

**MADCAP FLARE 2024**

# Scripting

Copyright © 2024 MadCap Software. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of MadCap Software.

MadCap Software  
9171 Towne Center Drive, Suite 335  
San Diego, California 92122  
858-320-0387  
[www.madcapsoftware.com](http://www.madcapsoftware.com)

**THIS PDF WAS CREATED USING MADCAP FLARE.**

# CONTENTS

---

## **CHAPTER 1**

Introduction .....	5
--------------------	---

## **CHAPTER 2**

Scripting Languages .....	7
JavaScript (and JQuery) .....	8
JScript .....	9
VBScript .....	10

## **CHAPTER 3**

Inserting Scripts .....	11
How to Insert a Script With the User Interface .....	12
How to Insert a Script Directly in the Markup .....	13
What's Noteworthy? .....	13

## **CHAPTER 4**

Editing Scripts .....	14
How to Edit a Script .....	14

## **CHAPTER 5**

Deleting Scripts .....	15
How to Delete a Script .....	15

## **APPENDIX**

PDFs .....	16
Tutorials .....	16
Cheat Sheets .....	17
User Guides .....	18

## CHAPTER 1

---

# Introduction


Supported In:




You can insert JavaScript, JQuery, JScript, or VBScript code into Flare content files. For example, you might have an image that you want to function as a button, performing a specific activity when users click it. There are all kinds of possibilities for scripting. If you can create a script that can be used in a website, you can create it in Flare as well.

### Main Activities and Information

- "Scripting Languages" on page 7
- "Inserting Scripts" on page 11
- "Editing Scripts" on page 14
- "Deleting Scripts" on page 15

 **NOTE** For Clean XHTML output, MadCap-injected JavaScript is removed. In other words, features such as drop-downs, Help controls, text popups, and footnotes rely on JavaScript to work. Therefore, with Clean XHTML output, these elements are converted to text. However, custom scripts that you might have added to topics will not automatically be removed when you generate Clean XHTML; if you want to exclude these custom scripts, you must do so manually (i.e., by deleting them or using conditions).

 **NOTE** If you run security scans on your HTML5 output, you may receive security messages during the scan if you use JavaScript to make context-sensitive (CSH) calls to your Help.

**CHAPTER 2**

---

# Scripting Languages

This chapter describes the main scripting languages that you can use in Flare.

This chapter discusses the following:

- JavaScript (and JQuery) ..... 8
- JScript ..... 9
- VBScript .....10

# I JavaScript (and JQuery)

JavaScript is a scripting language that lets authors design interactive sites. It shares many of the features and structures of the full Java language, but was developed independently. JavaScript can interact with HTML source code, enabling authors to include dynamic content in their sites.

JQuery is a JavaScript library intended to make client-side scripting simpler. You can integrate JQuery into a Flare project by following the same steps that you would for JavaScript.



# I JScript

JScript is Microsoft's extended implementation of ECMAScript (ECMA262), an international standard based on Netscape's JavaScript and Microsoft's JScript languages. JScript is implemented as a Windows Script engine. This means that it can be "plugged in" to any application that supports Windows Script, such as Active Server Pages and Windows Script Host. It also means that any application supporting Windows Script can use multiple languages—JScript, VBScript, Perl, and others. JScript (and the other languages) can be used for both simple tasks (such as mouseovers on web pages) and for more complex tasks (such as updating a database with ASP or running logon scripts for Windows NT). Windows Script relies on external "object models" to carry out much of its work.

# I VBScript

VBScript is a scripting language based on MS Visual Basic and, like JavaScript, is embedded in a web page. The interpretation and execution of scripts is controlled by the web client. Much like JavaScript, functions are most often executed by mouse functions, navigation buttons, Active X controls or by actions initiated by the user or by automated scripting such as retrieving user computer information.

# Inserting Scripts

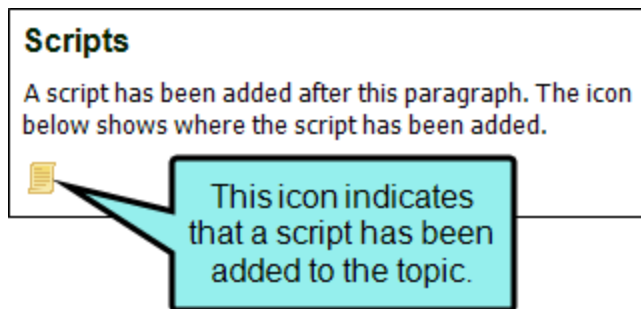
If you are an advanced user, you can insert a script (or a link to a script file) using JavaScript, VBScript, or JScript into your project. You can insert a script or a link directly into a content file (e.g., topic, template page) with the user interface or directly in the markup. If you insert a script into a topic, it is used just for that topic; if you insert a script into a template page, it is used for all topics associated with that template page.

This chapter discusses the following:

How to Insert a Script With the User Interface .....	12
How to Insert a Script Directly in the Markup .....	13
What's Noteworthy? .....	13


# I How to Insert a Script With the User Interface

1. Open the content file.
2. In the XML Editor, place your cursor where you want to insert the script.
3. Select **Insert > Script**. The Insert Script dialog opens.
4. From the **Language** drop-down menu, select **text/javascript**, **text/jscript**, or **text/vbscript**.
5. Do one of the following:
  - In the **Script Code** area, type the code for the script.
  - OR
  - Next to the **Script Link** field, click the **Browse** button to find and select a script file (which you would have stored in the Content Explorer).
6. Click **OK**. The script is added to the topic, with the script icon displayed at the spot of the insertion.




7. Click  to save your work.

# I How to Insert a Script Directly in the Markup

1. Open the content file.
2. At the bottom of the XML Editor, click the **Text Editor** tab to see the markup. You can also use split the editor between the XML Editor and the Internal Text Editor. .
3. Enter the script text—or a link to it—into the markup, usually between the <head> and </head> tags.
4. Click  to save your work.


## I What's Noteworthy?

 **NOTE** If you enter a script directly into the <head> tags of a topic (in the markup), it will be overwritten in the output if a template page is applied to the topic. However, if you add a script file to your project and link to it from the <head> area in the topic markup, the template page will not overwrite the topic script.

# Editing Scripts

After you insert a script into a topic or create a snippet from a script, you might decide later that the script needs to be altered. You can edit the script using the steps below. If you edit a script contained in a snippet, the changes are automatically reflected in any topics where you have inserted the snippet previously.


## I How to Edit a Script

1. Open the content file.
2. In the XML Editor, right-click the script that you want to modify.
3. From the context menu, select **Edit Script**. The Insert Script dialog opens.
4. Make the necessary changes to the script—either the language and/or the script code itself.
5. Click **OK**.
6. Click  to save your work.

# Deleting Scripts

You can delete a script that you have inserted into a topic.

## I How to Delete a Script

1. Open the content file.
2. In the XML Editor, right-click the script that you want to delete.
3. From the context menu, select **Delete**. The script is removed from the topic.
4. Click  to save your work.

## APPENDIX

---

# PDFs

The following PDFs are available for download from the online Help.

## I Tutorials

*Getting Started Tutorial*

*Autonumbers Tutorial*

*Back-to-Top Button Tutorial*

*Context-Sensitive Help Tutorial*

*Custom Toolbar Tutorial*

*eLearning Tutorial—Basic*

*eLearning Tutorial—Advanced*

*Image Tooltips Tutorial*

*Lists Tutorial*

*Meta Tags Tutorial*

*Micro Content Tutorial—Basic*

*Micro Content Tutorial—Advanced*

*Responsive Output Tutorial*

*Single-Sourcing Tutorial*

*Snippet Conditions Tutorial*

*Styles Tutorials*

*Tables Tutorial*

*Word Import Tutorial*



# | Cheat Sheets

*Context-Sensitive Help Cheat Sheet*

*Folders and Files Cheat Sheet*

*Learning & Development Cheat Sheet*

*Lists Cheat Sheet*

*Micro Content Cheat Sheet*

*Print-Based Output Cheat Sheet*

*Search Cheat Sheet*

*Shortcuts Cheat Sheet*

*Structure Bars Cheat Sheet*

*Styles Cheat Sheet*

# **I** User Guides

*Accessibility Guide*

*Analysis and Reports Guide*

*Architecture Guide*

*Autonumbers Guide*

*Branding Guide*

*Condition Tags Guide*

*Context-Sensitive Help Guide*

*Eclipse Help Guide*

*eLearning Guide*

*Getting Started Guide*

*Global Project Linking Guide*

*HTML5 Guide*

*Images Guide*

*Import Guide*

*Indexing Guide*

*Key Features Guide*

*Lists Guide*

*MadCap Central Integration  
Guide*

*Meta Tags Guide*

*Micro Content Guide*

*Navigation Links Guide*

*Plug-In API Guide*

*Print-Based Output Guide*

*Project Creation Guide*

*QR Codes Guide*

*Reviews & Contributions With  
Contributor Guide*

*Scripting Guide*

*Search Guide*

*SharePoint Guide*

*Skins Guide*

*Snippets Guide*

*Source Control Guide: Git*

*Source Control Guide:  
Perforce Helix Core*

*Source Control Guide:  
Subversion*

*Source Control Guide: Team  
Foundation Server*

*Styles Guide*

*Tables Guide*

*Tables of Contents Guide*

*Targets Guide*

*Template Pages Guide*

*Templates Guide*

*Topics Guide*

*Touring the Workspace Guide*

*Transition From FrameMaker  
Guide*

*Translation and Localization  
Guide*

*Variables Guide*

*Videos Guide*

*What's New Guide*