

**EXTENSION MANAGER CS6
CONFIGURATION REFERENCE
TECHNICAL NOTE**



© 2012 Adobe Systems Incorporated. All rights reserved.

Technical Note: Extension Manager CS Configuration Reference for CS6

Adobe, the Adobe logo, Acrobat, Creative Suite, Contribute, Dreamweaver, Fireworks, Flash, Illustrator, InCopy, InDesign, and Photoshop are either registered trademarks or trademarks of Adobe Systems Inc. in the United States and/or other countries. Microsoft and Windows are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Apple, Mac OS, and Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Inc. Adobe Systems Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. The software described in this document is furnished under license and may only be used or copied in accordance with the terms of such license.

Adobe Systems Inc., 345 Park Avenue, San Jose, California 95110, USA.

Extension Manager CS6 Configuration File Reference

Adobe® Creative Suite® extensions are packaged for installation into applications using Adobe Extension Manager. Extension Manager has a command-line interface (the UCF tool) to that you can use to package extensions for installation.

- ▶ For more information about packaging extensions, see *Technical Note: Packaging, Signing, and Deploying Extensions with Extension Manager CS6*.

When you package an extension for use with the UCF command-line tool, you must provide an installation configuration file, or *manifest*, in the MXI format. MXI is an XML schema that specifies various attributes of an extension. This document lists and describes the XML elements defined by the MXI format for CS6.

Target applications for extensions include:

- Dreamweaver®
- Fireworks®
- Flash®
- Photoshop®
- Illustrator®
- InDesign®
- InCopy®
- Premiere® Pro
- Prelude

Extension Manager CS6 is compatible with CS6 versions only. To install extensions in previous versions, use the corresponding version of Extension Manager. Any element that is present in the MXI file but is not supported by the product in which the extension is installed is ignored during installation.

Packaging products for Adobe Exchange

The new Adobe Exchange is a Creative Suite extension marketplace. It is available as a panel (palette) within a variety of CS6 applications. The Adobe Exchange panel provides a new way to search, discover, and install plug-ins, extensions, and other content for Creative Suite products. Users can browse through items that are available for their application; download, and install those items; and update acquired items when new versions are available.

Content that can be offered as an extension through Adobe Exchange can be any kind of file, customized panel, application, or plug-in that extends the functionality of the Creative Suite applications. Some of the tools that you can use to create Creative Suite extensions to be offered on Adobe Exchange contain their own packaging tools. If you are not using these tools to create the content, however, you must manually create a manifest MXI file and package your delivery files into a ZXP, using the command-line utility in the Packaging and Signing Toolkit.

You will need to create an MXI file in these cases:

- ▶ You have created a product for Adobe Exchange using a version of Configurator earlier than Configurator 3, or using a version of Extension Builder earlier than Extension Builder 2.
- ▶ You have created a product using the free Creative Suite SDK.

- ▶ The content is a script or C++ plug-in, or some combination of these things.

Adobe Exchange has specific requirements for packages that you submit through the Producers Portal. For details of these requirements and example of how to package various kinds of content, see "Packaging your Product for Adobe Exchange" at <https://www.adobeexchange.com/resources/7>.

About careful XML coding

XML files have strict syntax requirements. When you create or edit an extension installation file, make sure that you use correct XML syntax:

- ▶ Every attribute value must be enclosed in a single pair of double quotation marks. For example,
`version = "1.0.0".`
- ▶ A tag with no contents must end with `/>`. Do not include any spaces between the slash and the closing angle bracket.
- ▶ Each attribute name must be preceded by a space (or other form of white space). If you use more than one attribute in a tag, you must put a space between each attribute's value and the next attribute's name.
- ▶ XML uses the ampersand (&) as an escape character. To include an ampersand within a tag, use the code `&`. Similarly, use `<` and `>` for the `<` and `>` characters.
- ▶ It is recommended that you encode MXI with UTF-8 and explicitly declare the encoding.
 - ▷ In both Mac OS and Windows, include the UTF-8 encoding declaration at the head of the MXI file:
`<?xml version="1.0" encoding="UTF-8"?>`
 - ▷ In Windows, also include the BOM at the head of the MXI file. The easiest way to is to open the MXI file with Notepad and select File > Save As, then set Encoding as UTF-8.

MXI element summary

General elements

These elements are used in all target applications. Details are provided in [“General MXI elements” on page 10](#), where the tags are described in alphabetical order of containers.

Tag	Description
author	Name of the extension’s author.
dependency	Container for extensions upon which this one depends.
description	Describes what the extension does.
files	Container for tags describing the files an extension installs.
license-agreement	Allows a third-party developer to include a license agreement that is displayed at installation.
macromedia-extension	Main container tag for extension installation file.
products	Container for tags specifying an extension’s product compatibility.
ui-access	Text to appear in the Extension Manager window when the extension is selected.
update	Extension update information (CS5 and higher).

Product-specific elements

These tags are used only by Dreamweaver or Flash, and are ignored when an extension is installed in any other target application.

Details are provided in [“Product-specific MXI elements” on page 21](#), where the tags are described in alphabetical order of containers.

Dreamweaver-only elements

Element	Description
configuration-changes	Container for elements that modify the application’s configuration. These include menus, shortcuts, server behaviors, and data sources.
data-source-changes	Container for all changes to menus in the <code>menus.xml</code> file in any of the Dreamweaver MX Configuration/DataSources/document_type folders.
documenttype-changes documenttype-insert	Container for elements that describe changes made to the <code>MMdocumentTypes.xml</code> file.

Element	Description
extensions-changes extension-insert	Container for elements that describe changes to the <code>Extensions.txt</code> file, such as adding extensions that you can open in Dreamweaver.
format	Describes the data format to be inserted into the Dreamweaver Format menu during installation of the extension.
ftp-extension-map-changes ftp-extension-insert	Container for elements that describe changes to the <code>FTPExtensionMap.txt</code> file. This defines whether the file is downloaded or uploaded as an ASCII or binary file from Dreamweaver to an FTP server.
insertbar-changes insertbar-insert insertbar-item-insert	Container for elements that describe changes to be made to the <code>insertbar.xml</code> file and add new toolbars files.
menu-insert menu menubar menuitem comment separator	Container for elements that describe a menu or submenu to be inserted into the application's menu structure during installation of an extension.
server-behavior-changes server-format-changes server-format-definition-changes	Container for changes to menus in the <code>menus.xml</code> file in any of the Dreamweaver MX Configuration/ServerBehaviors/document_type or MX Configuration/ServerFormats/document_type folders
shortcut-insert shortcutlist shortcut	Container for elements that specify keyboard shortcuts to be added to the <code>menus.xml</code> file.
taglibrary-changes taglibrary-insert	Container for elements that describe changes to be made to the <code>TagLibraries.vtm</code> file.
toolbar-changes toolbar-insert toolbar-item-insert	Container for elements that specify changes to the tool bar.

Flash only elements

toolpanel-changes toolpanel-item-insert	Container for elements that specify Flash tool-panel changes.
--	---

Location specification

All file names and location specifications must conform to these standards.

- ▶ All file and folder names must be valid for all supported platforms.
- ▶ You can use colon (:), slash (/), or backslash (\) as a separator between path elements.
- ▶ Because some operating systems are case sensitive, make sure the capitalization you use in configuration attributes exactly matches the file-system names.
- ▶ File names can have a maximum of 30 characters.
- ▶ Do not use the same file names as Adobe extensions, unless your extension is intended to substitute for an Adobe extension.

Path tokens

When specifying paths, you can use path tokens, such as these globally available ones:

<code>\$System</code>	System folder.
<code>\$System64</code>	System folder for 64-bit operating system in Windows.
<code>\$Fonts</code>	Font folder on the local file system.
<code>\$ExtensionSpecificEMStore</code>	Folder that stores extension-specific files.

Extension Manager resolves these tokens to the appropriate location for the current platform and operating system. For example, in a Windows 32-bit environment, both `$system` and `$system64` resolve to the path "C:\Windows\system32\", while in a Windows 64-bit environment, `$system` resolves to the path "C:\Windows\SysWOW64\", and `$system64` resolves to the path "C:\Windows\system32\". In Mac OS, `$system` resolves to the path "/System/".

Many other application-specific path tokens are available as well, that allow you to specify locations relative to the target product's installation location, script or plug-in locations, and so on. For a complete list of tokens that are defined in different product environments and their values in Mac OS and in Windows 7 (x64), see ["Appendix A: Path Tokens" on page 39](#).

Making your extension updatable

Extension Manager CS5 and higher supports updatable extension packages, which allow an end user to update an installed extension when there is new version available. To enable this feature, include the [update](#) element in the MXI configuration file for your extension, with a link that points to the update information.

Update-information file XML elements

Provide update information in the form of an XML file that tells Extension Manager whether a new update is available, and provides information about how to update a particular extension.

The update information file must be UTF-8 encoded, and contain the following elements:

version

Required. Contains the version number of the latest version of the extension, in the format *major[.minor[.build[.misc]]]*, where major, minor, and build values are positive integers, and the optional misc value is alphanumeric. For example, "2", "2.1", or "2.1.2. test3".

The major version number should be incremented when you make substantial changes to the extension. The minor version number should be incremented for smaller changes, and the build number can be incremented for each build between releases.

download

Required. Contains a URL starting with "http" or "https". This can specify:

- ▷ A link to the latest extension itself, in the form of a ZXP or MXP installation file link;
—or—
- ▷ A web page containing descriptions and instructions for how to obtain and install the latest extension.

description

Required. A short description about what is new or changed in the update. Must contain a CDATA section, which you can format with any HTML tags. This is the update description that appears in Extension Manager. It can also point to more detailed release notes that you make available online.

Attributes

[url]

url

Optional. A URL for a web page that contains detailed release notes.

Example update information file

```
<ExtensionUpdateInformation>
  <version>1.5.0</version>
  <download>http://www.mysite.com/extensions/myExtension.zxp</download>

  <description url="http://www.mysite.com/extensions/myExt_15releasenotes.htm">
    <![CDATA[
      The 1.5 version fixes known problems.<br>
      New features include the ability to download updates. ]]>
  </description>

</ExtensionUpdateInformation>
```

The update process

When Extension Manager is launched, it follows the link in the [update](#) element of the MXI configuration file to retrieve the update information file. It then compares the `version` value in the update file with that of the installed version. If a newer version of your extension is available, it prompts the user to initiate the update.

If the user chooses to update your extension, Extension Manager goes to the URL given in the `download` element.

- ▶ If that link points to an installation package, Extension Manager downloads and installs the new version automatically.
- ▶ Otherwise, Extension Manager opens the given page in the default web browser.

General MXI elements

author

Required. Name of the author of the extension.

ATTRIBUTES: name, [author_resid]

name

Required. The author's name, a VARCHAR data type with a limit of 255 characters.

author_resid

Optional. A localized-string identifier for the author's name to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite 6 SDK*.

dependency

Optional. A container for one or more `extension` elements for other extensions that must be loaded for this extension to be loaded.

Contents If present, must contain at least one `extension` element.

extension

Each element describes one extension that must be loaded for this one to be loaded, along with required version information, if any. For example:

```
<dependency>
  <extension name="Sample1" />
  <extension name="Sample2" minversion="1.0" />
  <extension name="Sample3" minversion="1.0" maxversion="2.0" />
  <extension name="Sample4" version="2.1"/>
</dependency>
```

ATTRIBUTES: name, [version, minversion, maxversion]

name

Required. The name of the extension that depends on these listed extensions.

version

Required. The version number of the latest version of the extension, in the format `major[.minor[.build[.misc]]]`, where major, minor, and build values are positive integers, and the optional misc value is alphanumeric. For example, "2", "2.1", or "2.1.2.test3".

minVersion, maxVersion

Optional. Minimum and/or maximum versions of this extension that can be used, in the format `major[.minor][.build]`. If only the minimum is specified, you can use that version or higher; if both are specified, the version must fall into the (inclusive) range.

description

Required. Contains or points to HTML text that describes what the extension does or is used for. The text appears in the Extension Manager when the extension is selected.

- ▶ If the computer is online and a URL is provided, Extension Manager displays the referenced page.
- ▶ Otherwise, if a path to a local web page is provided, Extension Manager displays that page.
- ▶ Otherwise, Extension Manager displays the CDATA contents.

Contents Must contain a CDATA section, which you can format with any HTML tags. If text colors are not specified, the background is gray (62 62 62) and the text is black.

To display double-byte characters, include "charset=UTF-8". For example:

```
<description>
  <![CDATA[<meta http-equiv=Content-Type content="text/html; charset=UTF-8"><br>
    This is a sample Exchange item.<br>
    It is a sample library containing a single button.]]>
</description>
```

ATTRIBUTES: [href, resid, source, href_resid, source_resid]

href

Optional. A URL for the description to display in the Extension Manager. The value must start with either "http://" or "https://".

source

The path to an HTML file on the local computer, relative to the Extension Manager variable `$ExtensionSpecificEMStore`. See ["Path tokens" on page 7](#).

resid

Optional. A localized-string identifier for the description to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

href_resid

Optional. A localized-string identifier for the associated URL (`href` value) to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

source_resid

Optional. A localized-string identifier for the associated HTML file (`source` value) to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite CS6 SDK*.

files

Optional. A container for one or more [file](#) elements that describe specific files to be installed as part of the extension.

To create an extension as part of a bundle or framework in Mac OS, use a format like this, without wildcards:

```
<files>
  <file source="sourceFolder" destination="$photoshop/" />
</files>
```

or

```
<files>
  <file source="sourceFolder/" destination="$photoshop/" />
</files>
```

ATTRIBUTES: [xml:lang, products, default-file-type]

xml:lang

Optional. A locale code, such as "en_US". The language for this group of files. Extension Manager installs these files only when this is the current language. If Extension Manager cannot determine the current language, it ignores this value and installs the files.

products

Optional. The products for which these files should be installed. If not specified, the files are installed for all products. A comma-separated list of products; see [product](#).

default-file-type

Optional. The default type of the contained files. One of:

- ▷ csxs: A CS extension package.
- ▷ plugin: A native plug-in.
- ▷ ordinary: Ordinary files receive no special processing by Extension Manager.

This value is overridden by the `file-type` attribute of a contained file.

file

Describes a specific file to be installed as part of this extension.

If all files in a folder must be installed in the same destination folder, you do not need to specify each file. Use a single the `file` element to specify the folder. A slash at the end of the `source` value indicates that it is a folder, and all contained files are automatically included in the package. For example:

```
<file source="Resources/" destination="$dreamweaver/configuration/Resources" />
```

- ▶ Contained in a [files](#) element.

ATTRIBUTES (CHILD ELEMENT): source, destination, [products, platform, shared, systemfile, win-extension, isresourcefile, file-type, addToTrustFile, minVersion, maxVersion]

source

Required. The file name, and the path in which it is found relative to the installation file. Files can be in different folders. See ["Location specification" on page 7](#). For a folder whose entire contents should be included, this value ends with a slash ("/") character.

destination

Required. The name of the destination folder, in which to place the file during installation. If this folder does not exist, Extension Manager creates it during installation. This does *not* include the file name, which is specified by the `source` attribute. Generally, destination folders should be inside the application's configuration folder. See ["Location specification" on page 7](#).

products

Optional. The products for which this file should be installed. If not specified, the file is installed for all products. A comma-separated list of products; see [product](#).

platform

Optional. The platform for which this file is intended, one of "mac" or "win". This allows you to supply different versions of a file for different platforms. If not supplied, the file is installed on both platforms.

shared

Optional. True if file is used by more than one extension. Default is false.

When you use the Extension Manager to remove an extension, a shared file associated with that extension is not deleted as long as other installed extensions refer to that file.

If you install a newer version of a shared file and another extension is using the old version of the file, the new shared file must be backward compatible with the other extension, or must have a new filename so that the other extension continues to work properly.

systemfile

Optional. True if the file is used by anything other than extensions. For example, some extensions provide new versions of DLLs or other system files, or files that are used by other applications.

When you use the Extension Manager to remove an extension, a file marked as a system file is not deleted, even if no other extensions use it, and regardless of the `shared` attribute value.

win-extension

Optional. A file-name extension to use when a file generated in Mac OS that does not include the Windows extension, such as `.fla` or `.htm`.

If you create a file on Windows that does include the extension, such as "mypage.htm", and install it in Mac OS, this value is not needed.

If a platform attribute value is supplied, this attribute is ignored.

isresourcefile

Optional. True if this is a resource file containing language-specific text strings. Default is false.

Place resource files in a folder with the name `installerPrefix.mxi_Resources`. When the MXI file is loaded, Extension Manager copies this folder into the top-level Extension Manager folder, where it then looks for text strings. For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite CS6 SDK*.

file-type

Optional. The file type. One of:

▷ `csxs` : A CS extension package.

- ▷ `plugin`: A native plug-in.
- ▷ `ordinary`: Ordinary files receive no special processing by Extension Manager.

Default is the value specified in the container `files` element.

addToTrustFile

Optional. Boolean. When "true" for a SWF or HTML file, Extension Manager registers this file as locally trusted in the Flash Player Trust folder when it installs this extension. When the extension is removed, Extension Manager unregisters the file. Default is false.

minVersion, maxVersion

Optional. The minimum and maximum versions of the product in which this file can be installed. For example, if `minVersion` is 9 and `maxVersion` is 10, the file is not installed in product version 8 or 11. Use the same format as for the `version` attribute of [macromedia-extension](#), `major[.minor][.build]`.

license-agreement

Optional. Allows a third-party developer to include a license agreement with an extension. If supplied, the contents of this element are displayed under the heading Third Party License, at the end of the Adobe new-extension installation license.

Contents Must contain a CDATA section, which you can format with any HTML tags. If text colors are not specified, the background is gray (58 58 58) and the text is off-white (E0 E0 E0).

ATTRIBUTES: [`resid`]

resid

Optional. A localized-string identifier for the license agreement to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite CS6 SDK*.

Example

```
<license-agreement>
  <![CDATA[You are about to install an Extension from the Adobe Exchange.
    The Adobe Exchange is an area of the adobe.com website that allows
    third parties to submit extensions for posting to adobe.com.]]>
</license agreement>
```

macromedia-extension

The top-level container for an MXI file. The opening tag must be on the first line of the file.

Child elements

All other elements are children of this container. It must contain these elements:

[products](#), [product](#)

If you are installing files, it must contain these elements:

[files](#), [file](#)

Optional contained elements:

[author](#)
[dependency](#)
[description](#)
[license-agreement](#)
[ui-access](#)
[update](#)

ATTRIBUTES: *id, name, version, [mxiversion, icon, requires-restart, force-quit, ismultilingual, name_resid, type, plugin-manager-type, show-files]*

id

Required. The unique identifier assigned by Adobe when you submit your extension. Do not modify.

name

Required. The name of the extension, displayed in Extension Manager. A `VARCHAR` data type with a limit of 255 characters. You can localize the name using the `name_resid` attribute.

version

Required. The version number of the latest version of the extension, in the format `major[.minor[.build[.misc]]]`, where major, minor, and build values are positive integers, and the optional misc value is alphanumeric. For example, "2", "2.1", or "2.1.2. test3".

mxiversion

Optional. The version of MXI used for this file. Default is 1.0.

Extension Manager versions CS5.5 and later support configuration files created for earlier versions. If the installed Extension Manager does not support this version, an alert informs the user that a later version is required.

xmanversion

Optional. The minimum version of Extension Manager needed to install this extension. Default is 1.0.

Specify only if the extension is not compatible with newer versions of Extension Manager. See also the `maxversion` attribute of [product](#) element.

icon

Optional. The path to a customized icon for this extension, to display in Extension Manager. In order for a custom icon to be displayed, the icon file must be installed in the folder specified by `ExtensionSpecificEMStore` (see ["Location specification" on page 7](#)). For example:

```
<file source="myIcon.png" destination="$ExtensionSpecificEMStore" />
```

Icons are only shown in application versions CS4 and later. For CS3 or earlier, use the `type` attribute.

If not specified, a default icon is used.

requires-restart

Optional. When true, the target product must be restarted after the extension is installed. Default is false.

Superseded by `force-quit`, introduced in Extension Manager CS5.

force-quit

Optional. When true, the target product must quit before the extension can be installed or modified. The Extension Manager prompts the user to quit a running application before proceeding with the operation; in the case of Dreamweaver, the prompt provides an Exit Application button. Default is false.

Introduced in Extension Manager CS5, supercedes `requires-restart`.

ismultilingual

Optional. When true, installs language-specific files and applies localized UI strings in the extension. When false (the default), all multilingual elements are ignored.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite CS6 SDK*.

name_resid

Optional. A localized-string identifier for the extension name to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite CS6 SDK*.

type

Optional. Dreamweaver, Fireworks, and Flash only. The type of this extension. Values are not case-sensitive.

▷ Valid values for Dreamweaver:

`behavior, browserprofile, codehint codesnippet, coloringscheme, command, connection, datasource, dictionary, documenttype, encoding, flashbuttonstyle, flashelement, floater, insertbar, jsextension, keyboard shortcut, object, plugin, propertyinspector, report, referencebook, samplecontent, serverbehavior, serverformat, servermodel, site, suite, taglibrary, template, thirdpartytags, toolbar, translator, utility, query`

The value `suite` identifies a set of items released as a unit, with a single MXP file and a single MXI file. For example, you can create a set of objects, a command, a palette, and behaviors to make a process such as layer alignment easier to complete. Specify a single name and version for the entire suite.

▷ Valid values for Fireworks:

`autoshape, command, commandpanel, dictionary, keyboard shortcut, library, pattern, texture`

▷ Valid values for Flash:

`actionscript, flashcomponent, flashcustomaction, flashimporter, flashpanel, flashtemplate, keyboardshortcut, lesson, library, publishtemplate, sample, smartclip, utility`

`generatorobject` (*Flash 5 or earlier*)

plugin-manager-type

Optional. InDesign/InCopy CS5 and higher only. The type of included plug-ins, one of:

- ▷ `all-users`: When any user installs the plug-in or extension, it is available to all users. If any user disables or removes it, it is still available to other users. The destination of plug-in files must be the `$indesign/Plug-Ins` folder.
- ▷ `current-user`: When a user installs the plug-in or extension, it is available only to that user. If that user disables or removes it, it is no longer available to any user. The files should *not* reside in the `$indesign/Plug-Ins` folder.

show-files

Optional. CS5 and higher only. When true, Extension Manager's Advanced tab shows path information for all files installed with this extension. When false, the path information is not shown. Default is true.

Example

```
<macromedia-extension
  name = "My Command"
  version = "1.0.0"
  type = "command"
  requires-restart = "false" >
  mxiversion = "5.0"
  xmanversion = "5.0"
  icon = "command.png">

  <!-- all configuration elements-->

</macromedia-extension>
```

products

Required. A container for one or more [product](#) elements, each of which specifies an Adobe product in which this extension can be installed. The container has no attributes.

product

Specifies a product in which this extension can be installed.

► Contained in a [products](#) element.

ATTRIBUTES (CHILD ELEMENT): name, [version, primary, required, maxversion, familyname, platform, bit]

name

Required, except when `familyname` is supplied. The name of an Adobe product, a VARCHAR2 data type with a limit of 64 characters. One of:

```
Dreamweaver
Fireworks
Flash
Illustrator (Illustrator in Mac OS)
Illustrator32 (32-bit Illustrator in Windows)
Illustrator64 (64-bit Illustrator in Windows)
InCopy
InDesign
Photoshop (Photoshop in Mac OS)
Photoshop32 (32-bit Photoshop in Windows)
Photoshop64 (64-bit Photoshop in Windows)
Prelude
Premiere
```

version

The minimum version number of the product in which this extension can be installed. To install extensions for a particular CS version of a product, you must use the corresponding version of Extension Manager .

Valid version numbers are as follows:

Dreamweaver MX 2004	7	Illustrator CS4	14
Dreamweaver 8	8	Illustrator CS5	15
Dreamweaver CS3	9	Illustrator CS5.1	15.1
Dreamweaver CS4	10	Illustrator CS6	16
Dreamweaver CS5	11	InCopy CS4	6
Dreamweaver CS5.5	11.5	InCopy CS5	7
Dreamweaver CS6	12	InCopy CS5.5	7.5
Fireworks MX 2004	7	InCopy CS6	8
Fireworks 8	8	InDesign CS4	6
Fireworks CS3	9	InDesign CS5	7
Fireworks CS4	10	InDesign CS5.5	7.5
Fireworks CS5	11	InDesign CS6	8
Fireworks CS5.1	11.1	Photoshop CS4	11
Fireworks CS6	12	Photoshop CS5	12
Flash MX 2004	7	Photoshop CS5.1	12.1
Flash 8	8	Photoshop CS6	13
Flash CS3	9	Premiere Pro CS5	5
Flash CS4	10	Premiere Pro CS5.5	5.5
Flash CS5	11	Premiere Pro CS6	6
Flash CS5.5	11.5	Prelude	1
Flash CS6	12		

primary

Optional. True if this is the product for which the extension is primarily intended. For example, if the extension's user interface appears in Dreamweaver but the extension also uses Fireworks, Dreamweaver is the primary product. Can be true for more than one product.

required

Optional. True if this product is required for the extension to function properly. If the extension can function without this product, this should be false (the default).

If no product is explicitly required, the first product listed is assumed to be required.

maxversion

Optional. The highest version of this product in which this extension can be installed.

familyname

Optional. When supplied, `name` is not required. A value of "Photoshop" allows the extension to be installed in both Photohop and Photoshop Extended. A value of "Illustrator" is valid only for CS6, and allows the extension to be installed in both Illustrator and Illustrator (64-bit) in Windows.

platform

Optional. The platform on which this extension can be installed, oone of "mac" or "win". If not supplied, the extension is installed in this product on both platforms.

bit

Optional. The Win32 or Win64 version of this product for which this extension can be installed. One of "32" or "64". If not supplied, extension is installed in both the Win32 and Win64 versions of the product.

ui-access

Contains or points to HTML text that describes the extension's user interface. Together with the [description](#) element, determines the text that appears in the Extension Manager window when the extension is selected. You should include information about where to find the item in the product's user interface, as well as a brief description of the item's use.

- ▶ If the computer is online and a URL is provided, Extension Manager displays the referenced page.
- ▶ Otherwise, if a path to a local web page is provided, Extension Manager displays that page.
- ▶ Otherwise, Extension Manager displays the CDATA contents.

Contents

Must contain a CDATA section, which you can format with any HTML tags. The description is limited to 512 characters.

ATTRIBUTES: [`resid`]

resid

Optional. A localized-string identifier for the UI description to display in the Extension Manager. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite 6 SDK*.

Example

```
<ui-access>
  <![CDATA[You can run this extension by choosing<br>
    <b>Commands > Run My Extension.</b>]]>
</ui-access>
```

update

Provides update information for this extension. If supplied, Extension Manager checks the given site for updates, and when an update is available, prompts the user to update the extension.

ATTRIBUTES: `url`, [`method`]

url

Required. A URL for an extension update information file. The value must start with either "http://" or "https://". See details of this file format in ["Making your extension updatable" on page 8](#).

method

Optional. Reserved for future use in identifying an update-checking method. The only currently supported value is the default, `directlink`.

Product-specific MXI elements

These elements are used only by Dreamweaver (or Flash where specified), and are ignored when an extension is installed in any other target application.

Containment map

Elements that define changes to the menus, shortcuts, server behaviors or formats, or data sources for Dreamweaver are collected into various containers, depending on where the changes occur.

Container	Can contain
configuration-changes Top-level container for changes to different parts of the Dreamweaver (and Flash) configuration.	data-source-changes documenttype-changes extensions-changes insertbar-changes menu-insert server-behavior-changes server-format-changes server-format-definition-changes shortcut-insert toolbar-changes
data-source-changes	menu-insert
documenttype-changes	documenttype-insert
extensions-changes	extension-insert
ftp-extension-map-changes	ftp-extension-insert
insertbar-changes	insertbar-insert insertbar-item-insert
server-behavior-changes	menu-insert
server-format-changes	menu-insert
server-format-definition-changes	menu-insert
toolbar-changes	toolbar-insert toolbar-item-insert
toolbar-changes	toolbar-item-insert

configuration-changes

This is the top-level container for elements that define changes to the menus, shortcuts, server behaviors or formats, or data sources for Dreamweaver. The container element has no attributes.

Child elements

These elements can be children of this container:

- [data-source-changes](#)
- [documenttype-changes](#)
- [extensions-changes](#)
- [insertbar-changes](#)
- [menu-insert](#)
- [server-behavior-changes](#)
- [server-format-changes](#)
- [server-format-definition-changes](#)
- [shortcut-insert](#)
- [toolpanel-changes](#)

data-source-changes

Container for changes to menus in the `DataSources.xml` file in any of the Dreamweaver MX Configuration/DataSources/*servermodel* folders.

► Can contain any number of [menu-insert](#) elements.

ATTRIBUTES: `servermodelfolder`

servermodelfolder

Required. The name of the server-model folder in which the changes are to be made. Case sensitive. Can be the name of any installed server model, such as "ASP.NET_Csharp", "ASP.NET_VB", "ASP_Js", "ASP_Vbs", "ColdFusion", "UD4-ColdFusion", "PHP_MySQL" or "JSP".

documenttype-changes

Container for elements that describe changes to be made to the `MMDocumentTypes.xml` file.

- ▶ Can contain child elements [documenttype-insert](#).
- ▶ Contained in the [configuration-changes](#) element.

Example

```
<documenttype-changes>
  <documenttype-insert>
    <documenttype>
      ...
    </documenttype>
  </documenttype-insert>
</documenttype-changes>
```

documenttype-insert

One or more entries to be added to the `MMDocumentTypes.xml` file. Appends each specified tag library at the end of the file.

- ▶ Must contain one or more [documenttype](#) elements that describe tag libraries.
- ▶ Contained in the [documenttype-changes](#) element.

ATTRIBUTES: [`xml:lang`]

xml:lang

Optional. The language for this configuration value.

Example

```
<documenttype-insert>
  <documenttype>
    ...
  </documenttype>
</documenttype-insert>
```

documenttype

Describes a tag library to be added to the `MMDocumentTypes.xml` file.

- ▶ Contained in the [documenttype-insert](#) element.

For a complete description, see Dreamweaver help:

http://help.adobe.com/en_US/dreamweaver/cs/extend/index.html

extensions-changes

Container for elements that describe changes to be made to the `Extensions.xml` file, which contains file-name extensions that Dreamweaver can correlate with specific file types.

- ▶ Can contain child elements [extension-insert](#).
- ▶ Contained in the [configuration-changes](#) element.

Example

```
<extensions-changes>
  <extension-insert>
    <extension>
      ...
    </extension>
  </extension-insert>
</extensions-changes>
```

extension-insert

An entry to be added to the `Extensions.xml` file.

- ▶ Contained in the [extensions-changes](#) element.

ATTRIBUTES: `extension`, `description`, `[xml:lang]`

`extension`

Required. The file-name extension to add, a string such as "GIF" for a `.gif` extension.

`description`

Required. A description of the file type.

`xml:lang`

Optional. The language for this configuration value.

format

Describes a data format to be inserted into the Dreamweaver Format menu.

This element's attributes are difficult to write by hand. The best way to create a format tag is to use the interface within Dreamweaver. After you create a format, open the appropriate `Formats.xml` file in a text editor and copy the appropriate format tag, as generated by Dreamweaver. Paste this tag into the appropriate place in your extension installation file, then add the ID.

- ▶ Contained in the [configuration-changes](#) element.

ATTRIBUTES: `file`, `title`, `expression`, `strNamedFormat`, `nLCID`, `id`

This element's attributes are difficult to write by hand. We recommend that you create the format within Dreamweaver, then open the appropriate `Formats.xml` file in a text editor. Copy the appropriate `format` element, as generated by Dreamweaver. Paste this element into the appropriate place in your extension installation file, then add the ID.

id

Each format ID must be unique; it should start with a company name or other unique namespace prefix. Do not use DW as a prefix; it is reserved by the Dreamweaver. A convention is to use a domain name with the elements reversed; for example, `com.adobe`.

ftp-extension-map-changes

Container for elements that specify changes to the `FTPExtensionMap.txt` file in the `Configuration` folder.

- ▶ Can contain child elements [ftp-extension-insert](#).
- ▶ Contained in the [configuration-changes](#) element.

ftp-extension-insert

An entry to be added to the `FTPExtensionMap.txt` file in the `Configuration` folder. Defines whether a file of a given type is uploaded as an ASCII or binary file from Dreamweaver to an FTP server.

- ▶ Contained in the [ftp-extension-map-changes](#) element.

ATTRIBUTES: `extension`, `type`, `mac-creator`, `mac-file-type`

extension

Required. The file-name extension for this file type, such as GIF or PNG.

type

Required. The format to use to upload this type of file to the FTP server, one of "ASCII" and "Binary".

mac-creator

Required. The creator code for Mac OS. If you do not know the creator code, use "????".

mac-file-type

Required. The file type for Mac OS. If you do not know the file type, use "????".

Example

```
<ftp-extension-insert extension="JPG" type="ASCII"
  mac-creator ="MKBY" mac-file-type="JPEG" />
```

insertbar-changes

Contains elements that describe changes to the `Insertbar.xml` file. This file is automatically updated when objects are installed into Dreamweaver MX, so explicit change in the MXI file are not required.

- ▶ Can contain child elements [insertbar-insert](#) and [insertbar-item-insert](#).
- ▶ Contained in the [configuration-changes](#) element.

Example

```
<insertbar-changes>
  <insertbar-insert insertBefore|insertAfter="category_id">
    <category ...>
      <itemtype.../>
    </category>
  </insertbar-insert>

  <insertbar-item-insert
    insertBefore|insertAfter|appendTo|prependTo="category_or_item_id"
    category="category_id">
    <itemtype.../>
  </insertbar-item-insert>
</insertbar-changes>
```

insertbar-insert

Inserts a new category, as defined in the contained `category` element, into the `Insertbar.xml` file, at a position relative to an existing category.

- ▶ Must contain a [category](#) element.
- ▶ Contained in the [insertbar-changes](#) element.

ATTRIBUTES: `insertBefore|insertAfter`, [`xml:lang`]

`insertBefore|insertAfter`

The unique identifier of an existing category that determines the placement of this new one. One of these is required; do not specify both.

`xml:lang`

Optional. The language for this configuration value.

category

Defines a new category to be inserted into the `Insertbar.xml` file.

- ▶ Must contain an item-type element, such as a button. For a complete description, see Dreamweaver help: http://help.adobe.com/en_US/dreamweaver/cs/extend/index.html
- ▶ Contained in the [insertbar-insert](#) element.

ATTRIBUTES: `folder`, `id`

`folder`

Required. A subfolder in which to place the resources for the new category, which is added to the Dreamweaver configuration locations. See Dreamweaver documentation for details.

id
Required. A unique identifier for the new category.

Example

```
<insertbar-changes>
  <insertbar-insert>
    <category folder="Cat" id="DW_Insertbar_Cat">
      <button file="cat/cat.htm" id="DW_Insertbar_Cat_Cat1" image="cat
        cat.gif" />
    </category>
  </insertbar-insert>
</insertbar-changes>
```

insertbar-item-insert

Inserts an item into a specified position with respect to an existing item or category.

- ▶ Must contain an item-type element, such as a button, that describes the item to insert. Extension Manager verifies only that the XML is valid.
- ▶ Contained in the [insertbar-changes](#) element.

ATTRIBUTES: {appendTo|prependTo} | {insertBefore|insertAfter, category}, [xml:lang]

appendTo | prependTo

The unique identifier of an existing category that determines the placement of this item. Specify one of these, or the insertion and `category` attributes; do not use both placement methods.

insertBefore | insertAfter

The unique identifier of an existing item that determines the placement of this new one. Specify one of these with the `category` attribute, or one of the append/prepend attributes; do not use both placement methods.

category

The unique identifier of an existing category to which to append this item if the relative insertion item is not found.

xml:lang

Optional. The language for this configuration value.

menu-insert

Inserts a new menu feature into Dreamweaver's menu structure, at a position relative to an existing menu feature. The configuration file that it modifies is determined by the container.

Use this element to explicitly add your extension to menus, even if your extension is an object or a command; do not rely on Dreamweaver to automatically add objects or commands to its menus. To ensure that your extension is not automatically added to the menus, make an entry for your file in the `menus.xml` file, and add this line to the top of each of your extension's HTML files:

```
<!-- MENU-LOCATION=NONE -->
```

- ▶ Contained in a [configuration-changes](#), [server-behavior-changes](#), [server-format-changes](#), [server-format-definition-changes](#), or [data-source-changes](#) element.
- ▶ In all containers except [server-format-definition-changes](#), must contain one or more child elements [menu](#), [menubar](#), or [menuitem](#). Can also contain child elements [comment](#) and [separator](#).
 - ▷ When contained in [server-format-definition-changes](#), must contain one or more [format](#) child elements.

The attributes of this element determine where contained menus and items are placed with respect to existing menus and menu items. All child elements are inserted as a block, in the order they appear in this element.

A single `menu-insert` element can contain any number of menus or menu features. Submenus and menu items, however cannot be nested in menu or menubar elements; they must be inserted separately. Use a separate `menu-insert` element for each level of nesting; that is, use one `menu-insert` element for the parent menu, followed by another `menu-insert` element for the submenus in that menu, and another for the items in the submenus.

Example Suppose you want to add this menu with submenus and items to the main menu bar:

Get More Objects

Animals

Dog

Poodle

Cat

Use the following XML:

```
<menu-insert insertAfter="DWMenu_Insert_GetMoreObjects">
  <menu name="Animals" id="DWMenu_Insert_Animals" />
</menu-insert>

<menu-insert appendTo="DWMenu_Insert_Animals">
  <menu name="Dog" id="DWMenu_Insert_Animals_Dog" />
  <menuitem name="Cat" id="DWMenu_Insert_Animals_Cat" />
</menu-insert>

<menu-insert appendTo="DWMenu_Insert_Animals_Dog">
  <menuitem name="Poodle" id="DWMenu_Insert_Animals_Dog_Poodle" />
</menu-insert>
```

ATTRIBUTES: {insertAfter|insertBefore} | {appendTo, prependTo}, [skipSeparator, xml:lang]

insertBefore | insertAfter

The unique identifier of an existing item that determines the placement of this new one. Supply one of these, or one of the append/prepend attributes; do not use both placement methods.

Can identify an existing menu, menu bar, menu item, or format. No menu can appear to the right of the Help menu in Dreamweaver. If you insert a new menu after the Help menu, the application displays the new menu to the left of the Help menu.

appendTo | prependTo

The unique identifier of an existing menu or menubar that determines the placement of this item. Supply one of these, or one of the insertion attributes; do not use both placement methods.

Can identify an existing menu or menu bar. Cannot be a menu item or format.

skipSeparator

Optional. When true, and when the placement is `insertAfter`, inserts a separator in the parent menu before the new menu or item. Default is false. Ignored for other placement methods.

xml:lang

Optional. The language for this configuration value.

menu

Defines a menu or submenu to be inserted into the application's menu structure during installation of this extension.

- ▶ Contained in the [menu-insert](#) element.
- ▶ Does not contain any other elements, but must still be closed with the `</menu>` tag.

ATTRIBUTES: `name`, `id`, [`platform`]

name

Required. The display name of the menu. To set the menu's access key or mnemonic in Windows, use the underscore character (`_`) before the access letter; the underscore is automatically removed in Mac OS.

id

Required. A unique identifier for the new menu. Each ID must be unique; it should start with the Dreamweaver menu identifier for the parent menu.

platform

Optional. The platform in which this list appears, one of "mac" or "win". If not specified, the menu appears on both platforms.

Example

```
<menu name = "My _Menu", id = "DWMenu_myMenu", platform = "win">
</menu>
```

menubar

Defines a menu bar to be inserted into the application's menu structure during installation of this extension.

- ▶ Contained in the [menu-insert](#) element.
- ▶ Does not contain any other elements, but must still be closed with the `</menubar>` tag.

ATTRIBUTES: name, id, [platform]

name

Required. The display name of the menu. To set the menu's access key or mnemonic in Windows, use the underscore character (`_`) before the access letter; the underscore is automatically removed in Mac OS.

id

Required. A unique identifier for the new menu. Each ID must be unique; it should start with the Dreamweaver menu identifier for the parent menu.

platform

Optional. The platform in which this list appears, one of "mac" or "win". If not specified, the menu appears on both platforms.

Example

```
<menubar name = "My Menubar", id = "DWMenu_myMenubar" >
</menubar>
```

menuitem

Defines a menu item to be inserted into the application's menu structure during installation of this extension.

- ▶ Contained in a [menu-insert](#) element that is *not* contained in a [server-format-definition-changes](#) element.

ATTRIBUTES: name, id, file|command, [enabled, checked, dynamic, key, platform, arguments, resid:name]

name

Required. The display name of the item. To set the menu's access key or mnemonic in Windows, use the underscore character (`_`) before the access letter; the underscore is automatically removed in Mac OS. If two items have the same access key, it only works for the first one.

To make an underscore appear in the item's name, use percent (`%`) as an escape character; for example, "My%_Menu".

id

Required. A unique identifier for the new item. Each ID must be unique; it should start with the identifier of the parent menu.

key

Optional. A shortcut key for this menu item. See syntax details in Dreamweaver documentation.

platform

Optional. The platform in which this item appears, one of "mac" or "win". If not specified, the item appears on both platforms.

file

Optional. An HTML or JavaScript file which contains JavaScript code that implements the behavior of the menu item. The path is relative to the Configuration folder. Case sensitive.

When supplied, overrides the `command`, `enabled`, and `checked` attributes. Either `file` or `command` must be supplied.

command

Optional. JavaScript code that implements the behavior of the menu item. When `file` is supplied, it overrides the `command`, `enabled`, and `checked` attributes. Either `file` or `command` must be supplied.

enabled

Optional. JavaScript code that Dreamweaver executes before displaying the menu, to determine whether the menu item is enabled. The code must return a value of true to enable the item, or false to disable and dim it.

checked

Optional. JavaScript code that Dreamweaver executes before displaying the menu, to determine whether the menu item is in the selected state (checked). The code must return a value of true to select the item, or false to deselect it.

dynamic

Optional. True if the item's text and state are determined dynamically by the JavaScript in the associated file. Default is false. Ignored if no file is supplied.

arguments

Optional. Arguments to pass to the specified command file. Ignored if no file is supplied.

resid:name

Optional. A localized-string identifier for the item name to display. Used only when the extension is configured to be multilingual. The identified string must be defined in localization files.

For details of how to localize extensions, see Extension Builder help, or *Using the Adobe Creative Suite CS6 SDK*.

Example

```
<menuitem name = "My Menu Item", id = "DWMenu_myMenu_myMenuItem"
  key = "Cmd+Alt+Shift+M" platform = "mac"
  file = "commands/common/myMenuItem.htm" dynamic = "false" />
```

separator

Inserts a separator into a menu at the location determined by the parent insertion element and sibling menu or menu-item elements.

- ▶ Contained in the [menu-insert](#) element.

ATTRIBUTES: id, [platform]

id

Required. A unique identifier for the new item. Each ID must be unique; it should start with the Dreamweaver menu identifier for the parent menu.

platform

Optional. The platform in which this list appears, one of "mac" or "win". If not specified, the item appears on both platforms.

comment

Provides a comment about an item being inserted into the menu structure. Extension Manager inserts the contents as an XML comment element into the affected `menus.xml` file when it installs this extension.

- ▶ Contained in the [menu-insert](#) element.

Example

```
<comment>This command is part of the MyFeature extension.</comment>
```

server-behavior-changes

Container for elements that describe changes to menus in the `ServerBehaviors.xml` file in any of the Dreamweaver MX Configuration/ServerBehaviors/*servermodel* folders.

- ▶ Can contain [menu-insert](#) elements.
- ▶ Contained in the [configuration-changes](#) element.

ATTRIBUTES: `servermodelfolder`

servermodelfolder

Required. The name of the server-model folder in which the changes are to be made. Case sensitive. Can be the name of any installed server model, such as "ASP.NET_Csharp", "ASP.NET_VB", "ASP_Js", "ASP_Vbs", "ColdFusion", "UD4-ColdFusion", "PHP_MySQL" or "JSP".

server-format-changes

Container for elements that describe changes to menus in the `Formats.xml` file in any of the Dreamweaver MX Configuration/ServerBehaviors/*servermodel* folders.

- ▶ Can contain [menu-insert](#) elements.
- ▶ Contained in the [configuration-changes](#) element.

ATTRIBUTES: `servermodelfolder`

servermodelfolder

Required. The name of the server-model folder in which the changes are to be made. Case sensitive. Can be the name of any installed server model, such as "ASP.NET_Csharp", "ASP.NET_VB", "ASP_Js", "ASP_Vbs", "ColdFusion", "UD4-ColdFusion", "PHP_MySQL" or "JSP".

server-format-definition-changes

Container for elements that describe changes to menus in the `ServerFormats.xml` file in any of the Dreamweaver MX Configuration/ServerBehaviors/*servermodel* folders.

- ▶ Can contain [menu-insert](#) elements.
- ▶ Contained in the [configuration-changes](#) element.

ATTRIBUTES: `servermodelfolder`

servermodelfolder

Required. The name of the server-model folder in which the changes are to be made. Case sensitive. Can be the name of any installed server model, such as "ASP.NET_Csharp", "ASP.NET_VB", "ASP_Js", "ASP_Vbs", "ColdFusion", "UD4-ColdFusion", "PHP_MySQL" or "JSP".

shortcut-insert

Container for elements that describe additions to the shortcuts in the `menus.xml` file.

- ▶ Must contain either a [shortcutlist](#) or a [shortcut](#) element.
- ▶ Contained in the [configuration-changes](#) element.

ATTRIBUTES: `list_Id`, [`xml:lang`]

list_Id

The unique identifier of a shortcut list in which to add the contained [shortcut](#) element. Required for this case; do not use if this element contains a [shortcutlist](#) element.

xml:lang

Optional. The language for this configuration value.

shortcutlist

A shortcut list to add to the `menus.xml` file.

- ▶ Contained in the [shortcut-insert](#) element.

ATTRIBUTES: `id`, [`platform`]

id

Required. The unique identifier of the shortcut list, which matches the Dreamweaver window containing the menubar with which the shortcuts are associated. One of `DWMainWindow`, `DWMainSite`, `DWTimelineInspector`, and `DWHTMLInspector`.

platform

Optional. The platform in which this list appears, one of "mac" or "win". If not specified, the list appears on both platforms.

shortcut

A keyboard shortcut to add to the `menus.xml` file. The JavaScript to execute when the shortcut is activated can be contained directly in the `command` attribute, or in a specified file. One of these must be supplied; if both are supplied, the file takes precedence.

- ▶ Contained in the [shortcut-insert](#) element.

ATTRIBUTES: `key`, `id`, `command|file`, [`platform`]

key

Required. The key combination used to activate the associated command. Use the syntax specified for keyboard shortcuts in Dreamweaver documentation.

id

Required. A unique identifier for the new shortcut. Each ID must be unique; it should start with a company name or other unique namespace prefix. Do not use `DW` as a prefix; it is reserved by the Dreamweaver. A convention is to use a domain name with the elements reversed; for example, `com.adobe`.

command

Optional; if not supplied, `file` is required. JavaScript code to execute when the command is activated.

file

Optional; if not supplied, `command` is required. A file containing JavaScript code to execute when the command is activated.

platform

Optional. The platform in which this shortcut appears, one of "mac" or "win". If not specified, the shortcut appears on both platforms.

Example

```
<shortcut key = "Shift+F5" id = "ShortCutTest" command = "dw.newDocument()" />
```

taglibrary-changes

Container for elements that describe changes to the `TagLibraries.vtm` file.

- ▶ Contained in the [configuration-changes](#) element.
- ▶ Can contain [taglibrary-insert](#) elements.

taglibrary-insert

Appends a new tag-library entry to the `TagLibraries.vtm` file.

- ▶ Contained in the [taglibrary-changes](#) element.
- ▶ Must contain one or more [taglibrary](#) elements.

ATTRIBUTES: [`xml:lang`]

`xml:lang`

Optional. The language for this configuration value.

taglibrary

Describes a tag library to add to the `TagLibraries.vtm` file. Extension Manager verifies only that the XML structure is valid.

For a complete description, see Dreamweaver help:

http://help.adobe.com/en_US/dreamweaver/cs/extend/index.html

toolbar-changes

Container for elements that define changes to entries in toolbar configuration files.

- ▶ Contained in the [configuration-changes](#) element.
- ▶ Can contain any combination of [toolbar-insert](#) and [toolbar-item-insert](#) elements.

ATTRIBUTES: [file]

file

Optional. The name of the toolbar configuration file to modify. Default is `Toolbars.xml`.

Example

```
<toolbar-changes [file="file_name"]>
  <toolbar-insert>
    <toolbar ...>
      ...
    </toolbar>
  </toolbar-insert>

  <toolbar-item-insert
    insertBefore|insertAfter|appendTo|prependTo="toolbar_or_item_id"
    toolbar="toolbar_id">

    <itemtype.../>
  </toolbar-item-insert>
</toolbar-changes>
```

toolbar-insert

Describes a toolbar entry to append to the toolbar configuration file.

- ▶ Contained in the [toolbar-changes](#) element.
- ▶ Must contain one or more [toolbar](#) elements.

ATTRIBUTES: [xml:lang]

xml:lang

Optional. The language for this configuration value.

toolbar

Defines a toolbar. Extension Manager verifies only that the XML structure is valid. For a complete description, see Dreamweaver help: http://help.adobe.com/en_US/dreamweaver/cs/extend/index.html

toolbar-item-insert

Describes an item to be added to a toolbar, at a position relative to an existing item.

- ▶ Contained in the [toolbar-changes](#) element.
- ▶ Must contain one or more [itemtype](#) elements.

ATTRIBUTES: {insertAfter|insertBefore} | {appendTo, prependTo}, toolbar, [xml:lang]

insertBefore | insertAfter

The unique identifier of an existing item that determines the placement of this new one. Supply one of these, or one of the append/prepend attributes; do not use both placement methods.

If this item is not found in any existing toolbar, the new item is appended to the toolbar specified in the `toolbar` attribute.

appendTo | prependTo

The unique identifier of an existing toolbar that determines the placement of this item. Supply one of these, or one of the insertion attributes; do not use both placement methods.

toolbar

Required. The unique identifier of the toolbar to append to if the relative-placement item is not found.

xml:lang

Optional. The language for this configuration value.

itemtype

For a complete description, see Dreamweaver help:

http://help.adobe.com/en_US/dreamweaver/cs/extend/index.html

toolpanel-changes

Flash-only. Container for elements that modify the Flash tool panel.

- ▶ Contained in the [configuration-changes](#) element.
- ▶ Must contain one or more [toolpanel-item-insert](#) elements.

toolpanel-item-insert

Flash only. Inserts a tool into the Flash tool panel.

- ▶ Contained in the [toolpanel-changes](#) element.

ATTRIBUTES: name, [position, depth, xml:lang]

name

Required. The name of the tool to insert.

position

Optional. The 0-based index of the position at which to insert this tool, in the range [0..17]. If out of range or not supplied, the tool is inserted at the last position.

depth

Optional. The 0-based index of the depth in the menu at which to insert this tool. 0 is the top. If not supplied, or if the value is greater than the maximum depth, the tool is placed at the bottom of the menu.

xml:lang

Optional. The language for this configuration value.

Example

```
<toolpanel-changes>
  <toolpanel-item-insert name="polystar" position="7" />
</toolpanel-changes>
```

Appendix A: Path Tokens

When specifying locations, you can use path tokens. Different tokens are defined in different applications environments. The following tables show the tokens and the locations that they resolve to in a default installation in Mac OS and Windows 7 (64-bit).

Shared tokens

These tokens are available for all products.

Token	Path in Mac OS	Path in Windows 7 (64-bit)
\$system	/System	C:\Windows\SysWOW64
\$system64	N/A	C:\Windows\system32
\$fonts	/Library/Fonts	C:\Windows\Fonts
\$userhomefolder	/Users/<UserName>	C:\Users\<UserName>
\$userdatafolder	/Users/<UserName>/Library/ Application Support	C:\Users\<UserName>\AppData\Roaming (Flash only) C:\Users\<UserName>\ AppData\Local
\$shareddatafolder	/Library/Application Support	C:\ProgramData
\$sharedribsdatafolder	/Library/Application Support	C:\ProgramData
\$userlibraryfolder	/Users/<UserName>/Library	C:\Users\<UserName>\AppData\Roaming
\$userroamingfolder	/Users/<UserName>/Library/ Application Support	C:\Users\<UserName>\AppData\Roaming
\$sharedcommondatafolder	/Library/Application Support	C:\ProgramData
\$applicationsupport	/Library/Application Support	C:\Program Files (x86)
\$applicationsupport64	N/A	C:\Program Files
\$adobeccommon	/Library/Application Support/ Adobe	C:\Program Files (x86)\Common Files\Adobe
\$resourceroot	/Library/Application Support/ Adobe/Extension Manager CS6	C:\ProgramData\Adobe\Extension Manager CS6
\$startupscripts	/Library/Application Support/ Adobe/Startup Scripts CS6	C:\Program Files (x86)\Common Files\Adobe\Startup Scripts CS6
\$mediacoreplug-ins	/Library/Application Support/ Adobe/Common/Plug-ins/CS6/ MediaCore	C:\Program Files\Adobe\Common\ Plug-ins\CS6\MediaCore
\$installfolder	The folder where a specific product is installed. These examples show default installation locations; a product can be installed elsewhere.	
	/Applications/<Product Name>	C:\Program Files (x86)\Adobe\<Product Name>

Token	Path in Mac OS	Path in Windows 7 (64-bit)
\$ExtensionSpecificEMStore	Used for restore resource file, application-dependent	
\$locale	The installed language of application, application dependent	

Adobe Dreamweaver CS6 tokens

These additional tokens are defined when this product is installed.

Dreamweaver token	Path in Mac OS	Path in Windows 7 (64-bit)
\$dreamweaver	The folder where the product is installed. These examples show default installation locations; the product can be installed elsewhere.	
	/Applications/Adobe Dreamweaver CS6	C:\Program Files (x86)\Adobe\Adobe Dreamweaver CS6
\$dreamweaver/Configuration	/Users/<UserName>/Library/Application Support/Adobe/Dreamweaver CS6/\$LOCALE/Configuration	C:\Users\<UserName>\AppData\Roaming\Adobe\Dreamweaver CS6\\$LOCALE\Configuration
\$sharedextensionfolder	/Library/Application Support/Adobe/Dreamweaver CS6/\$LOCALE/Configuration/Extensions	C:\ProgramData\Adobe\Dreamweaver CS6\\$LOCALE\Configuration\Extensions
\$UserBinfolder	/Users/<UserName>/Library/Application Support/Adobe/Dreamweaver CS6/\$LOCALE	C:\Users\<UserName>\AppData\Roaming\Adobe\Dreamweaver CS6\\$LOCALE

Adobe Fireworks CS6 tokens

These additional tokens are defined when this product is installed.

Fireworks token	Path in Mac OS	Path in Windows 7 (64-bit)
\$fireworks	The folder where the product is installed. These examples show default installation locations; the product can be installed elsewhere.	
	/Applications/Adobe Fireworks CS6	C:\Program Files (x86)\Adobe\Adobe\Fireworks CS6
\$fireworksuser	/Users/<UserName>/Library/Application Support/Adobe/Fireworks CS6	C:\Users\<UserName>\AppData\Roaming\Adobe\Fireworks CS6

Adobe Flash CS6 tokens

For the product installation location, use the shared token `$installfolder`. These additional tokens are defined when this product is installed.

Flash token	Path in Mac OS	Path in Windows 7 (64-bit)
<code>\$RootFolder</code>	The folder where the product is installed. These examples show default installation locations; the product can be installed elsewhere.	
	<code>/Applications/Adobe Flash CS6</code>	<code>C:\Program Files (x86)\Adobe\Adobe Flash CS6</code>
<code>\$RootFolderLocal</code>	<code><product_root>/\$locale</code>	<code><product_root>\\$LOCALE</code>
<code>\$flash</code>	<code>/Users/<UserName>/Library/Application Support/Adobe/Flash CS6/\$LOCALE/Configuration</code>	<code>C:\Users\<UserName>\AppData\Local\Adobe\Flash CS6\\$LOCALE\Configuration</code>

Adobe Illustrator CS6 tokens

These additional tokens are defined when this product is installed.

Illustrator token	Path in Mac OS	Path in Windows 7 (64-bit)
<code>\$illustrator</code>	The folder where the product is installed. These examples show default installation locations; the product can be installed elsewhere.	
	<code>/Applications/Adobe Illustrator CS6</code>	<code>C:\Program Files [(x86)]\Adobe\Adobe Illustrator CS6</code>
<code>\$plugin</code>	<code><product_root>/Plug-ins.localized</code>	<code><product_root>\Plug-ins</code>
<code>\$presets</code>	<code><product_root>/Presets.localized</code>	<code><product_root>\Presets</code>
<code>\$scripting</code>	<code><product_root>/Scripting.localized</code>	<code><product_root>\Scripting</code>

Adobe InDesign CS6

These additional tokens are defined when this product is installed.

InDesign token	Path in Mac OS	Path in Windows 7 (64-bit)
<code>\$indesign</code>	The folder where the product is installed. These examples show default installation locations; the product can be installed elsewhere.	
	<code>/Applications/Adobe InDesign CS6</code>	<code>C:\Program Files (x86)\Adobe\Adobe InDesign CS6</code>
<code>\$indesign_user</code>	<code>/Users/<UserName>/Library/Application Support/Adobe/Adobe InDesign CS6</code>	<code>C:\Users\<UserName>\AppData\Roaming\Adobe\Adobe InDesign CS6</code>

Adobe Photoshop CS6 tokens

These additional tokens are defined when this product is installed.

Photoshop token	Path in Mac OS	Path in Windows 7 (64-bit)
\$photoshopappfolder	The folder where the product is installed. These examples show default installation locations; the product can be installed elsewhere.	
	/Applications/Adobe Photoshop CS6	C:\Program Files [(x86)]\Adobe\Adobe Photoshop CS6
\$imagestacks	<product_root>/Plug-Ins/Image Stacks	<product_root>\Plug-Ins\Image Stacks
\$3dengines	<product_root>/Plug-Ins/3D Engines	<product_root>\Plug-Ins\3D Engines
\$actions	<product_root>/Presets/Actions	<product_root>\Presets\Actions
\$additionalplugins	<product_root>/Locales/\$LOCALE/Additional Plug-ins	<product_root>\Locales\<LOCALE>\Additional Plug-ins
\$additionalpresets	<product_root>/Locales/\$LOCALE/Additional Presets	<product_root>\Locales\<LOCALE>\Additional Presets
\$automate	<product_root>/Plug-Ins/Automate	<product_root>\Plug-Ins\Automate
\$blackandwhite	<product_root>/Presets/Black and White	<product_root>\Presets\Black and White
\$brushes	<product_root>/Presets/Brushes	<product_root>\Presets\Brushes
\$channelmixer	<product_root>/Presets/Channel Mixer	<product_root>\Presets\Channel Mixer
\$colorbooks	<product_root>/Presets/Color Books	<product_root>\Presets\Color Books
\$colorrange	<product_root>/Presets/Color Range	<product_root>\Presets\Color Range
\$colorswatches	<product_root>/Presets/Color Swatches	<product_root>\Presets\Color Swatches
\$contours	<product_root>/Presets/Contours	<product_root>\Presets\Contours
\$curves	<product_root>/Presets/Curves	<product_root>\Presets\Curves
\$customshapes	<product_root>/Presets/Custom Shapes	<product_root>\Presets\Custom Shapes
\$digimarc	<product_root>/Plug-Ins/Digimarc	<product_root>\Plug-Ins\Digimarc
\$displacementmaps	<product_root>/Plug-Ins/Displacement Maps	<product_root>\Plug-Ins\Displacement Maps
\$duotones	<product_root>/Presets/Duotones	<product_root>\Presets\Duotones
\$effects	<product_root>/Plug-Ins/Effects	<product_root>\Plug-Ins\Effects
\$exposure	<product_root>/Presets/Exposure	<product_root>\Presets\Exposure

Photoshop token	Path in Mac OS	Path in Windows 7 (64-bit)
\$extensions	<product_root>/Plug-Ins/ Extensions	<product_root>\Plug-Ins\Extensions
\$fileformats	<product_root>/Plug-Ins/ File Formats	<product_root>\Plug-Ins\File Formats
\$filters	<product_root>/Plug-Ins/Filters	<product_root>\Plug-Ins\Filters
\$gradients	<product_root>/Presets/ Gradients	<product_root>\Presets\Gradients
\$hueandsaturation	<product_root>/Presets/ Hue and Saturation	<product_root>\Presets\Hue and Saturation
\$huesat	<product_root>/Presets/Hue Sat	<product_root>\Presets\Hue Sat
\$imagestacks	<product_root>/Plug-Ins/ Image Stacks	<product_root>\Plug-Ins\Image Stacks
\$imagestatistics	<product_root>/Presets/ Image Statistics	<product_root>\Presets\Image Statistics
\$importexport	<product_root>/Plug-Ins/ Import-Export	<product_root>\Plug-Ins\Import-Export
\$keyboardshortcuts	<product_root>/Presets/ Keyboard Shortcuts	<product_root>\Presets\Keyboard Shortcuts
\$layouts	<product_root>/Presets/Layouts	<product_root>\Presets/Layouts
\$lenscorrection	<product_root>/Presets/ Lens Correction	<product_root>\Presets\Lens Correction
\$levels	/Applications/Adobe Photoshop CS6/Presets/Levels	<product_root>\Presets\Levels
\$lightingstyles	<product_root>/Plug-Ins/Filters/ Lighting Styles	<product_root>\Plug-Ins\Filters\ Lighting Styles
\$lights	<product_root>/Presets/Lights	<product_root>\Presets\Lights
\$liquifymeshes	<product_root>/Presets/ Liquify Meshes	<product_root>\Presets\Liquify Meshes
\$localesfolder	<product_root>/Locales	<product_root>\Locales
\$localeskeyboard shortcuts	<product_root>/Locales/\$LOCALE/ Additional Presets/\$platform/ Keyboard Shortcuts	<product_root>\Locales\LOCALE\ Additional Presets\platform\ Keyboard Shortcuts
\$localesmenucustom ization	<product_root>/Locales/\$LOCALE/ Additional Presets/\$platform/ Menu Customization	<product_root>\Locales\LOCALE\ Additional Presets\platform\ Menu Customization
\$localesworkspaces	<product_root>/Locales/\$LOCALE/ Additional Presets/\$platform/ Workspaces	<product_root>\Locales\LOCALE\ Additional Presets\platform\ Workspaces
\$materials	<product_root>/Presets/Materials	<product_root>\Presets\Materials
\$matlab	<product_root>/MATLAB	<product_root>\MATLAB

Photoshop token	Path in Mac OS	Path in Windows 7 (64-bit)
\$measurements	<product_root>/Plug-Ins/ Measurements	<product_root>\Plug-Ins\Measurements
\$menucustomization	<product_root>/Presets/ Menu Customization	<product_root>\Presets\Menu Customization
\$meshes	<product_root>/Presets/Meshes	<product_root>\Presets\Meshes
\$optimizedcolors	<product_root>/Presets/ Optimized Colors	<product_root>\Presets\Optimized Colors
\$optimizedoutput Settings	<product_root>/Presets/ Optimized Output Settings	<product_root>\Presets\ Optimized Output Settings
\$optimizedsettings	<product_root>/Presets/ Optimized Settings	<product_root>\Presets\Optimized Settings
\$panels	<product_root>/Plug-Ins/Panels	<product_root>\Plug-Ins\Panels
\$parser	<product_root>/Plug-Ins/Parser	<product_root>\Plug-Ins\Parser
\$patterns	<product_root>/Presets/Patterns	<product_root>\Presets\Patterns
\$photoshop	/Users/<UserName>/Library/ Application Support/Adobe/Adobe Photoshop CS6/Configuration	C:\Users\<UserName>\AppData\ Roaming\Adobe\Adobe Photoshop CS6\ Configuration
\$platform	Mac	Win
\$pluginsfolder	<product_root>/Plug-Ins	<product_root>\Plug-Ins
\$presetsfolder	<product_root>/Presets	<product_root>\Presets
\$reducenoise	<product_root>/Presets/ Reduce Noise	<product_root>\Presets\Reduce Noise
\$rendersettings	<product_root>/Presets/ Render Settings	<product_root>\Presets\Render Settings
\$replacecolor	<product_root>/Presets/ Replace Color	<product_root>\Presets\Replace Color
\$scripts	<product_root>/Presets/Scripts	<product_root>\Presets\Scripts
\$selectivecolor	<product_root>/Presets/ Selective Color	<product_root>\Presets\Selective Color
\$shadowhighlight	<product_root>/Presets/ Shadow Highlight	<product_root>\Presets\Shadow Highlight
\$smartsharpen	<product_root>/Presets/ Smart Sharpen	<product_root>\Presets\Smart Sharpen
\$styles	<product_root>/Presets/Styles	<product_root>\Presets\Styles
\$textures	<product_root>/Presets/Textures	<product_root>\Presets\Textures
\$tools	<product_root>/Presets/Tools	<product_root>\Presets\Tools
\$variations	<product_root>/Presets/ Variations	<product_root>\Presets\Variations
\$volumes	<product_root>/Presets/Volumes	<product_root>\Presets\Volumes

Photoshop token	Path in Mac OS	Path in Windows 7 (64-bit)
\$webphotogallery	<product_root>/Presets/ Web Photo Gallery	<product_root>\Presets\Web Photo Gallery
\$widgets	<product_root>/Presets/Widgets	<product_root>\Presets\Widgets
\$workspaces	<product_root>/Presets/ Workspaces	<product_root>\Presets\Workspaces
\$zoomify	<product_root>/Presets/Zoomify	<product_root>\Preset\Zoomify