Toon Boom Harmony 12.2.1
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

Harmony is a trademark of Toon Boom Animation Inc.

Publication Date

7/6/2018

Contents

Chapter 1: Introduction

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

Detailed Scripting Method

For details about the scripting nodes, methods, and DbScript documentation, click the links below.

- Harmony Script Interface Documentation
- Script Module Documentation
- DbScript Documentation

You can also find the help files in the Script Editor view. From the Script Editor menu, select **Help > Scripting Interface Documentation**.

Move from QSA to QT Script

As of Toon Boom Harmony 10.3, Toon Boom Animation has changed the scripting language. The scripting language has changed to Qt Script from QSA for a better and faster scripting environment than before. Script storage is now done as plain text, `*.js` instead of `*.qsa` format.

This change to the API improves the software and supports 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 existing scripts may be modified.

NOTE:
Detailed instructions on moving QSA script to Qt Script can be found in the **"Moving from QSA to Qt Script"** documentation at this link: qt-project.org/doc/qt-4.8/porting-qsa.html

For a general overview of the QT UI toolkit, refer to qt-project.org/doc/qt-4.8/index.html

For a tutorial on coding in JavaScript, refer to docs.oracle.com/javase/tutorial/java

---

Chapter 2: Scripting Overview

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 the completion of various repetitive tasks.

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.

To use scripts prepared by other users, you must import those scripts. You can add buttons to the Scripting toolbar for quick and easy access.

This chapter is divided as follows:

- *Creating a Qt Script* on page 5
- *Using an External Script Editor* on page 10
- *Importing Scripts* on page 11
- *Linking a Script to a Toolbar Button* on page 13

Creating a Qt Script

Using the Script Editor, you can create Qt scripts for Harmony.

For details about the scripting nodes, methods, and DbScript documentation, click the links below.
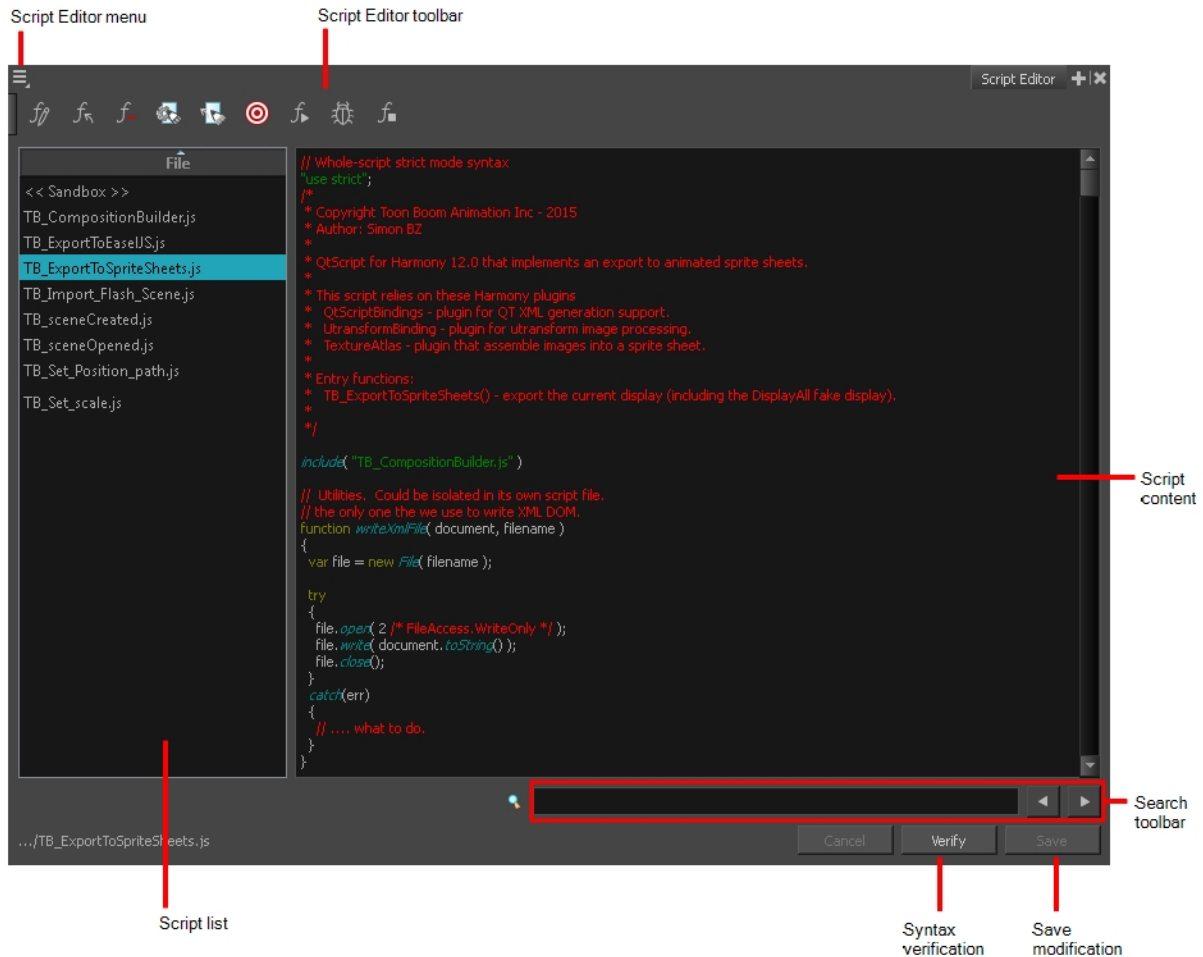
- Harmony Script Interface Documentation
- Script Module Documentation
- DbScript Documentation

You can also find the help files in the Script Editor view. From the Script Editor menu, select **Help > Scripting Interface Documentation**.

How to create a script

1. From the top menu, select **Windows > Script Editor**.

   The Script Editor view opens listing all the available JavaScript files.

Script Editor menu

Script Editor toolbar

Script content

Search toolbar

Script list

Syntax verification

Save modification

2. Do one o the folowing:

▸ From the Script Editor menu, select **File > New Script**

▸ In the Script Editor toolbar, click the New Script $f\!f$ button.

▸ From the top menu, select **Windows > Toolbars > Script Editor**.

The New Script dialog box appears.
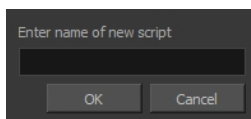
3. Enter a name and click on **OK**.

▸ When working with Harmony Server, you must set the location where the script will be save, either User level, Global level, Environment level, or Job level.
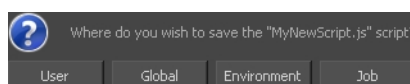
When working with Harmony Stand Alone, your custom scripts and default scripts are stored in:

- **Windows**:

  - **Toon Boom Harmony Premium**: C:\Users\[user_name]\AppData\Roaming\Toon Boom Anim-ation\Toon Boom Harmony\1200-scripts
  - **Toon Boom Harmony Advanced**: C:\Users\[user_name]\AppData\Roaming\Toon Boom Animation\Toon Boom Advanced\1200-scripts
  - **Toon Boom Harmony Essentials**: C:\Users[\user_name]\AppData\Roaming\Toon Boom Animation\Toon Boom Essentials\1200-scripts

- **Mac OS X**:

  - **Toon Boom Harmony Premium**: /Users/[user_name]/Library/Preferences/Toon Boom Anim-ation/Toon Boom Harmony/1200-scripts
  - **Toon Boom Harmony Advanced**: /Users/[user_name]/Library/Preferences/Toon Boom Anim-ation/Toon Boom Advanced/1200-scripts
  - **Toon Boom Harmony Essentials**: /Users/[user_name]/Library/Preferences/Toon Boom Anim-ation/Toon Boom Essentials/1200-scripts

 On Mac OS X, the Library folder is a hidden folder. To display the display the folder, told down the Alt key.

- **Linux**:

  - /home/[user_name]/Toon Boom Animation/Toon Boom Harmony/1200-scripts/
  - /home/[user_name]/Toon Boom Animation/Toon Boom StageAdvanced/1200-scripts/

  When working with Harmony Server, your custom scripts are stored in:

- **Global**: [Server_Name] > USA_DB > scripts
- **Environment**: [Server_Name] > USA_DB > environments > [environment_name]
- **Job**: [Server_Name] > USA_DB > jobs > [job name]
- **User**: [Server_Name] > USA_DB > users > [user_name] > stage > 1200-scripts

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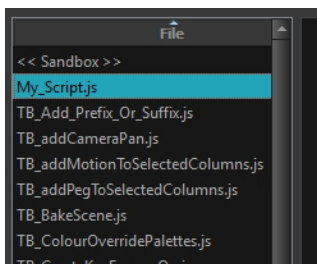
5. Click in the right side of the Script Editor and start writing your script. Try typing in the following script:

   ```
   function add3dPathCol()

   {
   ```

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

}
```

NOTE:
For a tutorial on coding in JavaScript, refer to docs.oracle.com/javase/tutorial/java

For a detailed Harmony script interface documentation, refer to Harmony Script Interface Documentation

You can also find the help files in the Script Editor view. In the Script Editor menu, select **Help > Scripting Interface Documentation**.

6. To check your syntax, click **Verify**.

    A window opens with syntax information.
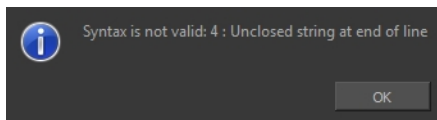
7. To test your script, select the script you want to run from the File column.

8. Do one of the following:

    ‣ From the Script Editor toolbar, click the Run 𝑓 button.

    ‣ From the Script Editor menu, select **Play/Debug > Run**.

    A window opens to ask you for the target function to run. In the window, select the function you want to execute.

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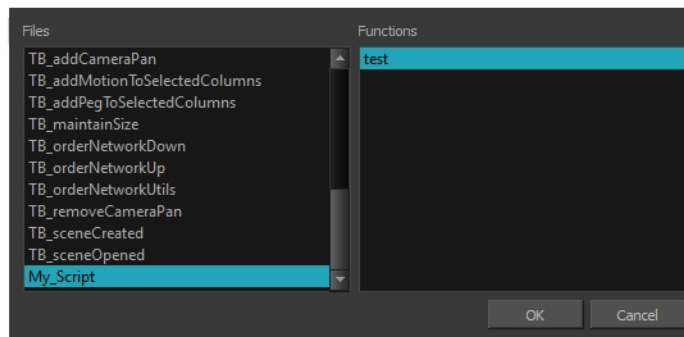

    ‣ To stop the execution of your script, click the Stop Execution 𝑓 button.

‣ You may receive this 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 wanted to run from the File column before running the script. When this happens, click **Save** and run your script again.

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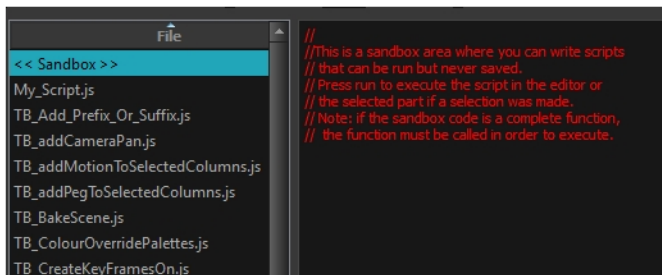
This is because the software jumped to the <<Sandbox>> item in the Function column. The <<Sandbox>> is provided to test scripts without saving them. Simply click on your newly created script and run it again.

How to delete a script

1. In the Script Editor, in the File column, select the script to delete.

2. Do one of the following:

   ‣ From the Script Editor toolbar, click the Delete Script ƒ button.

   ‣ From the Script Editor view menu, select **File > Delete Script**.

How to troubleshoot errors in your script

1. In the File list, select the script containing the function to debug.

2. Do one of the following:

   ‣ From the Script Editor toolbar, click the Debug 🐞 button.

   ‣ From the Script Editor menu, select **Play/Debug > Debug**.

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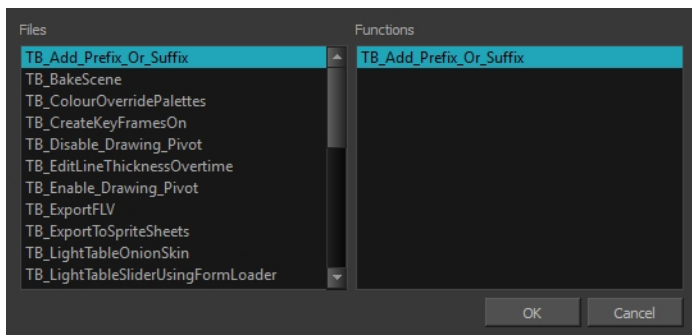
3. In the Functions column, select the function to debug and click **OK**.

The Qt Script Debugger opens.
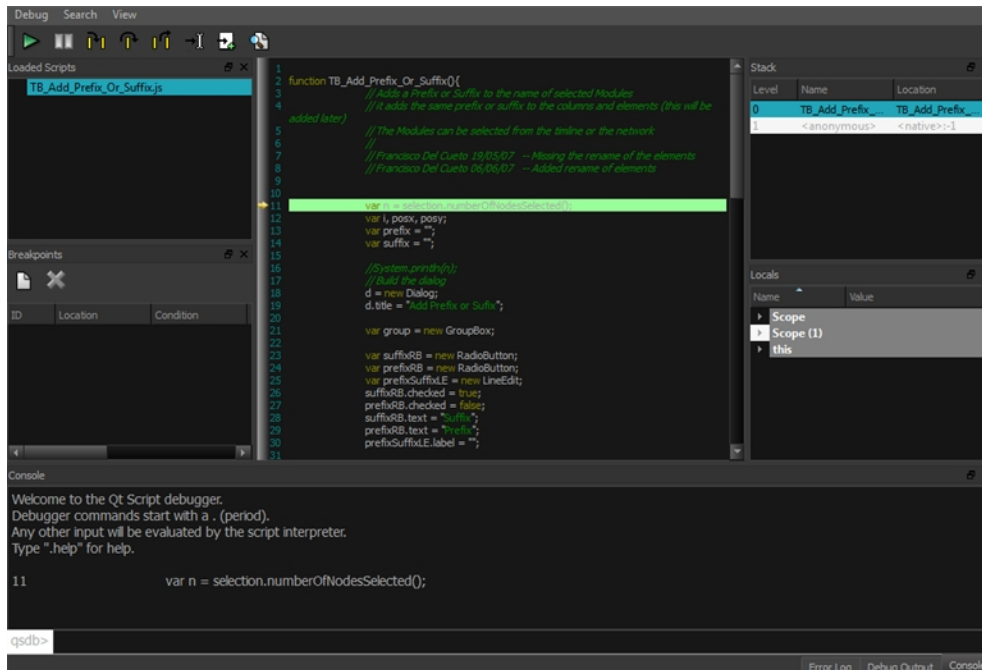
Using an External Script Editor

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.

> **NOTE:**
> 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.

How to set an external editor

1. Do one of the following:

   - From the Script Editor menu, select **Editor > Set External Editor**.
   - In the Script Editor toolbar, click the Set External Editor button.

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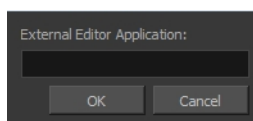
2. In the external editor, type or paste the complete path to the application, including its name and extension.

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. If the application supports the Applescript Open Document protocol, it will open.

3. Click **OK**.

How to edit a script in an external editor

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

1. Do one of the following:

   ‣ From the Script Editor menu, select **Editor > External Editor**.

   ‣ In the Script Editor toolbar, click the External Editor 🔲 button.

2. In the external editor that opens, edit the script and save the file.

3. In Harmony, refresh the Script Editor view by selecting another script and then 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.

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:

- **Windows**:

  - **Toon Boom Harmony Premium**: C:\Users\[user_name]\AppData\Roaming\Toon Boom Animation\Toon Boom Harmony\1200-scripts

  - **Toon Boom Harmony Advanced**: C:\Users\[user_name]\AppData\Roaming\Toon Boom Animation\Toon Boom Advanced\1200-scripts

  - **Toon Boom Harmony Essentials**: C:\Users\[user_name]\AppData\Roaming\Toon Boom Animation\Toon Boom Essentials\1200-scripts

- **Mac OS X**:

  - **Toon Boom Harmony Premium**: /Users/[user_name]/Library/Preferences/Toon Boom Animation/Toon Boom Harmony/1200-scripts

  - **Toon Boom Harmony Advanced**: /Users/[user_name]/Library/Preferences/Toon Boom Animation/Toon Boom Advanced/1200-scripts

- **Toon Boom Harmony Essentials**: /Users/[user_name]/Library/Preferences/Toon Boom Animation/Toon Boom Essentials/1200-scripts

  On Mac OS X, the Library folder is a hidden folder. To display the display the folder, told down the Alt key.

- **Linux**:

  - /home/[user_name]/Toon Boom Animation/Toon Boom Harmony/1200-scripts/
  - /home/[user_name]/Toon Boom Animation/Toon Boom StageAdvanced/1200-scripts/

With Harmony Server, scripts that you have been working on and imported are available at the level you saved them:

- **Global**: [Server_Name] > USA_DB > scripts
- **Environment**: [Server_Name] > USA_DB > environments > [environment_name]
- **Job**: [Server_Name] > USA_DB > jobs > [job name]
- **User**: [Server_Name] > USA_DB > users > [user_name] > stage > 1200-scripts

If you deleted default scripts and would like to reload them, you can find them in:

- **Windows**: Program Files (x86)> Toon Boom Animation > Harmony 12.2 [edition] > resources > scripts
- **Mac OS X**: Applications > Harmony 12.2 [edition] > tba > resources > scripts
- **Linux**: /usr/local/ToonBoomAnimation/harmony_12.2/resources/scripts

How to import scripts

1. Copy the `*.js` files that you want to transfer onto a USB key, or use the transfer method of your choice.

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

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

3. In Harmony, open a new or existing scene.

4. Do one of the following:

   ‣ In the Script Editor view, click the Import Scripts ƒ button.
   ‣ From the Script Editor menu, select **File > Import Script**.

   With Harmony Server, a window opens in which you can browse to the location on the server where the `*.js` files are saved. You can choose the User, Global, Environment, or Job folder.

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

   When working with Harmony Server, you must set the location where the script will be saved: User level, Global level, Environment level, or Job level.
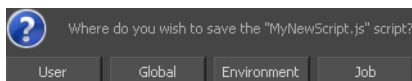
The script is imported and appears in the Files list of the Script Editor.

Linking a Script to a Toolbar Button

You can add buttons to the Scripting toolbar so you can access them easily. They will appear to the right of the default Scripting toolbar buttons.
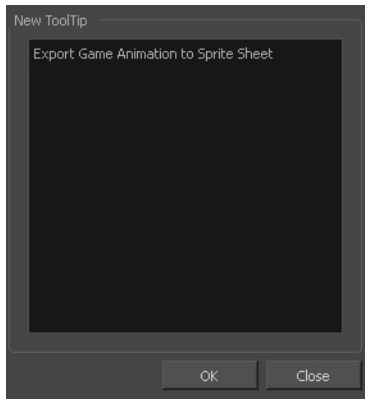
How to link a script to a toolbar button

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

   The Scripts Manager dialog box opens and displays the available scripts.
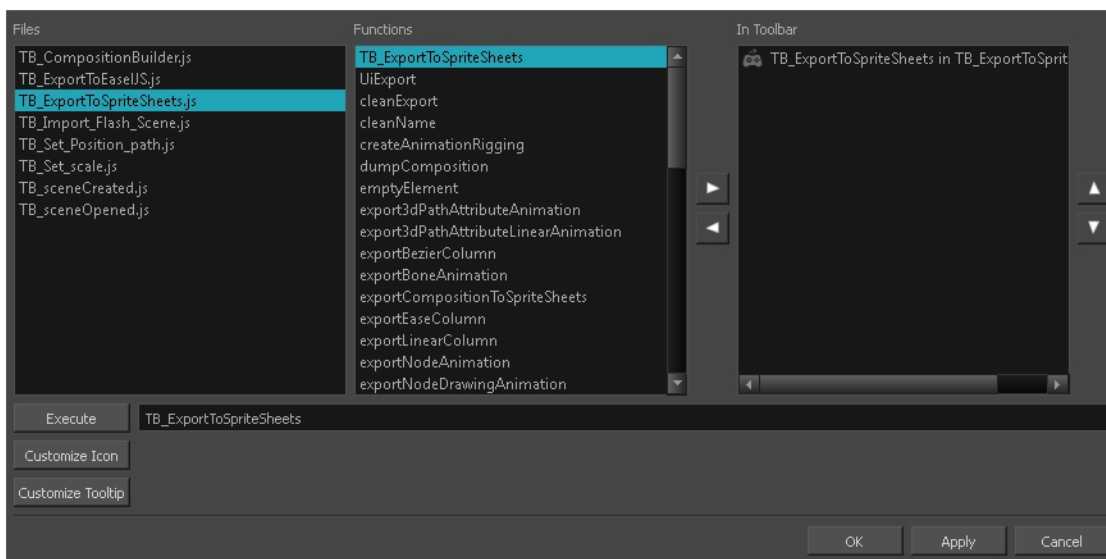
You can select a script from this dialog box and run it immediately by clicking **Execute**.

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

5. To load a custom icon, click **Customize Icon** and browser for a `*.png`, `*.jpg` or `*.xpm` file.

6. To customize the tooltip appearing in the Scripting toolbar, click **Customize Tooltip** and type in the new tooltip in the Function Tooltip window.

7. Click **OK**.

How 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 *How to link a script to a toolbar button* on page 13.

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