



Add-In Manager
User's Guide v2.0



Table of Contents

Terms and Conditions.....	3
System Requirements.....	3
Example Add-In Management Workflow	3
Create Development Source Folder	3
Define a New Add-In.....	4
Update General Info	5
Add Menu Item(s)	6
Building and Installing the JMP Add-In.....	7
Deploy Add-In to TEST.....	8
Deploy Add-In to PROD	8
Action Buttons	9
Add-In List	11
Status Icons.....	11
Re-Order Add-In List	11
Collapse the Add-In List.....	11
Preferences	12
General.....	12
Add-In	12
Deployment	13
Custom Metadata	14
Archive	14
Status Icons.....	15
General Info Tab.....	15
Addin.def Information.....	15
Folder Locations.....	16
Menu Items Tab.....	17
Menu Item Tree.....	17
Details.....	20
Files Tab	21
Archive Tab.....	22
Properties	22
Install.....	23
Restore.....	23
Custom Metadata.....	23

Included Metadata	23
Adding New Custom Metadata.....	23
Defining Default Custom Metadata.....	24
Deployment	24
TEST	24
PROD.....	24
Update Check Script	24
Deployment Status Bar	25
How Archiving Works	25
Installed Tab	26
Enable/Disable Add-Ins	27
Register/Unregister Add-Ins.....	27
Miscellaneous	27
Sorting Files and Archive Tables.....	27
Preferences and Internal Table File Locations	27
Setting up Startup and Exit Scripts	27
JMPCUST File Backups.....	27

Terms and Conditions

SAS INSTITUTE INC. IS PROVIDING YOU WITH THE COMPUTER SOFTWARE CODE INCLUDED WITH THIS AGREEMENT ("CODE") ON AN "AS IS" BASIS, AND AUTHORIZES YOU TO USE THE CODE SUBJECT TO THE TERMS HEREOF. BY USING THE CODE, YOU AGREE TO THESE TERMS. YOUR USE OF THE CODE IS AT YOUR OWN RISK. SAS INSTITUTE INC. MAKES NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT AND TITLE, WITH RESPECT TO THE CODE.

The Code is intended to be used solely as part of a product ("Software") you currently have licensed from SAS Institute Inc. or one of its subsidiaries or authorized agents ("SAS"). The Code is designed to either correct an error in the Software or to add functionality to the Software, but has not necessarily been tested. Accordingly, SAS makes no representation or warranty that the Code will operate error-free. SAS is under no obligation to maintain or support the Code.

Neither SAS nor its licensors shall be liable to you or any third party for any general, special, direct, indirect, consequential, incidental or other damages whatsoever arising out of or related to your use or inability to use the Code, even if SAS has been advised of the possibility of such damages.

Except as otherwise provided above, the Code is governed by the same agreement that governs the Software. If you do not have an existing agreement with SAS governing the Software, you may not use the Code.

System Requirements

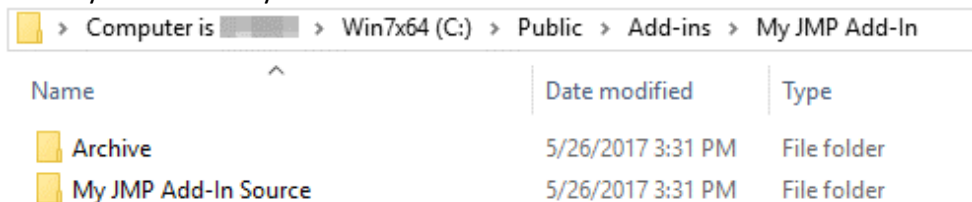
JMP 13 is required to use Add-In Manager. It can be used on either Windows or Mac operating systems.

Example Add-In Management Workflow

Create Development Source Folder

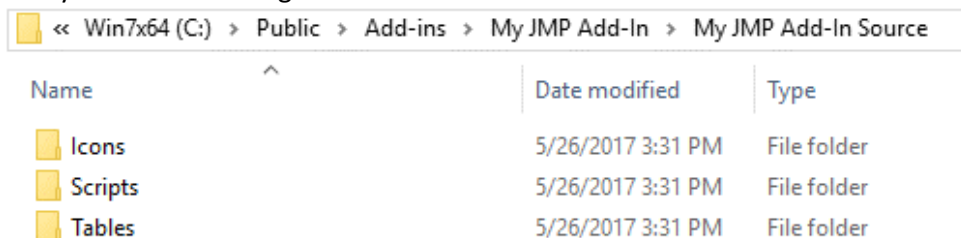
This step is just one suggested way to approach organizing your add-in files. You can do this part however works best for you.

1. Create a folder for your add-in. This folder will contain all versions of the add-in as well as the source files. Optionally create an Archive folder if you would like Add-In Manager to archive add-ins for you each time you build a new version.



Name	Date modified	Type
Archive	5/26/2017 3:31 PM	File folder
My JMP Add-In Source	5/26/2017 3:31 PM	File folder

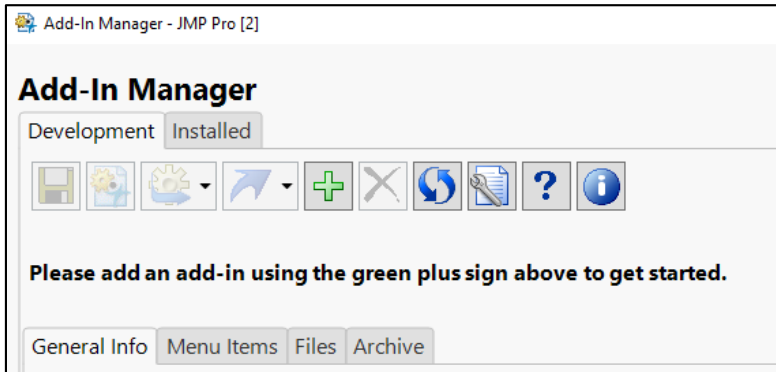
2. Optionally, within the source folder, create subfolders to group different types of files separately. This may be unnecessary for small add-ins, but can be very useful when there are many files used in a single add-in.




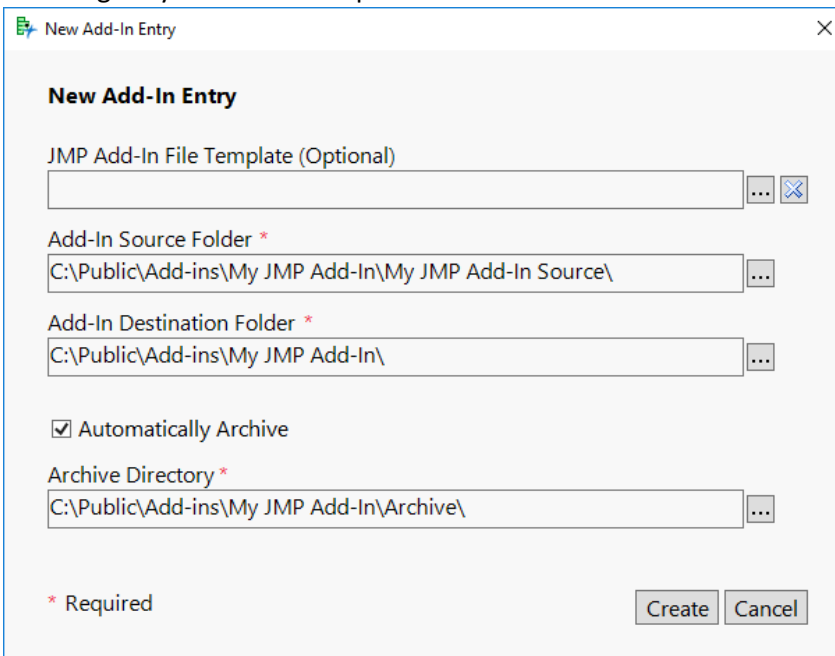
Name	Date modified	Type
Icons	5/26/2017 3:31 PM	File folder
Scripts	5/26/2017 3:31 PM	File folder
Tables	5/26/2017 3:31 PM	File folder

Define a New Add-In

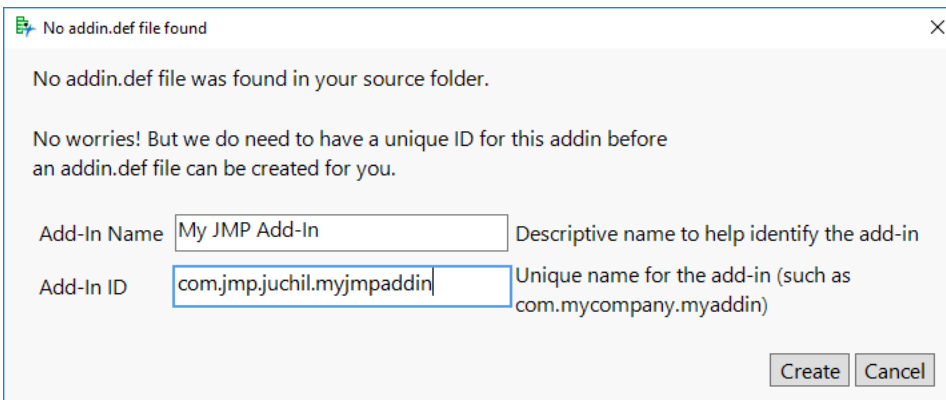
1. Before you define any add-ins, there will be a message as shown below.



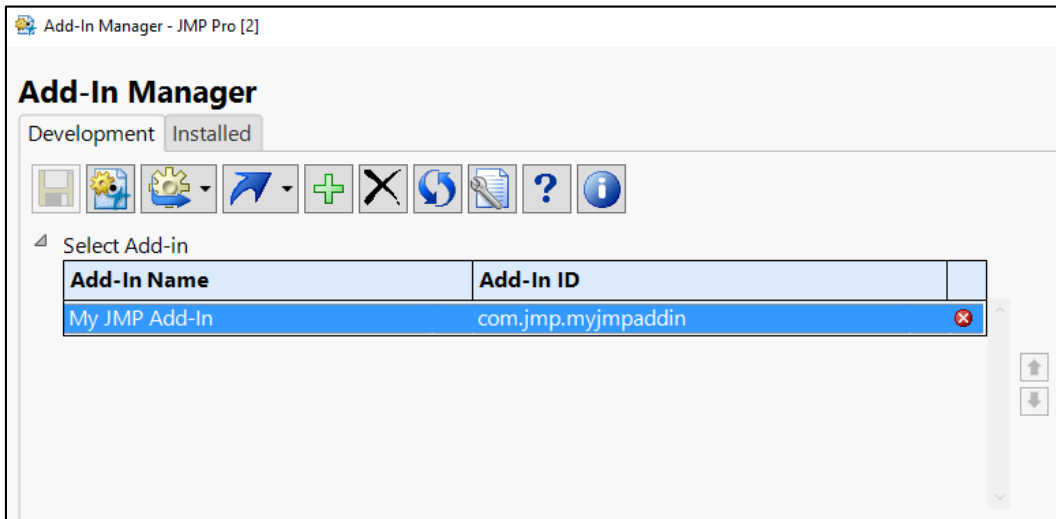
2. Click the New Add-In button  to define a new add-in.
3. Click the ellipsis buttons to add both a source and destination folders. You can optionally turn on archiving for your add-in and provide an archive folder.



4. If there was not an addin.def file located in the source folder, you will be prompted for a name and ID for the add-in. Enter a name and an ID and click "Create."

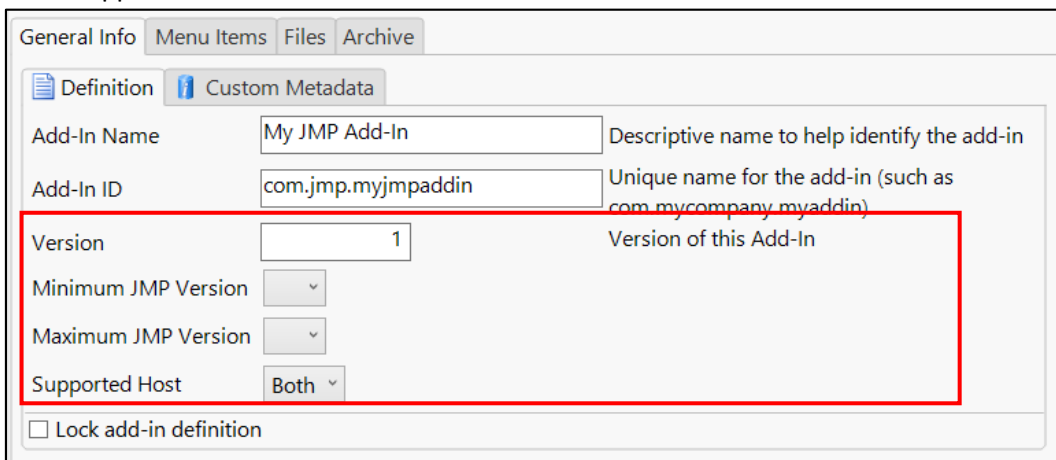


5. The add-in has now successfully been defined in Add-In Manager.

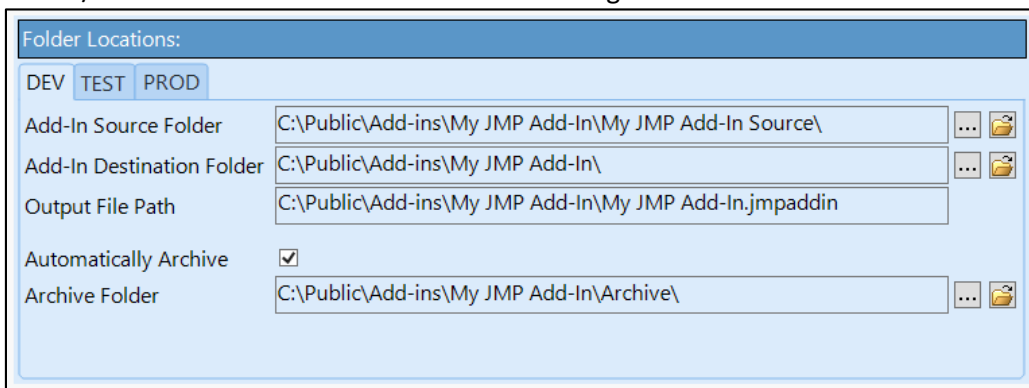



Update General Info

1. Under the General Info Tab, you can optionally add a version number, min/max JMP version, and a supported host.

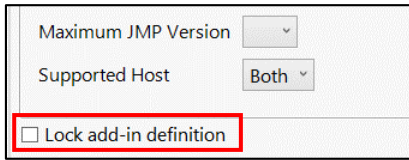


2. At the bottom of the tab, there is a Folder Locations section. Here you can update your source/destination folders as well as archive settings.



3. After updating, press the save button  to save this information.

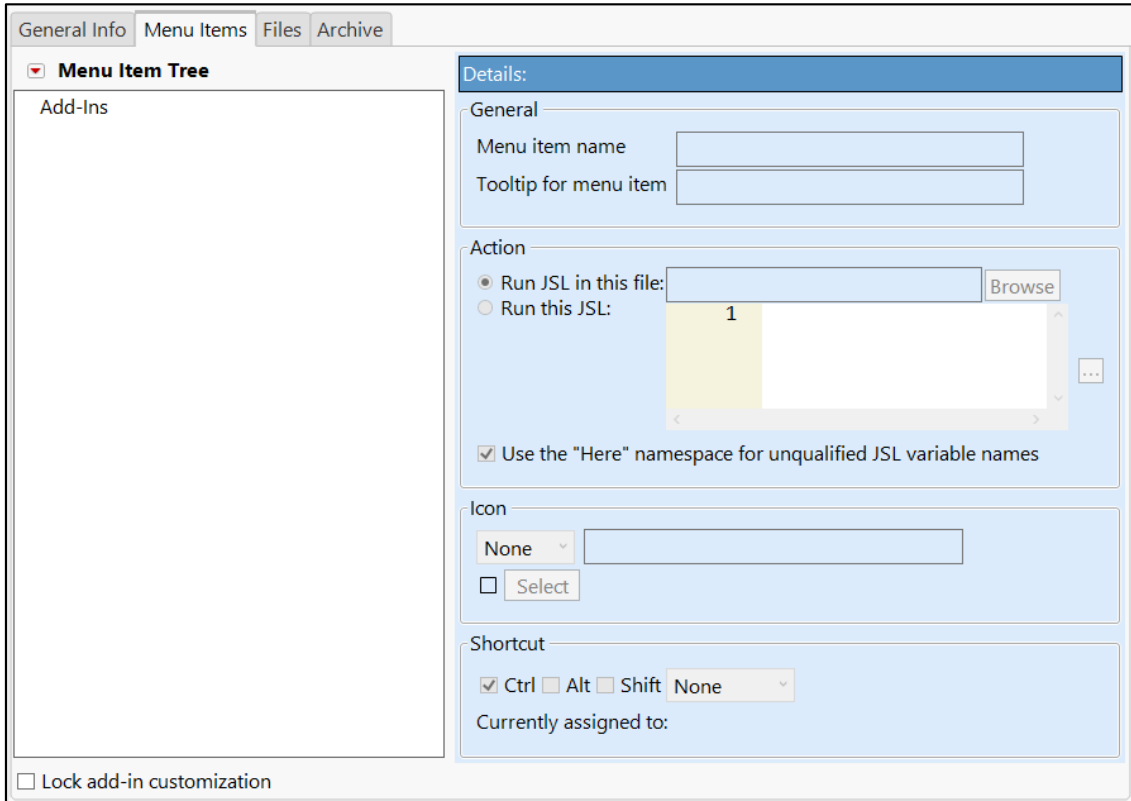
- Optionally, you can click the “Lock add-in definition” check box to ensure no changes are made to your addin.def file.



A screenshot of a configuration window showing three settings: 'Maximum JMP Version' with a dropdown arrow, 'Supported Host' with a dropdown menu set to 'Both', and 'Lock add-in definition' with an unchecked checkbox. The checkbox is highlighted with a red rectangular border.

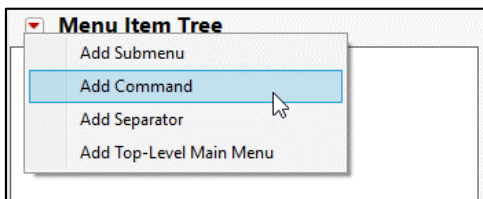
Add Menu Item(s)

- Click on the Menu Items tab to update the add-ins menu item commands. Before adding any submenus or commands, the tab appears as below.

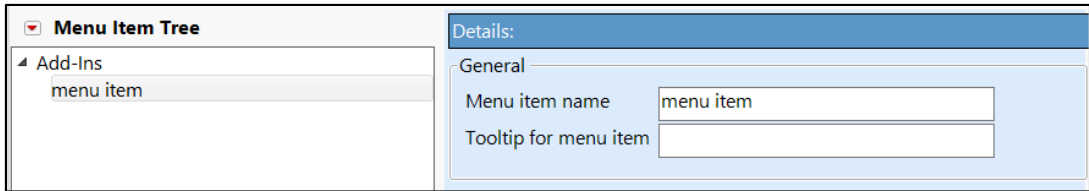


A screenshot of the 'Menu Items' tab in a software interface. The 'Menu Item Tree' on the left is empty under the 'Add-Ins' section. The 'Details' panel on the right contains the following fields: 'General' with 'Menu item name' and 'Tooltip for menu item' text boxes; 'Action' with radio buttons for 'Run JSL in this file:' (selected) and 'Run this JSL:', a file selection box with a 'Browse' button, and a checked checkbox for 'Use the "Here" namespace for unqualified JSL variable names'; 'Icon' with a 'None' dropdown and a 'Select' checkbox; and 'Shortcut' with checked 'Ctrl', 'Alt', and 'Shift' checkboxes and a 'None' dropdown. A 'Currently assigned to:' field is at the bottom. A 'Lock add-in customization' checkbox is at the bottom left of the window.

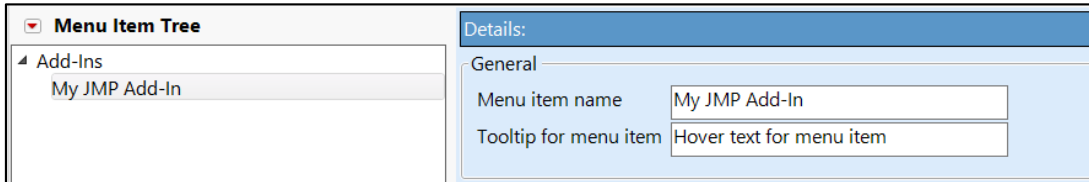
- You can then add either a submenu or command. In this example, there is just a single command. Click the red triangle menu next to “Menu Item Tree” and select “Add Command” to add a command menu item to the tree. After the command is created, it will have the below default values.



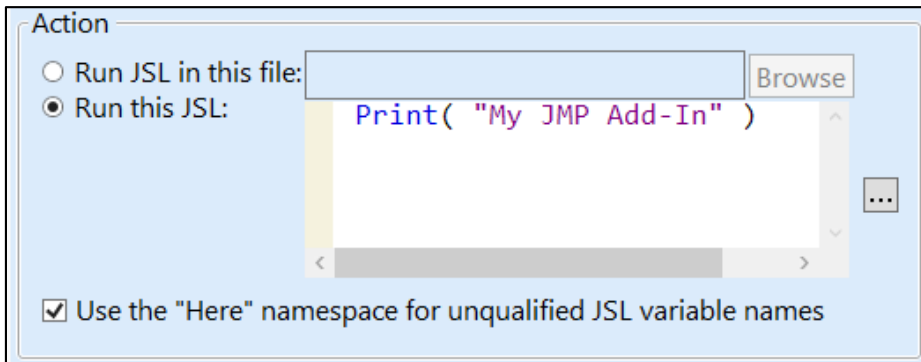
A screenshot showing the context menu for the 'Menu Item Tree'. The menu is open, displaying four options: 'Add Submenu', 'Add Command' (highlighted with a blue background and a mouse cursor), 'Add Separator', and 'Add Top-Level Main Menu'. A red triangle icon is visible next to the 'Menu Item Tree' header.




3. Rename the menu item and optionally give it a tooltip.



4. Add an action to run for this command either using a file or pasting the JSL in directly. In this example, we paste the script into the Run this JSL option as shown below. To do this, click the "..." button, paste the JSL into the script editor, and press OK. The results are shown below.




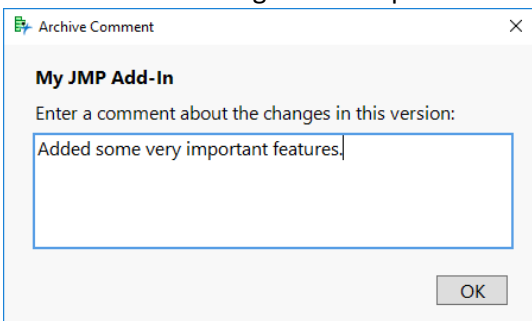
5. Optionally, set an Icon and Shortcuts for the menu item command.
6. When you are finished, press the save button  to save the menu item(s).


For more information about the Menu Item tab, see the [Menu Item section](#).

Building and Installing the JMP Add-In

With the add-in properly defined within Add-In Manager, you are now ready to build and install the add-in.

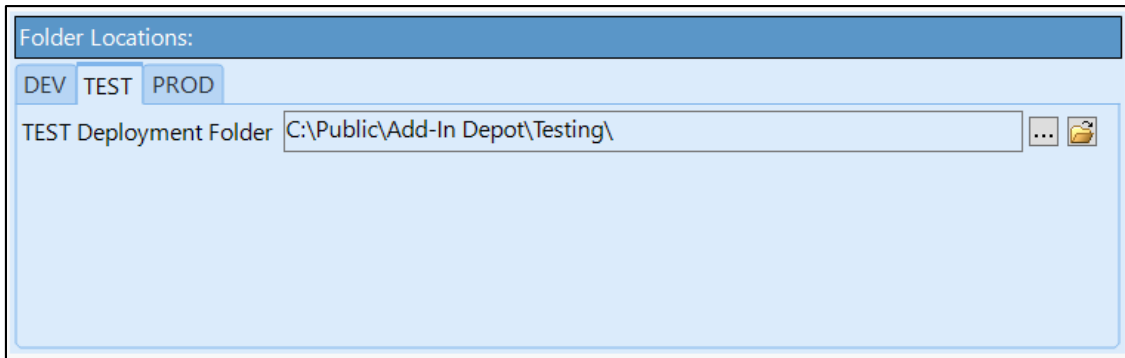
1. To build the add-in (i.e. creating the jmpaddin file), click the Build  button while the desired add-in is selected. When this button is clicked, the jmpaddin file is created within the Destination Folder.
2. If you have automatic archiving turned on, and the Prompt for Comment preference set, you will get a pop-up window with a text box for a comment. This is a good place to store information about what has changed in this specific build of the add-in.



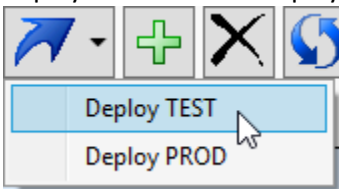
3. To install the add-in, simply click the Install  dropdown button and select “Install DEV” while the add-in is selected in the add-in list.

Deploy Add-In to TEST

1. Before deploying an add-in to a TEST location, ensure that you have a TEST deployment location defined in the TEST tab of the Folder Locations section. You can also set a default value within the [Deployment Preferences](#).

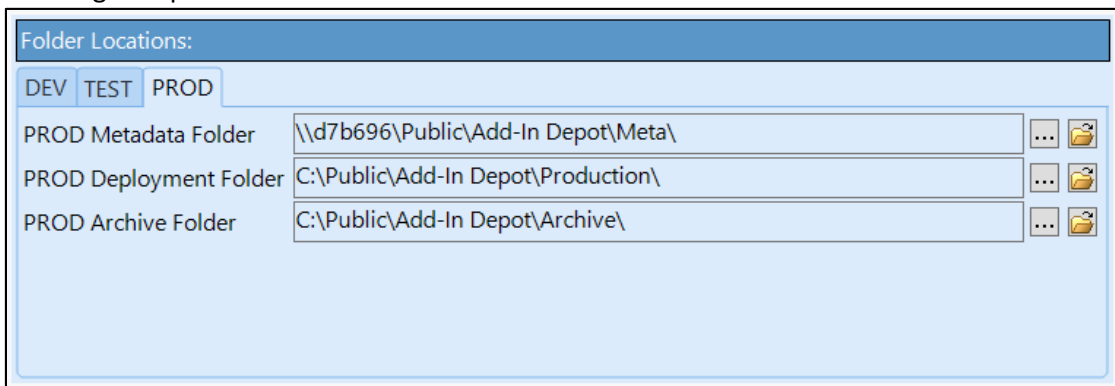


2. Click the Deploy dropdown button and select “Deploy TEST” to copy the add-in to the TEST Deployment Folder with a state set to “TEST.” The test add-in will get overwritten each time it is deployed to the TEST deployment folder.

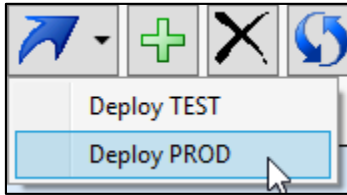


Deploy Add-In to PROD

1. Before deploying an add-in to a PROD location, ensure that you have a PROD deployment location defined in the PROD tab of the Folder Locations section. You can also set a default value within the [Deployment Preferences](#). If you use the update check script, make sure that the PROD Metadata Folder uses a server path as users will need to be able to access this path when checking for updates.













- Click the Deploy dropdown button and select “Deploy PROD” to copy the add-in to the PROD Deployment Folder with a state set to “PROD.” The previous PROD add-in will get archived to the PROD Archive folder and the new PROD add-in will take its place.



- When deploying to PROD, the table in the PROD Metadata Folder will be updated with the build date of the deployed add-in. This information can be queried using the updateCheck script, which you can turn on in the [Deployment Preferences](#). More information about the updateCheck script can be found in the [Update Check Script](#) section.

Action Buttons

Name	Button	Description
Save		<p>Saves the content from the General Info and Menu Item tabs.</p> <p>This includes doing the following:</p> <ul style="list-style-type: none"> • Creating a new addin.def file • Creating a new addin.jmpcust file • Updating Add-In Manager’s internal tables that contain folder locations.
Build		<p>Builds a new version of the currently selected add-in. This is done by zipping up the contents of the Source folder into a jmpaddin file within the Destination folder.</p> <p>If Automatic Archive is turned on, the current add-in file is archived before creating a new jmpaddin file. If Prompt for comment is on, there will be a prompt for a comment. If Automatic Archive is turned off, the old jmpaddin file is deleted in favor of the new source files.</p> <p>Check out the How Archiving Works section for more information about the archiving process.</p> <p>NOTE: This action automatically saves the currently selected add-in.</p>
Install		<p>Dropdown button to install the currently selected add-in.</p> <p>Dropdown options:</p> <ul style="list-style-type: none"> • Install DEV <ul style="list-style-type: none"> ○ Install the development version of the add-in, located in the Destination Folder (from the DEV tab in the Folder Locations section). • Install DEV <ul style="list-style-type: none"> ○ Install the test version of the add-in, located in the TEST Deployment Folder (from the TEST tab in the Folder Locations section).




		<ul style="list-style-type: none"> • Install PROD <ul style="list-style-type: none"> ○ Install the test version of the add-in, located in the TEST Deployment Folder (from the TEST tab in the Folder Locations section). <p>You will be prompted to install the add-in, as per the usual installation process in JMP.</p>
Deploy		<p>Dropdown button to deploy the selected add-in.</p> <p>Dropdown options:</p> <ul style="list-style-type: none"> • Deploy TEST <ul style="list-style-type: none"> ○ Deploy the add-in to the TEST Deployment Folder, as described in the TEST Deployment section. • Deploy PROD <ul style="list-style-type: none"> ○ Deploy the add-in to the PROD Deployment Folder, as described in the PROD Deployment section
Add New		<p>Define a new add-in. Involves setting source and destination folders. If the source files already contain an addin.def, it will be used to define the name and ID of the add-in. If the file is not found, a Name and ID are also prompted for.</p> <p>New in version 2.0: Specify a template JMP add-in file to use. You might use this for setting up a development folder if you only have the jmpaddin file, or if you want to establish a JMP add-in containing template scripts to use in all new add-ins. Note that this option requires you to specify an empty source folder as the contents of the specified jmpaddin file will be unzipped automatically into this folder.</p>
Delete		<p>Delete an add-in's definition from Add-In Manager.</p> <p>NOTE: This does not delete any files on your computer, only the add-in's record in the Add-In Manager's internal tables.</p>
Refresh		<p>Refresh the add-in list and contents in General Info/Menu Items/Files/Archive tabs.</p>
Preferences		<p>Open the Preferences window for Add-In Manager.</p> <p>Check out the Preferences section for more information</p>
Open Help		<p>Open the help documentation/User's Guide.</p>
About		<p>Open window containing information about Add-In Manager. Includes the license agreement, version, and a button to check for updates (pulled from the JMP Community).</p>

Add-In List

Status Icons

Status icons are used to tell you the current state of an add-in. These icons are located in the last column within the add-in list.

The severity of each item in the list below can be changed in the preferences window. Below are the options for severity.

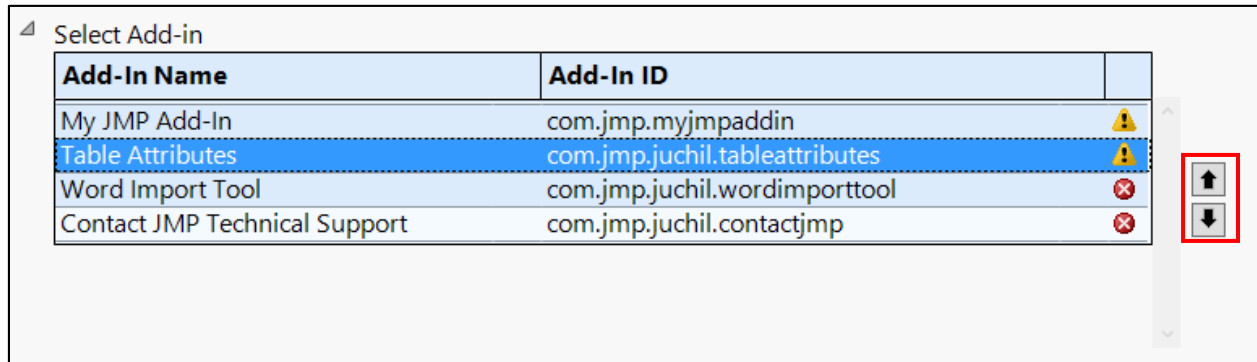
Icon	Severity
	Alert
	Warning
	All specified criteria are met.
N/A	Ignore – Does not check for this condition.

The items below are used for determining an add-in's status.

- Add-In file (.jmpaddin) does not exist
- Add-in source files are newer than the jmpaddin file
- addin.def file does not exist
- addin.jmpcust file does not exist
- Add-in is not installed
- Installed version of the add-in is out of date

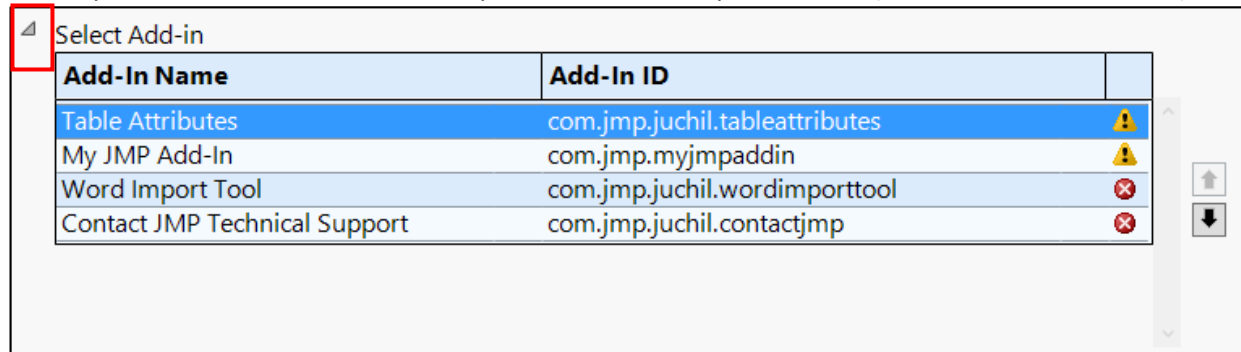
Re-Order Add-In List

To re-order the add-in list, select an add-in and press the up and down arrow buttons to the right of the list.

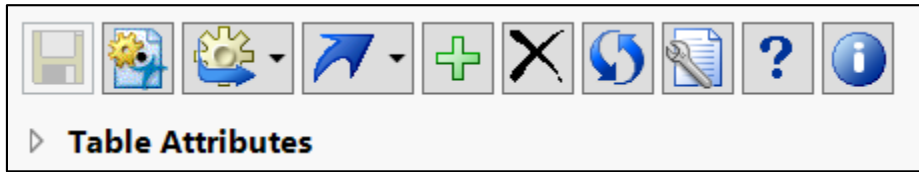


Collapse the Add-In List

To collapse the add-in list, click the collapse button in the top-left corner (below the action buttons).



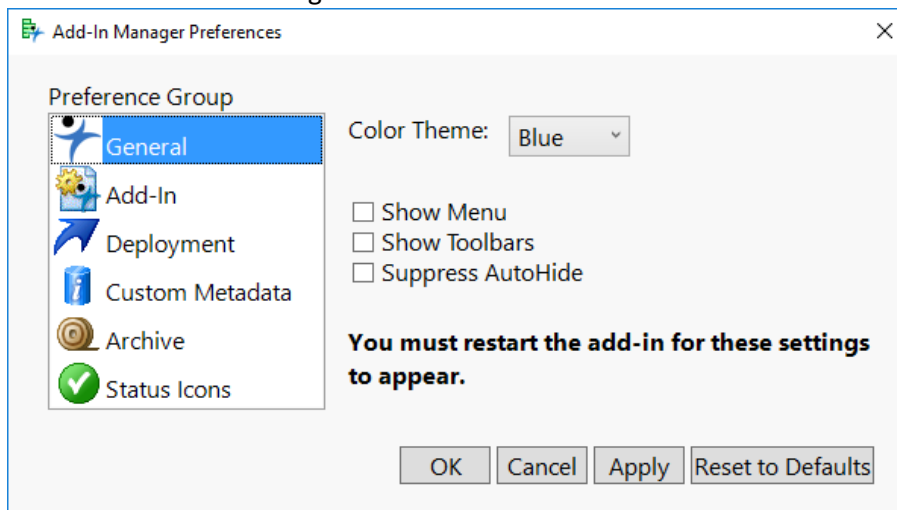
After collapsing the add-in list, the currently selected add-in's name will be shown next to the collapse button.



Preferences

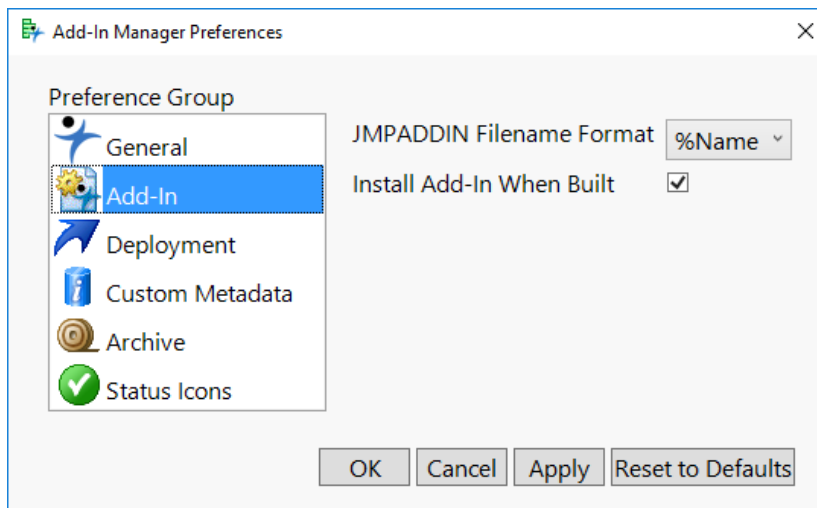
General

You can change the color theme used for the add-in within the General section. You can also choose to show the menus/toolbars and suppress the auto-hide of these menus/toolbars. You will need to restart the add-in for these changes to take effect.



Add-In

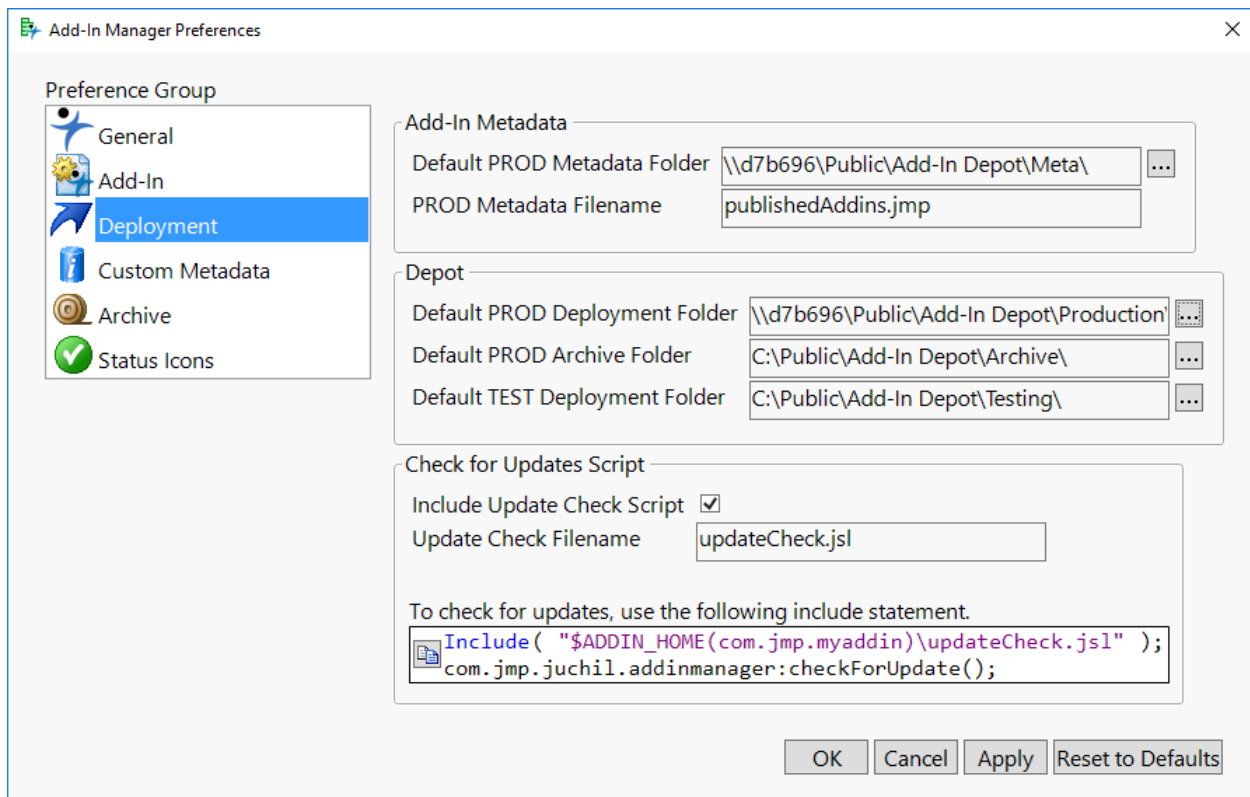
In the Add-In preference group, you can change the format of the jmpaddin file (either %Name or %Id). You can also turn in automatic installation of your add-in every time you build a new development version.



Deployment

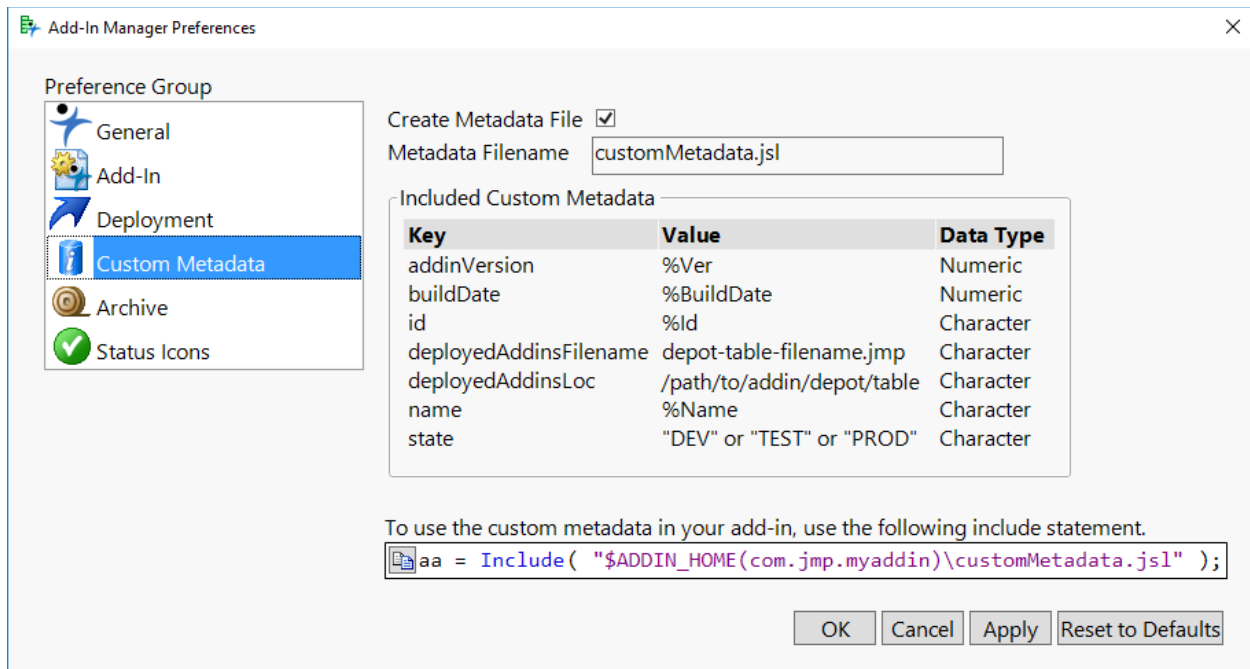
You can specify the following within the Deployment preference group:

- Default PROD Metadata Folder - Default Location for the add-in metadata table. This table is updated when deploying to PROD and can be changed for each add-in. This should be a network location so that other users can access the file using this path.
- Default PROD Deployment Folder – The location the jmpaddin file gets pushed to when deploying to PROD. This should be a network location so that other users can access the file using this path.
- Default PROD Archive Folder – The default location that PROD add-ins get archived.
- Default TEST Deployment Folder – The default location for deploying to TEST.
- Include Update Check Script – Yes or no, do you want the updateCheck.jsl script included in your add-ins



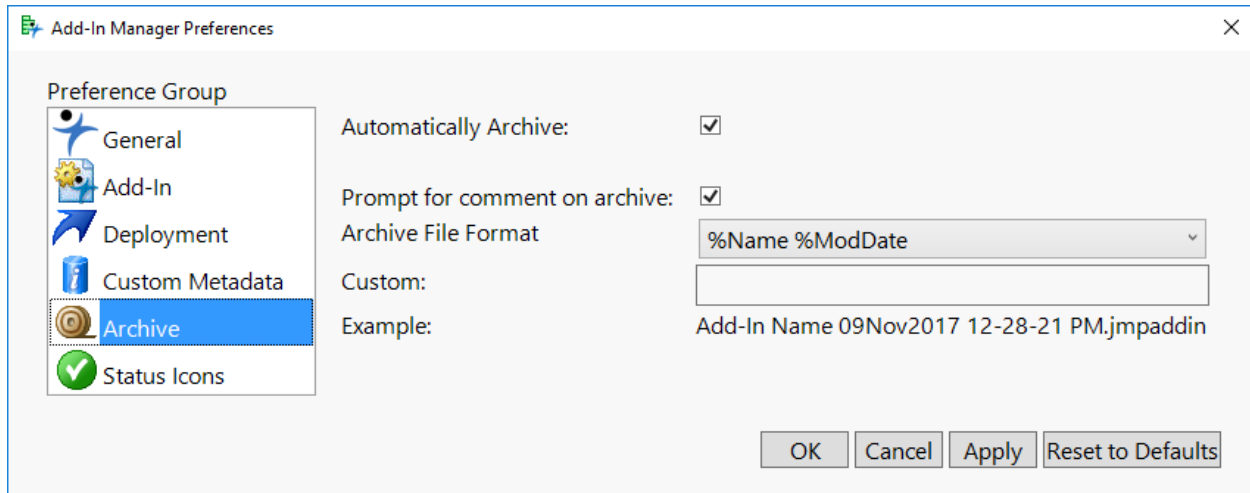
Custom Metadata

Within the Custom Metadata preference group, you choose whether or not the customMetadata.jsl script is copied to your add-in source directory.



Archive

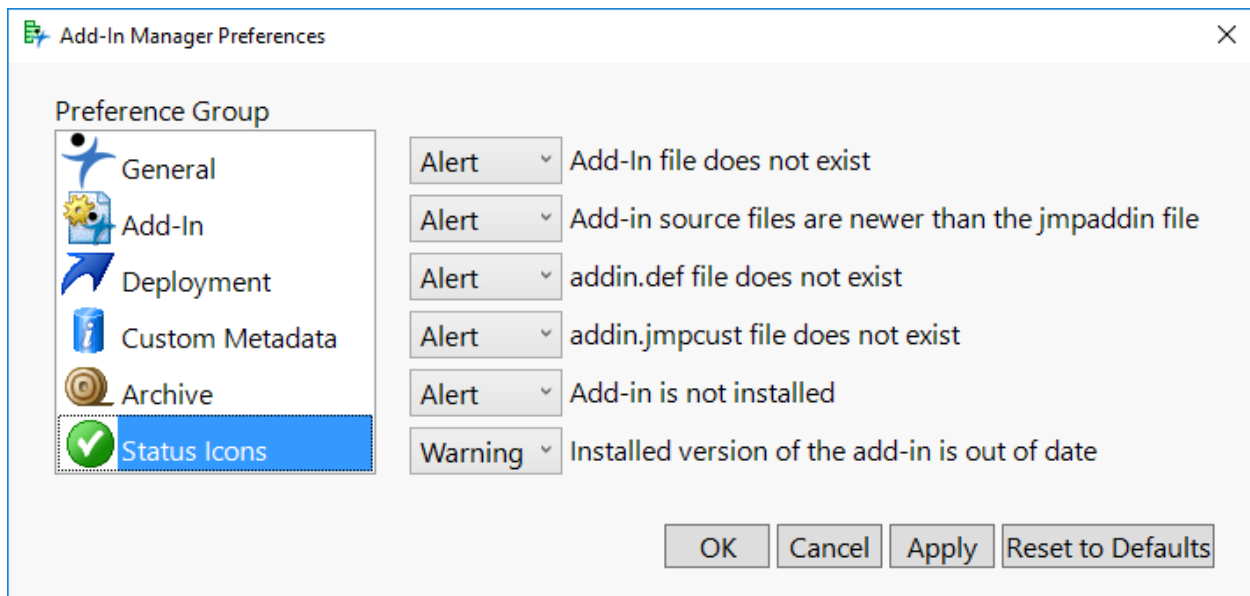
- Automatically Archive – the default value populated when defining a new add-in. Archiving is the process of saving a copy of an add-in each time a new one is built. See the [How Archiving Works section](#) for more information.
- Prompt for comment on archive – whether or not you want to be prompted for a comment each time a new add-in is built. Comments are a good place to keep track of what changes were made in each build of an add-in.
- Archive File Format – Choose a format for the file name from a preset list, or create your own using the Custom option. An example filename is shown after the custom text box at the bottom of the window.
 - Custom format placeholders (case sensitive):
 - %Name – The name of the add-in
 - %Id – The ID of the add-in
 - %ModDate – The build date/time for the add-in
 - %ArchiveDate – The date/time when the add-in was archived
 - %Ver – The version of the add-in
 - %BuildNum – The build number of the add-in (resets for each version).



Status Icons

Define what should be considered an alert or warning, and which should not be considered (ignored). You may want to change the default if you would like to have a higher notification level for certain events, or if you do not wish to check for an event at all.

The icons are described in more detail in the Add-In List [Status Icons](#) section.



General Info Tab

Addin.def Information

The information within the “General” Panel Box will be used to create the addin.def file, which is used to register your add-in.

- Add-In Name – The name of your add-in; used for the file name for your add-in.
- Add-ID – a unique ID for your add-in. This ID is used when determining if an add-in should be overwritten (i.e. re-installed).

- Version – The version for your add-in; must be numeric
- Minimum JMP Version – Restrict your add-in from being installed on older versions of JMP.
- Maximum JMP Version – Restrict your add-in from being installed on new versions of JMP.
- Supported Host – Is your add-in able to run only on Windows, Mac, or Both?

Definition Custom Metadata

Add-In Name: My JMP Add-In Descriptive name to help identify the add-in

Add-In ID: com.jmp.myjmpaddin Unique name for the add-in (such as com.mycompany.myaddin)

Version: 1 Version of this Add-In

Minimum JMP Version: 13

Maximum JMP Version:

Supported Host: Both

Lock add-in definition

Folder Locations

This information is specifically for use by Add-In Manager.

- DEV
 - Add-In Source Folder – The location of your add-in’s source files. These are the files that will be zipped into the jmpaddin file.
 - Add-In Destination Folder – The folder where the jmpaddin file will be created.
 - Automatically Archive – Do you want to save off a copy of your add-in each time you build a new one?
 - Archive Folder – The folder where the archived files will be moved to.

Folder Locations:

DEV TEST PROD

Add-In Source Folder: C:\Public\Add-ins\My JMP Add-In\My JMP Add-In Source\

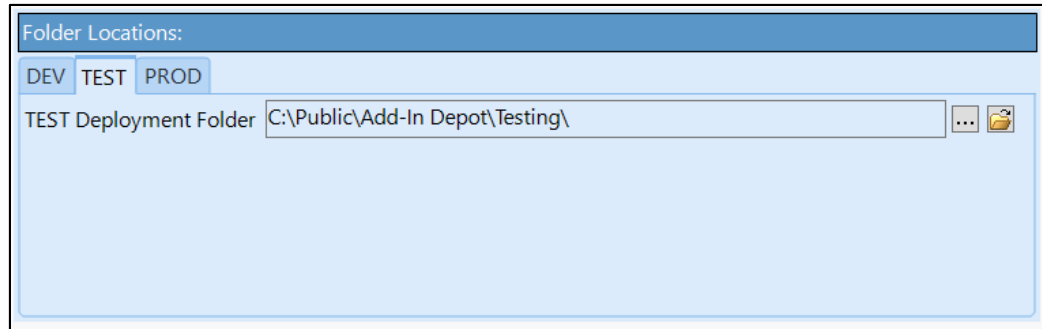
Add-In Destination Folder: C:\Public\Add-ins\My JMP Add-In\

Output File Path: C:\Public\Add-ins\My JMP Add-In\My JMP Add-In.jmpaddin

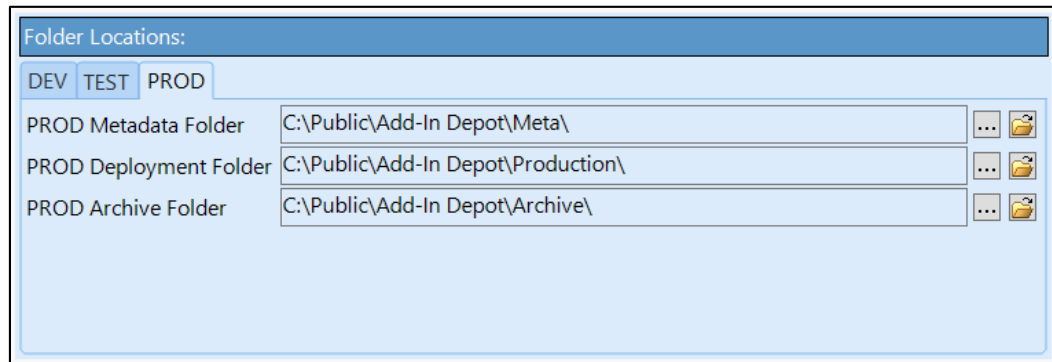
Automatically Archive:

Archive Folder: C:\Public\Add-ins\My JMP Add-In\Archive\

- TEST
 - TEST Deployment folder – the folder where the jmpaddin file gets created when deploying to TEST.



- PROD
 - PROD Metadata Folder – the folder where the metadata table with information about add-ins deployed to the PROD location is saved.
 - PROD Deployment folder – the folder where the jmpaddin file gets created when deploying to PROD.
 - PROD Archive Folder – the folder where add-ins are archived from the PROD location when a new add-in is deployed.



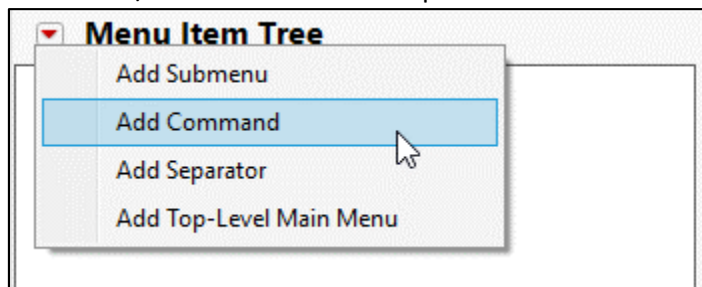
NOTE: If any of these fields are highlighted in red, it means the folder no longer exists and you need to fix the path.

Menu Items Tab

The Menu Items tab works very similar to how the same tab in Add-In Builder works, but with a few differences.

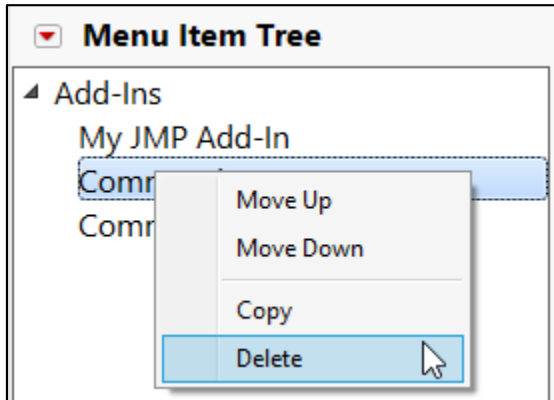
Menu Item Tree

You can add submenus, commands, separators, and main menus using the red triangle menu next to “Menu Item Tree.” The “Add Top-Level Main Menu” option is new in version 2.0 and allows you to have commands/submenus within multiple different main menus.

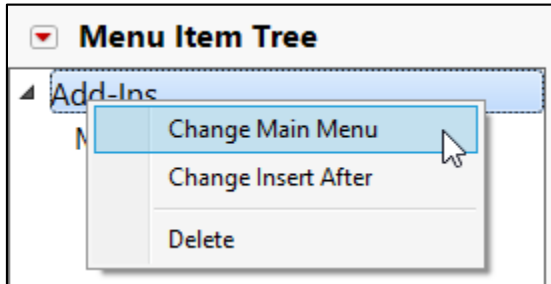


In Add-In Manager, the Move Up, Move Down, Copy and Delete actions are available on the right-click menu. A paste action is available once you have selected to copy a node. You can copy a node (along

with any children) except for main menu nodes. However, you can paste nodes by right-click on any node.

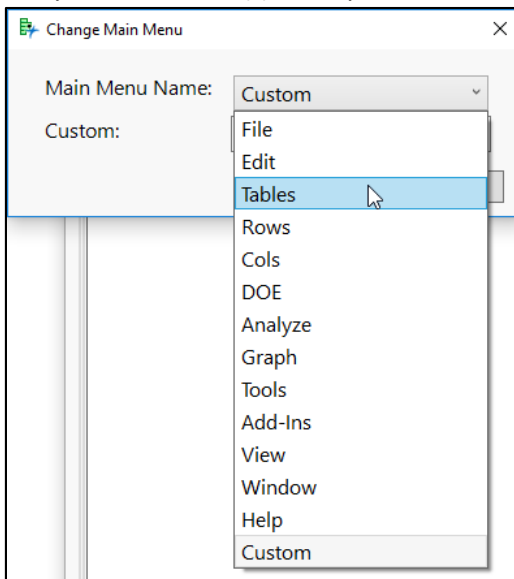


You can change the Main Menu (Add-Ins by default) by right-clicking the top-level item in the tree. You can also change the Menu Item to insert in this same menu. This would be useful if you are using an existing main menu item, like Analyze, and want to place the menu item in a specific position, such as after Distribution.



Changing Main Menu to a Built-In Menu:

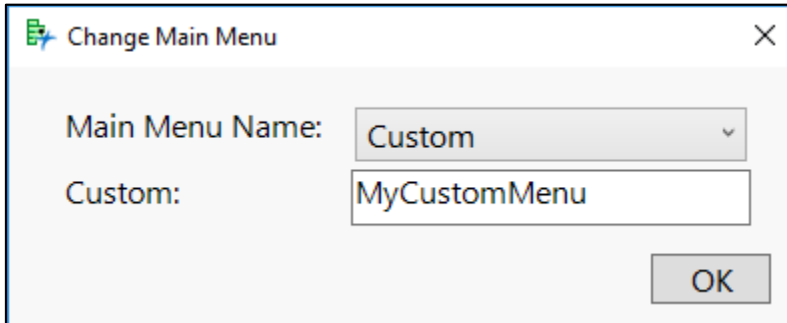
1. Right click on the top item in the Menu Item Tree and select "Change Main Menu."
2. From the "Change Main Menu" window, select the menu from the ComboBox that you would like your menu item(s) to be placed.



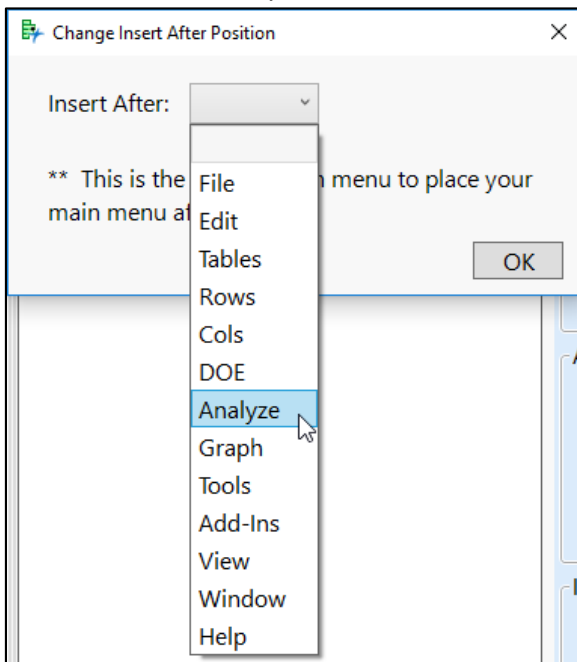
3. If there is a value already set in the “Insert After” property (and the previous menu was a custom one), you will be prompted to change the property to a new value. The item you select here will determine where your custom menu will be placed within the Main Menu bar.
4. If you were not prompted for an Insert After value, you can right-click on the top item in the Menu Item Tree and select “Change Insert After” to update this value.

Changing Main Menu to a Custom Menu:

1. Right click on the top item in the Menu Item Tree and select “Change Main Menu.”
2. From the “Change Main Menu” window, select “Custom” from the ComboBox and type the name of your desired Menu in the TextBox, as shown below.



3. If there is a value already set in the “Insert After” property (and the previous menu was a built-in one), you will be prompted to change the property to a new value. The item you select here will determine where your custom menu will be placed within the Main Menu bar.



4. If you were not prompted for an Insert After value, you can right-click on the top item in the Menu Item Tree and select “Change Insert After” to update this value.

Details

Details:

General

Menu item name

Tooltip for menu item

Action

Run JSL in this file:

Run this JSL:

```
Print( "My JMP Add-In" )
```

Use the "Here" namespace for unqualified JSL variable names

Icon

Shortcut

Ctrl Alt Shift

Currently assigned to:

General

- Menu item name: Name for the selected command or submenu
- Tooltip for menu item: Tooltip to appear when hovering over the selected menu item

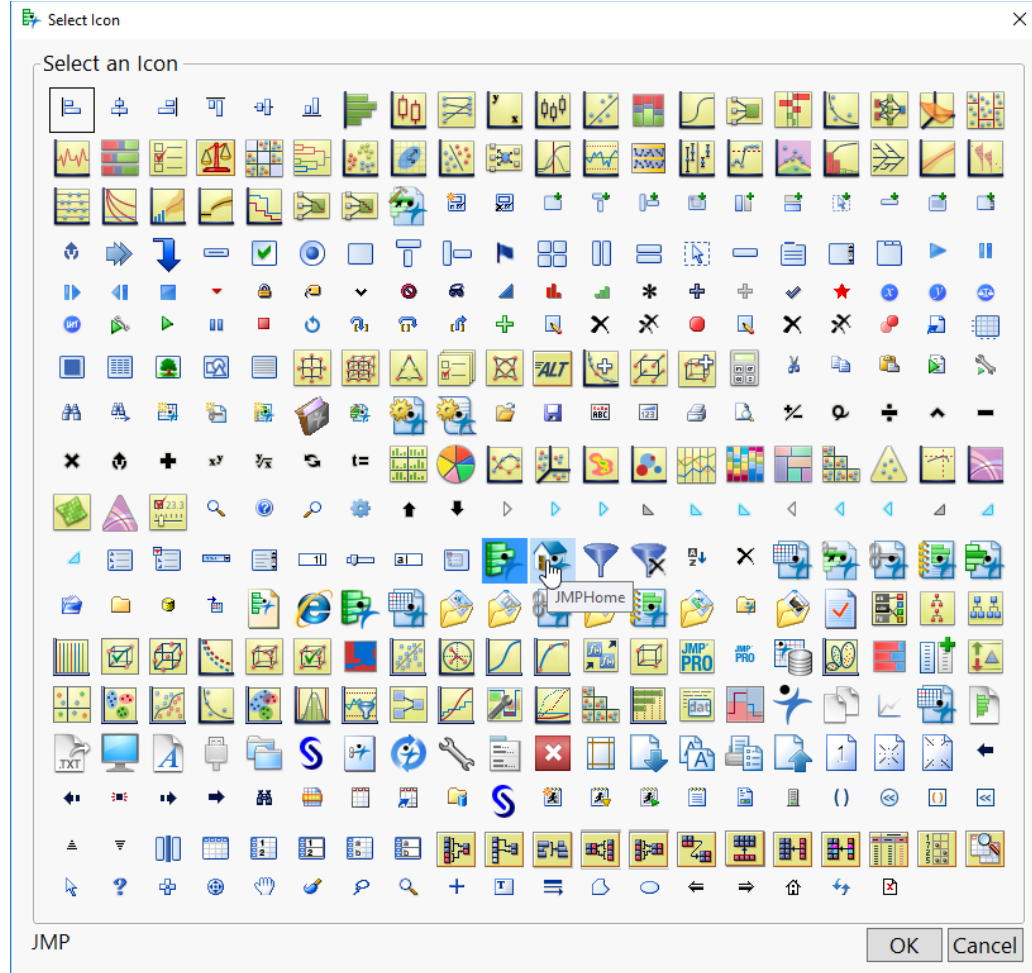
Action

- Run JSL in this File: Run a JSL script from a file for the selected menu item command. This script must be located within the add-in's source folder. If it is not, you will be prompted to copy the file to the source folder.
- Run this JSL: Paste in the JSL to run for the selected command. Note that this script will be embedded in the addin. jmpcust XML.
- Use the "Here" namespace for unqualified JSL variable names

Icon

- Combo Box: specify None, File, or Built-in.
- Select Button: Depending on the Combo Box selection, the following will occur
 - File: You will be prompted for a file location of an image file. If the selected file is not located in the source folder, you will be prompted if you want to copy the file to the source directory.

- Built-in: You will be prompted to select an icon from a window of IconBoxes as shown below.



Shortcut

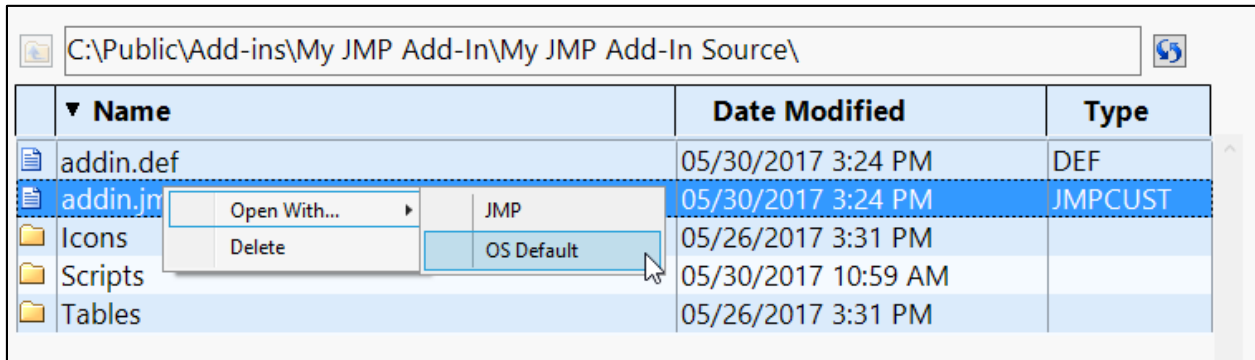
- Specify a combination of Ctrl, Alt and Shift, along with an additional key to act as a shortcut for the selected menu item command.

Files Tab

On the Files tab, you can view and open any of the files within your add-in's source folder.

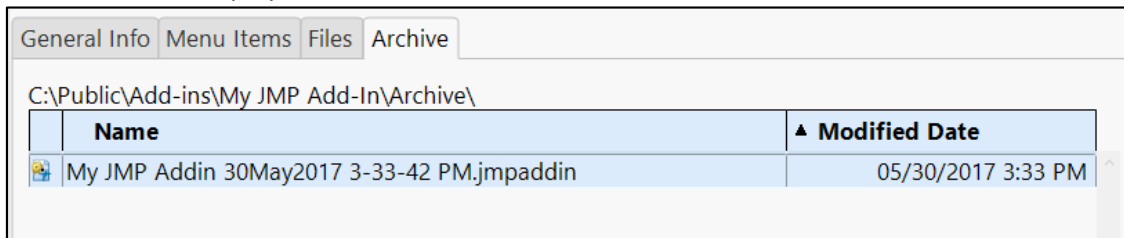
General Info Menu Items Files Archive			
C:\Public\Add-ins\My JMP Add-In\My JMP Add-In Source\			
	▼ Name	Date Modified	Type
	addin.def	05/30/2017 3:24 PM	DEF
	addin.jmpcust	05/30/2017 3:24 PM	JMPCUST
	Icons	05/26/2017 3:31 PM	
	Scripts	05/30/2017 10:59 AM	
	Tables	05/26/2017 3:31 PM	

To open a file, you can double-click on the file, or, with the file selected, right click and choose Open With... > JMP /OS Default. Choosing JMP will attempt to open the file using JMP's Open command and will defer to the OS if that fails. The OS Default uses a command line operation to open the files.



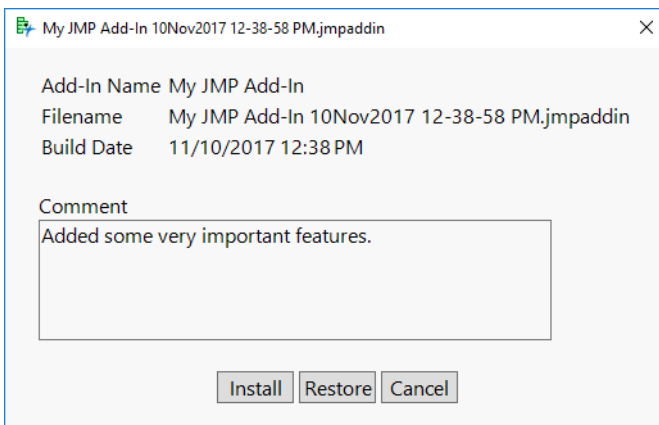
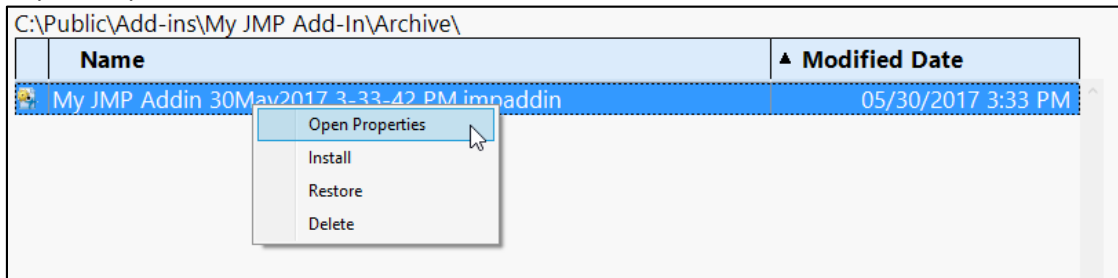
Archive Tab

The Archive tab displays all of the archived files for the current add-in.



Properties

You can view the properties for an add-in by double clicking on the add-in or right-clicking and choosing "Open Properties."



Install

You can install an archived add-in directly from the right-click menu or from the add-in properties window. This installs the archived version of the add-in and does not affect any of the add-ins source folder. This can be helpful when comparing behavior to a previous version of your add-in.

Restore

You can restore an archived add-in directly from the right-click menu or from the add-in properties window. Restoring an add-in is the process of reverting back to a previous version of an add-in.

WARNING: Restoring an archived add-in will automatically build the current add-in files and will replace those files with the files with the contents of the archive file being restored.

Custom Metadata

Custom metadata can be included in your add-in in the form of an Associative Array. The associative array will be written out to a "customMetadata.jsl" file in the top-level folder of your add-in. Check out the Scripting Index (Help > Scripting Index) for usage examples. The associative array can be imported in your add-in using the below include statement (after replacing 'com.jmp.myaddin' with your add-in's ID).

```
aa = Include( "$ADDIN_HOME(com.jmp.myaddin)\customMetadata.jsl" );
```

Once you have imported the Associative Array, you can use the following syntax to reference a value:

```
aa["author"]
```

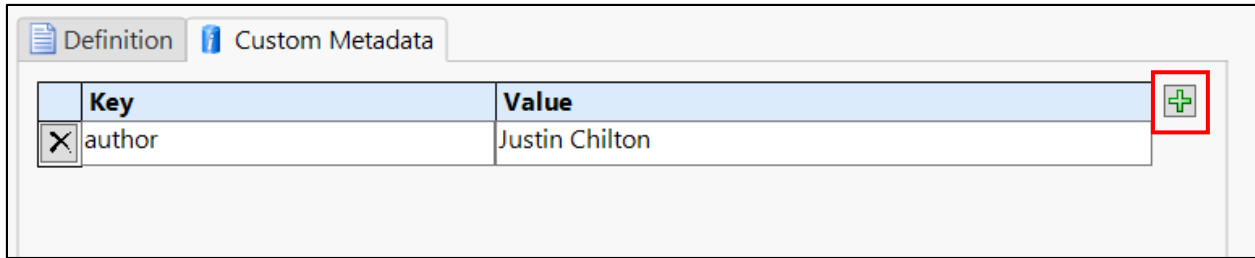
Included Metadata

Below is a list of all key/value pairs that are automatically managed by Add-In Manager. Do not use any of these keys in your own custom metadata.

Key	Value	Data Type
addinVersion	%Ver	Numeric
buildDate	%BuildDate	Numeric
id	%Id	Character
deployedAddinsFilename	depot-table-filename.jmp	Character
deployedAddinsLoc	/path/to/addin/depot/table	Character
name	%Name	Character
state	"DEV" or "TEST" or "PROD"	Character

Adding New Custom Metadata


Custom metadata can be added by clicking the green plus icon next to the table on the Custom Metadata tab (within the General Info tab).

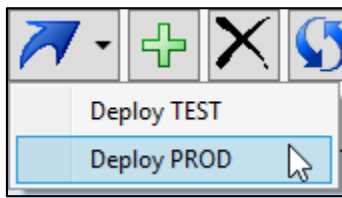


Defining Default Custom Metadata

You can define default custom metadata, to be used as a starting point for all new add-ins, by editing the “customMetadata.jsl” file located in the “Template Scripts” folder within Add-In Manager’s home folder. Be sure to make a copy of this file before editing it, as incorrect changes can break Add-In Manager.

Deployment

Your add-ins can be deployed to either a TEST or PROD location using the  dropdown button, as shown below.



TEST

Clicking the “Deploy TEST” option will copy the selected add-in to the folder specified on the TEST tab of the [Folder Locations](#) section. If you have “Create Metadata File” preference turned on, the “state” key in the associative array will be changed to “TEST.” TEST add-ins are not archived and are therefore overwritten each time a new version of an add-in is deployed.

PROD

Clicking the “Deploy PROD” option will copy the selected add-in to the folder specified on the PROD tab of the [Folder Locations](#) section. If you have “Create Metadata File” preference turned on, the “state” key in the associative array will be changed to “PROD.” Each time a new version of an add-in is deployed, the previous version is archived to the PROD Archive Folder.

There is an Associative Array (in a JSL file), containing file locations and build dates, that gets updated when deploying an add-in to PROD. This associative array contains the ID, file location, and publish date for an add-in. You can change the location of this associative array by changing the PROD Metadata Folder on the PROD tab of the [Folder Locations](#) section. The value of the “pubDate” key of the publish dates associative array will match the value of the “buildDate” key of the custom metadata associative array.

Update Check Script

If you have the “Include Update Check Script” preference turned on, a script called “updateCheck.jsl” will be included in the top-level folder of your add-in. You can change the default script by editing the

“updateCheck.jsl” file in the “Template Scripts” directory within Add-In Manager’s home directory. Be sure to make a copy of this file before editing it.

The existing “updateCheck.jsl” file checks the PROD metadata table discussed in the previous section. It uses the custom metadata file’s “buildDate” attribute to check if the installed version is older than an available published version. Because the update script relies on the custom metadata file, is important to keep the “Create Metadata File” turned on if you use the Deployment features.

You can use this script using the Include statement below. You will need to change the ID to match the ID of your add-in. You can use this update check at the beginning of your command’s script or in a ButtonBox script to check for a new version. However, as of JMP 13, using this script within an addinLoad.jsl script will not work.

```
Include( "$ADDIN_HOME(com.jmp.myaddin)\updateCheck.jsl" );  
com.jmp.juchil.addinmanager:checkForUpdate();
```

You can edit the updateCheck.jsl script within your source folder once it has been copied over, or you can edit the original copy used from the Template Scripts folder within Add-In Manager’s home folder.

Deployment Status Bar

You can see the status of the add-ins deployed to the TEST and PROD locations within the status bar at the bottom of the window, as shown below.

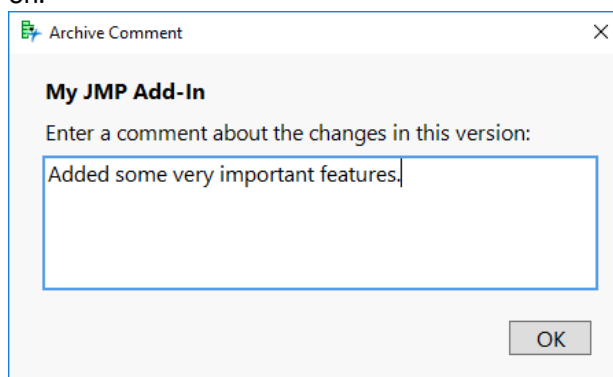


The status bar checks for the following criteria:

- Does the jmpaddin file exist?
- Are the source files newer than the deployed add-in?

How Archiving Works

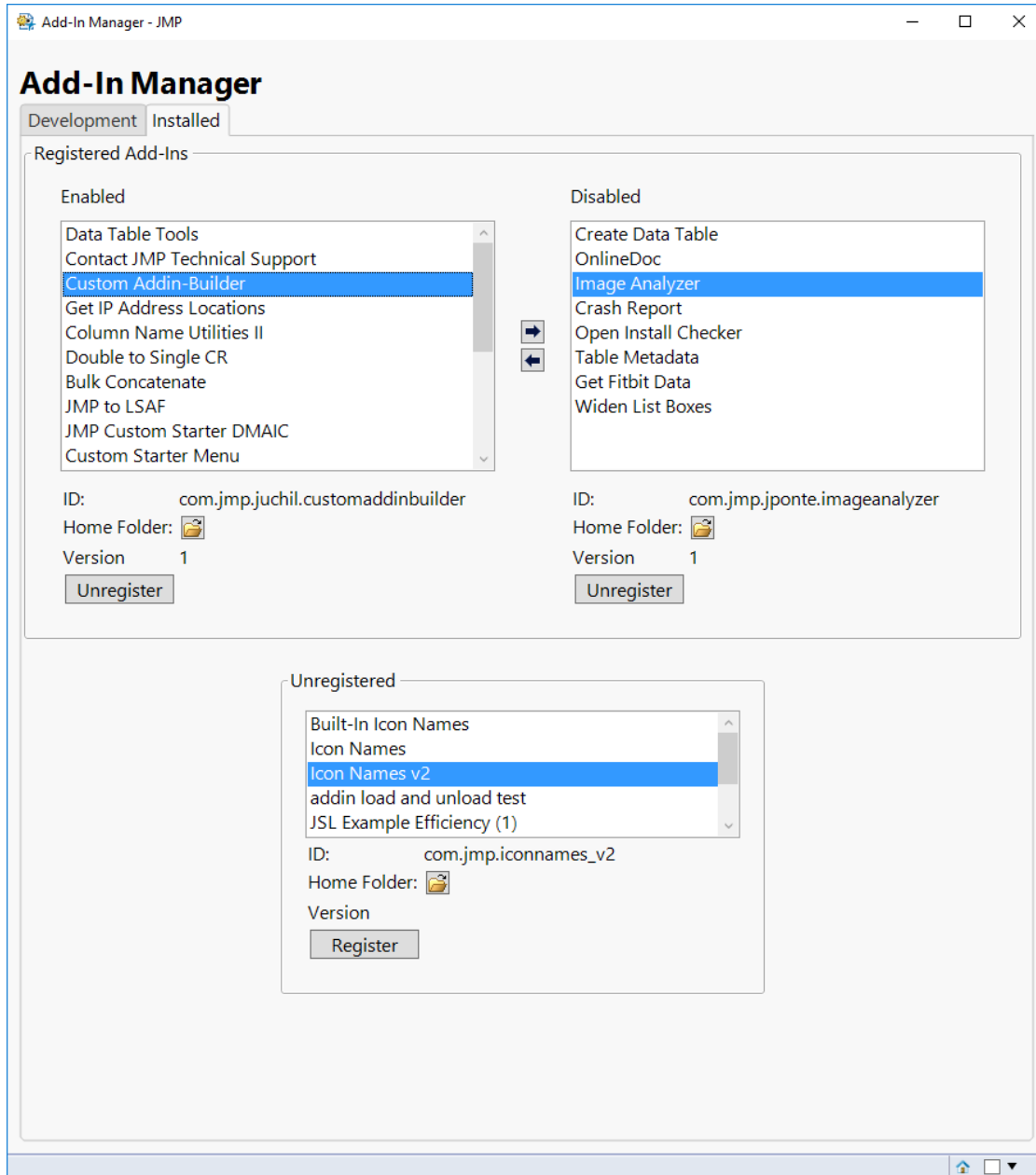
1. When you build an add-in using Add-In Manager, you will be prompted for a comment if you have the “Prompt for Comment on Archive” preference set and Automatic Archiving is turned on.



2. After clicking OK within the “Archive Comment” window, the jmpaddin file located within the Destination Folder is moved to the Archive Folder with the format specified in the Archive File Format preference. For more information, check out the [Archive preferences section](#). If there is no existing add-in file, this step will be skipped.
3. Once the previous add-in file has been moved successfully, a new jmpaddin file will be created in the Destination Folder.

Installed Tab

To manage your installed add-ins, click on the “Installed” tab at the top of the window.

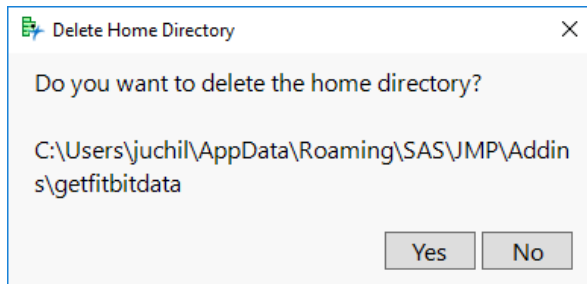


Enable/Disable Add-Ins

To enable or disable an add-in, use the left and right arrow buttons to move the add-in between the enabled and disabled lists.

Register/Unregister Add-Ins

To register an add-in, click on the item in the list box within the “Unregistered” section at the bottom of the window and click “Register.” To unregister an add-in, click on the add-in within the “Registered” section and click “Unregister.” When you unregister an add-in, you will be prompted regarding the deletion of the add-in’s home folder. Below is an example of this prompt.



Miscellaneous

Sorting Files and Archive Tables

You can sort the tables within the Files and Archive tabs by clicking on the column headers. The up and down arrows within the headers will signify the current sort method (ascending or descending).

Preferences and Internal Table File Locations

The preferences file and supporting tables are located in the following folder.

On Windows: %APPDATA%\SAS\JMP Add-In Manager

On Mac: ~/Library/Application Support/JMP Add-In Manager

Deleting these files will cause the add-in to revert to default preferences and will clear the add-in list. You may need to delete this if you need to start fresh with Add-In Manager.

Setting up Startup and Exit Scripts

To create a startup script, place a JSL script file within the add-in’s source folder (not within any subfolders) with the name “addinLoad.jsl.” For an exit script, follow the same process, but name the file “addinUnload.jsl.”

JMPCUST File Backups

Each time a new addin.jmpcust file is written, the old file is copied to the Add-In Manager AppData/Application Support folder.

On Windows: %APPDATA%\SAS\JMP Add-In Manager\Jmpcust Backups\<add-in-id>

On Mac: ~/Library/Application Support/JMP Add-In Manager/Jmpcust Backups/<add-in-id>