



*Harmony*

---

# Toon Boom Harmony V15.0 Scripting Guide



---

TOON BOOM ANIMATION INC.  
4200 Saint-Laurent, Suite 1020  
Montreal, Quebec, Canada  
H2W 2R2

---

+1 514 278 8666  
contact@toonboom.com  
toonboom.com

## Legal Notices

Toon Boom Animation Inc.  
4200 Saint-Laurent, Suite 1020  
Montreal, Quebec, Canada  
H2W 2R2

Tel: +1 514 278 8666

Fax: +1 514 278 2666

[toonboom.com](http://toonboom.com)

### **Disclaimer**

The content of this guide is covered by a specific limited warranty and exclusions and limit of liability under the applicable License Agreement as supplemented by the special terms and conditions for Adobe®Flash® File Format (SWF). For details, refer to the License Agreement and to those special terms and conditions.

The content of this guide is the property of Toon Boom Animation Inc. and is copyrighted.

Any reproduction in whole or in part is strictly prohibited.

### **Trademarks**

Harmony is a trademark of Toon Boom Animation Inc.

### **Publication Date**

01-15-2020

Copyright © 2018 Toon Boom Animation Inc., a Corus Entertainment Inc. company. All rights reserved.

---

# Table of Contents

Table of Contents .....	2
About Scripting .....	4
Chapter 1: Scripting Reference .....	6
Chapter 2: Creating Scripts .....	8
Chapter 2: Checking Script Syntax .....	13
Chapter 2: Testing a Script .....	14
Chapter 3: Debugging Scripts .....	16
Chapter 3: Using the Scripting Sandbox .....	18
Chapter 4: Importing Scripts .....	20
Chapter 5: Deleting Scripts .....	22
Chapter 5: Restoring a Harmony Script .....	23
Chapter 6: Using an External Script Editor .....	26
Chapter 7: Adding Scripts as Toolbar Buttons .....	28
Chapter 8: Exposing Node Attribute Names for Scripting .....	30



# About Scripting

T-ANIMPA-007-001

The Harmony Scripting guide contains information to help you use Qt Script, the scripting language used by Toon Boom Harmony.

Qt Script provides access to many of the functions supported in the Harmony interface. With Qt Script, you can automate a number of Harmony functions to speed up the completion of various repetitive tasks.

Harmony has three different scripting interfaces:

- Harmony Scripts are scripts that you can write and use directly in the Harmony user interface. Among many possible uses, Harmony scripts can automate workflows and functionalities in Harmony, create dialogs and perform batch operations on a scene.
- The Script Module is a special node which can take input from image or peg nodes, process its information in a custom-made script and output the processed peg or image information in the desired way. It allows you to create nodes that have custom behaviors. For example, the Maya Render Batch node is actually a Script Module template that is configured to render your scene through an external render.
- DbScript are scripts than can be launched in a command line interface via the Control Center application. They can be used to perform operations on the different elements of a Harmony Server database.

Qt Script is an object-oriented scripting language based on the ECMAScript standard, like JavaScript and JScript. However, there are some differences that distinguish Qt Script from these other scripting languages, which are familiar to web programmers. For a reference on how to write Qt Scripts for the different scripting interfaces in Harmony , see [Scripting Reference on page 6](#).

**NOTE**

For a reference of the QT development environment, refer to the [Qt Documentation](#) website.

**NOTE**

For a tutorial on writing JavaScript, refer to the [JavaScript Tutorial](#) section of W3Schools.




**NOTE**

As of Toon Boom Harmony 10.3, Toon Boom Animation has changed its scripting language from QSA script to Qt Script for a better and faster scripting environment. Qt Script is very similar to JavaScript. Scripts are stored in plain text in files with the `.js` extension, instead of `.qsa`. This change to the API improves the software and gives access to 64-bit capabilities of the scripting interface.

To avoid complications for users of earlier version, every effort was made to enable backward compatibility with QSA scripts. Simple functions can be ported without making any changes. However, class definitions must conform to the Qt Script syntax. Therefore, scripts that define classes must be updated. For detailed instructions on how to update from QSA to Qt scripts, see the [Moving from QSA to Qt Script](#) documentation of the Qt website.

# Chapter 1: Scripting Reference

T-HSCP-001-001

When scripting in Harmony, you can open the scripting reference guide from the Script Editor by opening the Script Editor view menu  and selecting **Help > Scripting Interface Documentation**.

You can also open the reference guide for the Harmony scripting interface, the Script module and for Control Center scripting from the installation directory's help sub-directory, at the following locations:

On Windows:

- Harmony scripting: `C:\Program Files (x86)\Toon Boom Animation\Toon Boom Harmony 15.0 [Edition]\help\script\index.html`
- Script node: `C:\Program Files (x86)\Toon Boom Animation\Toon Boom Harmony 15.0 [Edition]\help\scriptmodule\index.html`
- Control Center scripting: `C:\Program Files (x86)\Toon Boom Animation\Toon Boom Harmony 15.0 [Edition]\help\dbscript`

On macOS (Stand Alone):

- Harmony scripting: `/Applications/Toon Boom Harmony 15.0 [Edition]/tba/help/script/index.html`
- Script node: `/Applications/Toon Boom Harmony 15.0 [Edition]/tba/help/scriptmodule/index.html`
- Control Center scripting: `/Applications/Toon Boom Harmony 15.0 [Edition]/tba/help/dbscript/index.html`

On macOS (Server):

- Harmony scripting: `/Applications/Toon Boom Harmony 15.0 [Edition] Network/tba/help/script/index.html`
- Script node: `/Applications/Toon Boom Harmony 15.0 [Edition] Network/tba/help/scriptmodule/index.html`
- Control Center scripting: `/Applications/Toon Boom Harmony 15.0 [Edition] Network/tba/help/dbscript/index.html`

On GNU/Linux:

- Harmony scripting: `/usr/local/ToonBoomAnimation/harmony[Edition]_15.0/help/script/index.html`
- Script node: `/usr/local/ToonBoomAnimation/harmony[Edition]_15.0/help/scriptmodule/index.html`
- Control Center scripting: `/usr/local/ToonBoomAnimation/harmony[Edition]_15.0/help/dbscript/index.html`

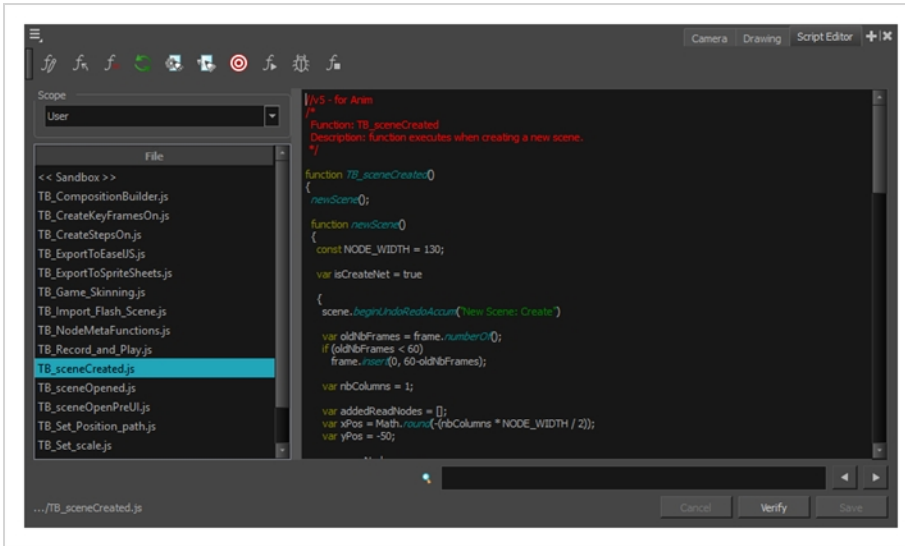




## Chapter 2: Creating Scripts

T-HSCP-001-002

You can create scripts for Harmony by using the Script Editor view.



### TIP

To help familiarize yourself with Harmony scripts, you can open the Script Editor view and look into the **TB\_sceneCreated** script. This script is automatically executed when you create a scene. By default, it sets the scene's length to 60 frames and creates a basic layout containing a single drawing layer. You can customize this script to automatically set up scenes you create to fit your project's needs.

**NOTE**

When using Harmony Stand Alone, user scripts are stored in the following location:

- **Windows:** `C:\Users\[username]\AppData\Roaming\Toon Boom Animation\Toon Boom Harmony [edition]\1500-scripts`
- **macOS:** `/Users/[username]/Library/Preferences/Toon Boom Animation/Toon Boom Harmony [edition]/1500-scripts`
- **GNU/Linux:** `/home/[username]/Toon Boom Animation/Toon Boom Harmony [edition]/1500-scripts`

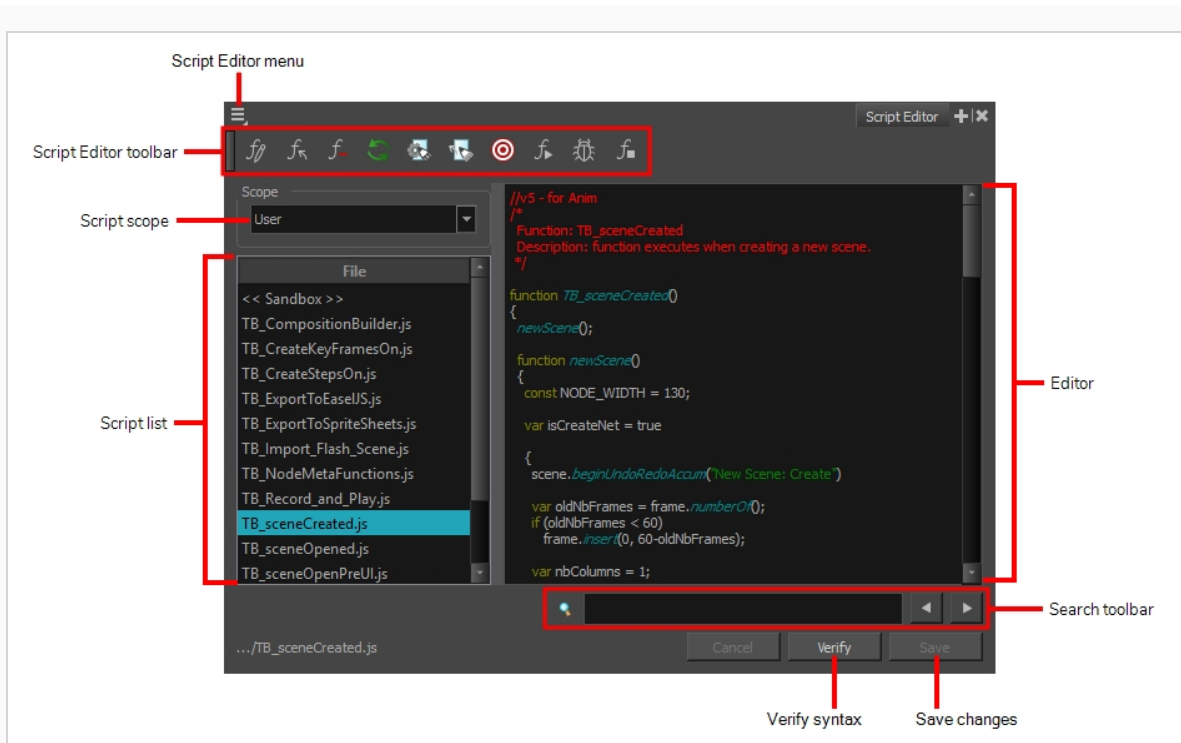
When using Harmony Server, user scripts are stored in the following location, depending on the level at which you save them:

- **User:** `/USA_DB/users/[username]/Harmony [edition]/1500-scripts`
- **Job:** `/USA_DB/jobs/[job_name]`
- **Environment:** `/USA_DB/environments/[environment name]`
- **Global:** `/USA_DB/scripts`



## How to create a script

1. In the top-right corner of the Camera or Drawing view, click on the **+** Add View button and select **Script Editor**.

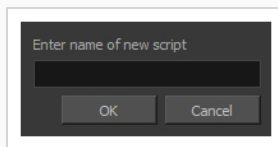
The Script Editor view is added to your workspace. It lists all the scripts packaged with Harmony as well as any script you previously created.



2. Do one o the following:

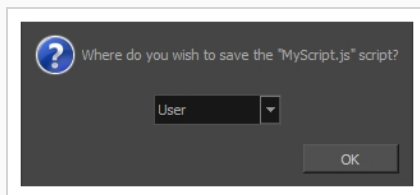
- In the Script Editor toolbar, click the  New Script button.
- From the top-left corner of the Script Editor, click on the  Menu button, and select **File > New Script**.

The New Script dialog box appears.



3. Type in a name for your script and click on **OK**.

A dialog prompts you to select the location in which to create the script.



4. In the drop-down menu, select the location in which you want to create the script:

- **User:** Saves the script to your user preferences. You will be able to run and edit the script from any scene as long as you are logged in as the same user.



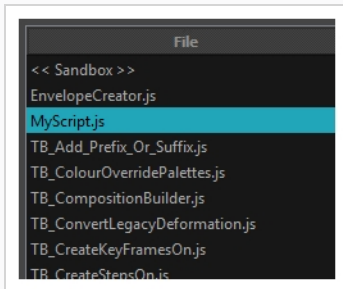
#### NOTE

If you are using Harmony Stand Alone, the script will be saved to the preferences for your account on your computer. If you are using Harmony Server, the script will be saved to the preferences for your Harmony database user.

- **Global:** If you are using Harmony Server, select this option to save the script to the Harmony database's global script directory. Anyone who is logged in to the database will be able to edit and run this script.
- **Environment:** If you are using Harmony Server, select this option to save the script to the current scene's environment. Anyone will be able to edit and run the script as long as they are working on a scene inside this environment.
- **Job:** If you are using Harmony Server, select this option to save the script to the current scene's job. Anyone will be able to edit and run the script as long as they are working on a scene inside this job.
- **Scene:** Save the script to the scene's **scripts** sub-folder. Anyone will be able to edit and run the script when working on this specific scene.

Your script is added to the list of scripts in the Script Editor.

5. Select the script you just created in the list.



You can now start editing your script in the script editor.


6. Click on the text field in the right area of the Script Editor and start writing your script.

The following is an example of a script that will display a message box saying "Hello World!"

```
function HelloWorld()
{
```

```
MessageBox.information("Hello World!");  
}
```

**NOTE**

For information on how to use the Qt scripting language in Harmony, see [Scripting Reference on page 6](#). You can also open the scripting reference guide directly from the Script Editor by clicking on the  Menu button in the top-left corner and selecting **Help > Scripting Interface Documentation**.

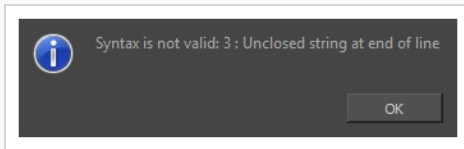
## Chapter 2: Checking Script Syntax

You can verify the syntax of your script before you attempt to run it.


### How to verify the syntax of a script




1. In the **File** list of the Script Editor, select the script you want to check.
2. Under the editor, click on the **Verify** button.

A dialog appears and displays whether or not it found a syntax error in the script, and if it did, the location and type of error.






## Chapter 2: Testing a Script

You can test a script using the  Run command, which is available in the Script Editor toolbar and menu. The run command will execute a single function in one of your scripts. The function it runs is called the target.



To select which function in which script the Run command should execute, you must use the  Set Target command, which will prompt you to select a script and a function. Once you've done this, every time you use the  Run command, it will execute the function you set as your target. If you need to test a different function or script later, you will have to use the  Set Target command again to change the target script and function. This allows you to work on and repeatedly test one function at a time when working on a script.



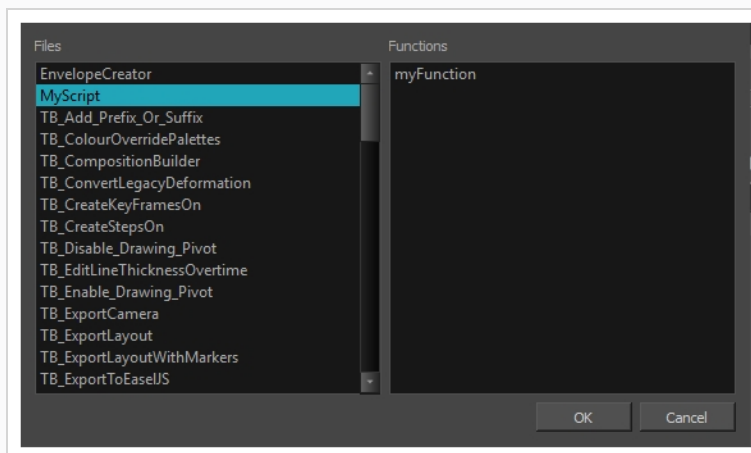
### NOTES

- If you use the  Run command without setting a target first, you will be prompted to set a target. The selected target will persist until you use the  Set Target command to change it.
- The target script and function is also used by the  Debug command.



### How to test a script

1. If you are currently editing the script you want to test, save it by clicking on the **Save** button underneath the editor.
2. Do one of the following:
  - In the Script Editor toolbar, click on the  Set Target button.
  - In the top-left corner of the Script Editor, click on the  Menu button and select **Play/Debug > SetTarget**.

A dialog appears listing all the scripts as well as all the functions in the current script.



3. In the **Files** list of the dialog, select the script you want to test.


4. In the **Functions** list of the dialog, select the function you want to test.
5. Click on **OK**.
6. Do one of the following to execute the target function:
  - In the Script Editor toolbar, click on the  Run button.
  - In the top-left corner of the Script Editor, click on the  Menu button and select **Play/Debug > Run**.






# Chapter 3: Debugging Scripts

T-HSCP-001-003




You can use the Debug command in the Script Editor to troubleshoot errors and problems in your scripts.

You can debug a script using the  Debug command, which allows you to run a function through a standard script debugging interface.



To select which function in which script the Debug command should debug, you must use the  Set Target command, which will prompt you to select a script and a function. Once you've done this, every time you use the  Debug command, it will debug the function you set as your target. If you need to debug a different function or script later, you will have to use the  Set Target command again to change the target script and function. This allows you to work on and repeatedly debug one function at a time when working on a script.



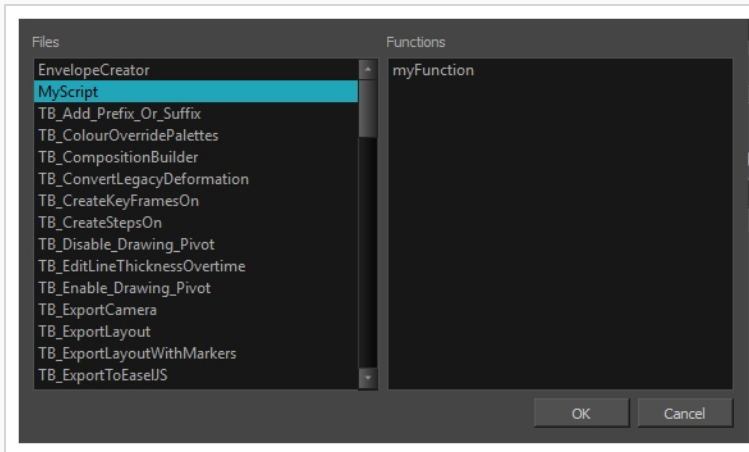
## NOTES



- If you use the  Debug command without setting a target first, you will be prompted to set a target. The selected target will persist until you use the  Set Target command to change it.
- The target script and function is also used by the  Run command.

## How to debug a script

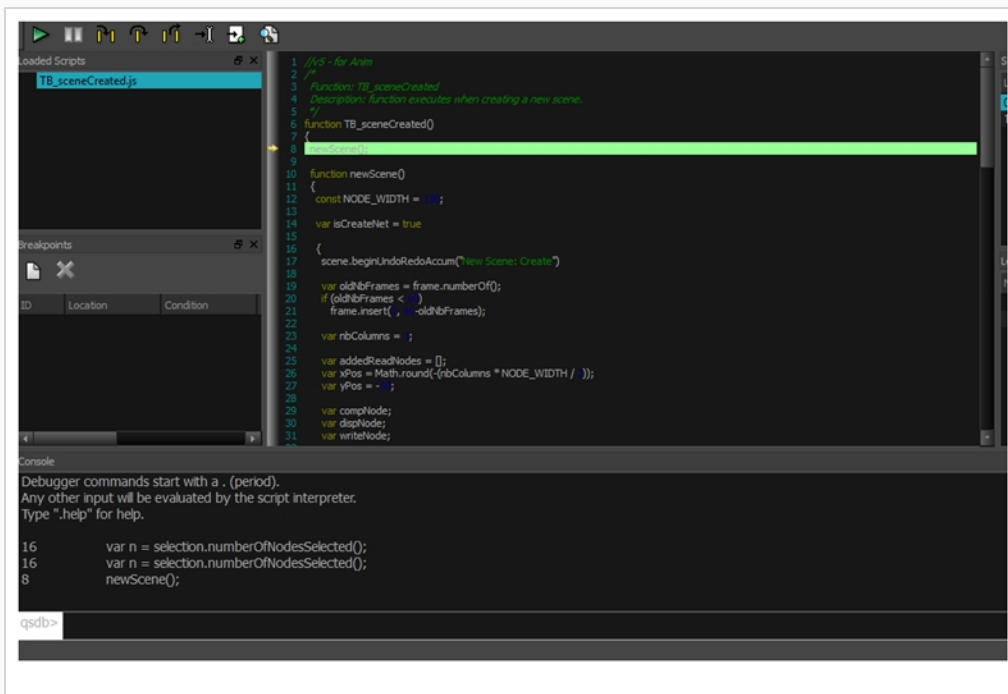
1. If you are currently editing the script you want to debug, save it by clicking on the **Save** button underneath the editor.
2. Do one of the following:
  - In the Script Editor toolbar, click on the  Set Target button.
  - In the top-left corner of the Script Editor, click on the  Menu button and select **Play/Debug > SetTarget**.

A dialog appears listing all the scripts as well as all the functions in the current script.



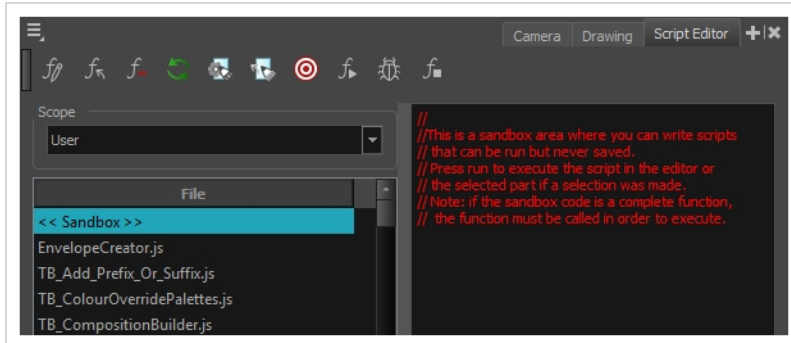
3. In the **Files** list of the dialog, select the script you want to debug.
4. In the **Functions** list of the dialog, select the function you want to debug.
5. Do one of the following to execute the target function:
  - In the Script Editor toolbar, click on the  Run button.
  - In the top-left corner of the Script Editor, click on the  Menu button and select **Play/Debug > Debug**.

The Qt Script Debugger opens. In it, you will find standard debugging controls.



## Chapter 3: Using the Scripting Sandbox

The sandbox is a space in the Script Editor in which you can type or paste code snippets and test or debug them on the fly, without having to put them in the context of a function, and without having to create a script or to modify an existing script. It's also a useful tool to get yourself familiarized with the Harmony scripting interface.







### How to use the Script Editor sandbox

1. In the File list, select <<<Sandbox>>>.
2. In the text field of the script editor, type in your code.

The following sample code will display a message box saying "Hello World!"

```
MessageBox.information("Hello World!");
```

3. Do one of the following:
  - To verify the syntax of your script, click on the **Verify** button under the text field.
  - To test your script, click on the  Run button in the Scripting toolbar or, in the top-left corner of the Script Editor, click on the  Menu button and select **Play/Debug > Run**.
  - To debug your script, click on the  Debug button in the Scripting toolbar or, in the top-left corner of the Script Editor, click on the  Menu button and select **Play/Debug > Debug**.



# Chapter 4: Importing Scripts

T-ANIMPA-007-002

To use scripts prepared by other users, transfer the `.js` files from the computer where the scripts were created to the one that you intend to use. Then use the Script Editor to import the `.js` files from where they are saved on your computer.

When working with Harmony Stand Alone, scripts that you created while working on a scene or imported are available for all other projects created on the same computer; there's no need for any kind of transfer or import. You can find your custom, default and imported scripts in:



## NOTE


When using Harmony Stand Alone, user scripts are stored in the following location:

- **Windows:** `C:\Users\[username]\AppData\Roaming\Toon Boom Animation\Toon Boom Harmony [edition]\1500-scripts`
- **macOS:** `/Users/[username]/Library/Preferences/Toon Boom Animation/Toon Boom Harmony [edition]/1500-scripts`
- **GNU/Linux:** `/home/[username]/Toon Boom Animation/Toon Boom Harmony [edition]/1500-scripts`

When using Harmony Server, user scripts are stored in the following location, depending on the level at which you save them:

- **User:** `/USA_DB/users/[username]/Harmony [edition]/1500-scripts`
- **Job:** `/USA_DB/jobs/[job_name]`
- **Environment:** `/USA_DB/environments/[environment name]`
- **Global:** `/USA_DB/scripts`

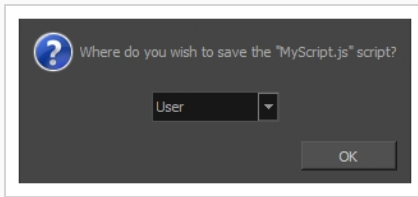
## How to import scripts

1. Do one of the following:
  - In the Script Editor toolbar, click on  Import Script.
  - In the top-left corner of the Script Editor view, click on the  Menu button and select **File > Import Script**.

An Open dialog appears.

2. In the Open dialog, browse to the location of the script you want to import. Select it, then click on **Open**.

A dialog prompts you to select the location in which to import the script.



3. In the drop-down menu, select the location in which you want to import the script:
  - **User:** Saves the script to your user preferences. You will be able to run and edit the script from any scene as long as you are logged in as the same user.



#### NOTE

If you are using Harmony Stand Alone, the script will be saved to the preferences for your account on your computer. If you are using Harmony Server, the script will be saved to the preferences for your Harmony database user.

- **Global:** If you are using Harmony Server, select this option to save the script to the Harmony database's global script directory. Anyone who is logged in to the database will be able to edit and run this script.
  - **Environment:** If you are using Harmony Server, select this option to save the script to the current scene's environment. Anyone will be able to edit and run the script as long as they are working on a scene inside this environment.
  - **Job:** If you are using Harmony Server, select this option to save the script to the current scene's job. Anyone will be able to edit and run the script as long as they are working on a scene inside this job.
  - **Scene:** Save the script to the scene's **scripts** sub-folder. Anyone will be able to edit and run the script when working on this specific scene.
4. Click on **OK**.

The script is copied to the chosen location and can now be edited and executed in the Script Editor view. It can also be added as a button to the Scripting toolbar—see [Adding Scripts as Toolbar Buttons on page 28](#).

# Chapter 5: Deleting Scripts



You can delete scripts from the list of scripts file in the Script Editor.



## IMPORTANT

This will permanently delete the script file containing the script from your file system.

## How to delete a script

1. In the Script Editor, in the File list, select the script you want to delete.
2. Do one of the following:
  - From the Script Editor toolbar, click the  Delete Script button.
  - In the top-left corner of the Script Editor view, click on the  Menu button and select **File > Delete Script**.



## TIP



If you have deleted one of the scripts that are packaged with Harmony and wish to restore it, see [Restoring a Harmony Script](#) on page 23

## Chapter 5: Restoring a Harmony Script

If you deleted one of the scripts that was packaged with Harmony, or if you wish to restore a script that was packaged with Harmony to its original version after you made changes to it, you can simply re-import it from the Harmony installation directory

### How to restore a script packaged with Harmony to its original version

1. Do one of the following:

- In the Script Editor toolbar, click on  Import Script.
- In the top-left corner of the Script Editor view, click on the  Menu button and select **File > Import Script**.

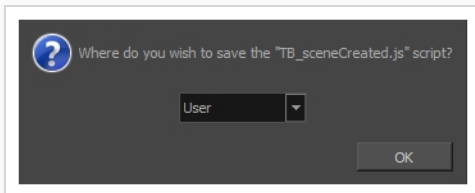
An Open dialog appears.

1. In the Open dialog, browse to the following location, depending on your operating system:

- **Windows:** `C:\Program Files (x86)\Toon Boom Animation\Toon Boom Harmony 15.0 [Edition]\resources\scripts`
- **macOS (Stand Alone):** `/Applications/Toon Boom Harmony 15.0 [Edition]/tba/resources/scripts`
- **macOS (Server):** `/Applications/Toon Boom Harmony 15.0 [Edition] Network/tba/resources/scripts`
- **GNU/Linux:** `/usr/local/ToonBoomAnimation/harmony[Edition]_15.0/resources/scripts`

2. Select the script file that you want to restore.
3. Click on **Open**.

A dialog prompts you to select the location in which to import the script.



4. In the drop-down menu, select the location in which you want to import the script:

- **User:** Saves the script to your user preferences. You will be able to run and edit the script from any scene as long as you are logged in as the same user.



**NOTE**

If you are using Harmony Stand Alone, the script will be saved to the preferences for your account on your computer. If you are using Harmony Server, the script will be saved to the preferences for your Harmony database user.

- **Global:** If you are using Harmony Server, select this option to save the script to the Harmony database's global script directory. Anyone who is logged in to the database will be able to edit and run this script.
  - **Environment:** If you are using Harmony Server, select this option to save the script to the current scene's environment. Anyone will be able to edit and run the script as long as they are working on a scene inside this environment.
  - **Job:** If you are using Harmony Server, select this option to save the script to the current scene's job. Anyone will be able to edit and run the script as long as they are working on a scene inside this job.
  - **Scene:** Save the script to the scene's **scripts** sub-folder. Anyone will be able to edit and run the script when working on this specific scene.
5. Click on **OK**.

The original Harmony script is copied to the chosen location and can now be edited and executed in the Script Editor view.



# Chapter 6: Using an External Script Editor

T-HSCP-001-004



If you are more comfortable editing your scripts using another text editor or syntax highlighter, Harmony allows you to set an external script editor and use that one instead.



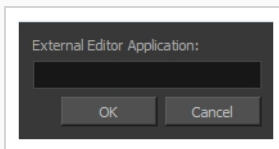
## NOTE

On Windows and macOS, if you try to open a script in the external editor without setting it first, Harmony will open the script in the default editor for JavaScript (.js) files on your machine.

## How to set an external editor

1. Do one of the following:
  - In the Script Editor toolbar, click the  Set External Editor button.
  - In the top-left corner of the Script Editor view, click on the  Menu button and select **Editor > Set External Editor**.

The Set External Editor window opens.



2. In the external editor, type the complete and absolute path to the text editing application you want to use, including its name and extension. For example, if you want to use notepad++ on Windows, you could type:

**C:\Program Files\Notepad++\notepad++.exe**

On macOS, if you want to use an application that supports the AppleScript Open Document protocol, simply typing the name of the application might work, for example:



**TextWrangler**

Typing the name of the application you want to use will also work if that application has its path in the **PATH** environment variable, such as notepad on Windows or gedit on GNU/Linux.

3. Click on **OK**.

## How to edit a script in an external editor

1. In the File list, select the script you want to edit.

2. Do one of the following:
  - In the Script Editor toolbar, click the  Open with External Editor button.
  - In the top-left corner of the Script Editor view, click on the  Menu button and select **Editor** > **External Editor**.
3. In the external editor that opens, make the desired changes to your script, save the script and close the external editor.
4. In Harmony, the changes you made to your script in the external editor will not appear immediately. You can load those changes by selecting another script in the File list, then selecting the script you changed again.

## Chapter 7: Adding Scripts as Toolbar Buttons

To use scripts in your regular workflow, you can add them as buttons to the Scripting toolbar. You can add buttons to the Scripting toolbar that are configured to run a specific function in a script and, if you want, give these button a specific icon and tool tip.



### NOTE


For more information on the Scripting toolbar and the Scripts Manager dialog, see the Reference Guide.

### How to link a script to a toolbar button

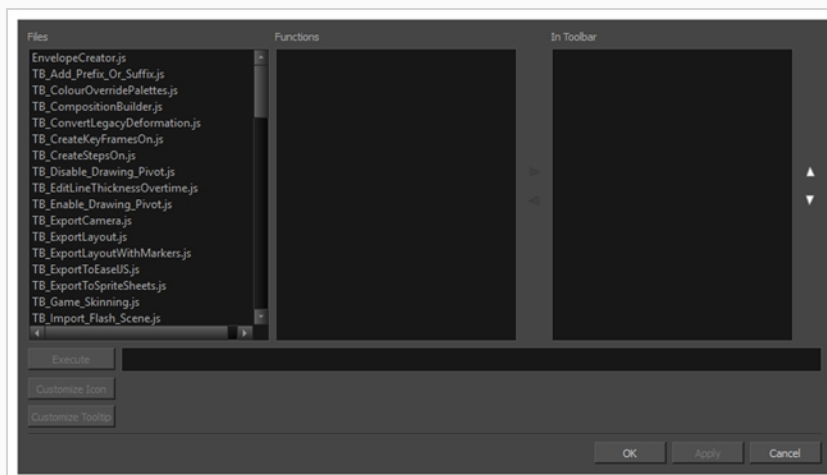
1. Add the Scripting toolbar to your workspace by doing one of the following:
  - From the top menu, select **Windows > Toolbars > Scripting**.
  - Right-click on any toolbar and select **Scripting**.

The Scripting toolbar is added to your workspace.



1. In the Scripting toolbar, click the  Manage Scripts button.

The Scripts Manager dialog box opens and displays the available scripts.



2. In the **Files** list, select the script file containing the function you want to add to the toolbar.  
The functions in the selected script appear in the **Functions** list.
3. In the **Functions** list, select the function that you want to add to the toolbar.

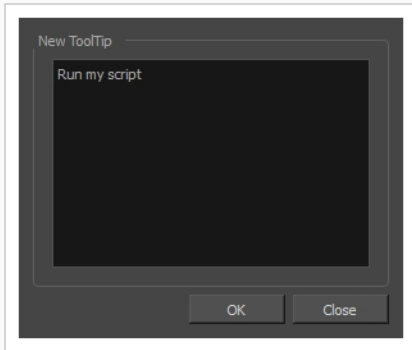
4. Between the **Functions** and **In Toolbar** list, click on the ► Right Arrow button.

The selected function is added to the **In Toolbar** list.

5. If you want to add a custom icon to the toolbar button for your script, click on the **Customize Icon** button and, in the Open dialog that appears, browse to and select the icon file that you want the button to have.

You can use **.png**, **.jpg**, **.xpm** or **.svg** files as toolbar icons.

6. If you want to add a custom tooltip to the toolbar button for your script, click on the **Customize Tooltip** button and, in the dialog that appears, type in the tooltip for your toolbar button.




7. Click on **OK**.

A button for the selected function is added to the Scripting toolbar. If you click on this button, the function your selected will be executed.



**TIP**



You can stop a script while it is running by pressing the Esc key or clicking on the  Stop button on the Scripting toolbar.

# Chapter 8: Exposing Node Attribute Names for Scripting

If you want to make scripts that manipulate nodes in your scene, you will need to use the real names of the attributes for your nodes as understood by the Harmony scripting interface. For example, to refer to the position of a peg node on the (x) axis, you would need to refer to the real name of the corresponding attribute in your script, which is `position.x`.

You can easily obtain the real names of a node's attributes by exposing them in the Layer Properties dialog. To do this, you must enable Publish Attribute Mode. On top of allowing you to add certain attributes in a node to the list of attributes in its parent group's properties, Publish Attribute Mode also displays the name of each attribute as understood by the Harmony scripting interface.

## How to see the real names of a node's attributes for scripting

1. In the top-left corner of the Node view, open the View Menu  and select **View > Publish Attribute Mode**.
2. In the Node view, locate the node for which you want the list of scriptable attributes.
3. Click on the Layer Properties  button for this node.

The Layer Properties dialog for this node appears. Right of each attribute is the option to publish

them and to set their published name, and then the name of the attribute in the scripting interface:

