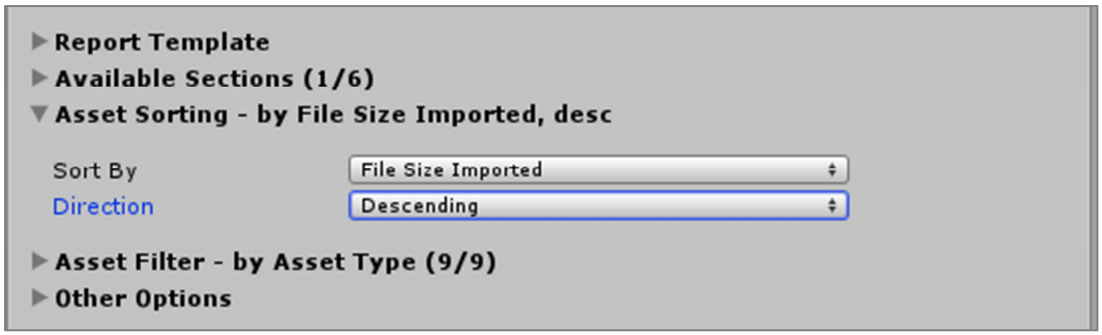


Although this template defines the visual presentation of all 6 sections you can uncheck the sections you don't need in this report. As we are planning to investigate the texture sizes of our game we only need to check the *Assets* section. For a description of the other sections please read the *Asset Report* documentation later in this manual.

Setting asset sorting

In order to make it easier to identify the big ones from our textures we should change the sort criteria of the asset list in our report. By default the assets are sorted by asset name alphabetically. We change this to *File Size Imported* to order by file size. There are 2 file size properties: *File Size* and *File Size Imported*. The first one is the size of the original asset, e.g. a PSD file for a texture. The second one is the size of the asset after Unity imported it to the project for the current target platform. This is the one we are interested in.

In order to avoid reading the whole report and marking the big textures we set the sort order to *Descending*. So the assets are sorted by their file size from big down to small and we only have to deal with the top of the list.



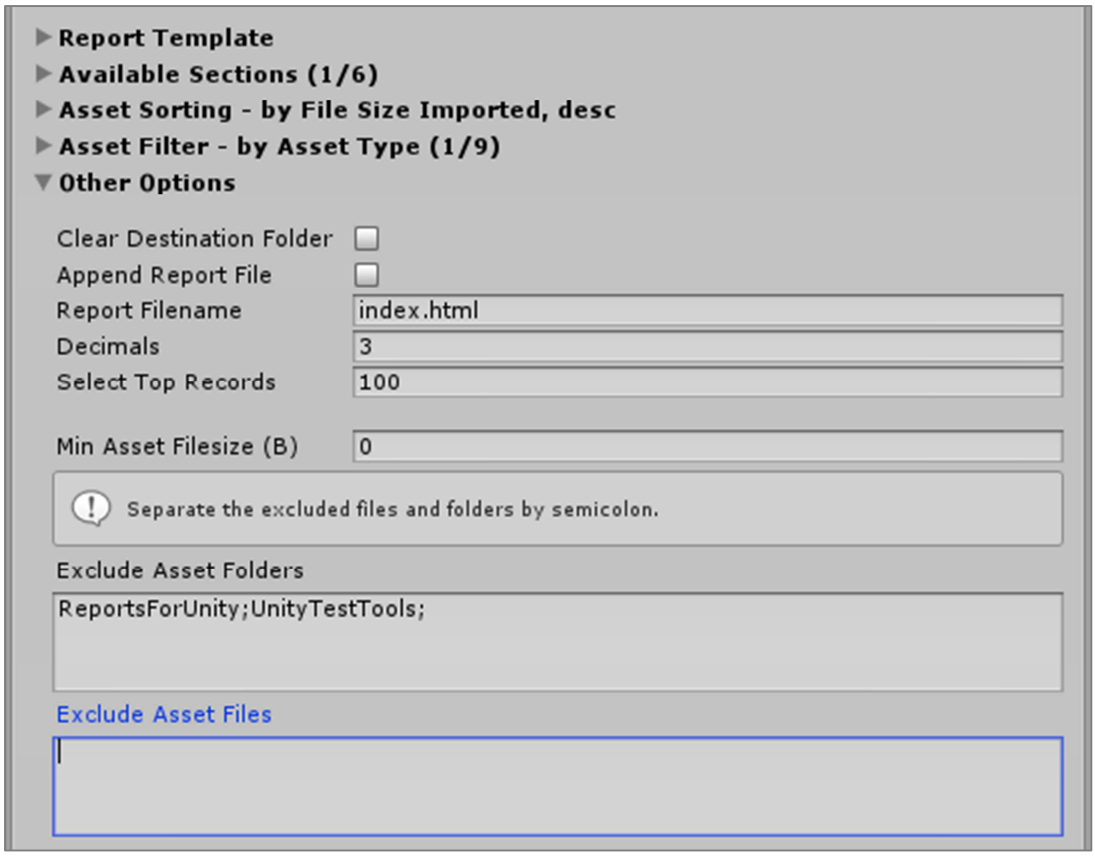
Filtering assets

As we are only interested in textures we need to exclude all the other asset types from the report. The foldout *Asset Filter* lists all asset types in our project so we can select the asset types to include in the report.

Other options to filter by are *File Extension* and *Asset Label*. So we could limit the reported assets to certain image file extension like PSD, PNG, TGA et cetera.

Other options

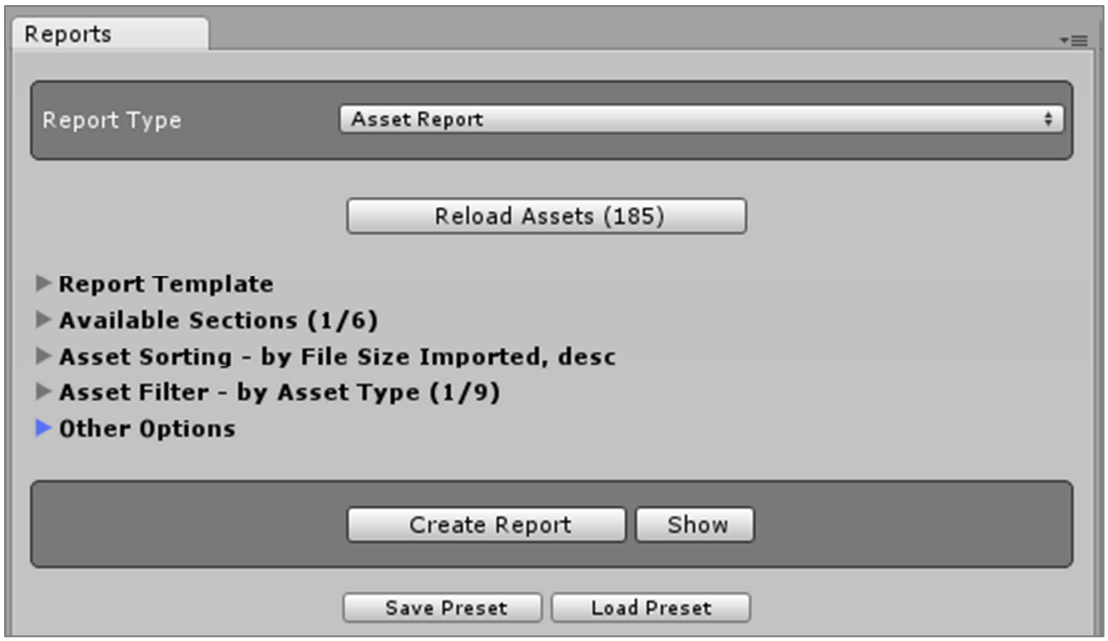
The last foldout in the window contains some other options you might find useful in order to customize your report.



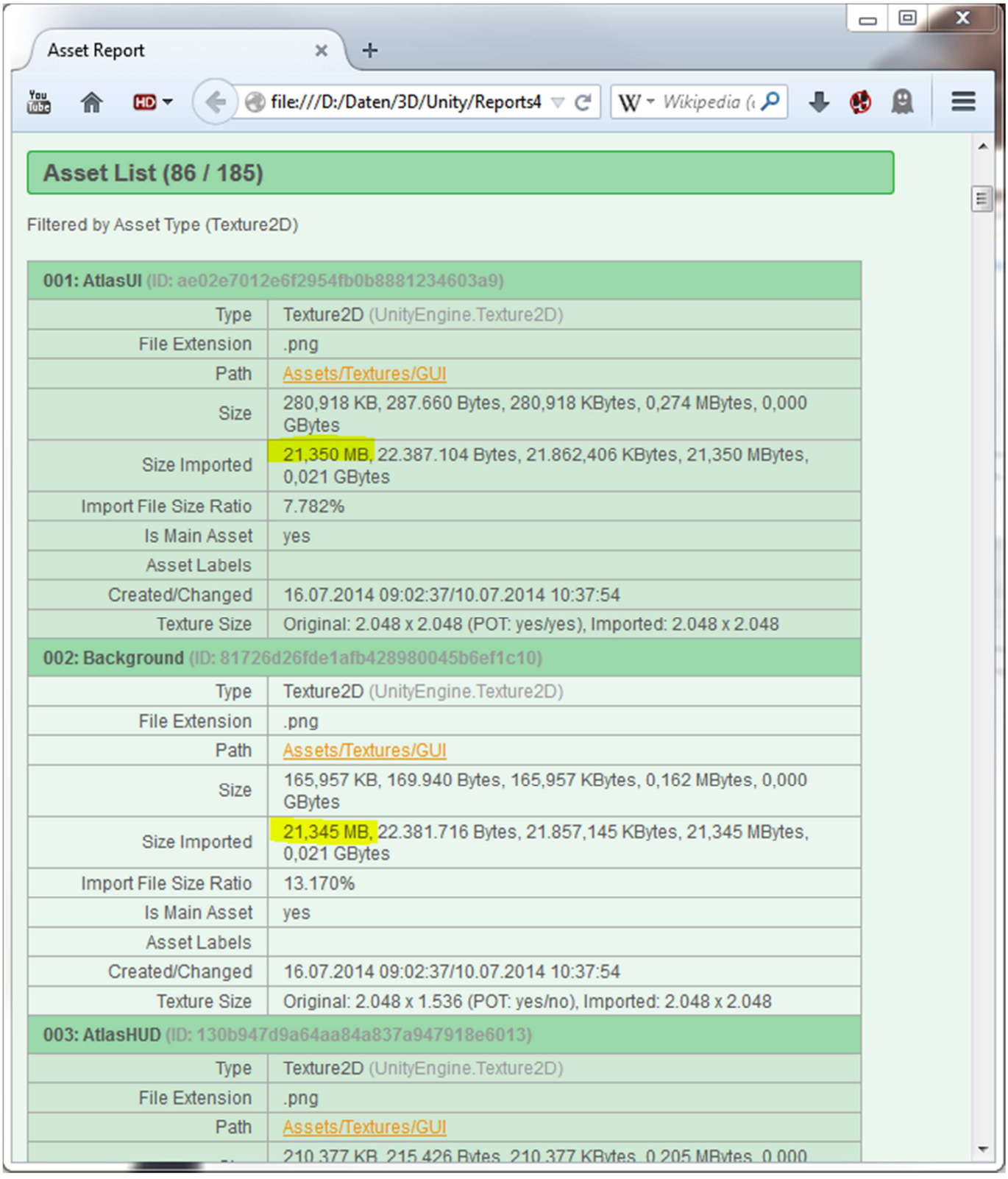
You can set the number of decimals, limit the list to assets with a minimum file size and exclude files and folders from processing. We leave these options at their default values for this example. But the value *Select Top Records* we set to 100. This limits the reported assets to the top 100. As we are only interested in the biggest textures of our project, this should be enough.

Creating the report

After setting all the options for our report we could save this set of options as a preset for reusing them in the future. But for now click the button *Create Report*. You have to select the destination folder on your computer where the report files are saved. After the report has been created and saved to disk you'll notice a new button next to the *Create Report* button labeled *Show*.



If you click this button the report system tries to open the file with the default application on your operating system. For our HTML file this will open your default browser.

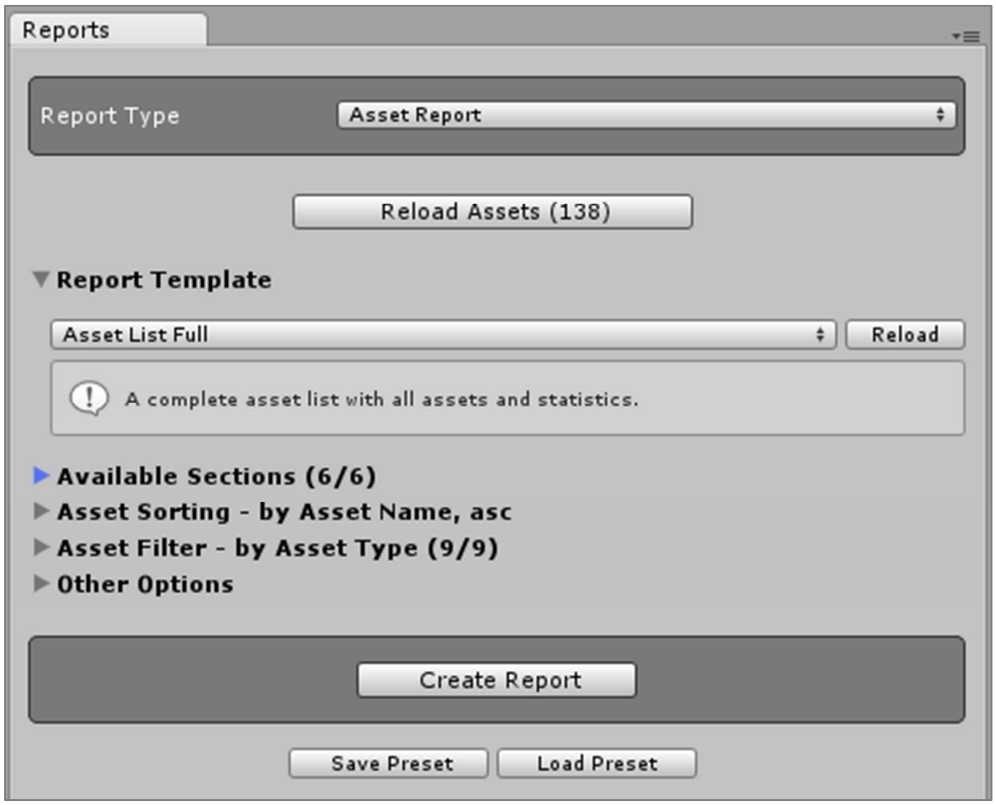


In this example you can see some assets with several data (defined in the templates). You see the texture file sizes of the top assets (21.350 MB and 21.345 MB, marked in this screen shot). You now can consider optimizing these textures for a smaller size. If you cannot get a better import size by changing the texture itself you maybe want to run an *Asset Importer Report* to tweak the importer settings for these textures.

Using the Asset Report

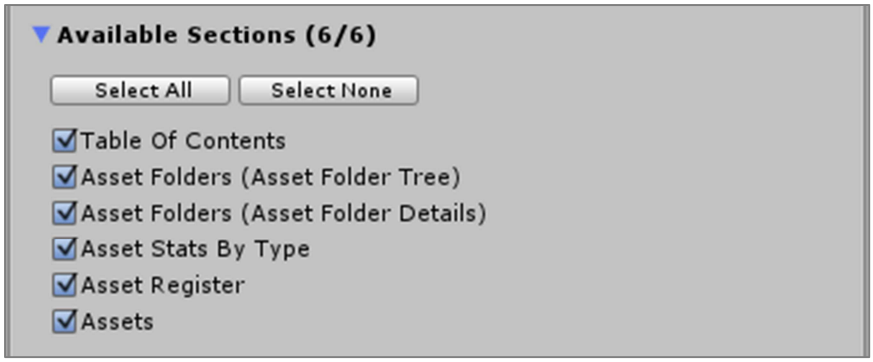
The *Asset Report* collects all assets and asset folders of your Unity project and a lot of information about the assets.

You can exclude files and folders from the report by defining some special tags in the preferences or the template.ini file. For more information about excluding files and folders please see the chapters about preferences and template customizing later in this manual.



Available Sections

The following sections are available for this report. If some of these sections are not defined in the current template file then they will not show up in this editor foldout. All selected sections will be used to create the report.



As you can see in the screenshot above a section may appear more than once in a template. It then holds the same data but with a different appearance and format (defined in the template). In the example above the *Asset Folders* section is displayed as a tree and a list with detailed information.

Option	Description
Table Of Contents	A list with links to all sections in the report.
Asset Folders	The list of asset folders in the project.
Asset Stats by Type	Statistics for each asset type like total file sizes and asset counts.
Asset Register	An alphabetical register with links to the first asset in the list that starts with this letter (depending on the current sort criteria).
Assets	The list of all assets that matching the filter criteria.

Asset Sorting

Set the sort behavior of the resulting asset list in the report.

Option	Description
Sort By	Select the asset field to sort by.
Direction	Select *Ascending* or *Descending* for the sort direction.

Asset Filter

If you only want to include a subset of assets in the report you can filter the list by

- Asset Type
- File Extension
- Asset Label

Just select the filter criteria in the dropdown and then toggle the checkboxes for the filter values you want to include in the report.

Other Options

These are some general options for the report generation defined in the system preferences or the current template. You can override them for the current report by editing the textbox content.

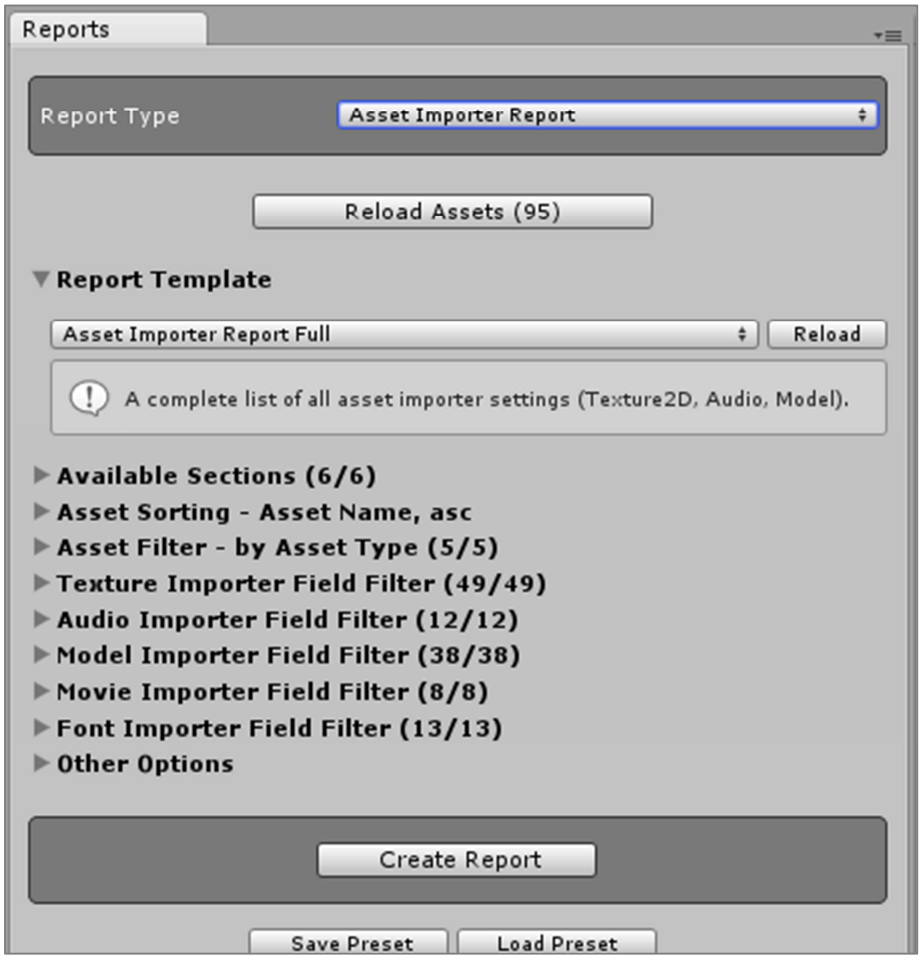
Option	Description
Clear Destination Folder	Set this option to delete all files from the folder you will select to save the report files. After deleting all the files and folders in the destination folder the report system copies all files and folders from the selected template source folder to this destination. So you are able to setup an HTML template with included images, CCS files and Javascript files.
Append Report File	Set this option to append the generated report data to an existing report file in the destination folder. This can be used to append data to a CSV file to open and edit in EXCEL.
Report Filename	This is the file name of the report file defined in the template.
Decimals	The number of decimals for numeric values as defined in the preferences or the template.
Select Top Records	If you want to display just the first assets, just set the amount of records you want to generate. For example: if you want to identify your TOP 20 assets in

	file size, you don't need to create the complete asset report. If you enter a value of 0, all assets are reported.
Min Asset File Size (B)	Another way to exclude small assets from the report is to set a minimal asset file size. Enter a value in Bytes the file size has to be to be included in the report. If you enter a value of 0, all assets are reported.
Exclude Asset Folders	To exclude some folders and their content from the report enter a list of folder names, separated by a semicolon.
Exclude Asset Files	To exclude some asset files from the report enter a list of file names, separated by a semicolon.

Using the Asset Importer Report

The *Asset Importer Report* collects all asset importer settings of your Unity project. These settings appear in the Unity inspector if you select an asset in the project view. Comparing and checking them one by one would be very time consuming and error prone. On the other hand these settings have a big impact on the performance of your game, so you should be sure they are correct. Listing certain properties of the import settings with the Asset Importer Report will help you to spot some areas to optimize.

You can exclude files and folders from the report by defining some special tags in the preferences or the template.ini file. For more information about excluding files and folders please see the chapters about preferences and template customizing later in this manual.



Available Sections

The following sections are available for this report. If some of these sections are not defined in the current template file then they will not show up in this editor foldout. All selected sections will be used to create the report.

Option	Description
Table Of Contents	A list with links to all sections in the report.
Texture Import Settings	The list of import settings for textures.
Audio Import Settings	The list of import settings for audio files.
Model Import Settings	The list of import settings for models.
Movie Import Settings	The list of import settings for movies.
Font Import Settings	The list of import settings for fonts.

Asset Sorting

Set the sort behavior of the resulting asset list in the report.

Option	Description
Sort By	Select the asset field to sort by.
Direction	Select *Ascending* or *Descending* for the sort direction.

Asset Filter

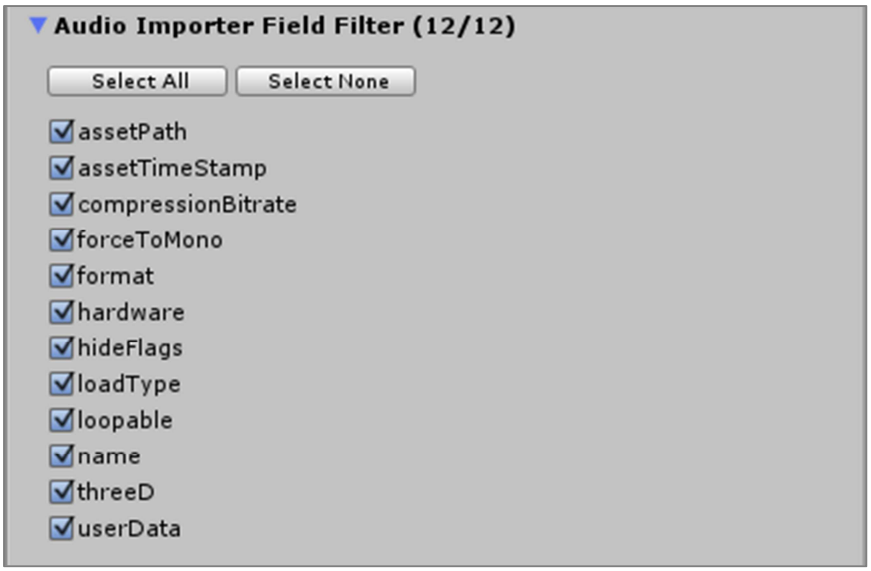
If you only want to include a subset of assets in the report you can filter the list by

- Asset Type
- File Extension
- Asset Label

Just select the filter criteria in the dropdown and then toggle the checkboxes for the filter values you want to include in the report.

Importer Field Settings

There are 5 foldouts with a list of the public fields of each importer type (texture, audio, model, movie and font). These fields are extracted from the Unity importer class using reflection. In the example below you can see the fields of the audio importer class.



By selecting one or more of these fields they can be printed in the resulting report if there is a corresponding placeholder. You will find more details about placeholders in the Customizing Templates chapter later in this manual.

Other Options

These are some general options for the report generation defined in the system preferences or the current template. You can override them for the current report by editing the textbox content.

Option	Description
Clear Destination Folder	Set this option to delete all files from the folder you will select to save the report files. After deleting all the files and folders in the destination folder the report system copies all files and folders from the selected template source folder to this destination. So you are able to setup an HTML template with included images, CCS files and Javascript files.
Append Report File	Set this option to append the generated report data to an existing report file in the destination folder. This can be used to append data to a CSV file to open and edit in EXCEL.
Report Filename	This is the file name of the report file defined in the template.
Decimals	The number of decimals for numeric values as defined in the preferences or the template.
Select Top Records	If you want to display just the first assets, just set the amount of records you want to generate.
Min Asset File Size (B)	Another way to exclude small assets from the report is to set a minimal asset file size. Enter a value in Bytes the file size has to be to be included in the report. If you enter a value of 0, all assets are reported.
Exclude Asset Folders	To exclude some folders and their content from the report enter a list of folder names, separated by a semicolon.
Exclude Asset Files	To exclude some asset files from the report enter a list of file names, separated by a semicolon.

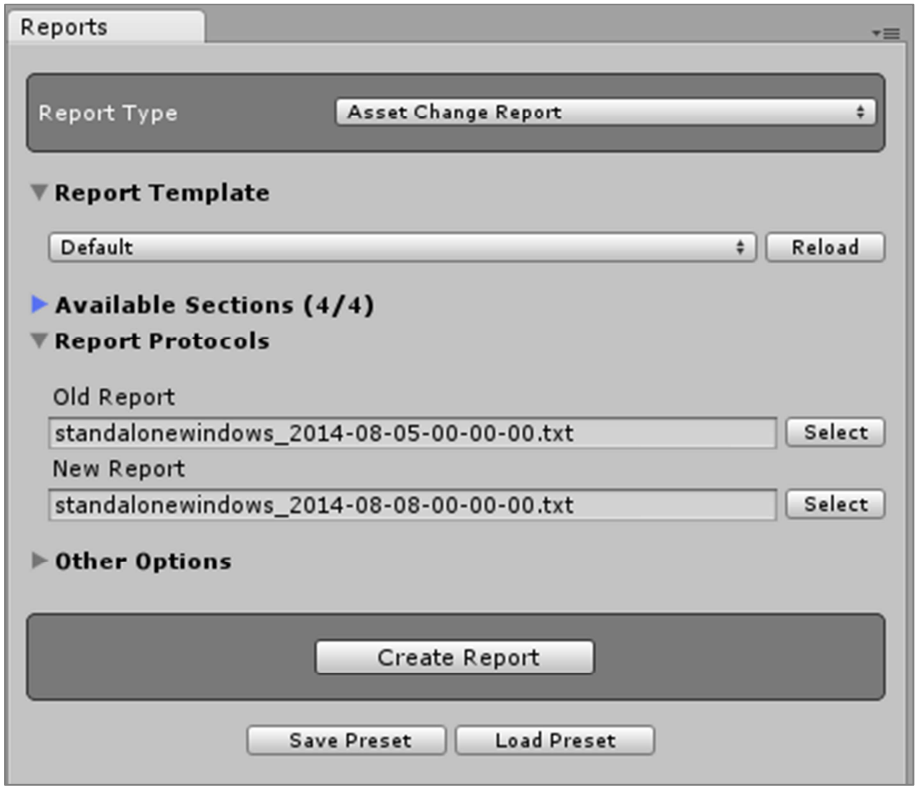
Using the Asset Change Report

The *Asset Change Report* can be used to compare one project state to another regarding added, changed and deleted assets. With this report it is very easy to e.g. concentrate quality assurance efforts only on new and changed assets. It also helps you noticing when assets were accidently deleted by yourself or a team mate.

Every time you create an *Asset Report* the system will save an asset protocol, a snapshot of the current assets in the project with its file sizes (by platform). To generate an *Asset Change Report* you need to select 2 snapshots to compare to each other. By default the last two snapshots are selected, but you can change this by clicking the *Select* button next to the file name textbox.

You can change the way of asset protocol saving in the preferences window. Please see the chapter about preferences for more details.

You can exclude files and folders from the report by defining some special tags in the preferences or the template.ini file. For more information about excluding files and folders please see the chapters about preferences and template customizing later in this manual.



Available Sections

The following sections are available for this report. If some of these sections are not defined in the current template file then they will not show up in this editor foldout. All selected sections will be used to create the report.

Option	Description
Table Of Contents	A list with links to all sections in the report.
New Assets	The list of all new assets in the project.

Missing Assets	The list of all missing assets.
Changed Assets	The list of all changed assets in the project (changed file size).

Report Protocols

Select two protocol files to compare. By default the last two snapshots are selected.

Other Options

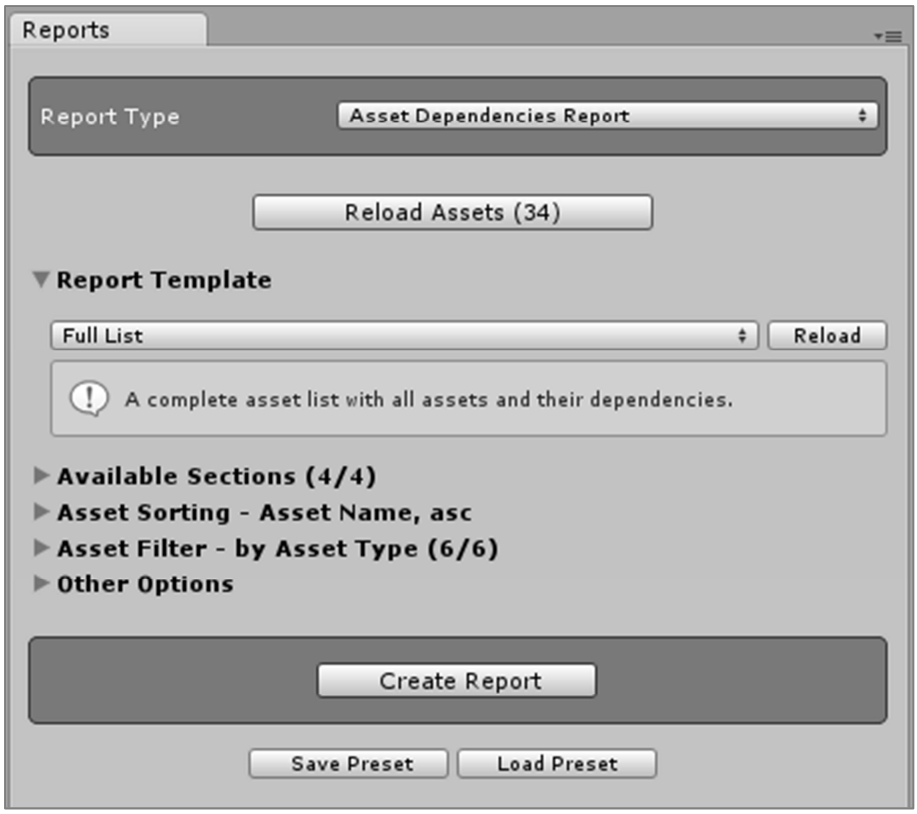
These are some general options for the report generation defined in the system preferences or the current template. You can override them for the current report by editing the textbox content.

Option	Description
Clear Destination Folder	Set this option to delete all files from the folder you will select to save the report files. After deleting all the files and folders in the destination folder the report system copies all files and folders from the selected template source folder to this destination. So you are able to setup an HTML template with included images, CCS files and Javascript files.
Append Report File	Set this option to append the generated report data to an existing report file in the destination folder. This can be used to append data to a CSV file to open and edit in EXCEL.
Report Filename	This is the file name of the report file defined in the template.
Decimals	The number of decimals for numeric values as defined in the preferences or the template.
Select Top Records	If you want to display just the first assets, just set the amount of records you want to generate.

Using the Asset Dependencies Report

The *Asset Dependencies Report* collects the dependencies and reverse dependencies of all assets of your Unity project. The main reason for doing this is to find missing dependencies. But it can also help you finding some optimizing potential in your assets. For example: you quickly find materials or textures that are used only by one other asset and you may replace it by another material that is already used. Sharing textures and materials helps you increase the performance of your game.

You can exclude files and folders from the report by defining some special tags in the preferences or the template.ini file. For more information about excluding files and folders please see the chapters about preferences and template customizing later in this manual.



Available Sections

The following sections are available for this report. If some of these sections are not defined in the current template file then they will not show up in this editor foldout. All selected sections will be used to create the report.

Option	Description
Table Of Contents	A list with links to all sections in the report.
Assets With Dependencies	A list of all assets that have dependencies.
Assets With Missing Dependencies	A list of all assets that miss one or more dependencies.
Assets Reverse Dependencies	A list of all assets that other assets are depending on.

Asset Sorting

Set the sort behavior of the resulting asset list in the report.

Option	Description
Sort By	Select the asset field to sort by.
Direction	Select *Ascending* or *Descending* for the sort direction.

Asset Filter

If you only want to include a subset of assets in the report you can filter the list by

- Asset Type

- File Extension
- Asset Label

Just select the filter criteria in the dropdown and then toggle the checkboxes for the filter values you want to include in the report.

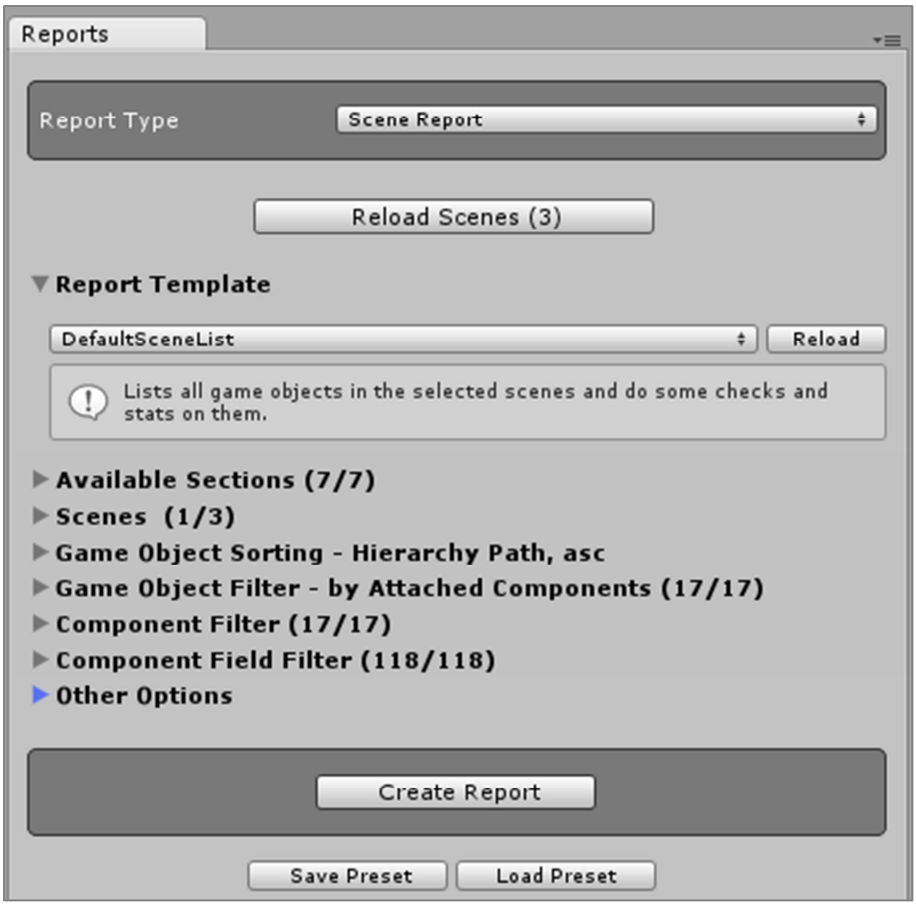
Other Options

These are some general options for the report generation defined in the system preferences or the current template. You can override them for the current report by editing the textbox content.

Option	Description
Display Short List	When creating this report all asset dependencies are listed with the main asset and its sub assets. By checking this option only the main assets are listed. If the asset is a sprite sheet for example, then with this option checked only the sprite sheet file itself will be listed, not the single sprites it contains.
Clear Destination Folder	Set this option to delete all files from the folder you will select to save the report files. After deleting all the files and folders in the destination folder the report system copies all files and folders from the selected template source folder to this destination. So you are able to setup an HTML template with included images, CCS files and Javascript files.
Append Report File	Set this option to append the generated report data to an existing report file in the destination folder. This can be used to append data to a CSV file to open and edit in EXCEL.
Report Filename	This is the file name of the report file defined in the template.
Decimals	The number of decimals for numeric values as defined in the preferences or the template.
Select Top Records	If you want to display just the first assets, just set the amount of records you want to generate.
Min Asset File Size (B)	Another way to exclude small assets from the report is to set a minimal asset file size. Enter a value in Bytes the file size has to be to be included in the report. If you enter a value of 0, all assets are reported.
Exclude Asset Folders	To exclude some folders and their content from the report enter a list of folder names, separated by a semicolon.
Exclude Asset Files	To exclude some asset files from the report enter a list of file names, separated by a semicolon.

Using the Scene Report

The *Scene Report* collects the all information about the game objects in your scenes. It prints all game objects and their attached components with the public fields.



Available Sections

The following sections are available for this report. If some of these sections are not defined in the current template file then they will not show up in this editor foldout. All selected sections will be used to create the report.

Option	Description
Table Of Contents	A list with links to all sections in the report.
Missing Scripts	A list of all game objects with missing scripts.
Missing Prefabs	A list of all game objects with missing prefabs.
Missing Meshes	A list of all game objects with missing meshes.
Scenes	A list of all selected scenes and their game objects.
Game Object Register	An alphabetical register with links to the first game object in the list that starts with this letter (depending on the current sort criteria).
Game Objects	A list of all game objects in all selected scenes.

Scenes

This foldout lists all scenes available in the project. The currently loaded scene in the editor is selected by default. You can select all the scenes you want to be processed. There are 4 buttons helping you selecting scenes:

- *Select All*: Selects all scenes.
- *Select None*: Removes the selection from all scene checkboxes.

- *Scenes in Build*: Selects all scenes you have currently in your build settings.
- *Current*: Selects the scene currently loaded in the editor.

Game Object Sorting

Set the sort behavior of the resulting asset list in the report.

Option	Description
Sort By	Select the game object field to sort by.
Direction	Select *Ascending* or *Descending* for the sort direction.

Game Object Filter

If you only want to include a subset of game objects in the report you can filter the list by

- Attached components
- Layer name
- Sorting layer name
- Tag

Just select the filter criteria in the dropdown and then toggle the checkboxes for the filter values you want to include in the report.

For example: if you only want to see game objects with a light on it, select the filter criteria *Attached Components* and then check *Light* as the only component. You can now check the settings for all lights in your scenes without selecting every game object in the hierarchy view of Unity. The list of game objects is now limited to game objects with a light component attached. But the list of components displayed for each of this game objects is still complete. To limit this you can use the component filter in the next foldout.

Component Filter

While you can filter the reported list of game objects with the game object filter, you limit the displayed components for each game object by selecting them in the component filter. So in this list the game object not having the selected components will still be displayed but without components.

Component Field Filter

With this filter you can select the fields displayed for each component in the report. As this filter collects all fields of all the different components of your game objects by reflection, this list can be quite long. This filter is a good way to check the correct state of some special fields on components at a glance, e.g. the *isTrigger* property of all colliders in a scene. Or you list all game objects in a scene displaying the *isVisible* field to check if you accidently made a game object invisible in the inspector.

Other Options

These are some general options for the report generation defined in the system preferences or the current template. You can override them for the current report by editing the textbox content.

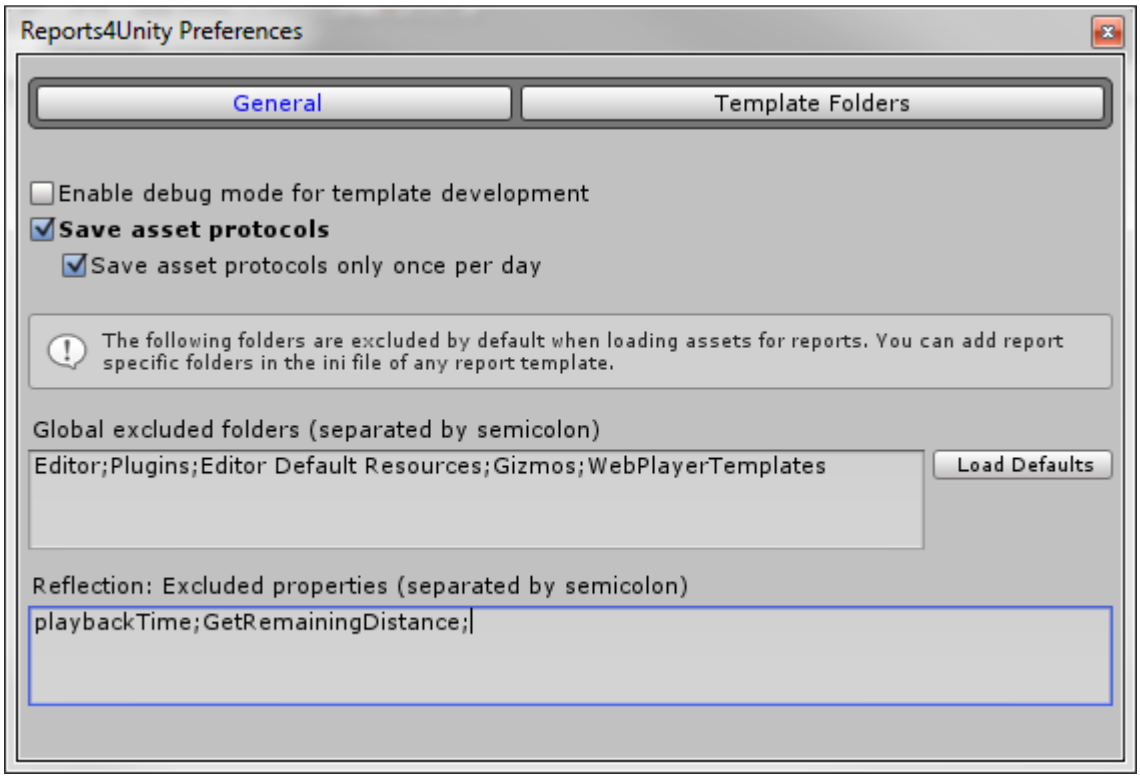
Option	Description
Display Short List	When creating this report all asset dependencies are listed with the main asset and its sub assets. By checking this option only the main assets are listed. If the asset is a sprite sheet for example, then with this option

	checked only the sprite sheet file itself will be listed, not the single sprites it contains.
Clear Destination Folder	Set this option to delete all files from the folder you will select to save the report files. After deleting all the files and folders in the destination folder the report system copies all files and folders from the selected template source folder to this destination. So you are able to setup an HTML template with included images, CCS files and Javascript files.
Append Report File	Set this option to append the generated report data to an existing report file in the destination folder. This can be used to append data to a CSV file to open and edit in EXCEL.
Report Filename	This is the file name of the report file defined in the template.
Decimals	The number of decimals for numeric values as defined in the preferences or the template.
Select Top Records	If you want to display just the first assets, just set the amount of records you want to generate.

Preferences

By opening the preferences window (*Window/Reports4Unity/Asset Report*) you can set some global options for the report system.

General



Enabled debug mode for template development

When the debug mode is enabled, the report system writes some information and warnings to the *Unity* console window. This helps you to identify and fix invalid tag definitions or other mistakes in your templates. You can disable the debug mode to keep the console output clean if you are not actively working on a template.

Save asset protocols

In order to report asset changes with the *Asset Change Report* the system writes some protocol files with the current state of your assets at a particular time. This folder is where they are stored. You can choose to write an asset protocol every time you create an *Asset Report* or only once per day, if you create more than one *Asset Report* a day. If you select the option "*Save asset protocols only once per day"* the system updates the recent protocol file of the day, if there is one.

Global excluded folders

When scanning your asset folder for files to create an *Asset Report* you may want to exclude some folders from this process. For example, if you have some third party asset packs in your project, especially editor extensions that are not part of your game, you can rule them out with this option. By default the *Unity* special folders will not be parsed by the report system.

Reflection: Excluded properties

For the *Scene Report* reflection is used to get all public fields and properties of the components in a scene. There are cases where the particular classes perform some plausibility checks in the *getValue*-method of the property and throw custom exceptions, if the conditions are not true. In this case *Report4Unity* cannot report this property. To generate the scene report anyway, please check the thrown exception and add the property name to the list of excluded properties. They will be ignored in the future.

Example exception:

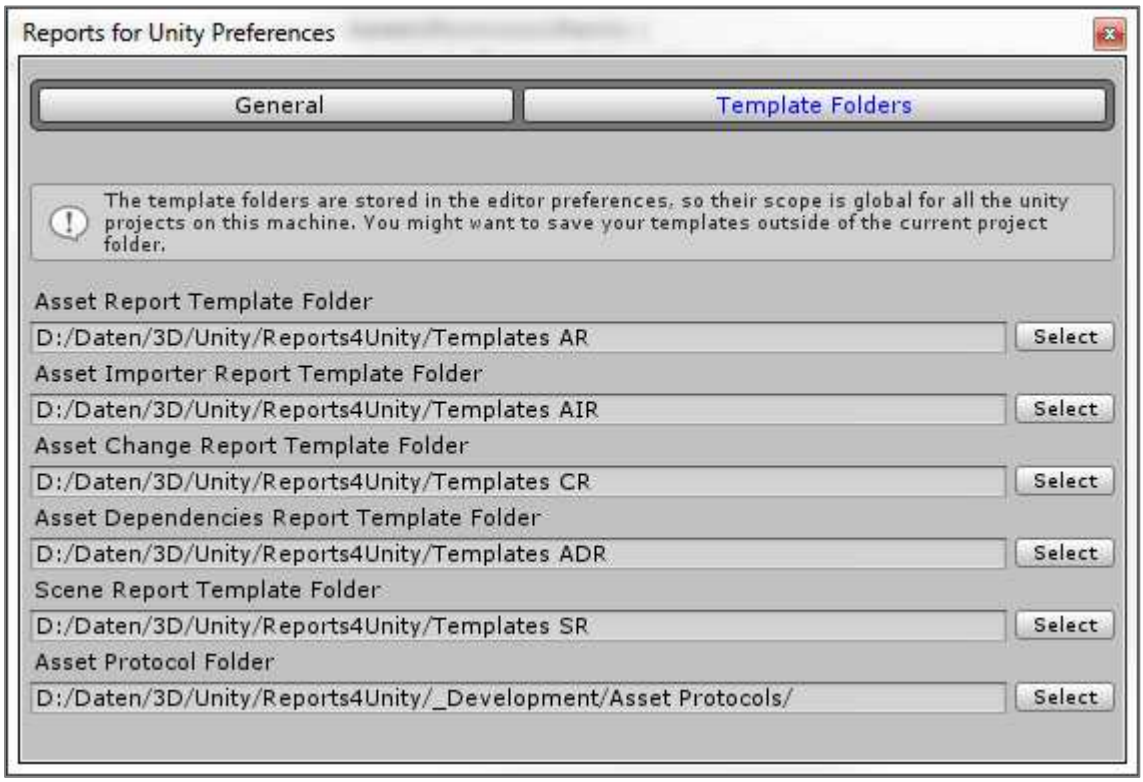
```
"GetRemainingDistance" can only be called on an active agent that has been
placed on a NavMesh.
System.Reflection.MonoProperty:GetValue(Object, Object[])
VirtualPlaygrounds.Reports4Unity.ReflectionHelper:GetPrimitiveFields(Object
)
VirtualPlaygrounds.Reports4Unity.ComponentItem:GetComponentFields(Component
, ComponentItem)
VirtualPlaygrounds.Reports4Unity.ComponentItem:GetComponentItems(GameObject
, GameObjectItem)
VirtualPlaygrounds.Reports4Unity.SceneManager:LoadGameObjectsFromScene(Scen
eItem)
VirtualPlaygrounds.Reports4Unity.SceneManager:LoadGameObjectItems()
VirtualPlaygrounds.Reports4Unity.SceneManager:Refresh(Boolean)
VirtualPlaygrounds.Reports4Unity.ReportWindow:ButtonSaveReportClick()
VirtualPlaygrounds.Reports4Unity.ReportWindow:BuildGuiCreateReportButton()
VirtualPlaygrounds.Reports4Unity.ReportWindow:BuildGuiReportTypeScenes()
VirtualPlaygrounds.Reports4Unity.ReportWindow:OnGUI()
UnityEditor.DockArea:OnGUI()
```

Template Folders

Reports4Unity uses a special folder setup for its templates. A template folder has to contain at least to files:

- The template itself (HTML or plain text file)
- A template ini-file called template.ini

In order to display the template selection dropdown in the report window, the report system searches for those ini-files containing the report type of the template, a description of the template and a few other properties. The name of the folder containing these two files is the name of the template displayed in the dropdown.



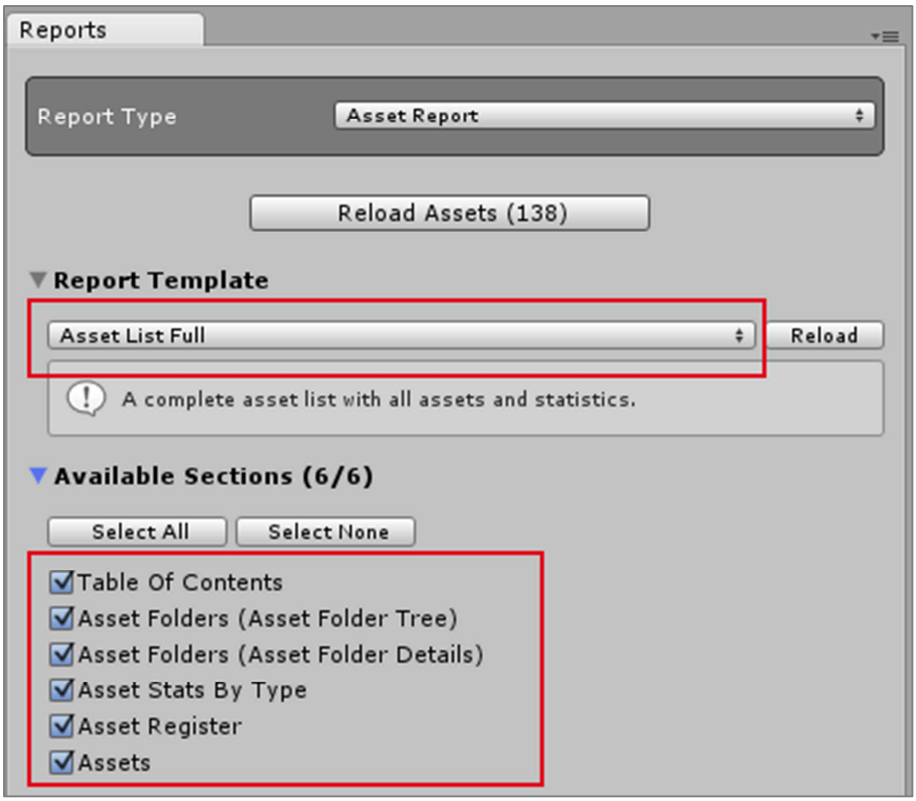
For each report type the folder with template definitions can be setup here. Of course you can store all of your different report templates in the same folder (with subfolder for every template). The template type will be pulled from the particular ini-file. You should use a location outside of your asset folder. Otherwise you have to exclude the template folder from the asset scan process as mentioned in the previous chapter.

Asset Protocol Folder

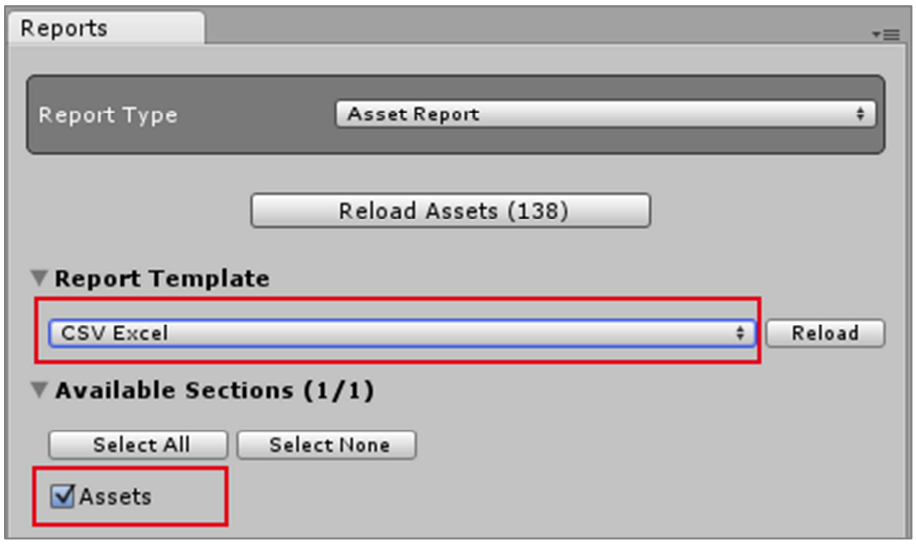
The asset protocol folder is where all the asset protocols are stored. The *Asset Change Report* gets the latest protocols from there to compare.

Customizing Templates

The *Reports4Unity* system is designed to format the reported data in a very flexible way. To achieve this it uses templates (HTML or plain text files) with placeholders in them to hold the generated data. To get the reports in a format that suits your needs you can change the included templates or create your own additional templates in minutes. The particular report will generate all data for the available sections (report data sections) every time. Your template controls, which of these sections are processed and how the data will be displayed. For example, if you do not need information about the asset folders in the Asset Report but only the asset data, just omit the asset folder section in the template. The following screenshots show the Asset Report with all available section (6) in the template “Asset List Full” and another template with just one section (“CSV Excel”).



Asset Report with full template (6 sections)



Asset Report with tailored template (1 section)

Download Free Templates

Reports4Unity comes with some templates as a starting point for you, when you download it from the Asset Store. You can develop your own templates as described in the following chapters, but you there are more templates to download from our website www.unity-playgrounds.com. If there are templates you are interested in, just download them and copy the unzipped files to the templates folder on your machine.

If you developed templates you want to share with other Unity developers, you can upload them to this website and we will check them and make them available for download.

The Template Structure

A template contains placeholders that will be filled with data during the report creation. There are essentially two types of placeholders.

Slots

Slots will be replaced by data only once. A slot can exist multiple times in a template but it will always be filled with the same data.

Slots are enclosed by “[--“ and “--]”.

Example: [--ProjectName--]

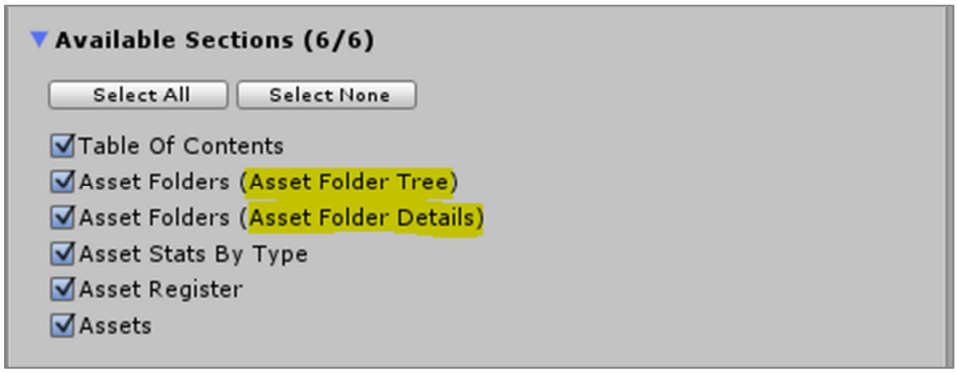
Sections

Sections define a set of content and slots that will be repeated for every record of data, for example every asset in the particular report.

They are enclosed by “[++“ and “++]” and contain the section name and a reference to its block definition. All sections are displayed in the report window for you to select. There are predefined sections for each report type you can use in your templates. You will find a complete list in the reference chapters later in this manual.

Example: [++Assets Block=AssetList ++]

You may use sections multiple times in your templates with different block definitions to display the same data in a different format. In the image below the section Asset Folders occurs 2 times with different block definitions (*Asset Folder Tree* and *Asset Folder Details*). At first the folder hierarchy is reported as a tree, the second sections prints details about each folder.



Example Template

The example below contains a global slot (*ProjectName*) and two sections (*AssetFolders* and *Assets*). Let’s have a closer look at it.


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Asset Report</title>
  </head>
  <body style="font-family: Helvetica, Arial, sans-serif;font-size: 12px;">

  <h1>[--ProjectName--]</h1>

  <h2>Asset Folders</h2>
  <table cellpadding="2" cellspacing="1" style="width:50%;">
  [++AssetFolders Block=AssetFolderDetails++]
  </table>

  <h2>Asset List ([--AssetCountFiltered--] / [--AssetCount--])</h2>
  <table cellpadding="2" cellspacing="1" style="width:50%;">
  [++Assets Block=AssetList++]
  </table>

  </body>
</html>

[--EndOfContent--]
<!--

[--BlockStart AssetFolderDetails--]
  <tr>
    <td><strong>[--FolderRelativePath--]</strong></td>
  </tr>
[--BlockEnd--]

[--BlockStart AssetList--]
  <tr>
    <td colspan="2" style="font-weight:bold;background-color:#eeeeee;">
      [--IndexFormatted--]: [--AssetName--]
    </td>
  </tr>
  <tr>
    <td>Type</td>
    <td>[--AssetTypeShort--]</td>
  </tr>
  <tr>
    <td>Size</td>
    <td>
      Original: [--AssetFileSizeAuto--],
      Imported: [--AssetFileSizeImportedAuto--]
    </td>
  </tr>
[--BlockEnd--]

-->
```

The slot (line 8) will be replaced by the name of the current project while parsing the template and generating the report.

The section *Assets* (line 17) is the place where all the assets are listed. It refers to the block definition *AssetList* at the bottom of the template (line 32). For every asset in the report the block will be filled with data and the result will be appended to the list. In this case the block *AssetList* defines 3 HTML table rows with several slots for the asset data in it (line 33 to 48). These table rows will be appended to the HTML table (line 16) that encloses the section (line 17). So for every asset the following data is reported in this example:

- [--IndexFormatted--] The index of the asset in the list, starting at 1.
- [--AssetName--] The name of the asset.
- [--AssetTypeShort--] The asset type, e.g. *Texture2D*, *Material* or *AudioClip*
- [--AssetFileSizeAuto--] The original file size of the asset. The unit (Bytes, Kilobytes etc.) is assigned automatically.
- [--AssetFileSizeImportedAuto--] The file size of the asset after it is imported by Unity.

For a complete list of available data slots please read the reference chapters later in this manual.

The result of this asset report looks like this (only a part of the asset list):

006: Animal4	
Type	Texture2D
Size	Original: 24.84 KB, Imported: 274.81 KB
007: Animal5	
Type	Texture2D
Size	Original: 20.53 KB, Imported: 273.32 KB
008: ArrowLeft	
Type	Material
Size	Original: 4.14 KB, Imported: 7.41 KB
009: ArrowLeft	
Type	Texture2D
Size	Original: 22.57 KB, Imported: 274.54 KB

There is a special slot in the template at line 23 named *EndOfContent*. This defines the end of the content part of the template. All lines below this slot will not be printed to the resulting report. This is the place where you want to put your block definitions.

Note: If you changed the current template file while it is selected, you can reload the template by clicking the button *Reload* next to the template selection dropdown. Otherwise the changes will not be reflected in the report you create.

Advanced Template Customizing

With the information above you are able to create and customize templates for an unlimited number of output formats. However, there are some more template features giving you more control over the report output.

Block Visibility

If you disable the data generation for a section in the report window, you don't want to appear any of the section related data in the report at all. In the following example 3 of the 7 sections are disabled.

To avoid the appearance of any related data or HTML code in the report, you can use the *BlockVisible* tag. This prints the string yes or no depending on the visibility of the section. You can use this behavior for defining CSS styles in your template as in the code snippet below.

```
27
28        .sectionVisible_yes { display: block; }
29        .sectionVisible_no { display: none; }
30
```

These styles are used in the class tag of a div surrounding the whole data section (line 118). If the section is disabled in the report window, the tag *[++BlockVisible Block=MissingScripts++]* is replaced by "*no*", so the CSS style *sectionVisible_no* is applied to the div.

```
117
118   <div class="sectionVisible_[++BlockVisible Block=MissingScripts++]">
119     <h2>Game Objects Missing Scripts</h2>
120     <table cellpadding="2" cellspacing="1" style="width:500px;">
121     [++MissingScripts Block=MissingScripts++]
122     </table>
123   </div>
124
```

To hide parts of your template you can also use conditional statements with the *BlockVisible* tag like this:

```
85
86    [==IF [++BlockVisible Block=TableOfContents++] = yes ==]
87
88      <h2>Table of Contents</h2>
89      [++TableOfContents Block=TableOfContents++]
90      <br /><br />
91
92    [==ENDIF==]
93
```

Only if the *BlockVisible* tag is equal to “yes” (line 86), the content between line 86 and line 92 will be printed. For more information see the chapter about conditional statements. Note that if you use conditional statements for hiding disabled sections, you cannot use other conditional statements inside this section because they cannot be nested.

NoData Blocks

If a section of your reports has no records to show you can define a special block definition that will be printed in this case instead of the block definition assigned to this section. Please have a look at the following example:

```
117
118   <div class="sectionVisible_[++BlockVisible Block=MissingScripts++]">
119     <h2>Game Objects Missing Scripts</h2>
120     <table cellpadding="2" cellspacing="1" style="width:500px;">
121     [++MissingScripts Block=MissingScripts++]
122     </table>
123   </div>
124
```

This code snippet defines a section for missing scripts in a *Scene Report* (line 121), surrounded by some HTML. If there are game objects with missing scripts, they will be listed as rows in a HTML table (line 120). The rows are defined with the block definition *MissingScripts* (line 197) as shown in the image below.

```
196
197 [--BlockStart MissingScripts--]
198   <tr>
199     <td class="index_[--OddEven--]">[--NameWithHierarchyPathFull--]</td>
200   </tr>
201 [--BlockEnd--]
202 [--BlockStart MissingScripts_NoData--]
203   <tr>
204     <td class="index_even">No missing scripts.</td>
205   </tr>
206 [--BlockEnd--]
207
```

If there are no records (game objects with missing scripts) the report will show a headline (line 119 in the first snippet) and a HTML table without rows, which is invalid HTML code. But if you define a special *NoData*-block somewhere in your template (line 202), this block definition is used instead of the original block definition. This special block definition has to have the same name as the block definition it substitutes, but with *_NoData* at the end of its name (line 202).

Conditional Statements

If you want to show or hide parts of your template you can use conditional statements as shown in the following example.

```
 99   <tr>
100     <td class="index_odd">Game Object Count Total:</td>
101     <td class="index_odd" style="width:150px;text-align:right;">
102
103       [==IF [--GameObjectCount--] GT 50 ==]
104         <br /><span style="color:red;">[--GameObjectCount--]</span>
105       [==Endif==]
106       [==IF [--GameObjectCount--] LT 51 ==]
107         <br /><span style="color:green;">[--GameObjectCount--]</span>
108       [==Endif==]
109
110   </td>
111   </tr>
```

This template defines 2 blocks enclosed by a *[==IF==]* and *[==ENDIF==]* tag. The first *IF* condition tests if the *GameObjectCount* slot contains a value greater than 50 (GT 50). If this is true, the line 104 is printed in the report and the game object count is colored red. The second statement checks for a *GameObjectCount* less than 51 (LT 519) and prints line 107 if the condition is true.

A conditional statement has the following structure with spaces between the parts of the term:

`[==IF <Operand1> <Operator> <Operand2> ==]`

Note:

- **IF and ENDIF statements have to be on a separate line!**
- **IF and ENDIF statements cannot be nested!**

Operator can be one of the following

Operator	Description
=	True, if operator 1 is equal to operator 2. Use for strings and numeric values.
!=	True, if operator 1 is not equal to operator 2. Use for strings and numeric values.
LT	True, if operator 1 is less than operator 2. Only use for numeric values.
GT	True, if operator 1 is greater than operator 2. Only use for numeric values.

Operand 1 and 2 can be hard coded values or the output of a slot after it has been processed.

Content Substitutions

You can substitute any string in the processed template by defining a substitution rule like this:

```
183
184   [**Vertex Count: Block=SubstVertexCount**]
185
186   [--BlockStart SubstVertexCount--]
187   <u>Vertex Count:</u>
188   [--BlockEnd--]
189
```

Use a tag with the format

```
[**<Original Content> Block=<Blockdefinition>**]
```

In your template after the [--EndOfContent--] slot to define the original content to be replaced and the block definition holding the substitution for this content. It will replace any occurrence of this content in the processed template.

In the example above the string “Vertex Count:” will be replaced by “<u>Vertex Count:</u>”, so the string is underlined.

Mathematical Operations

You can use simple mathematical operations in your templates. The following example shows some of them with slots and constant values (line 292-296).

```
285   <tr>
286     <td  class="index_[--OddEven--]" style="width:150px;text-align:right;padding
287     <td  class="index_[--OddEven--]" style="padding-left:10px;">[--AssetFileSize
288   </tr>
289   <tr>
290     <td  class="index_[--OddEven--]" style="width:150px;text-align:right;padding
291     <td  class="index_[--OddEven--]" style="padding-left:10px;">
292       [##ADD part1=[--AssetFileSize--] part2=[--AssetFileSize--]##]<br />
293       [##SUB part1=[--AssetFileSize--] part2=[--AssetFileSizeKB--]##]<br />
294       [##MUL part1=[--AssetFileSize--] part2=[--AssetFileSize--]##]<br />
295       [##MUL part1=[--AssetFileSize--] part2=10##]<br />
296       [##DIV part1=[--AssetFileSize--] part2=[--AssetFileSize--]##]
297     </td>
298   </tr>
```

The tags have the following structure with spaces between the parts of the term:

```
[##<Operation> part1=<Operand1> part2=<Operand2> ##]
```

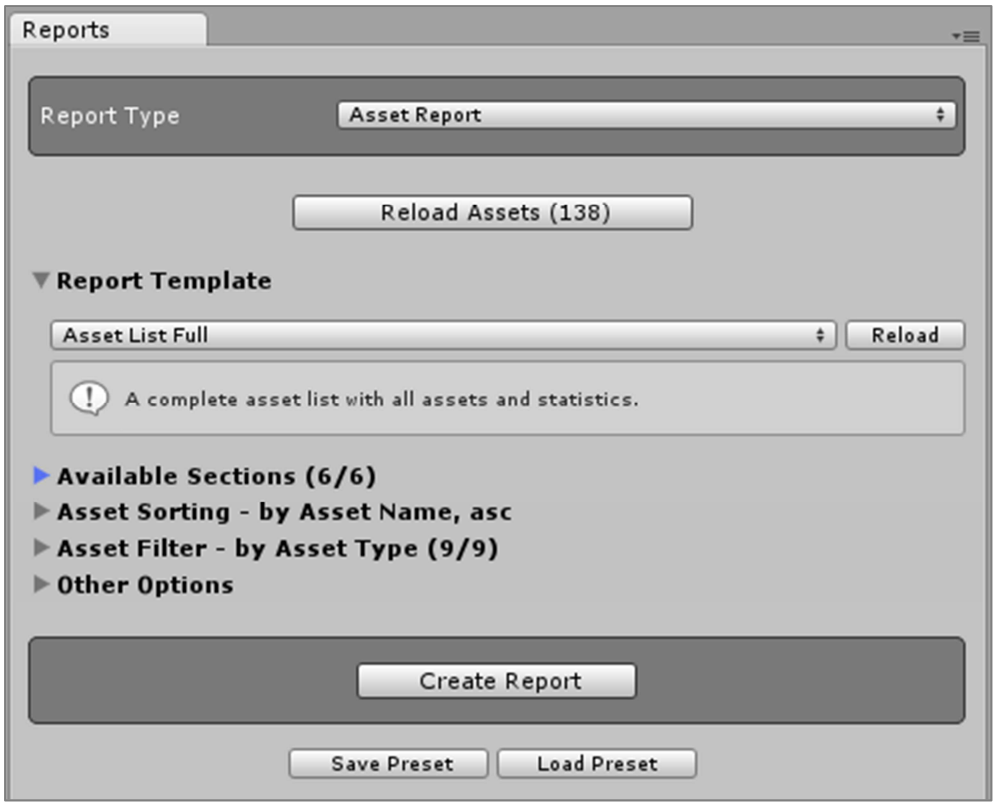
Operation can be one of the following

Operator	Description
ADD	Adds operand 2 to operand 1.
SUB	Subtracts operand 2 from operand 1.
MUL	Multiplies operand 1 with operand 2.
DIV	Divides operand 1 by operand 2.

Presets

Because there are tons of options and checkboxes you have to set for some reports, you can save all settings made in the report window as a preset. You don’t have to go through all settings again when creating the next report with the same settings. Just load the previously saved preset file and you’re done. Presets in combination with custom templates are a great way to have your own special reports at hand with just a few mouse clicks.

To load and save presets please use the buttons at the bottom of the report window.



The Template.ini File

In order to create your own templates you have to create a subfolder for each template in the template directory you have selected in the *Unity* editor when you first worked with the reporting system. The name of the subfolder acts as the template name in the reporting window. This subfolder is the place to save your template file and it has to contain a file called template.ini. In this file some global properties for the template are defined, at least its report type. The property name and value are separated by an equal sign.

```
1   TemplateType=AssetReport
2   Description=An example asset report for the docs.
3
```

Here is the list of available properties. Note that most of these properties can be overridden in the foldout *Other Options* when creating the actual report. So the properties in the template.ini file are default values for this particular template. You only need to set them if they are different from the default values.

Property	Description	Default Value
ReportType	The report type of this template. Values can be • AssetReport • AssetImporterReport • AssetChangeReport • AssetDependenciesReport • SceneReport	

Description	An optional description that will be shown in the editor window below the template selection dropdown.	
Culture	The culture for the numerical and date and time values in the report. It defines the formatting regarding the decimal separators in floating point numbers for example.	en-US
Decimals	The decimals in floating point numbers, like file sizes.	2
SelectTop	In some cases you don't want to report all assets or game objects but only the first 10 or 100. Set this number with this property. If it's 0 then all data records will be shown.	0
MinAssetFileSizeByte	You can set the minimal file size (in bytes) an asset has to have in order to be included in the report. If it's 0 then all assets will be included.	0
DestinationFileName	Set the name of the report file if you want it to be different from the default value. This is useful if you want to create a CSV file for Excel.	index.html
ClearDestinationFolder	If you set this option to *True* all files and folders in the selected destination folder will be deleted before the report is saved.	False
AppendDestinationFile	Set this option to *True* to append the report data to an existing report.	False
ExcludeTemplateFolders	When the system creates a report all files and folders from the template folder are copied to the destination folder. This allows you to have external CSS and Javascript files or images included in your template file. If there are folders that you don't want to be copied, set them here as a list, separated by a semicolon.	
ExcludeTemplateFiles	When the system creates a report all files and folders from the template folder are copied to the destination folder. This allows you to have external CSS and Javascript files or images included in your template file. If there are files that you don't want to be copied, set them here as a list, separated by a semicolon.	
ExcludeAssetFolders	Define a list of folders in the asset folder of your project that have to be excluded from an asset report. This may be third party extensions. This list extends the global excluded folders you set in the preferences window.	
ExcludeAssetFiles	Define a list of files in the asset folder of your project that have to be excluded from an asset report.	

Another example with more properties:

```
TemplateType=AssetReport
Description=A complete asset list with all assets and statistics.
Culture=de-DE
Decimals=3
SelectTop=100
MinAssetFileSizeByte=0
ExcludeTemplateFiles=Readme.txt;
ExcludeTemplateFolders=Dummy;_svn;.svn
ExcludeAssetFolders=ReportsForUnity;UnityTestTools;
ExcludeAssetFiles=FingerLeft.png
```

Note: If the template sets one of the following properties, the assets have to be reloaded by clicking the button *Reload Assets*.

- MinAssetFileSizeByte
- ExcludeAssetFolders
- ExcludeAssetFiles

That's because these properties affect the assets to include in the asset list.

Presets in template.ini file

If you develop your own templates you might want to set some options for sorting and filtering in the report window as soon as the template is loaded. For example, a report listing the vertex and triangle count of meshes should only include game objects with a *MeshFilter* component. To avoid setting all these options manually you can define them in the template.ini file as well. So if you want to set only some of the available checkboxes in a particular foldout of the report window, write the captions of the checkboxes as a list (separated by a semicolon) like this:

```
ReportType=SceneReport
Description=A report with the vertex and triangle counts of all meshes.

GameObjectFilterOptions=MeshFilter;
ComponentFilterOptions=MeshFilter;

```

This will uncheck all components and component fields in the game object and component filter options foldout.

The presets in the template.ini file use the same format as in a regular preset file. The easiest way to define the settings for a template you are developing is to set all options manually in the report window and save it as a preset. Then open the preset file with a text editor and copy all lines with the

needed settings from there. The option keys allowed in the template.ini file for different report types are the following:

Asset Report

Key	Description
AssetsSortBy	Sets the sort field. Possible values are • Asset Name • Asset Path • File Size • File Size Imported • Import File Size Ratio • File Extension • Asset Type • Asset Type (Short) • Creation Time • Last Access Time • Last Write Time • Texture Width Imported • Texture Height Imported • Texture Width Original • Texture Height Original
AssetsSortDirection	The asset sort direction. Possible values are • 0 = Ascending • 1 = Descending
AssetsFilterBy	The asset filter field. Possible values are • Asset Type • File Extension • Asset Label
AssetFilterOptions	List of filter options, separated by semicolon, e.g. *AudioClip;Material;MonoScript;Texture2D;*

Asset Importer Report

Key	Description
AssetsSortBy	Sets the sort field. Possible values are • Asset Name • Asset Path • File Size • File Size Imported • Import File Size Ratio • File Extension • Asset Type • Asset Type (Short) • Creation Time • Last Access Time • Last Write Time • Texture Width Imported

	• Texture Height Imported • Texture Width Original • Texture Height Original
AssetsSortDirection	The asset sort direction. Possible values are • 0 = Ascending • 1 = Descending
AssetsFilterBy	The asset filter field. Possible values are • Asset Type • File Extension • Asset Label
AssetFilterOptions	List of filter options, separated by semicolon, e.g. *AudioClip;Material;MonoScript;Texture2D;*
TextureImporterFieldFilterValues	List of field names, separated by semicolon.
AudioImporterFieldFilterValues	List of field names, separated by semicolon.
ModelImporterFieldFilterValues	List of field names, separated by semicolon.
MovieImporterFieldFilterValues	List of field names, separated by semicolon.
FontImporterFieldFilterValues	List of field names, separated by semicolon.

Asset Change Report

There are no preset values for this report type.

Asset Dependencies Report

Key	Description
AssetsSortBy	Sets the sort field. Possible values are • Asset Name • Asset Path • File Size • File Size Imported • Import File Size Ratio • File Extension • Asset Type • Asset Type (Short) • Creation Time • Last Access Time • Last Write Time • Texture Width Imported • Texture Height Imported • Texture Width Original • Texture Height Original
AssetsSortDirection	The asset sort direction. Possible values are • 0 = Ascending • 1 = Descending
AssetsFilterBy	The asset filter field. Possible values are • Asset Type

	• File Extension • Asset Label
AssetFilterOptions	List of filter options, separated by semicolon, e.g. *AudioClip;Material;MonoScript;Texture2D;*
DisplayShortDependencyList	The string *True* if you want to show the short list version, the string *False* otherwise.

Scene Report

Key	Description
Scenes	List of names, separated by semicolon.
GameObjectsSortBy	The game object sort field. Possible values are • Hierarchy Path • Name • Tag • Layer
GameObjectsSortDirection	The game object sort direction. Possible values are • 0 = Ascending • 1 = Descending
GameObjectsFilterBy	The game object filter field. Possible values are • Attached Components • Layer • Sorting Layer • Tag
GameObjectFilterOptions	List of filter options, separated by semicolon.
ComponentFilterOptions	List of filter options, separated by semicolon.
ComponentFieldFilterOptions	List of filter options, separated by semicolon.

General Slot Reference

These are the slots you can use in all report templates and everywhere in your template.

Global Slots

Slot	Description
ProjectName	The name of the *Unity* project.
ReportDate	The date of the report.
ReportTime	The time of the report.
BuildTarget	The current build target.
SelectTop	The number of displayed records. This can be configured in the template.ini file.
ExcludeAssetFiles	The excluded asset files. This can be configured in the template.ini file.
ExcludeAssetFolders	The excluded asset folders. This can be configured in the template.ini file.
ExcludeTemplateFiles	The excluded files from the template folder. This can be configured in the template.ini file.
ExcludeTemplateFolders	The excluded subfolders from the template folder. This can be

	configured in the template.ini file.
MinAssetFileSizeByte	The minimal asset file size an asset needs to have to be included in the report. This can be configured in the template.ini file.

Section: Table of Contents

The location of this section is marked by: `[++TableOfContents Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
TOCItemName	The complete entry name (section name and block definition name.
TOCItemSectionName	The section name of the entry.
TOCItemBlockDefinitionName	The block definition name of the entry.

Asset Report Slot Reference

General Slots

These slots can be placed everywhere in the template.

Slot	Description
FilterField	The asset filter field. Possible values are • Asset Type • File Extension • Asset Label
FilterValues	The list of filter values you selected in the report window.
SortField	The asset sort field. Possible values are • Asset Name • Asset Path • File Size • File Size Imported • Import File Size Ratio • File Extension • Asset Type • Asset Type (Short) • Creation Time • Last Access Time • Last Write Time • Texture Width Imported • Texture Height Imported • Texture Width Original • Texture Height Original

SortDirection	Depending on the sort direction: the string “asc” or “desc”
AssetFolderCount	The number of asset folders in the project asset folder without the excluded folders.
AssetCount	The number of assets in the project asset folder without the excluded folders and files.
AssetCountFiltered	The number of assets in the project asset folder without the excluded folders and files after the filter has been applied.
AssetTotalFileSize	The file size of all assets in bytes.
AssetTotalFileSizeKB	The file size of all assets in KB.
AssetTotalFileSizeMB	The file size of all assets in MB.
AssetTotalFileSizeGB	The file size of all assets in GB.
AssetTotalFileSizeAuto	The file size of all assets in a fitting unit (B, KB, MB, GB).
AssetTotalFileSizeFiltered	The file size of all filtered assets in bytes.
AssetTotalFileSizeFilteredKB	The file size of all filtered assets in KB.
AssetTotalFileSizeFilteredMB	The file size of all filtered assets in MB.
AssetTotalFileSizeFilteredGB	The file size of all filtered assets in GB.
AssetTotalFileSizeFilteredAuto	The file size of all filtered assets in a fitting unit (B, KB, MB, GB).
AssetTotalFileSizeImported	The file size of all assets in bytes after importing to *Unity*.
AssetTotalFileSizeImportedKB	The file size of all assets in KB after importing to *Unity*.
AssetTotalFileSizeImportedMB	The file size of all assets in MB after importing to *Unity*.
AssetTotalFileSizeImportedGB	The file size of all assets in GB after importing to *Unity*.
AssetTotalFileSizeImportedAuto	The file size of all assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
AssetTotalFileSizeImportedFiltered	The file size of all filtered assets in bytes after importing to *Unity*.
AssetTotalFileSizeImportedFilteredKB	The file size of all filtered assets in KB after importing to *Unity*.
AssetTotalFileSizeImportedFilteredMB	The file size of all filtered assets in MB after importing to *Unity*.
AssetTotalFileSizeImportedFilteredGB	The file size of all filtered assets in GB after importing to *Unity*.
AssetTotalFileSizeImportedFilteredAuto	The file size of all filtered assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.

Section: Asset Folders

The location of this section is marked by: `[++AssetFolders Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
FolderRelativePath	The relative path of the folder, e.g. *Assets/Textures*

FolderAbsolutePath	The absolute path of the folderin the file system, e.g. *D:/MyUnityProject/Assets/Textures.*
FolderName	The name of the folder.
FolderCreationTime	The creation time of the folder.
FolderLastAccessTime	The last access time of the folder.
FolderLevel	The level of the folder in the hierarchy, starting with 1.
FolderSubFolderCount	The subfolder count of the folder.
FolderHasSubFolders	The string *yes*, if the folder has subfolders, *no* otherwise.

Section: Assets

The location of this section is marked by: `[++Assets Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
AssetGUID	The GUID of the asset. *Unity* saves the asset file with the GUID as the file name when importing it to the *Unity* asset library.
AssetName	The name of the asset.
AssetFileExtension	The file extension of the asset.
AssetPath	The relative path of the asset.
AssetCreationTime	The creation time of the asset file.
AssetLastAccessTime	The last access time of the asset file.
AssetLastWriteTime	The last write time of the asset file.
AssetType	The type of the asset, e.g. *UnityEngine.Texture2D*.
AssetTypeShort	The short type of the asset, e.g. *Texture2D*.
AssetFullPath	The full path of the asset in the file system. Can be used to set a link to the asset file in the report.
IsMainAsset	The string *yes*, if the asset is a main asset, *no* otherwise.
IsSubAsset	The string *yes*, if the asset is a sub asset, *no* otherwise.
AssetLabels	The asset labels of the asset, separated by a comma.
AssetFileSize	The file size of the asset in bytes.
AssetFileSizeKB	The file size of the asset in KB.
AssetFileSizeMB	The file size of the asset in MB.
AssetFileSizeGB	The file size of the asset in GB.
AssetFileSizeAuto	The file size of the asset in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
AssetFileSizeImported	The file size of the asset in bytes after importing to *Unity*.
AssetFileSizeImportedKB	The file size of the asset in KB after importing to *Unity*.
AssetFileSizeImportedMB	The file size of the asset in MB after importing to *Unity*.
AssetFileSizeImportedGB	The file size of the asset in GB after importing to *Unity*.
AssetFileSizeImportedAuto	The file size of the asset in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
ImportFileSizeRatio	The ratio between the original file size and the file size of the imported asset. A value less than 100 means that the asset has been compressed.
TextureWidth	The width of a texture asset after importing to *Unity*.
TextureHeight	The height of a texture asset after importing to *Unity*.
TextureWidthOriginal	The original width of a texture asset file.

TextureHeightOriginal	The original height of a texture asset file.
IsTextureWidthPOT	The string *yes*, if the texture width is a power of 2, *no* otherwise.
IsTextureHeightPOT	The string *yes*, if the texture height is a power of 2, *no* otherwise.
IsTextureWidthAndHeightPOT	The string *yes*, if the texture width and height is a power of 2, *no* otherwise.
IsTextureWidthOrHeightPOT	The string *yes*, if the texture width or height is a power of 2, *no* otherwise.

Section: Asset Register

The location of this section is marked by: `[++AssetRegister Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
RegisterChar	The register char. You can set a link to a corresponding HTML anchor in the report: `<a href="#[--RegisterChar--]">[--RegisterChar--]</a>`

Section: Asset Stats by Type

The location of this section is marked by: *[++AssetStatsByType Block=<Blockdefinition>++]*

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
AssetType	The type of the asset, e.g. *UnityEngine.Texture2D*.
AssetTypeShort	The short type of the asset, e.g. *Texture2D*.
Count	The number of assets with the current asset type.
TotalFileSize	The file size of the assets with the current asset type in bytes.
TotalFileSizeKB	The file size of the assets with the current asset type in KB.
TotalFileSizeMB	The file size of the assets with the current asset type in MB.
TotalFileSizeGB	The file size of the assets with the current asset type in GB.
TotalFileSizeAuto	The file size of the assets with the current asset type in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
TotalFileSizeImported	The file size of the assets with the current asset type in bytes after importing to *Unity*.
TotalFileSizeImportedKB	The file size of the assets with the current asset type in KB after importing to *Unity*.
TotalFileSizeImportedMB	The file size of the assets with the current asset type in MB after importing to *Unity*.
TotalFileSizeImportedGB	The file size of the assets with the current asset type in GB after importing to *Unity*.
TotalFileSizeImportedAuto	The file size of the assets with the current asset type in a fitting unit (B,

KB, MB, GB) after importing to *Unity*.

Asset Importer Report Slot Reference

General Slots

These slots can be placed everywhere in the template.

Slot	Description
FilterField	The asset filter field. Possible values are • Asset Type • File Extension • Asset Label
FilterValues	The list of filter values you selected in the report window.
SortField	The asset sort field. Possible values are • Asset Name • Asset Path • File Size • File Size Imported • Import File Size Ratio • File Extension • Asset Type • Asset Type (Short) • Creation Time • Last Access Time • Last Write Time • Texture Width Imported • Texture Height Imported • Texture Width Original • Texture Height Original
SortDirection	Depending on the sort direction: the string “asc” or “desc”
AssetFolderCount	The number of asset folders in the project asset folder without the excluded folders.
AssetCount	The number of assets in the project asset folder without the excluded folders and files.
AssetCountFiltered	The number of assets in the project asset folder without the excluded folders and files after the filter has been applied.
AssetTotalFileSize	The file size of all assets in bytes.
AssetTotalFileSizeKB	The file size of all assets in KB.
AssetTotalFileSizeMB	The file size of all assets in MB.
AssetTotalFileSizeGB	The file size of all assets in GB.
AssetTotalFileSizeAuto	The file size of all assets in a fitting unit (B, KB, MB, GB).
AssetTotalFileSizeFiltered	The file size of all filtered assets in bytes.

AssetTotalFileSizeFilteredKB	The file size of all filtered assets in KB.
AssetTotalFileSizeFilteredMB	The file size of all filtered assets in MB.
AssetTotalFileSizeFilteredGB	The file size of all filtered assets in GB.
AssetTotalFileSizeFilteredAuto	The file size of all filtered assets in a fitting unit (B, KB, MB, GB).
AssetTotalFileSizeImported	The file size of all assets in bytes after importing to *Unity*.
AssetTotalFileSizeImportedKB	The file size of all assets in KB after importing to *Unity*.
AssetTotalFileSizeImportedMB	The file size of all assets in MB after importing to *Unity*.
AssetTotalFileSizeImportedGB	The file size of all assets in GB after importing to *Unity*.
AssetTotalFileSizeImportedAuto	The file size of all assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
AssetTotalFileSizeImportedFiltered	The file size of all filtered assets in bytes after importing to *Unity*.
AssetTotalFileSizeImportedFilteredKB	The file size of all filtered assets in KB after importing to *Unity*.
AssetTotalFileSizeImportedFilteredMB	The file size of all filtered assets in MB after importing to *Unity*.
AssetTotalFileSizeImportedFilteredGB	The file size of all filtered assets in GB after importing to *Unity*.
AssetTotalFileSizeImportedFilteredAuto	The file size of all filtered assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.

Section: Texture Import Settings

The location of this section is marked by: `[++TextureImportSettings Block=<Blockdefinition>++]`

You can use all common asset slots. In addition you can use slots for the importer public fields in the format:

```
TextureImporter.<FieldName>
```

For the list of fields you can define another section inside the used block definition:

```
[++TextureImportSettingFields Block=<Blockdefinition++]
```

This section will fill the following slots for each field.

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
FieldName	The name of the field.
FieldValue	The value of the field.

Section: Audio Import Settings

The location of this section is marked by: `[++AudioImportSettings Block=<Blockdefinition>++]`

You can use all common asset slots. In addition you can use slots for the importer public fields in the format:

```
AudioImporter.<FieldName>
```

For the list of fields you can define another section inside the used block definition:

```
[++AudioImportSettingFields Block=<Blockdefinition++]
```

This section will fill the following slots for each field.

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
FieldName	The name of the field.
FieldValue	The value of the field.

Section: Model Import Settings

The location of this section is marked by: `[++ModelImportSettings Block=<Blockdefinition>++]`

You can use all common asset slots. In addition you can use slots for the importer public fields in the format:

```
ModelImporter.<FieldName>
```

For the list of fields you can define another section inside the used block definition:

```
[++ModelImportSettingFields Block=<Blockdefinition++]
```

This section will fill the following slots for each field.

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
FieldName	The name of the field.
FieldValue	The value of the field.

Section: Movie Import Settings

The location of this section is marked by: `[++MovieImportSettings Block=<Blockdefinition>++]`

You can use all common asset slots. In addition you can use slots for the importer public fields in the format:

```
MovieImporter.<FieldName>
```

For the list of fields you can define another section inside the used block definition:

`[++MovieImportSettingFields Block=<Blockdefinition++]`

This section will fill the following slots for each field.

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
FieldName	The name of the field.
FieldValue	The value of the field.

Section: Font Import Settings

The location of this section is marked by: `[++FontImportSettings Block=<Blockdefinition>++]`

You can use all common asset slots in this section. In addition you can use slots for the importer public fields in the format:

`FontImporter.<FieldName>`

For the list of fields you can define another section inside the used block definition:

`[++FontImportSettingFields Block=<Blockdefinition++]`

This section will fill the following slots for each field.

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
FieldName	The name of the field.
FieldValue	The value of the field.

Scene Report Slot Reference

General Slots

These slots can be placed everywhere in the template.

Slot	Description
FilterField	The game object filter field. Possible values are • Attached Components • Layer • Sorting Layer • Tag
FilterValues	The list of filter values you selected in the report window.
SortField	The game object sort field. Possible values are • Hierarchy Path

	• Name • Tag • Layer
SortDirection	Depending on the sort direction: the string “asc” or “desc”
GameObjectCount	The number of game objects in the selected scenes.
GameObjectCountFiltered	The number of game objects in the selected scenes after the game object filter has been applied.
MissingScriptsCount	The total number of missing scripts in the selected scenes.
MissingPrefabsCount	The total number of missing prefabs in the selected scenes.
MissingMeshesCount	The total number of missing meshes in the selected scenes.
MeshCount	The total number of meshes in the selected scenes.
MeshVertexCount	The total vertex count of all meshes in the selected scenes.
MeshTriangleCount	The total triangle count of all meshes in the selected scenes.
MeshColliderCount	The total number of mesh colliders in the selected scenes.
PolygonCollider2DCount	The total number of polygon colliders (*PolygonCollider2D*) in the selected scenes.
PolygonCollider2DPointsCount	The total point count of all polygon colliders (*PolygonCollider2D*) in the selected scenes.

Section: Game Object Register

The location of this section is marked by: `[++GameObjectRegister Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
RegisterChar	The register char. You can set a link to a corresponding HTML anchor in the report: `<a href="#[--RegisterChar--]">[--RegisterChar--]</a>`

Section: Game Objects

The location of this section is marked by: `[++GameObjects Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
Name	The name of the game object.
NameWithHierarchyPath	The name of the game object with the hierarchy path without the scene name.
NameWithHierarchyPathFull	The name of the game object with the hierarchy path, starting with

	the scene name.
Scene	The name of the scene.
ActiveSelf	The string *yes*, if the property *ActiveSelf* is true, *no* otherwise.
ActiveInHierarchy	The string *yes*, if the property *ActiveInHierarchy* is true, *no* otherwise.
HierarchyPath	The hierarchy path of the game object without the scene name.
HierarchyPathFull	The hierarchy path of the game object, starting with the scene name.
HierarchyLevel	The level of the game object in the hierarchy, starting with 1.
HasMissingScripts	The string *yes*, if the game object has missing scripts, *no* otherwise.
HasMissingPrefab	The string *yes*, if the game object has a missing prefab, *no* otherwise.
HasMissingMeshes	The string *yes*, if the game object has missing meshes, *no* otherwise.
IsStatic	The string *yes*, if the property *IsStatic* is true, *no* otherwise.
Layer	The layer name of the game object.
Tag	The tag of the game object.
IsPrefab	The string *yes*, if the game object has a prefab, *no* otherwise.
PrefabPath	The prefab path.

Subsection: Game Object Components

For every game object a list of its attached components can be reported. Use this section between the opening and closing tag of a game object block definition.

The location of this section is marked by: `[++Components Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
ComponentName	The name of the component, e.g. *BoxCollider*.

In this section you can also report all component fields one by one, not as a complete list as described in the next chapter. To show a components field in the component section use the following format:

`[--Component.<Componentname>.Fieldname--]`

Example: `[--Component.Meshfilter.MeshVertexCount--]`

If the current component has the type *Componentname* and the public field *Fieldname* the value of this field will be printed at the slots position. It’s a good idea to use this type of slots in report templates together with a filter including only the components specified in the slot.

Subsection: Component Fields

For every component a list of its public fields and properties (as shown in the inspector) can be reported. Use this section between the opening and closing tag of a game object component block definition.

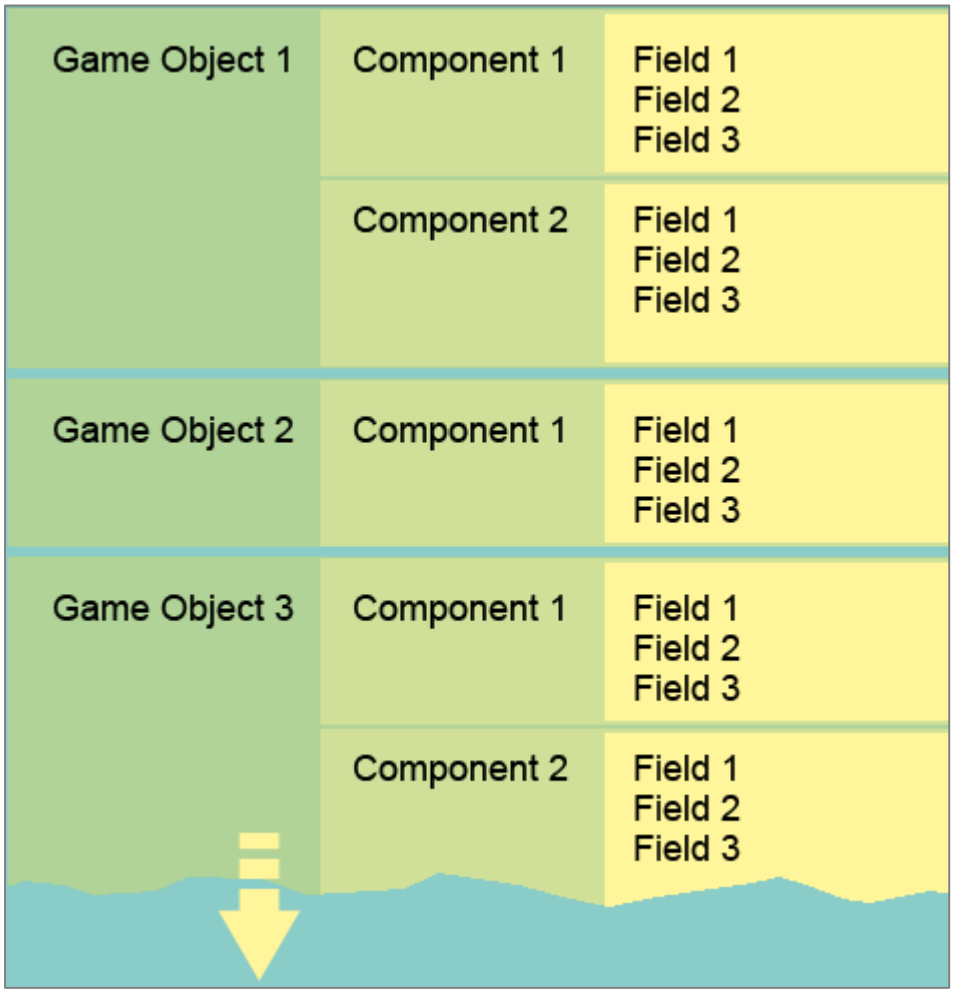
Note that the names of the fields are not always the same as the field captions in the inspector, because they are retrieved from the C# classes using reflection. But it’s quite easy to identify the

relations between the captions and fields. There are also fields that are not displayed for editing in the inspector at all.

The location of this section is marked by: `[++ComponentFields Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
ComponentFieldName	The name of the component field.
ComponentFieldValue	The value of the component field.

You can see an overview of the nested sections and subsections in the image below.



Section: Scenes

The location of this section is marked by: `[++Scenes Block=<Blockdefinition>++]`

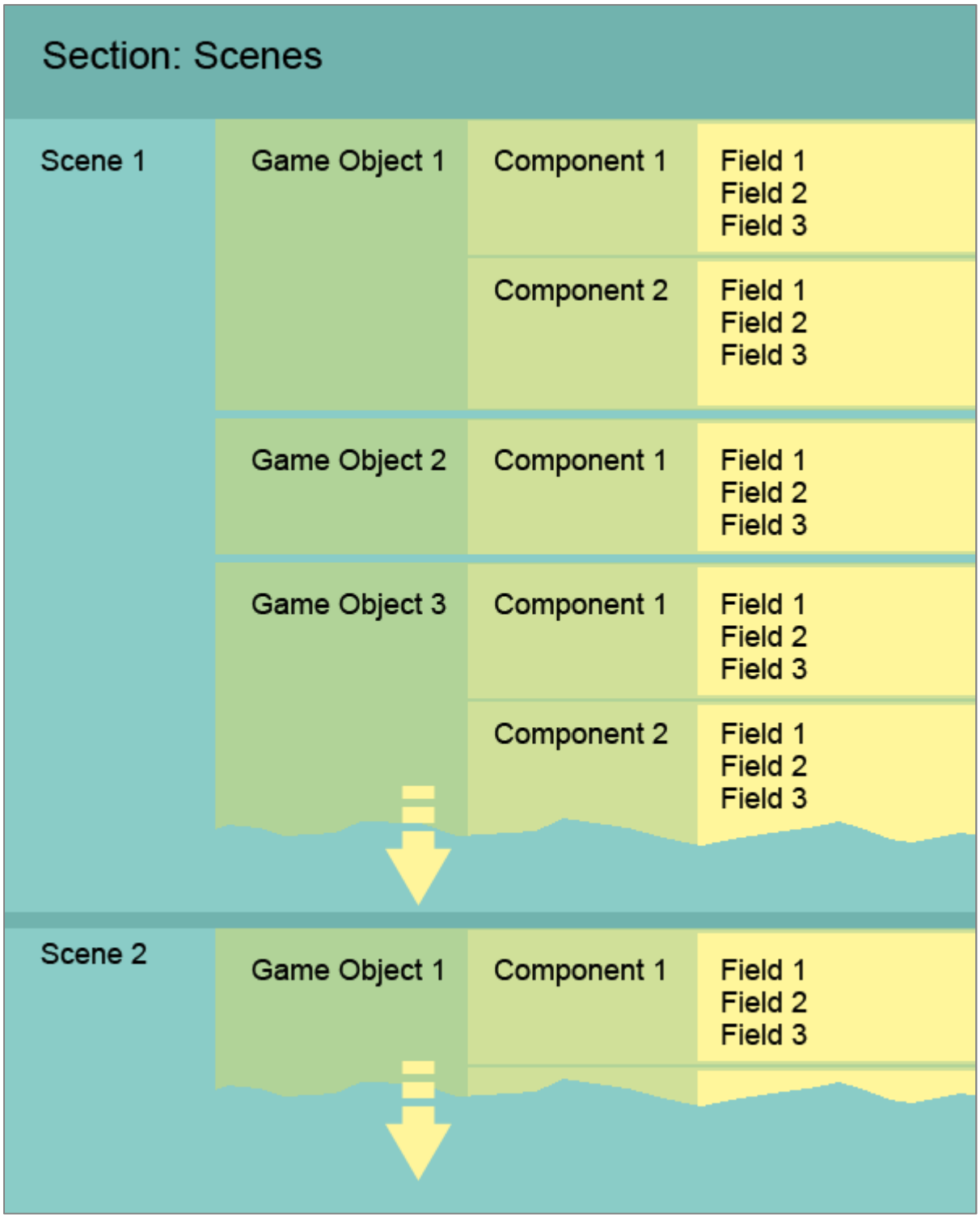
Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.

OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
ScenePath	The path of the scene, e.g. *Assets/Scenes/MainScene.unity*.
ScenePathShort	The short path of the scene, e.g. *Scenes/MainScene*.
MissingScriptsCount	The number of missing scripts in the scene.
MissingPrefabsCount	The number of missing prefabs in the scene.
MissingMeshesCount	The number of missing meshes in the scene.
GameObjectCount	The number of game objects in the scene.
MeshCount	The number of meshes in the scene.
MeshVertexCount	The total vertex count of all meshes in the scene.
MeshTriangleCount	The total triangle count of all meshes in the scene.
MeshColliderCount	The number of mesh colliders in the scene.
PolygonCollider2DCount	The number of polygon colliders (*PolygonCollider2D*) in the scene.
PolygonCollider2DPointsCount	The total point count of all polygon colliders (*PolygonCollider2D*) in the scene.

Subsection: Game Objects

For each scene you can report its game objects. Use the same slots as for the complete game object section including the components with their fields.

You can see an overview of the nested sections and subsections in the image below.



Section: Missing Scripts

The location of this section is marked by: `[++MissingScripts Block=<Blockdefinition>++]`

You can use all common game object slots in this section including the components and component fields.

Section: Missing Prefabs

The location of this section is marked by: `[++MissingPrefabs Block=<Blockdefinition>++]`

You can use all common game object slots in this section including the components and component fields.

Section: Missing Meshes

The location of this section is marked by: `[++MissingMeshes Block=<Blockdefinition>++]`

You can use all common game object slots in this section including the components and component fields.

Asset Change Report Slot Reference

General Slots

These slots can be placed everywhere in the template.

Slot	Description
MissingAssetCount	The number of missing (deleted) assets.
NewAssetCount	The number of new assets.
ChangedAssetCount	The number of changed assets.
MissingAssetTotalFileSize	The file size of the missing assets in bytes.
MissingAssetTotalFileSizeKB	The file size of the missing assets in KB.
MissingAssetTotalFileSizeMB	The file size of the missing assets in MB.
MissingAssetTotalFileSizeGB	The file size of the missing assets in GB.
MissingAssetTotalFileSizeAuto	The file size of the missing assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
MissingAssetTotalFileSizeImported	The file size of the missing assets in bytes after importing to *Unity*.
MissingAssetTotalFileSizeImportedKB	The file size of the missing assets in KB after importing to *Unity*.
MissingAssetTotalFileSizeImportedMB	The file size of the missing assets in MB after importing to *Unity*.
MissingAssetTotalFileSizeImportedGB	The file size of the missing assets in GB after importing to *Unity*.
MissingAssetTotalFileSizeImportedAuto	The file size of the missing assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
NewAssetTotalFileSize	The file size of the new assets in bytes.
NewAssetTotalFileSizeKB	The file size of the new assets in KB.
NewAssetTotalFileSizeMB	The file size of the new assets in MB.
NewAssetTotalFileSizeGB	The file size of the new assets in GB.
NewAssetTotalFileSizeAuto	The file size of the new assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
NewAssetTotalFileSizeImported	The file size of the new assets in bytes after importing to *Unity*.

NewAssetTotalFileSizeImportedKB	The file size of the new assets in KB after importing to *Unity*.
NewssetTotalFileSizeImportedMB	The file size of the new assets in MB after importing to *Unity*.
NewAssetTotalFileSizeImportedGB	The file size of the new assets in GB after importing to *Unity*.
NewAssetTotalFileSizeImportedAuto	The file size of the new assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
ChangedAssetTotalFileSize	The file size of the changed assets in bytes.
ChangedAssetTotalFileSizeKB	The file size of the changed assets in KB.
ChangedAssetTotalFileSizeMB	The file size of the changed assets in MB.
ChangedAssetTotalFileSizeGB	The file size of the changed assets in GB.
ChangedAssetTotalFileSizeAuto	The file size of the changed assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
ChangedAssetTotalFileSizeImported	The file size of the changed assets in bytes after importing to *Unity*.
ChangedAssetTotalFileSizeImportedKB	The file size of the changed assets in KB after importing to *Unity*.
ChangedAssetTotalFileSizeImportedMB	The file size of the changed assets in MB after importing to *Unity*.
ChangedAssetTotalFileSizeImportedGB	The file size of the changed assets in GB after importing to *Unity*.
ChangedAssetTotalFileSizeImportedAuto	The file size of the changed assets in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
FileSizeChangeTypeTotal	The string *up*, if the total file size of all changed assets has increased. The string *down*, if it has decreased and *even*, if there is no change at all.
FileSizeChangeValueTotal	The value of the total file size change in bytes.
FileSizeChangeValueTotalKB	The value of the total file size change in KB.
FileSizeChangeValueTotalMB	The value of the total file size change in MB.
FileSizeChangeValueTotalGB	The value of the total file size change in GB.
FileSizeChangeValueTotalAuto	The value of the total file size change in a fitting unit (B, KB, MB, GB).
FileSizeChangeTypeTotalImported	The string *up*, if the total file size of all changed assets (after the import to *Unity*) has increased. The string *down*, if it has decreased and *even*, if there is no change at all.
FileSizeChangeValueTotalImported	The value of the total file size change in bytes after importing to *Unity*.
FileSizeChangeValueTotalImportedKB	The value of the total file size change in KB after importing to *Unity*.
FileSizeChangeValueTotalImportedMB	The value of the total file size change in MB after importing to *Unity*.
FileSizeChangeValueTotalImportedGB	The value of the total file size change in GB after importing to *Unity*.
FileSizeChangeValueTotalImportedAuto	The value of the total file size change in a fitting unit (B, KB, MB, GB) after importing to *Unity*.

Section: Missing Assets

The location of this section is marked by: `[++MissingAssets Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
AssetPath	The path of the asset.
AssetFileSize	The file size of the asset in bytes.
AssetFileSizeKB	The file size of the asset in KB.
AssetFileSizeMB	The file size of the asset in MB.
AssetFileSizeGB	The file size of the asset in GB.
AssetFileSizeAuto	The file size of the asset in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
AssetFileSizeImported	The file size of the asset in bytes after importing to *Unity*.
AssetFileSizeImportedKB	The file size of the asset in KB after importing to *Unity*.
AssetFileSizeImportedMB	The file size of the asset in MB after importing to *Unity*.
AssetFileSizeImportedGB	The file size of the asset in GB after importing to *Unity*.
AssetFileSizeImportedAuto	The file size of the asset in a fitting unit (B, KB, MB, GB) after importing to *Unity*.

Section: New Assets

The location of this section is marked by: `[++NewAssets Block=<Blockdefinition>++]`

You can use all common asset slots in this section.

Section: Changed Assets

The location of this section is marked by: `[++ChangedAssets Block=<Blockdefinition>++]`

You can use all common asset slots in this section. In addition you can use the following slots.

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
OldAssetFileSize	The file size of the old asset in bytes.
OldAssetFileSizeKB	The file size of the old asset in KB.
OldAssetFileSizeMB	The file size of the old asset in MB.
OldAssetFileSizeGB	The file size of the old asset in GB.
OldAssetFileSizeAuto	The file size of the old asset in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
OldAssetFileSizeImported	The file size of the old asset in bytes after importing to *Unity*.
OldAssetFileSizeImportedKB	The file size of the old asset in KB after importing to *Unity*.
OldAssetFileSizeImportedMB	The file size of the old asset in MB after importing to *Unity*.

OldAssetFileSizeImportedGB	The file size of the old asset in GB after importing to *Unity*.
OldAssetFileSizeImportedAuto	The file size of the old asset in a fitting unit (B, KB, MB, GB) after importing to *Unity*.
FileSizeChangeType	The string *up*, if the file size of the changed asset has increased. The string *down*, if it has decreased and *even*, if there is no change at all.
FileSizeChangeValue	The value of the file size change in bytes.
FileSizeChangeValueKB	The value of the file size change in KB.
FileSizeChangeValueMB	The value of the file size change in MB.
FileSizeChangeValueGB	The value of the file size change in GB.
FileSizeChangeValueAuto	The value of the file size change in a fitting unit (B, KB, MB, GB).
FileSizeChangeTypeImported	The string *up*, if the file size of the changed asset has increased after importing to *Unity*. The string *down*, if it has decreased and *even*, if there is no change at all.
FileSizeChangeValueImported	The value of the file size change in bytes after importing to *Unity*.
FileSizeChangeValueImportedKB	The value of the file size change in KB after importing to *Unity*.
FileSizeChangeValueImportedMB	The value of the file size change in MB after importing to *Unity*.
FileSizeChangeValueImportedGB	The value of the file size change in GB after importing to *Unity*.
FileSizeChangeValueImportedAuto	The value of the file size change in a fitting unit (B, KB, MB, GB) after importing to *Unity*.

Asset Dependencies Report Slot Reference

General Slots

These slots can be placed everywhere in the template.

Slot	Description
FilterField	The game object filter field. Possible values are • Attached Components • Layer • Sorting Layer • Tag
FilterValues	The list of filter values you selected in the report window.
SortField	The asset sort field. Possible values are • Asset Name • Asset Path • File Size • File Size Imported • Import File Size Ratio • File Extension • Asset Type

	• Asset Type (Short) • Creation Time • Last Access Time • Last Write Time • Texture Width Imported • Texture Height Imported • Texture Width Original • Texture Height Original
SortDirection	Depending on the sort direction: the string “asc” or “desc”
AssetFolderCount	The number of asset folders in the project asset folder without the excluded folders.
AssetCount	The number of assets in the project asset folder without the excluded folders and files.
AssetCountFiltered	The number of assets in the project asset folder without the excluded folders and files after the filter has been applied.

Section: Assets with Dependencies

The location of this section is marked by: `[++AssetsWithDependencies Block=<Blockdefinition>++]`

You can use all common asset slots in this section. In addition you can use the following slots per asset.

Slot	Description
HasMissingDependencies	The string *yes*, if the asset has missing dependencies, *no* otherwise.
MissingDependencyCount	The number of missing dependencies.

Inside the block definition you used for this section you can have the section

`[++AssetDependencies Block=<Blockdefinition>++]`

This section reports all dependencies of an asset in the list (section *AssetsWithDependencies*).

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
DependencyName	The name of the dependent asset.
DependencyAssetPath	The path of the dependent asset.

Section: Assets with Missing Dependencies

The location of this section is marked by: `[++AssetsWithMissingDependencies Block=<Blockdefinition>++]`

You can use all common asset slots in this section. In addition you can use the following slots per asset.

Slot	Description
HasMissingDependencies	The string *yes*, if the asset has missing dependencies, *no* otherwise.
MissingDependencyCount	The number of missing dependencies.

Inside the block definition you used for this section you can have the section

`[++AssetDependencies Block=<Blockdefinition>++]`

This section reports all dependencies of an asset in the list (section *AssetsWithMissingDependencies*).

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
DependencyName	The name of the dependent asset.
DependencyAssetPath	The path of the dependent asset.

Section: Assets Reverse Dependencies

The location of this section is marked by: `[++AssetsReverseDependencies Block=<Blockdefinition>++]`

Slot	Description
Index	The current row number, starting with 1.
IndexFormatted	The current row number, filled left with zeros.
OddEven	If the row number is odd, it prints “odd”, otherwise “even”.
AssetPath	The path of the asset, that depends in the current asset.

Inside the block definition you used for this section you can have the section

`[++AssetReverseDependencies Block=<Blockdefinition>++]`

This section reports all depending assets of a dependency in the list (section *AssetsReverseDependencies*).

You can use all common asset slots in this section.