HARMONY

Toon Boom Harmony 11.1
Scripting Guide

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

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

Toon Boom® Harmony™

Publication Date

11/29/2018

Contents

Chapter 1: Introduction

In the Scripting guide you will find information which helps you to understand how to use Qt Script, the scripting language supported by Toon Boom Harmony.

Detailed Scripting Method

For all details about the scripting modules, methods, and DbScript documentation, click on the links below.

- [Harmony Script Interface Documentation](#)
- [Script Module Documentation](#)
- [DbScript Documentation](#)

Move from QSA to QT Script

Beginning with this release of Toon Boom Harmony 10.3, Toon Boom Animation has changed the scripting language.

- Scripting language used in Harmony 11.1**: Qt script**
- Reason for changing the scripting language: **better and faster scripting environment**

Toon Boom Animation has changed the scripting language to Qt Script from QSA used in earlier versions. Script's storage is now done as plain text, `*.js` instead of `*.qsa` format.

This change to the api was made in the interests of improving the software so that it could support 64-bit platforms. As always, we have made every effort to enable backward compatibility between this version and earlier versions which used QSA script.

- Simple functions should port seamlessly.
- Class definitions must follow the new QT Script syntax, so that existing scripts may be modified.

Detailed instructions on moving QSA script to Qt Script can be found in the **"Moving from QSA to Qt Script"** documentation at this link:

- http://qt-project.org/doc/qt-4.8/porting-qsa.html

For a general overview of the QT UI toolkit, refer to the following:

- http://qt-project.org/doc/qt-4.8/index.html

For a tutorial on coding in JavaScript, refer to the following:

- http://docs.oracle.com/javase/tutorial/java

Chapter 2: Scripting Overview

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

To use scripts prepared by other users, you must import those scripts into your preferences.

You can add buttons to the Scripting Tools toolbar so that you can access them easily. They will appear to the right of the default Scripting Tools toolbar buttons.

Using Qt Script to Automate Toon Boom Harmony Functions

Qt Script provides access to many of the functionalities supported in the Harmony interface.

Qt Script is an object-oriented scripting language based on the ECMAScript standard, like JavaScript and JScript. However, there are some differences that distinguish it from these scripting languages, which are familiar to web programmers.

If you are working in a network environment, refer to the toHarmony Server Network Environment

Creating a Qt Script

Use the Script Editor to create Qt scripts for Harmony.

For all details about the scripting modules, methods, and DbScript documentation, click on the links below.

- Harmony Script Interface Documentation

- Script Module Documentation

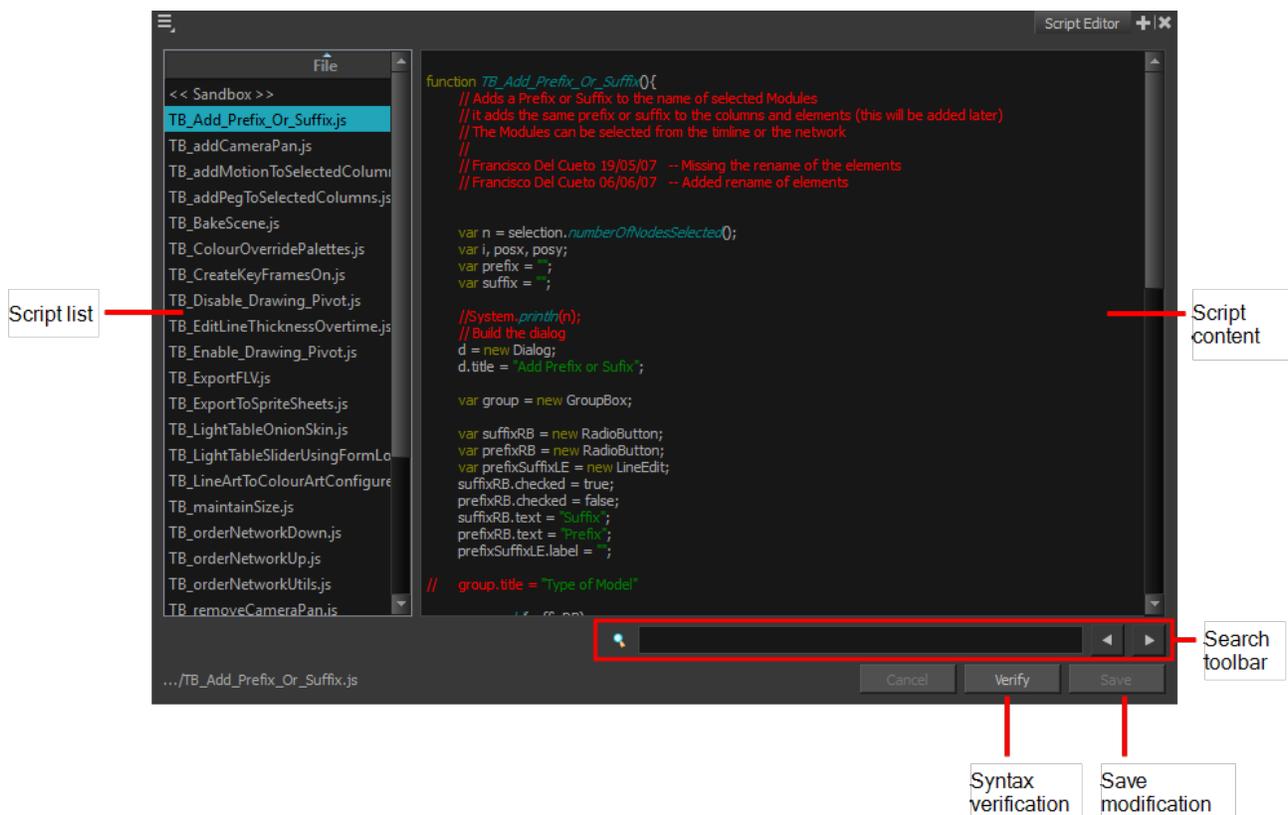- DbScript Documentation

This section is divided as follows:

- *Creating a Script* on page 7

- *Deleting a Script* on page 10

- *Troubleshooting a Script* on page 10

# Creating a Script

To create a script:

1. Open the Script Editor by going to the top menu and selecting **Windows > Script Editor**.
   The Script Editor view opens listing all the JavaScript files.

2. From the view menu, select **File > New Script** or from the Script Editor toolbar, click on the New Script button. (If you do not see the Script Editor toolbar, go to the top menu and select **Windows > Toolbars > Script Editor**).

The New Script dialog box appears.
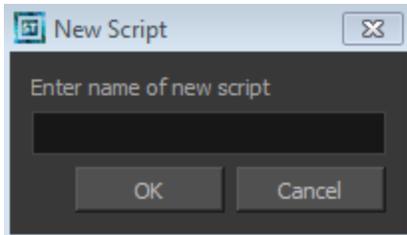
3. Enter a name in the Enter name of new script field and click on **OK**.

The name of your script appears in the File column of the Script Editor view.
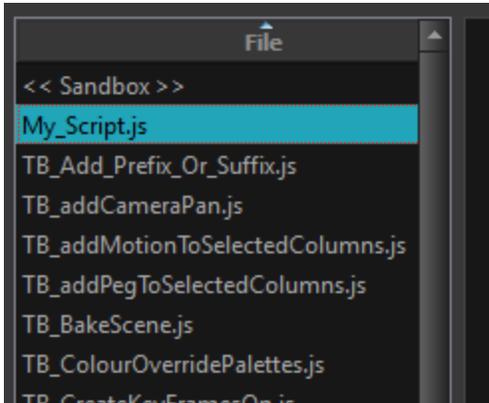
4. Click in the right side of the Script Editor and start writing your script. Try typing in the following script:

```
function add3dPathCol()

{

column.add("ColumnName", "3DPATH");

}
```

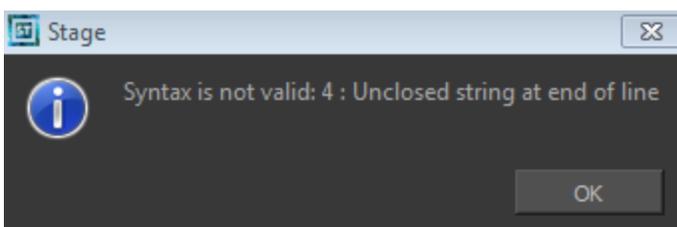For a tutorial on coding in JavaScript, refer to the following:

- http://docs.oracle.com/javase/tutorial/java

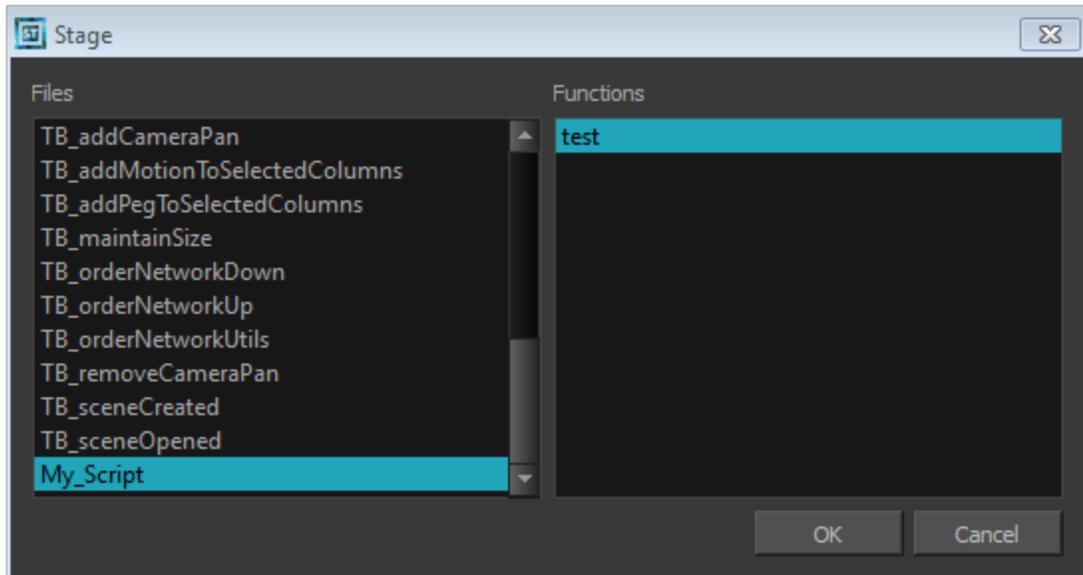For a detailed Harmony script interface documentation, refer to the following:

- Harmony Script Interface Documentation

5. To check your syntax, click on the **Verify** button.

A window opens with syntax information.

8

6. To test your script, select the script you would like to run from the File column, then, from the Script Editor toolbar, click on the **Run** ⨍ button, or from the view menu select **Play/Debug > Run**.

‣ A window opens to ask you for the target function to run. In the window, select which function you want to execute. To avoid selecting the function every time you want to run your script, you can set a target function. In the Script Editor toolbar, click on the **Set Target** ◎ button, or from the view menu select **Play/Debug > Set Target**. In the Function column, select the function to target and press OK.

‣ To stop the execution of your script, click on the **Stop Execution** ⨍ button.

‣ You may receive the message: Only functions that have been saved or are explicitly called will be executed. This usually occurs if you did not select the function you wished to run from the File column before running the script. In this case, click on the **Save Changes** button, then try running your script again.

‣ You may also see this message on the right side of the Script Editor:
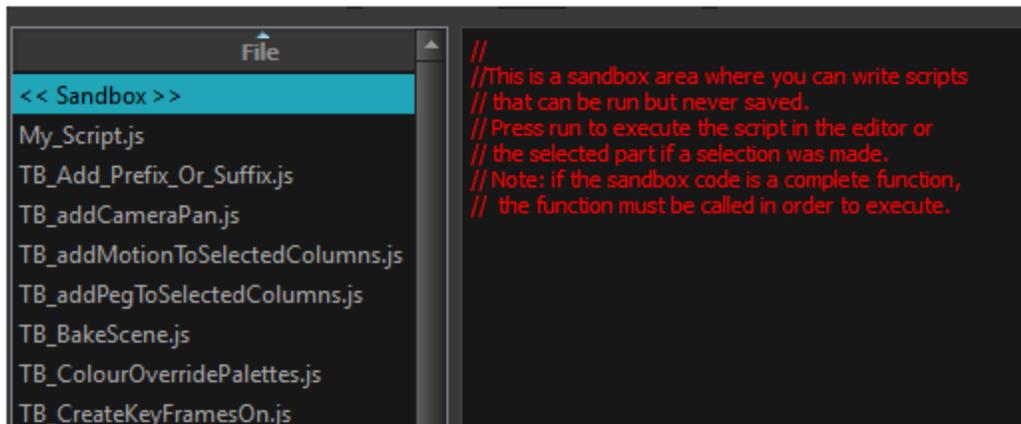
This is because the software jumped to the <<Sandbox>> item in the Function column. The <<Sandbox>> is provided to test scripts without saving them. Simply click on your newly created script and then try running it again.

Deleting a Script

How to delete a script

1. In the Script Editor, in the File column, select the script that you would like to delete.
2. From the Script Editor toolbar, click on the Delete Script ![f-] button or from the view menu, select **File > Delete Script**.

Troubleshooting a Script
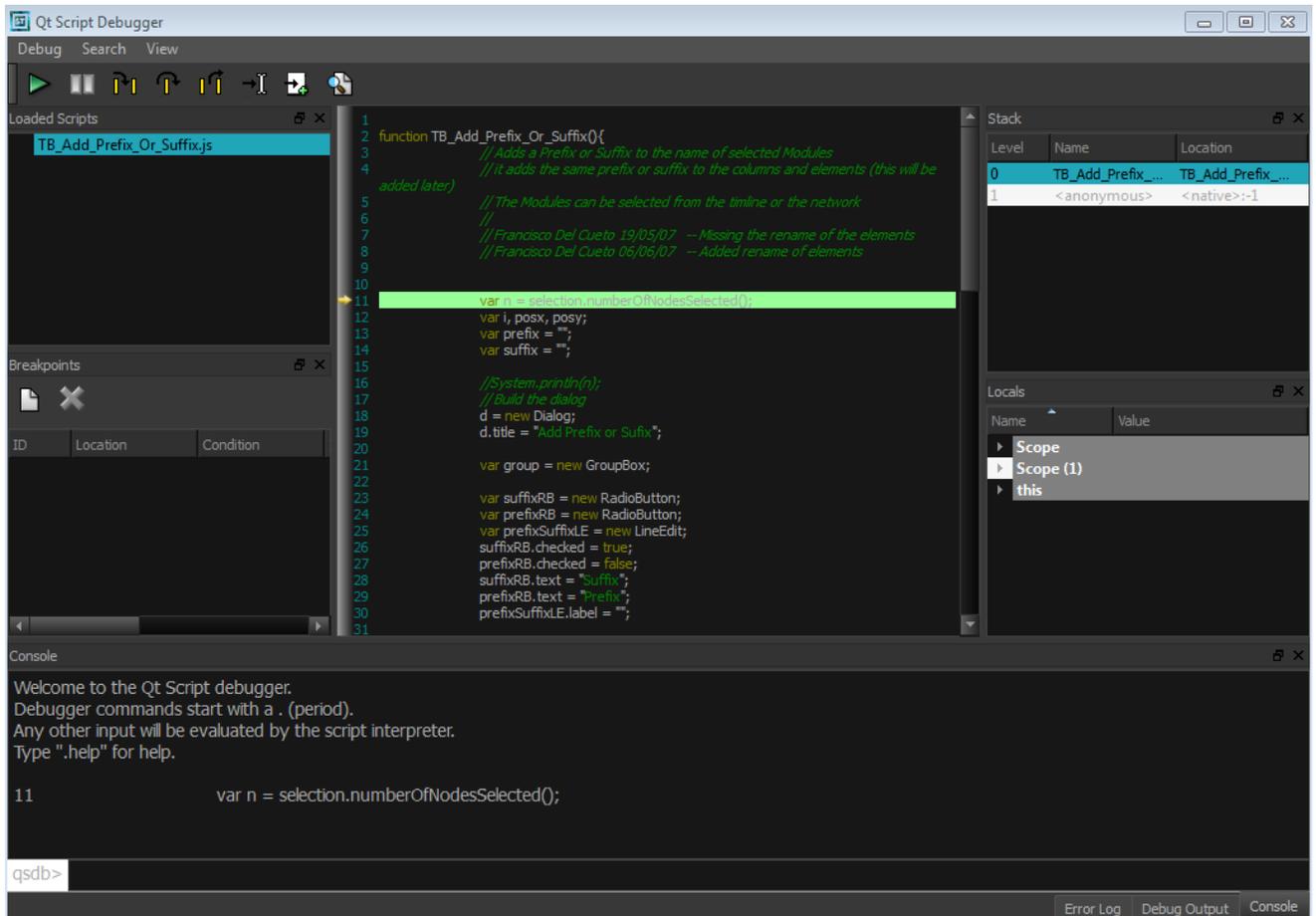
How to troubleshoot errors in your script

1. Debug: In the File list, select the script containing the function to debug.From the Script Editor toolbar, click on the Debug ![bug] button, or from the view menu, select **Play/Debug > Debug**.

    A window opens listing the scripts and functions available.
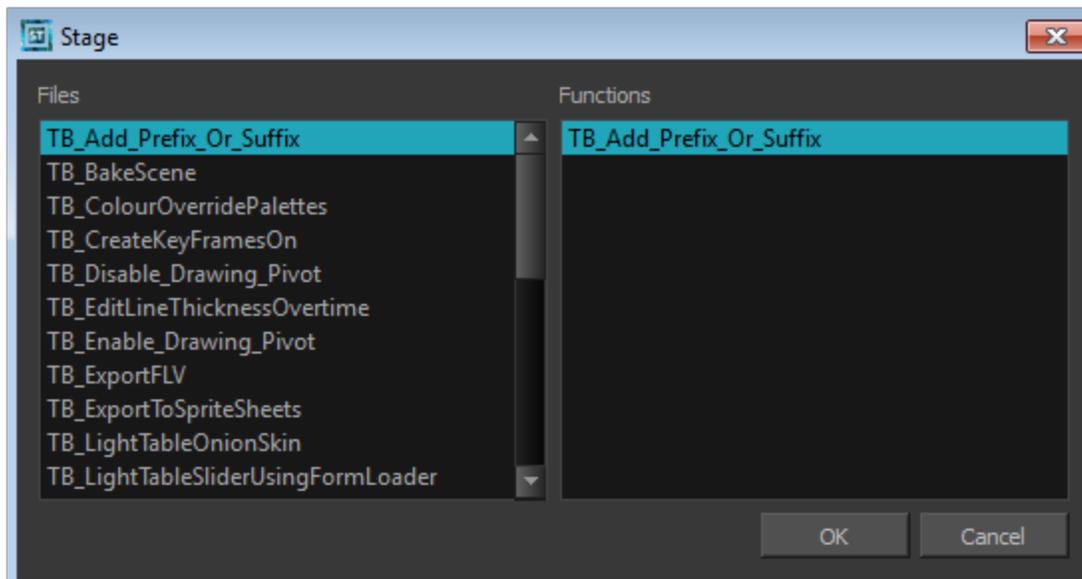
2. In the Functions column, select the function to debug.
3. Click **OK**.

The Qt Script Debugger opens.

If you are working in a Network Environment, see *Creating a Qt Script* on page 16

Using an External Script Editor

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

> If you do not set an external editor, on Windows and Mac OS X, when selecting the External Editor option, it will launch the registered editor for JavaScript program. (e.g. Visual Studio , XCode or Brackets).
>
> Multiple files can be opened in the same editor, instead of starting a different copy of the editor each time a `*.js` script is selected.

To set an external editor:

1. In the Script Editor view, in the View menu, select **Editor > Set External Editor**. In the Script Editor window, click on the **Set External Editor** button.

   The Set External Editor window opens.
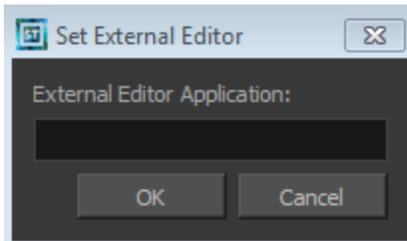
2. In the External Editor Application, type or paste the complete path to the application, including its name and extension.

   ‣ For example, if you want to set Adobe Dreamweaver as your external editor, a Windows path would look like this:
   `C:\Program Files (x86)\Adobe\Adobe Dreamweaver CS5\Dreamweaver.exe`

   ‣ On Mac OS X, simply specify the application name, and if this application supports the Apple SCript Open Document protocol, it will open in this app.

3. Click on the **OK** button.

To edit a script in an external editor:

1. In the Script Editor view, in the File list, select the script to edit.

2. In the View menu, select **Editor > External Editor** or in the Script Editor window, click on the External Editor button.

   The external editor opens.

3. In the external editor, edit the script and save the file.

4. In Harmony, refresh the Script Editor view by selecting another script and selecting your edited script again to see the modifications.

Importing Scripts

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.

Scripts that you have created while working on a scene or project will be available for all other projects created on the same computer, with no need for any kind of transfer or import.

To import a script:

1. On the computer that contains the `*.js` files, open My Computer (Windows) or Finder (Mac OS X), then follow this path:

   To the default scripts:

   ‣ Windows: **C:\Program Files (x86)\Toon Boom Animation\Toon Boom Harmony 11.1\resources\scripts**

   ‣ Mac OS X: **Home > Library > Preferences > Toon Boom Animation > Toon Boom Harmony > 1110-scripts**

   ‣ Linux: **/Applications/Toon Boom Harmony 11.1/tba/resources/scripts**

   To your custom scripts:

   ‣ Windows: **C:\Users\(User Name)\AppData\Roaming\Toon Boom Animation\Toon Boom Harmony\1030-scripts**

   ‣ Mac OS X: **/home/(User Name)/Toon Boom Animation/Toon Boom Harmony/1110-scripts**

   ‣ Linux: **/usr/local/ToonBoomAnimation/harmony_10.3/resources/scripts**

2. Copy the `*.js` files that you would like to transfer to another computer onto a USB key, or use another transfer method of your choice.

3. Transfer and save the `*.js` files onto the computer that you intend to use. Be sure that you save them in a logical location. You can save them directly in the locations indicated in Step 1, in which case they will automatically appear in the Script Editor. If for some reason you do not wish to mix them up with the default scripts, save them to a different location and continue following the steps below.

4. Open a new, or existing, scene in Harmony.

5. In the Script Editor, click on the Import Scripts ⬚ button, or from the view menu, select **File > Import Script**.A browser window will open and allow you to browse to the location on the server where the `*.js` files are saved.

To save time, you can copy all the files and icons located in the user's preferences and paste them back in your own preferences to avoid having to import all scripts one by one.

- Windows: **C:\Users\(User Name)\AppData\Roaming\Toon Boom Animation\Toon Boom Harmony\1110-scripts**

- Mac OS X: **/home/(User Name)/Toon Boom Animation/Toon Boom Harmony/1110-scripts**

- Linux: **/usr/local/ToonBoomAnimation/harmony_11.1/resources/scripts**

---

6. Select the `*.js` file and click **Open**.

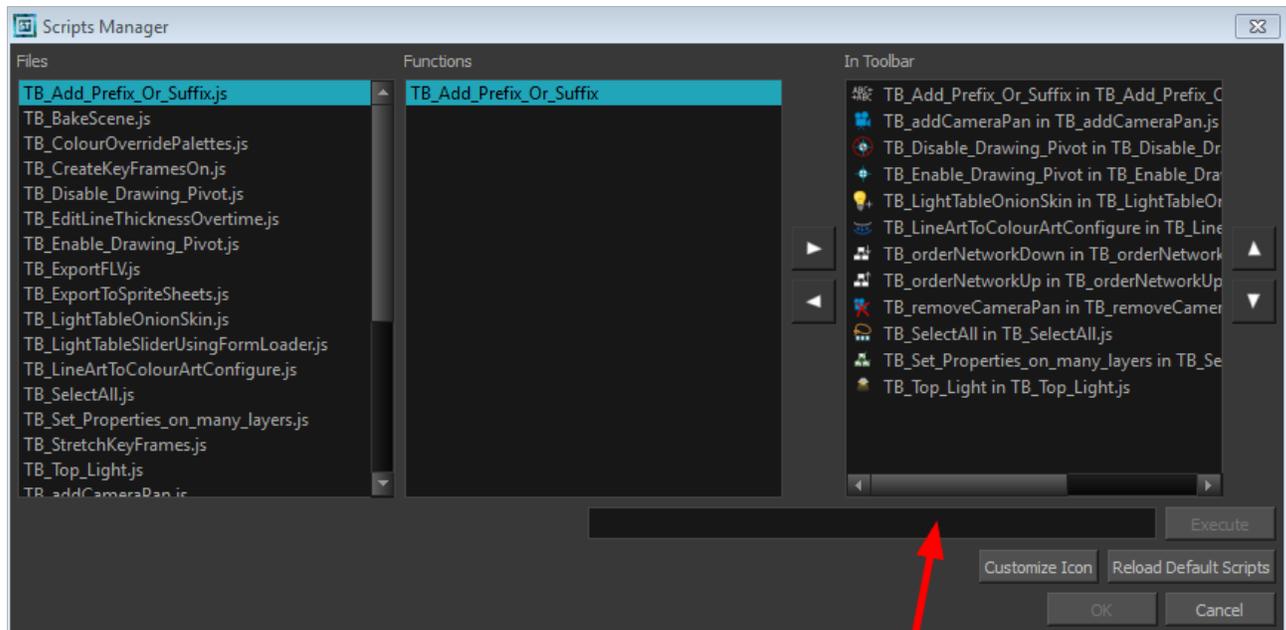   The script is imported and appears in the File list of the Script Editor File list.

If you are working in a network environment, refer to the topic *Importing Scripts* on page 21.

Linking a Script to a Toolbar Button

You can add buttons to the Scripting Tools toolbar so that you can access them easily. They will appear to the right of the default Scripting Toolbar buttons.

To link a script to a toolbar button:

1. In the Scripting Tools toolbar, click on the Manage Scripts f_{x} button.

   A dialog box containing all of the available scripts opens.

You can select a script from this dialog box and run it immediately by clicking on the Execute button. The field displays the name of the currently selected script.

2. In the Files list, select the JavaScript file containing the function you want to add to the toolbar.

3. In the Functions list, select the function to add to the toolbar.

4. Click on the Right Arrow ▷ button to add the function to the In Toolbar list.

5. Click on **OK** to add the button and close the dialog box.

To execute a script:

1. Once the script is added to the Scripting toolbar, you can press on the corresponding script button to execute the action–see *Linking a Script to a Toolbar Button* on page 15.

2. You can press the Esc key to interrupt the execution of a script.

If you are working in a network environment, refer to the topic *Linking a Script to a Toolbar Button* on page 21.

Harmony Server Network Environment

When you are working in a network environment, use these procedures:

Using Qt Script to Automate Toon Boom Harmony Functions - server functions

Qt Script provides access to many of the functionalities supported in the Toon Boom Harmony interface.

Qt Script is an object-oriented scripting language based on the ECMAScript standard, like JavaScript and JScript. However, there are some differences that distinguish it from these scripting languages, which are familiar to web programmers.

Creating a Qt Script

Use the Script Editor to create Qt scripts for Harmony.

For all details about the scripting modules, methods, and DbScript documentation, click on the links below.

- Harmony Script Interface Documentation

- Script Module Documentation

- DbScript Documentation

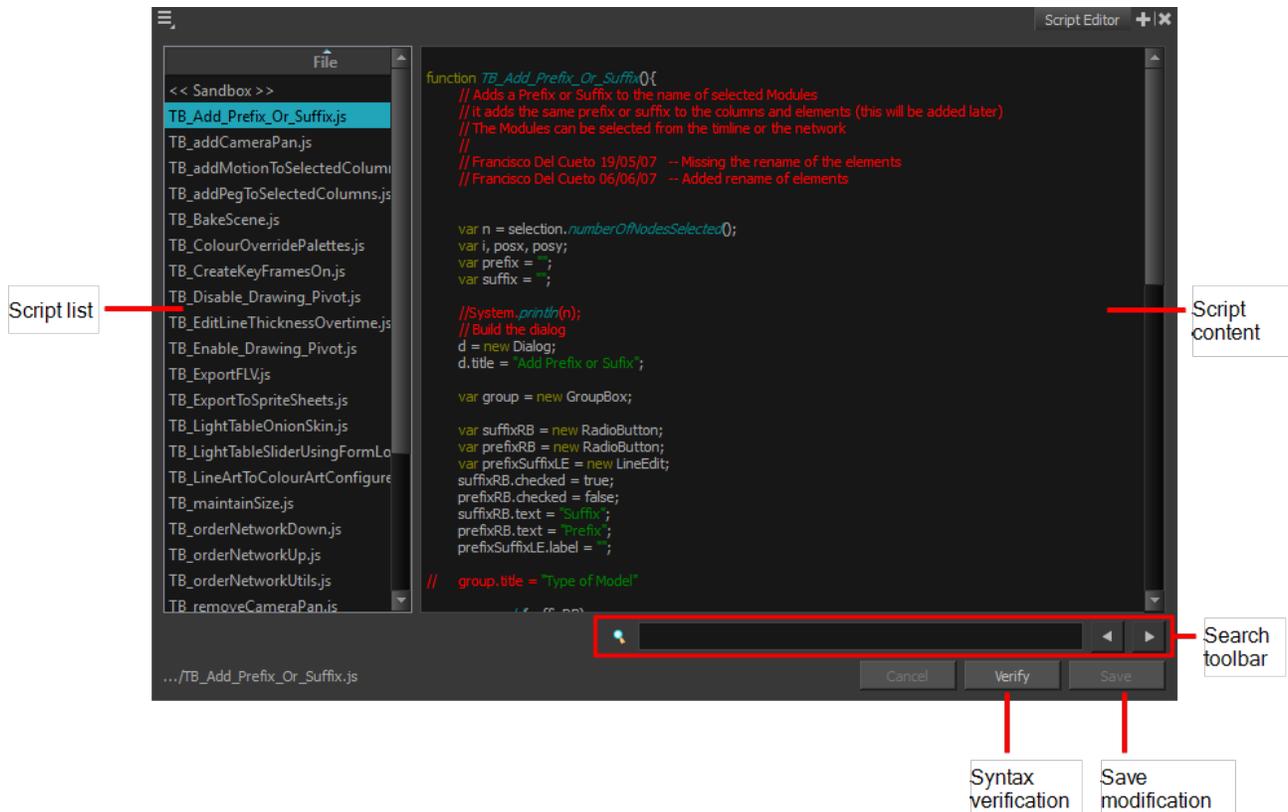This topic is divided as follows:

Creating a Script

How to create a script

1. Open the Script Editor by going to the top menu and selecting **Windows > Script Editor**.
   The Script Editor view opens listing all the JavaScript files.

2. From the view menu, select **File > New Script** or from the Script Editor toolbar, click on the New Script button. (If you do not see the Script Editor toolbar, go to the top menu and select **Windows > Toolbars > Script Editor**).

The New Script dialog box appears.
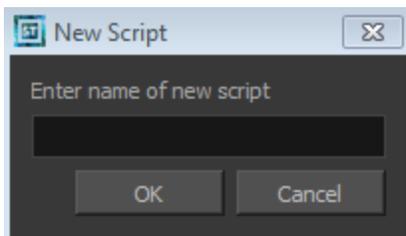
3. Enter a name in the Enter name of new script field and click on **OK**.

The Save Script window opens.
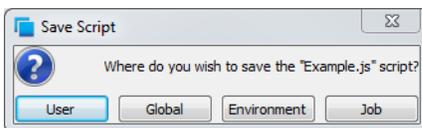
4. To save the script:

▸ Click on the **User** button to store the script into the user profile so the user can always access the script (**/usa_db/users/user_name**).

▸ Click on the **Global** button to make the script available to all users from any scene, job and environment (**/usa_db/scripts**).

‣   Click on the **Environment** button to make the script available to all users when working in a scene from this particular environment (**/usa_db/environments/environments_name**).

‣   Click on the **Job** button to make the script available to all users when working on scenes from a particular job (**/usa_db/jobs/job_name**).

The name of your script appears in the File column of the Script Editor view.
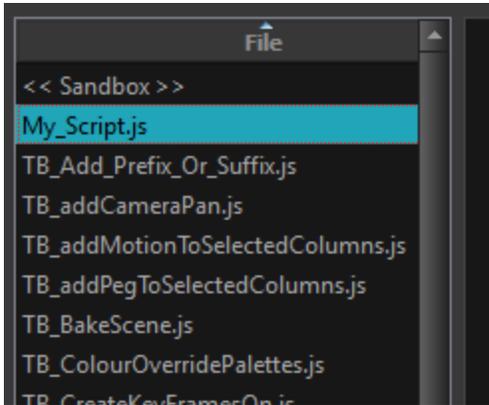
5.   Click in the right side of the Script Editor and start writing your script. Try typing in the following script:

```
function add3dPathCol()

{

column.add("ColumnName", "3DPATH");

}
```

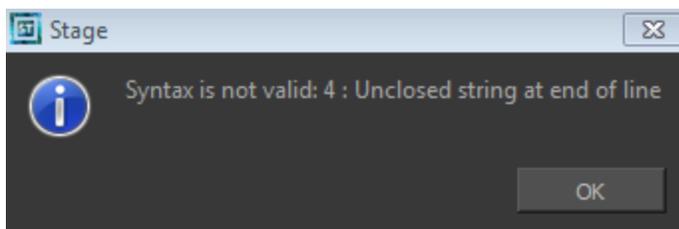For a tutorial on coding in JavaScript, refer to the following:

•   http://docs.oracle.com/javase/tutorial/java

For a detailed Harmony script interface documentation, refer to the following:
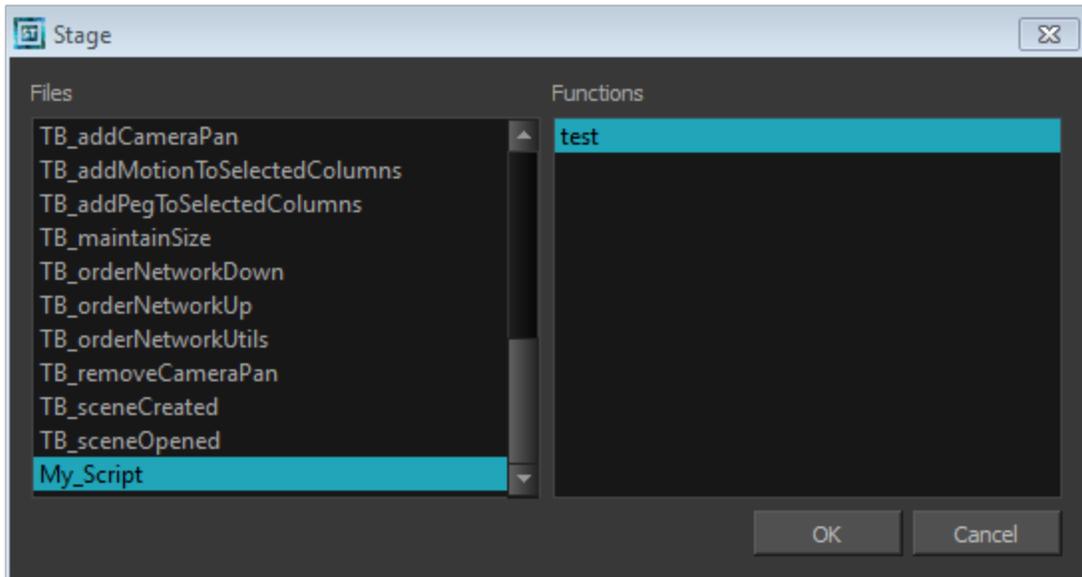
•   Harmony Script Interface Documentation

6.   To check your syntax, click on the **Verify** button.

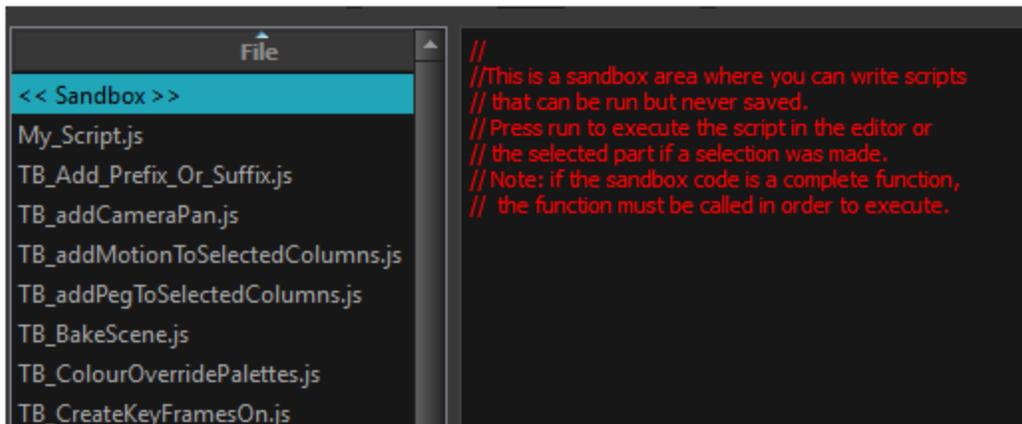A window opens with syntax information.

7.   To test your script, select the script you would like to run from the File column, then, from the Script Editor toolbar, click on the **Run** 🔨 button, or from the view menu select **Play/Debug > Run**.

‣   A window opens to ask you for the target function to run. In the window, select which function you want to execute. To avoid selecting the function every time you want to run your script, you can set a target

function. In the Script Editor toolbar, click on the **Set Target** ◎ button, or from the view menu select **Play/Debug > Set Target**. In the Function column, select the function to target and press OK.

▸ To stop the execution of your script, click on the **Stop Execution** *fx* button.

▸ You may receive the message: Only functions that have been saved or are explicitly called will be executed. This usually occurs if you did not select the function you wished to run from the File column before running the script. In this case, click on the **Save Changes** button, then try running your script again.

▸ You may also see this message on the right side of the Script Editor:

This is because the software jumped to the <<Sandbox>> item in the Function column. The <<Sandbox>> is provided to test scripts without saving them. Simply click on your newly created script and then try running it again.

> You should back up your scripts if they are placed at the Global location because every user opening a scene gets the rights to them and if a problem arises it can corrupt the saved scripts.

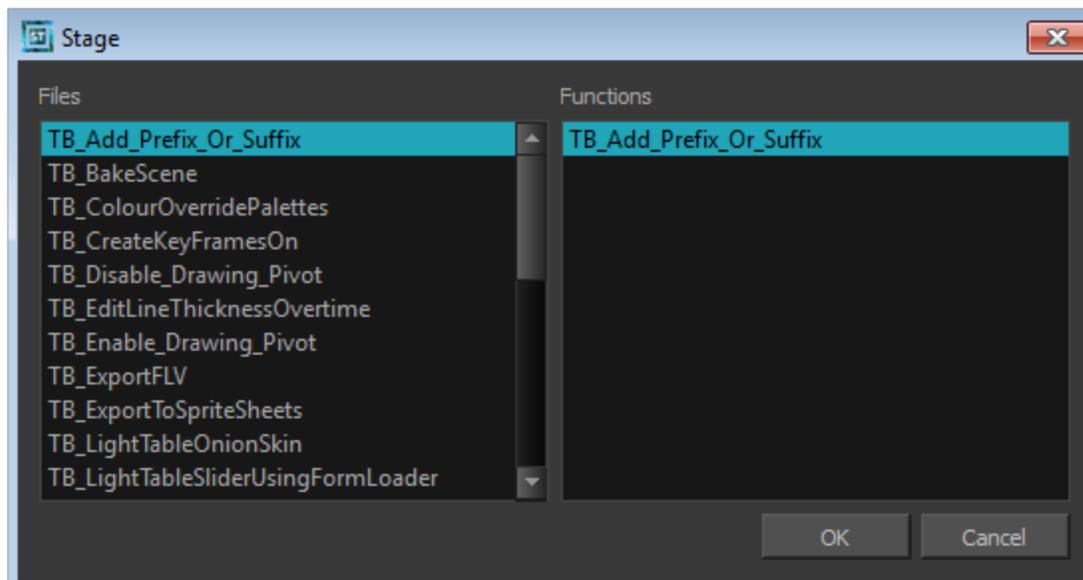# Deleting a Script

How to delete a script

1. In the Script Editor, in the File column, select the script that you would like to delete.
2. From the Script Editor toolbar, click on the Delete Script ![icon] button or from the view menu, select **File > Delete Script**.

Troubleshooting a Script

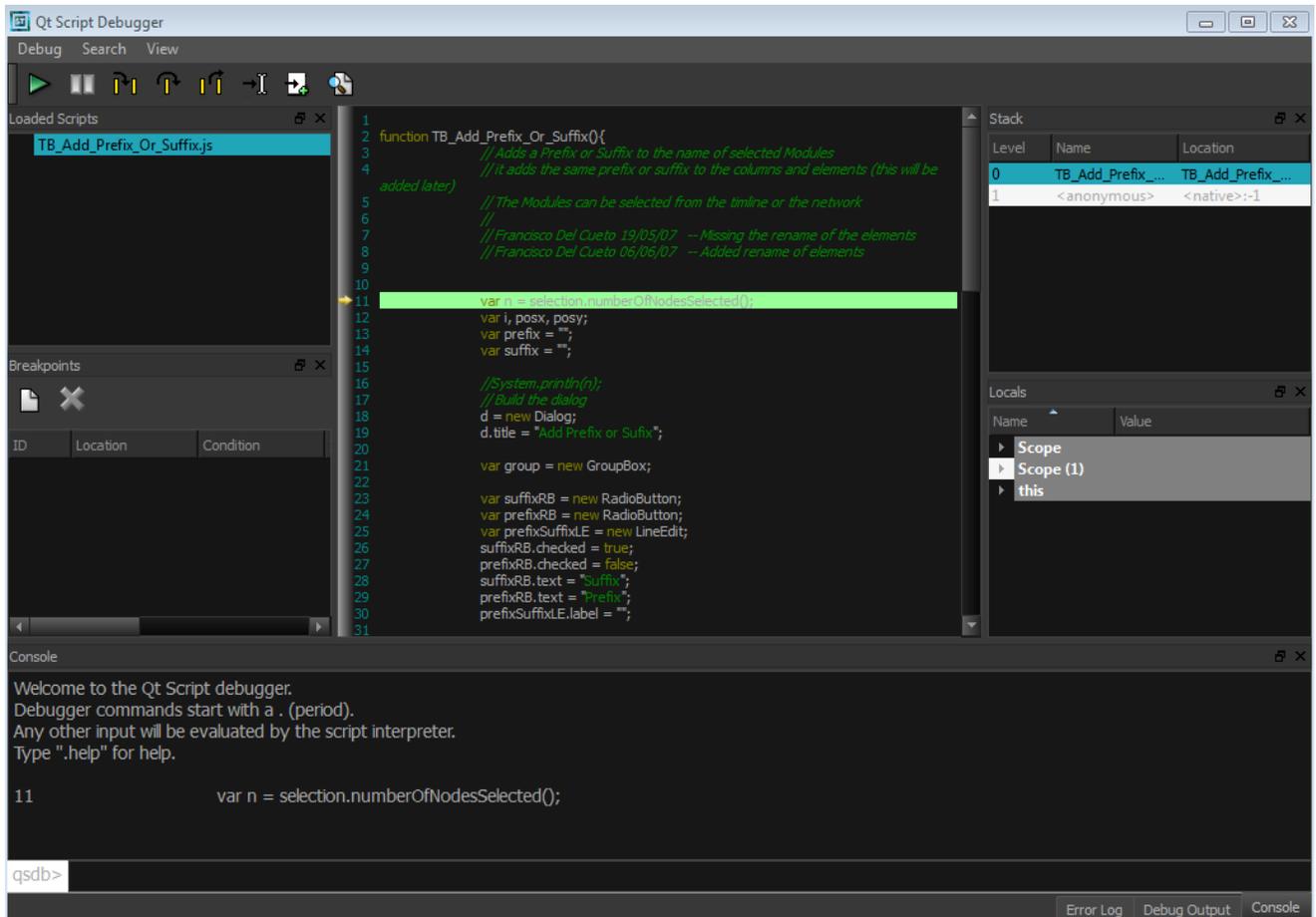## How to troubleshoot errors in your script

1. Debug: In the File list, select the script containing the function to debug.From the Script Editor toolbar, click on the Debug ![icon] button, or from the view menu, select **Play/Debug > Debug**.

   A window opens listing the scripts and functions available.

2. In the Functions column, select the function to debug.
3. Click **OK**.

The Qt Script Debugger opens.

Importing Scripts

To use scripts prepared by other users, you can simply import the `*.js` from where it is saved on the server.

To import a script:

1. In the Script Editor, click on the **Import Scripts** ![icon] button, or from the view menu, select **File > Import Script**.

   A browser window will open and allow you to browse to the location on the server where the `*.js` files are saved. You can choose either the User, Global, Environment, or Job folder.

   Here is an example of where those scripts can be found:

   ‣ **Global**: \\(Server_Name)\USA_DB\scripts

   ‣ **Environment**: \\(Server_Name)\USA_DB\environments\(user name)

   ‣ **Job**: \\(Server_Name)\USA_DB\jobs\(user name)

   ‣ **User**: \\(Server_Name)\USA_DB\users\(user name)\stage\1030-scripts

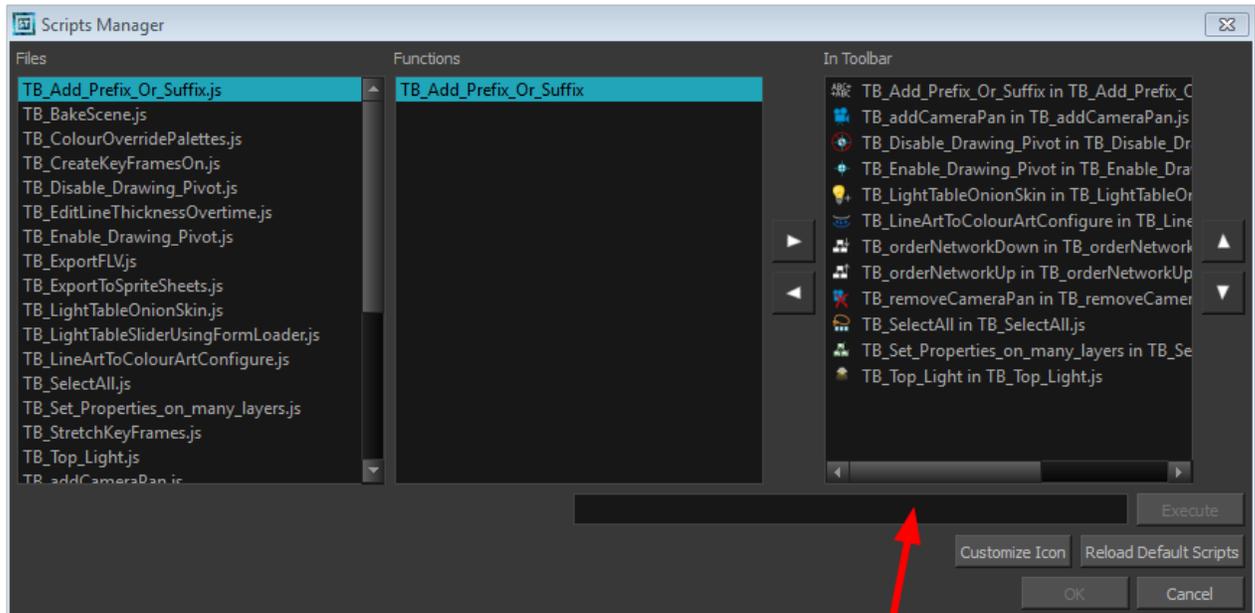2. Select the `*.js` file and click on the **Open** button.

   The script is imported and appears in the File list of the Script Editor.

Linking a Script to a Toolbar Button

You can add buttons to the Scripting Tools toolbar so that you can access them easily. They will appear to the right of the default Scripting Toolbar buttons.

To link a script to a toolbar button:

1. In the Scripting Tools toolbar, click on the Manage Scripts ⨍ button.

   A dialog box containing all of the available scripts opens.

You can select a script from this dialog box and run it immediately by clicking on the Execute button. The field displays the name of the currently selected script.

2. In the Files list, select the JavaScript file containing the function you want to add to the toolbar.

3. In the Functions list, select the function to add to the toolbar.

4. Click on the Right Arrow ▷ button to add the function to the In Toolbar list.

5. Click on **OK** to add the button and close the dialog box.

To execute a script:

1. Once the script is added to the Scripting toolbar, you can press on the corresponding script button to execute the action–see *Linking a Script to a Toolbar Button* on page 21.

2. You can press the Esc key to interrupt the execution of a script.