



TI-*nspire*™

## Writing Lua Scripts Guide

## Important Information

Except as otherwise expressly stated in the License that accompanies a program, Texas Instruments makes no warranty, either express or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis. In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability of Texas Instruments, regardless of the form of action, shall not exceed the amount set forth in the license for the program. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

© 2023 Texas Instruments Incorporated

The TI-Nspire™ software uses Lua as scripting environment. For copyright and license information, see <http://www.lua.org/license.html>.

The TI-Nspire™ software uses Chipmunk Physics version 5.3.4 as simulation environment. For license information, see <http://chipmunk-physics.net/release/Chipmunk-5.x/Chipmunk-5.3.4-Docs/>.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Mac OS®, iPad® and OS X® are registered trademarks of Apple Inc.

Unicode® is a registered trademark of Unicode, Inc. in the United States and other countries.

Bluetooth® word mark and logos are registered trademark owned by Bluetooth SIG, Inc.

Actual products may vary slightly from provided images.

# Contents

<b>Writing Lua Scripts</b> .....	<b>1</b>
Overview of the Script Editor .....	1
Exploring the Script Editor Interface .....	1
Using the Toolbar .....	2
Using the Tools Panel .....	2
Using the Status Bar .....	3
Inserting New Scripts .....	3
Zooming Text in the Script Editor .....	3
Editing Scripts .....	4
Changing View Options .....	4
Setting Minimum API Level .....	4
Saving Script Applications .....	5
Managing Images .....	5
Add an Image to Resources .....	5
Add Multiple Images to Resources .....	5
Create a Script to Call Up an Image .....	5
Create a Script to Call Up Multiple Images .....	6
Setting Script Permissions .....	6
Debugging Scripts .....	6

# Writing Lua Scripts

The Script Editor allows you to create and deliver dynamically linked simulations, powerful and flexible utilities, and other educational content for exploring math and science concepts. When you open a document containing a script, the script runs automatically as programmed. To see the running script application, the page containing the script application must be active.

The Script Editor is directed toward teachers and other authors who are comfortable working in a Lua scripting environment. Lua is a powerful, fast, lightweight scripting language that is fully supported in TI-Nspire™ documents. Documents containing script applications can be opened on TI-Nspire™ CX II handhelds. The script application runs on a handheld, but you cannot view or edit the script.

Note these resources for using the Script Editor and creating scripts:

- Press **F1** to access the TI-Nspire™ help, which includes the Script Editor help.
- Press **F2** for additional TI-Nspire™ resources such as scripting samples and a link to the TI-Nspire™ Scripting API library. (This information is also available at [education.ti.com/nspire/scripting](http://education.ti.com/nspire/scripting).)
- Go to [lua.org](http://lua.org) for more information about Lua.

## Overview of the Script Editor

With the Script Editor, you can insert, edit, save, run, and debug script applications in TI-Nspire™ (.tns files) documents.

- Script applications function within documents, problems, and pages the same way that other TI-Nspire™ applications do.
- When you create a new document or open an existing document, you can insert or edit a script application within a page or within a work area of a split page.
- In a split page layout, you can add a script application to each work area of a page. A page can be split into a maximum of four quadrants.
- Images can be added to script applications. See the *Inserting Images* section.
- All work done in the Script Editor is lost if you close the TI-Nspire™ document without saving it.

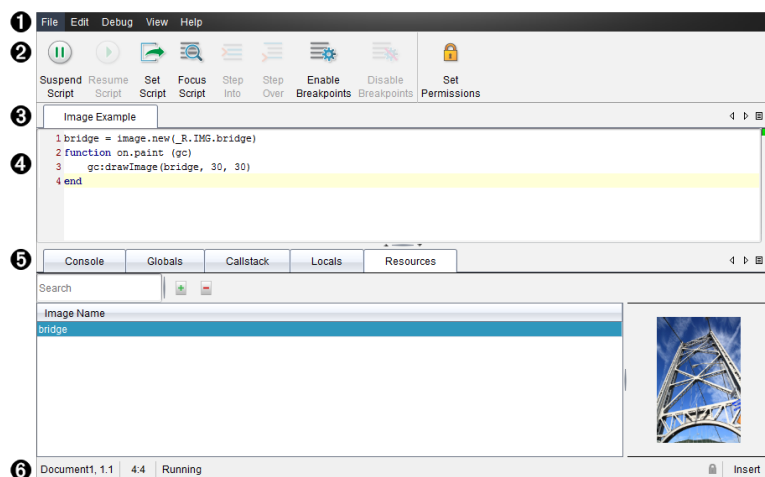
## Exploring the Script Editor Interface

The Script Editor window opens when you insert a new script application or edit an existing script application in a TI-Nspire™ document.

Select options for creating new scripts or editing scripts from the Insert menu in the Documents Workspace when a document is open.

**Note:** Although not labeled, the Documents Workspace is the default workspace in the TI-Nspire™ CX Student Software and TI-Nspire™ CX CAS Student Software.










The following figure shows the Script Editor with an existing script.



- ❶ **Menu bar.** Contains options for working with the Script Editor.
- ❷ **Toolbar.** Provides tools for common Script Editor functions. See *Using the Toolbar*.
- ❸ **Script title.** Shows script title. Right-click the title to change it or by clicking **Edit > Set Script Title**.
- ❹ **Text box.** Provides a space to type script text.
- ❺ **Tools panel.** Shows script data. See *Using the Tools Panel*.
- ❻ **Status bar.** Displays the operational state of the script. See *Using the Status Bar*.

## Using the Toolbar

The following table describes the toolbar options.

Tool name		Tool function
	Suspend Script	Pauses the script execution.
	Resume Script	Resumes the script execution. While debugging, the script continues to execute until the next breakpoint or the end of the script.
	Set Script	Starts the script execution.
	Focus Script	Sets the focus to the page in the document where the script application is attached: <ul style="list-style-type: none"> <li>• In a TI-Nspire™ document, sets the focus to the page.</li> </ul>
	Step Into	While debugging, executes the current statement. If the statement calls any functions, the debugger stops at the first line of each function.
	Step Over	While debugging, executes the current statement. If the statement calls any functions, the debugger does not stop within the function unless the function contains a breakpoint.
	Enable Breakpoints	Switches from normal mode to debugging mode.
	Disable Breakpoints	Switches back from debugging mode to normal mode and resumes the script.
	Set Permissions	Sets permission levels to Protected, View Only, or Unprotected and allows you to set a password for the script.

## Using the Tools Panel

At the bottom of the window, the Tools panel shows the scripting data. For more information, see the *Debugging Scripts* section.

Tab	Window Display
Console	Script errors are printed here. Print statements embedded in the script also print here.
Globals	Selected global variables are displayed. To select a global variable for display, select <b>Click to add new watch variable</b> at the bottom of the Tools panel.

Tab	Window Display
Call Stack	Displays the call tree for the currently executing function.
Locals	Local variables in the current function's scope are displayed.
Resources	Upload, insert, and manage images here.

## Using the Status Bar

The status bar at the bottom of the window shows basic script data, as described in this example:

**stopwatch, 1.1, 4:1, Running**

- Name of the document that the script application is attached to (**stopwatch**)
- Problem and page number (**1.1**)
- Script line and character, (**4:1** describing line 4 and character 1)
- Operational state of the script (**Running**). Note the possible states:
  - Normal mode: Running, Paused, or Error
  - Debugging mode: Running (debugging), Paused, or Error

## Inserting New Scripts

To insert a new script application and script, follow these steps.

1. Open the document where you want to insert the script. It can be a new or existing document.
2. Click **Insert > Script Editor > Insert Script**.

A script application is inserted, and the Script Title dialog box opens.

**Note:** The TI-Nspire™ CX Student Software and the TI-Nspire™ CX CAS Student Software open automatically within the Documents Workspace.

3. Type a script title. (The maximum number of characters is 32.)
4. Click **OK**.

The Script Editor window opens showing a blank script.

5. Type your text in the script lines.

**Note:** Some nonstandard UTF-8 wide characters may not be displayed correctly. For these characters, it is highly recommended that you use the `string.uchar` function.

6. When the script is complete, click **Set Script** to execute it.
  - In a TI-Nspire™ document, the script application is inserted in a new page. When the page containing the script application is active, the Documents Toolbox is empty.
7. To view the script application, click **Focus Script**.

## Zooming Text in the Script Editor

You can adjust the size of the text in your script and in any of the tool panels.

### Menu Method

1. Click in the area of the text to zoom.
2. On the **View** menu, select **Zoom**, and then select **Zoom In**, **Zoom Out**, or **Restore**.

Note that the menu also displays keyboard shortcuts for the Zoom commands.

### Mouse Method

1. Position the mouse pointer over the area to zoom.
2. Hold **Ctrl**, and roll the mouse wheel forward or back.

## Editing Scripts

To edit an existing script, follow these steps.

1. Open the TI-Nspire™ document that contains the script. The page containing the script must be active.
2. Select the page and the work area that contains the script.
3. Click **Insert > Script Editor > Edit Script**.

The Script Editor opens showing the script. If the selected work area on the page does not contain a script, Edit Script is dimmed.

If the script is password protected, the Password Protected dialog box opens prompting for a password.

4. Make any desired changes.
  - To designate comments, use double hyphens (--) at the start of each comment line.
  - To change the title, click **Edit > Set Script Title** or right-click the title and click **Set Script Title**.

### Notes:

- Some nonstandard UTF-8 characters may not be displayed correctly. For these characters, it is highly recommended that you use the `string.uchar` function.
  - The print function may yield unexpected results for non-UTF-8 characters.
  - Some nonprintable characters returned by the `on.save` function will be discarded.
5. To execute the script, click **Set Script**.

Any errors are displayed in the Console area in the Tools panel.

6. To view the script application (running script), click **Focus Script**.

## Changing View Options

To change viewing options:

- ▶ To clear the scripting data in the Tools panel and restore the editor defaults, click **View > Restore Editor to Defaults**.
- ▶ To view the script title in the document and before each print statement in the Console, click **View > Title in Document View**.
- ▶ To hide or show toolbar labels, click **View > Toolbar Text Labels**.
- ▶ To show or hide the Tools panel or its areas, click **View > Tools Panel** and click the appropriate option.
- ▶ To create tab groups when multiple scripts are open, right-click one of the titles and click **New Horizontal Group** or **New Vertical Group**.

## Setting Minimum API Level

Each release of the T-Nspire™ software includes API support for a specific set of Lua scripting features. Setting the minimum API level for a script lets you specify a minimum set of features that you require for your script.

If a user tries to run the script on a system that does not meet the script's minimum API level, a message notifies the user and prevents the script from running.

### To Set the Minimum API Level for a Script:

1. Determine the minimum level that you want your script to require.
  - Setting the level too low for the script's feature set can result in a script error on older software.
  - Setting the level too high can result in the script refusing to start on older software that supports the feature set.
2. On the Script Editor **File** menu, select **Set Minimum API Level**.
3. In the dialog box, type the minimum level using the format *major . minor*. For example, you might type `2 . 3`.

This API level or higher will be required for running the script.

## Saving Script Applications

Clicking Set Script resets (updates) a script application in a TI-Nspire™ document. However, the script and script application are not saved until you save the document. If you close the document or close the TI-Nspire™ software without saving, work on the script is lost.

To ensure the script application is saved after all work is complete, follow these steps.


1. From the Script Editor window, click **Set Script** to reset (update) the script application in the document.
2. From an open document, click **File > Save Document** to save changes to the TI-Nspire™ document.

**Note:** To ensure work is backed up, set the script and save the document frequently.


## Managing Images

To insert an image into a script application, follow these steps.


### Add an Image to Resources

1. Click the **Resource** tab.
2. Click the  button.
3. Click on an image file name.
4. Click **Open**.
5. Accept the default image name or rename the image by typing a new name into the box. (Ex: newimage)
6. Click **OK**.

**Note:** You will see the image thumbnail in the bottom-right corner of your screen. Your image file name will appear in a list of images at the bottom-left of your screen.

7. Right-click an image name to Rename, Copy Name, Preview, or Remove the image. You may also click the  button to remove an image.

### Add Multiple Images to Resources


1. Click the **Resource** tab.
2. Click the  button.
3. Select **Comma Separated Values (\*.csv)** from the **Files of Type** menu.
4. Select your **.csv file**.

**Note:** The .csv format consists of two comma-separated columns. The first column is the local name of the image resource used in the code. The second column is the absolute path to the image on the current system.

**Example for Windows®:**

bridge,C:\images\bridge.jpg

house,C:\images\house.jpg

5. Click **Open**.
6. Right-click an image name to Rename, Copy Name, Preview, or Remove the image. You may also click the  button to remove an image.

### Create a Script to Call Up an Image

1. Type a script in the script line box as follows:

```
myimage = image.new(_R.IMG.img_1)
function on.paint (gc)
  gc:drawImage (myimage, 30, 30)
end
```

**Note:** Replace img\_1 (above) with the name of your image.



2. Click **Set Script** to save the script. You will see your image in the Document Preview screen.
3. Click **Focus Script** to set the focus to the page in the document where you want to attach the script application.

**Note:** A TI-Nspire™ document sets the focus to the page.

## Create a Script to Call Up Multiple Images

1. Type a script in the script line box as follows:

```
myimg = { }
for name, data in pairs (_R.IMG)
  myimg [name] = image.new(data)
end
function on.paint (gc)
  gc.drawImage (myimg[imagenam], 30, 30)
end
```

2. Click **Set Script** to save the script. You will see your image in the Document Preview screen.
3. Click **Focus Script** to set the focus to the page in the document where you want to attach the script application.

**Note:** A TI-Nspire™ document sets the focus to the page.

## Setting Script Permissions

You can set permission levels for a script and specify a password to protect a script. Follow these steps.

1. In the Script Editor window, click **File > Set Permissions**.

The Set Permissions dialog box opens.

2. In the Permissions Level area, select the appropriate security level:

- **Protected.** The script can be run, but not viewed or edited.
- **View only.** The script can be viewed, but not edited.
- **Unprotected.** The script can be viewed and edited.

3. To secure a script, designate a password in the Security area.

**Note:** Use caution when setting passwords because they cannot be recovered.

4. Click **OK**.

The next time you click **Insert > Script Editor > Edit Script**, a Password Protected dialog box opens prompting for the password. Choose one of these options:

- To edit the script, enter the password and click **OK**.
- To view the script only, do not enter the password and click **View**.

## Debugging Scripts

You can debug your script to investigate runtime errors and trace the execution flow. While debugging, data is displayed in the Tools panel.

- ▶ To enable debugging mode or disable it and return to normal mode, click **Debug > Enable Breakpoints** or **Disable Breakpoints**.

**Note:** Disabling breakpoints always resumes the script execution.

- ▶ While debugging, click **Step Into** and **Step Over** as appropriate. See the *Exploring the Script Editor Interface* section.
- ▶ To set breakpoints, double-click in the space to the far left of the line number. Breakpoints are disabled until you click **EnableBreakpoints**.
- ▶ Note these factors when debugging:
  - Breakpoints in coroutines are not supported.
  - If a breakpoint is set in a function that is a callback, the debugger may not stop at the breakpoint.

- The debugger may not stop at functions such as `on.save`, `on.restore`, and `on.destroy`.

On the toolbar, Step Into and Step Over are enabled when breakpoints are enabled.

- ▶ To suspend and resume the script execution, click **Suspend Script** and **Resume Script**. When the script resumes, it runs until the next breakpoint is encountered or to the end of the script. A script can be suspended in normal mode or debugging mode.