madcap
software

MADCAP FLARE 2024

Variables

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 4

APPENDIX

Introduction

Supported In:

A variable is a small piece of plain text or auto-generated content that can be edited in one place but used in many places throughout your project. Variables are especially useful for text that might change frequently, such as version numbers and dates.

HOW VARIABLES MIGHT LOOK IN THE SOURCE FILE

MadCap Product:[Product Name] ■ VersionNumberDot:2024

Copyright © Yr: 2024 CompanyName: MadCap Software
Help System Last Updated on LastUpdated: February 12, 2024 at 09:14 AM

HOW VARIABLES MIGHT LOOK IN THE OUTPUT

MadCap Flare 2024

Copyright © 2024 MadCap Software
Help System Last Updated on February 12, 2024 at 08:53 AM

General Information

- "Types of Variables" on page 8
- "Variable Components" on page 11
- "Visual Cues for Variables" on page 12
- "Initial Variables" on page 14
- "Variables and Auto Suggestion" on page 14
- "Stylesheet (CSS) Variables" on page 14
- "Variable Features Per Output Type" on page 15

Main Activities

- "Adding Variable Sets to Projects" on page 17
- "Creating Variables" on page 18 and "Creating Custom Date/Time Variables" on page 19
- "Inserting Variables" on page 23
- "Editing Variables" on page 28 and "Editing Date/Time Variables" on page 32
- "Pinning Variables" on page 38
- "Overriding Variable Definitions in Targets" on page 68

Other Activities

- "Chap/Sec/Vol Variables in Frames" on page 51
- "Converting Variables to Text" on page 54
- "Deleting Variables" on page 55
- "Hiding and Showing Variable Names" on page 56
- "Date/Time Variables" on page 57
- "Variable Definitions" on page 62
- "Heading Variables" on page 72
- "Running Head Variables" on page 85
- "Variable Sets" on page 92
- "Styled Variables" on page 96

CHAPTER 2

General Information for Variables

There are various pieces of general information you should know if you plan to use this feature.

This chapter discusses the following:

| Types of Variables

Following are the main categories of variables that you can use.

Custom Basic Variables

These are basic variables that you can create in variable sets. They can be used for virtually any purpose (product names, company information, terms that are used frequently). See "Creating Variables" on page 18.

Custom Date/Time Variables

You can create date and time variables based on Microsoft's "Custom Date and Time Format Strings." You can include custom date/time variables in template pages, which is useful for showing when each topic in your output was created. See "Creating Custom Date/Time Variables" on page 19.

System Variables

You can add a system variable by selecting **Insert > Variable**. Then select **System** and choose the specific variable.

- **Chapter/Section/Volume Number** For Adobe PDF output, you can insert system variables in page layout frames that display your chapter, section, or volume numbers (if you are using autonumbers to identify the various parts of a manual). See "Chap/Sec/Vol Variables in Frames" on page 51.

  - **ChapterNumber** This displays the current chapter number.

  - **SectionNumber** This displays the current section number.

  - **VolumeNumber** This displays the current volume number.

- **Date/Time** The global format in windows controls the format dates and times in variables.
  - **LongDate** Example: Thursday, December 8, 2017
  - **LongTime** Example: 2:49:00 PM
  - **ShortDate** Example: 12/8/2017
  - **ShortTime** Example: 2:50 PM
- **Linked Title/Header/File** In tables of contents (TOC) and browse sequences, you can insert system variables that are tied to information in the linked topic. These variables ensure that your TOC or browse sequence entries are always in sync with your topic titles, headers, or file names. When one of these variables is used in the TOC or Browse Sequence, the text displays in blue to indicate a system variable is being used (the text will not be blue in the output).

  These variables can be seen in the Variables dialog only if you access the dialog from the Properties dialog for a TOC or browse sequence entry.
  - **LinkedFile** This displays the entry using the name of linked source file, without the file extension.
  - **LinkedHeader** This displays the entry using the name of the first heading in the linked source file.
  - **LinkedTitle** This displays the entry using the title information within the title tag in the linked source file.
- **Page** For print-based outputs, you can use variables to show the page count (total number of pages in the output) or the page number.
- **SourceFile** This displays the name of the file where the variable is inserted.
- **Title** This lets you display the title of a glossary, if you've added one, or the title of a topic.

☐ **NOTE** The LinkedFile and LinkedTitle system variables are different from the Title and SourceFile system variables. While the LinkedFile and LinkedTitle variables show information for *the linked source file*, the Title and SourceFile variables show information for *the file where the variable is inserted*. If you use Title or SourceFile in your TOC, you will see the TOC's title or file path in the TOC entry.

Heading Variables

You can insert Heading variables into page layouts or template pages in order to automatically display text based on the mc-heading-level style property, which by default is applied to the h1 through h6 heading styles that you use in your project. Like Running Head variables, they are useful when creating print-based output. It's an easy way, for example, to automatically display a chapter title in the header of a chapter. For Adobe PDF output, you can also use Heading variables to automatically display glossary headings/terms and index headings/terms in a page layout frame. See "Heading Variables" on page 72.

Running Head Variables

A Running Head (or Running HF) variable is a special variable that you can insert into a header or footer in a page layout or a print template page. It lets you display certain text in the header or footer automatically, based on the style associated with the variable. For example, Running Head variables are useful if you want to include the title of each chapter in a document in a header or footer without having to type them into multiple pages. For more information see "Running Head Variables" on page 85.

> **NOTE** Running Head variables are supported only in Microsoft Word output.

❙ Variable Components

A variable has two main components—the variable *name* and the variable *definition.*

⭐ **EXAMPLE** An example of a variable name is "CompanyName." The definition for that variable name might be "ACME Incorporated." Using that example, if you were to insert the CompanyName variable into a paragraph of a topic, the phrase "ACME Incorporated" would be added at that spot and shown in the output.
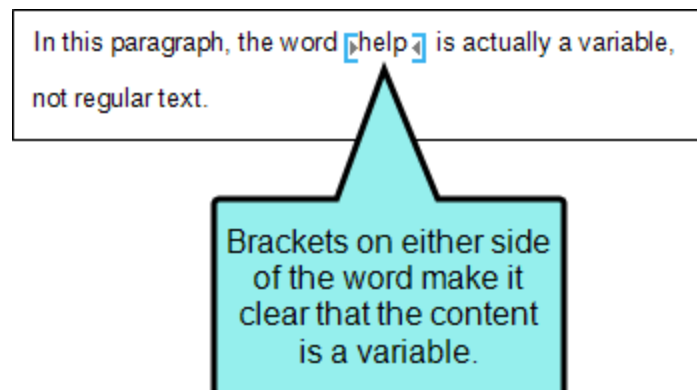
▎Visual Cues for Variables

When you insert variables into content, there are a few ways to discern that certain content is actually a variable, as opposed to regular text.

Markers

First, markers can be turned on or off, providing a visual cue about the content in the form of brackets. When you insert a variable into a topic with markers turned on, the variable definition is displayed in a marker represented by brackets on either side of the definition. If you cannot see all of the information in a marker, you can adjust the marker width. (Click the down arrow next to the **Show tags** button 👁 ▾ in the XML Editor. The marker width field is at the bottom of the drop-down menu.)

Here is an example of a variable with markers turned on. (To turn markers on, click the down arrow next to the **Show tags** button 👁 ▾ in the XML Editor. Then select **Show Markers**.)

In this paragraph, the word ⸢help⸣ is actually a variable, not regular text.

Brackets on either side of the word make it clear that the content is a variable.

Variable Names

In addition to markers, you can select an option to show variable names. Here is an example of a variable with variable names turned on. (To show variable names, click the down arrow next to the **Show tags** button ![icon] in the XML Editor. Then select **Show Variable Names**.)

In this paragraph, the word ⟨Help: help⟩ is actually a variable, not regular text.

In this example, the markers are turned on (showing brackets), and the variable names option is also enabled (showing the name of the variable inserted; in this case, "Help").

Highlighted Names

You can also select an option to display variables with a highlighted background.(To highlight variables, click the down arrow next to the **Show tags** button ![icon] in the XML Editor. Then select **Show Highlight Variables**.) Here is an example.

In this paragraph, the word ⟨Help: help⟩ is actually a variable, not regular text.

With the highlight option turned on, you can see a gray background behind the variable definition.

▌Initial Variables

There are usually some initial custom variables (e.g., CompanyName and PhoneNumber) already provided when you start a new project or add a new variable set (using the "MyVariables" template). You can use these variables, modifying their names and/or definitions to meet your needs. You can also create additional variables. A variable set file has an .flvar extension and is stored in the Project Organizer under the Variables folder.

▌Variables and Auto Suggestion

In addition to inserting variables the usual way, you can also insert variables when using Auto Suggestion (a popup window that predicts content as you type).

▌Stylesheet (CSS) Variables

You also might hear about CSS variables. These are not the same as the regular variables that are stored in the Project Organizer. Instead, CSS variables are created in a stylesheet. But like regular variables, CSS variables can be used to create information in one place and then re-use it in many places. In the case of CSS variables, you specify a style value once and then insert that variable in any style properties where you want to use it.

CSS Variables and Branding

CSS variables are the foundation of branding functionality. The Start New Project Wizard and the Branding Editor group together certain CSS variables (e.g., logo, hero image, font, color palette) that relate to branding. The branding values you identify in the Branding.css file are those associated with those CSS variables in the project. If you start your project with the wizard, the CSS variables are automatically linked to various places throughout the project where they point to the branding stylesheet.

Variable Features Per Output Type

Following are variable features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help
Basic Variables	✔	✔	✔	✔	✔	✔	✔
Date & Time Variables	✔	✔	✔	✔	✔	✔	✔
Heading Variables	✔	✔	✔	✔	✔	✘	✔
Running HF Variables	✘	✘	✔	✘	✘	✘	✘
Snippet Variables	✔	✔	✔	✔	✔	✔	✔
System Variables	✔	✔	✔	✔	✔	✔	✔

CHAPTER 3

Main Activities for Variables

Some activities are particularly common and important when it comes to this feature.

This chapter discusses the following:

▌Adding Variable Sets to Projects

You can add a new variable set to your project. You can then open the variable set and create new variables.

How to Add a Variable Set to a Project

1. In the Project Organizer, right-click on the **Variables** folder and from the context menu select **Add Variable Set**.

2. In the **File Type** field at the top, make sure **Variable Set** is selected.

3. In the **Source** area, choose to create the new file based on a template or an existing file.

   - **New From Template** Choose either a factory template file or one of your own custom template files as a starting point. The new file will take on all of the settings contained in the template. If you want to use the factory template provided by Flare, expand the **Factory Templates** folder and click on a template file. If you want to use your own custom template file, expand the appropriate folder and click on a file. For more information about templates, see the online Help.

   - **New From Existing** Choose an existing file of the same type as a starting point for your new file. As with template files, your new file will take on all of the settings contained in the file you select. To use this option, click ▦, use the Open File dialog to find a file, and double-click it.

4. (Optional) If you want to place the file into a subfolder previously created in the Content Explorer or Project Organizer, in the **Folder** field click ▦ and select the subfolder. Otherwise, keep the default location.

5. In the **File Name** field, type a new name for the variable set.

6. (Optional) If you want to apply condition tags to the file, expand the **Attributes** section at the bottom of the dialog. Next to the **Condition Tags** field, click ▦ and select the conditions you want to apply.Click **OK**.

7. (Optional) If you want to apply file tags, expand the **Attributes** section at the bottom of the dialog. Next to the **File Tags** field, click ▦ and select the file tags you want to apply.Click **OK**.

8. Click **Add**. The variable set is added to the Variables folder in the Project Organizer. The Variable Set Editor opens to the right, with the variable entries shown.

❙ Creating Variables

You can create a new variable within a variable set.

How to Create a Variable

1. Open the Project Organizer.

2. Double-click the **Variables** folder.

3. Double-click a variable set that you have added.

4. To create an additional variable, click ▤ in the local toolbar. A new variable row is added, with a temporary name for the variable.

5. Click in a field and press **F2** on your keyboard. Then type the name, definition, or comment. If you want to insert a variable in a field, you can click ▣. The variable will appear as syntax in the field, but in the output the variable definition will be shown.

    > 🗒 **NOTE** If you want to see the variable syntax replaced by the variable definition, click ▣ in the local toolbar of the editor. This shows you the "Evaluated Definition," which determines the actual text of the variable.

6. Click 💾 to save your work.

Creating Custom Date/Time Variables

You can create date and time variables based on Microsoft's "Custom Date and Time Format Strings." You can add date/time variables in a topic or a template page. See "Custom Date/Time Variables Into Template Pages" on page 57.

How to Create Custom Date/Time Variables

1. In the Project Organizer, expand the **Variables** folder and double-click a variable set. The Variable Set Editor opens.

2. In the local toolbar click ⬛. A new row is added for the date/time variable.
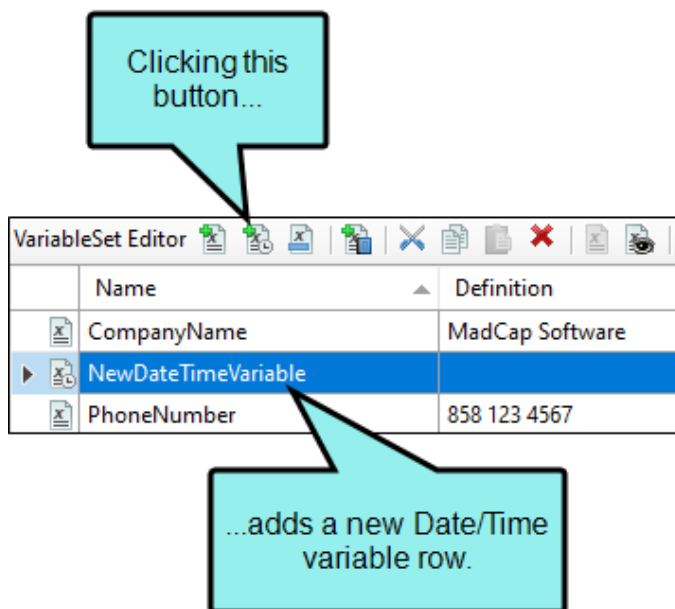
3. To enter a new name or comment, do one of the following:

   - Double-click in the appropriate cell and type the name or comment.

     OR

   - Click once in the appropriate cell and press **F2** on your keyboard. Then type the name or comment.

4. Click in the **Definition** cell. The Edit Format dialog opens.

5. In the field, enter a combination of format specifiers. As you do this, the current date and time are displayed below the field in that format.

    Specifiers are based on Microsoft's custom date and time format strings. For details of the specifiers available, see:

    `http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx`

    Here are some of the more common specifier combinations:

Specifier Combination	Result
MMMM dd, yyyy hh:mm:ss tt	July 09, 2013 03:55:53 PM
hh:mm:ss tt dd MMMM, yyyy	03:55:53 PM 09 July, 2013
dddd MMMM dd, yyyy	Tuesday July 09, 2013
ddd MM/dd/yy	Tue 07/09/13

> 📝 **NOTE** You can make a word part of the definition. For example, you might want the word Date with a colon and a space before a date string (e.g., Date: October 25, 2014). The issue is that certain characters are translated automatically as specifiers. So just typing the word Date wouldn't work. The easiest solution is to put the non-specifier characters within quotation marks. Therefore, you might enter "Date:" MMMM dd, yyyy.

6. In the **Update** field, select how you want the variable to display the date and time. You can choose any of the following:

    ▪ **Manually** The variable displays the date and time when the variable was created. You can update the variable manually, and it will display the date and time when it was most recently updated. See "Updating Manual Date/Time Variables" on page 61.

    ▪ **On File Creation** The variable displays the date and time that you created the file.

- **On File Save** The variable displays the date and time that you last saved the file.

- **On Project Save** The variable displays the date and time that you last saved all the files in the project.

- **On Build** The variable displays the date and time of the most recent project build. This is the default date/time variable type.

7. Click **OK**. It might look something like this:

8. Click 🖫 to save your work.

NOTE System variables use the language set in a tag, in the target, or at the project level. For example, you might have an English operating system, but in a Flare target you have French set as the language. If you insert a date/time variable, the day and month will display in French when you generate that target. Flare will also adjust the format of the variable if necessary (e.g., date first vs. month first).

NOTE Like standard variables, you can override custom date/time variables in a target.

NOTE If you want to edit a date/time variable, click on the **Definition** cell in the Variable Set Editor and make changes in the Edit Format dialog. You can also add multiple definitions to a variable, using them as a way to override variables on a target.

NOTE You can add custom date/time variables to your template pages to show when each topic in your output was created. These variables will update as if they are part of the topic, so you do not need to add a variable to each individual topic in your output. See "Custom Date/Time Variables Into Template Pages" on page 57.

❙ Inserting Variables

After you create or modify variables, you can insert them into any content file (e.g., topic, snippet) in your project. You can insert variables by dragging them from the Project Organizer, or you can use the Variables dialog.

How to Insert a Variable—Drag-and-Drop

1. Open the content file.

2. Open the Project Organizer.

3. Expand the **Variables** folder and expand the appropriate variable set under it.

4. Click on the variable and drag it to the XML Editor.

5. Click 💾 to save your work.

How to Insert a Variable—Variables Dialog

1. Open the content file.

2. Place your cursor where you want to add a variable.

3. Do one of the following, depending on the part of the user interface you are using:

   - **Ribbon** Select **Insert > Variable**.

   - **Local Toolbar** Click the face of the **Insert a Variable** button ⬚ .

     OR

     Click the down arrow next to ⬚ , then select **More Variables**.

   - **Keyboard Shortcut** On your keyboard press **CTRL+SHIFT+V**.

   - **Right-Click** Right-click in the editor and select **Insert > Variable**. This context menu option is available only when your cursor is located in certain places in the editor (e.g., not in a drop-down effect).

   The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

   > **NOTE** There are various additional places throughout the user interface where you can select variables (e.g., Properties dialog for TOC entry, Target Editor). Next to certain fields, you can click ⬚ to insert a variable into that field.

   > **NOTE** If a variable contains multiple definitions, the default definition is displayed in the Definitions column. The additional definitions are not shown. You can view and select alternative definitions from a drop-down list in the Target Editor when overriding definitions.

4. On the left, select the appropriate variable set.

5. On the right, select the variable you want to insert.

6. (Optional) If you want to add one of your 20 most recently used variables to the project, you can select a variable from the **Recently Used Variables** group. If you want to add one of your pinned—or favorite—variables to the project, you can select a variable from the **Pinned Variables** group.
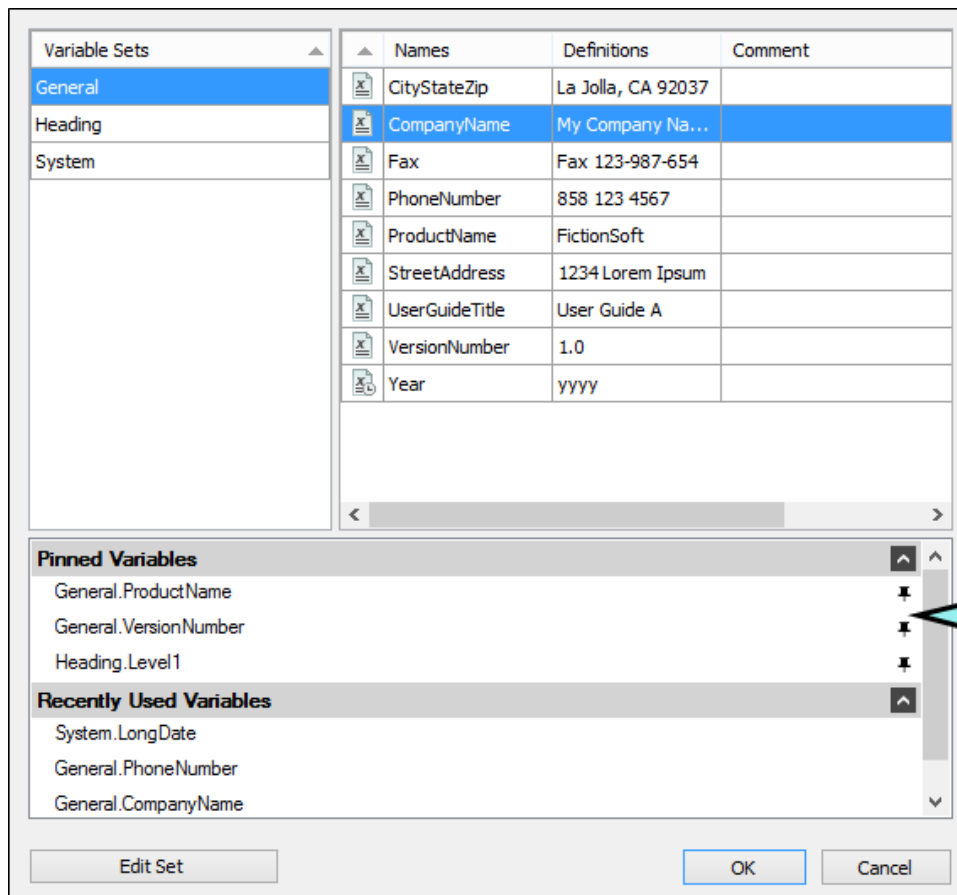

7. Click **OK**. The variable is added, with brackets surrounding it. The bracket shows the variable name and the variable definition (as long as markers are turned on and the set marker width allows it).

    You can adjust the marker width from the **Show tags** button to see more or less of the variable information, and you can also tell Flare to turn off the variable names so that you only see the variable definition. For more details about markers see the online Help.

8. Click to save your work.

How to Insert a Variable—Local Toolbar

1. Open the content file.

2. Place your cursor where you want to insert the variable.

3. In the local toolbar at the top of the XML Editor click the down arrow next to ⬛ ▾. From the drop-down, you can see your pinned variables and the 20 most recently used variables.

4. Navigate to the variable that you want to insert and select it.

5. Click 💾 to save your work.

What's Noteworthy?

NOTE If you need to insert a variable that does not appear in the Pinned Variables or Recently Used Variables groups, you can select **More Variables** to open the Variables dialog.
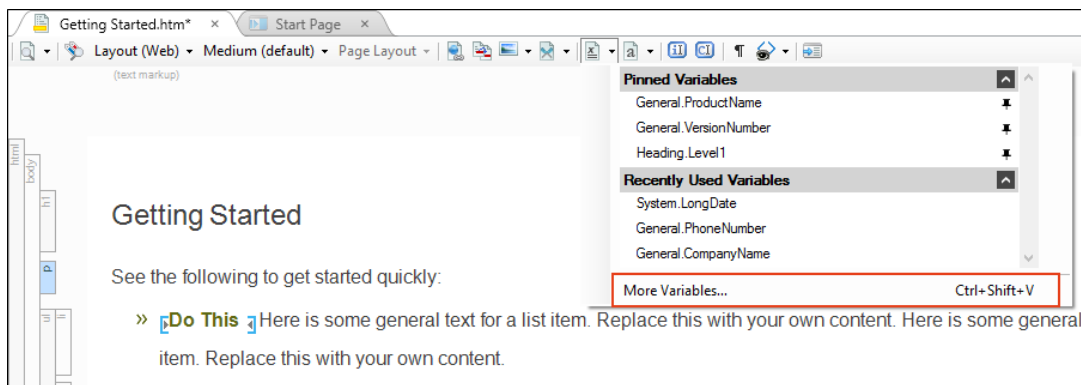
NOTE Variables must appear in the Recently Used Variables group before they can be pinned. If the variable you want to pin does not appear in the Recently Used Variables group, you can add it to a content file by dragging it from the Project Organizer or using the Variables dialog.

NOTE You can expand or collapse the Recently Used Variables and Pinned Variables groups by clicking the arrows next to each group name.
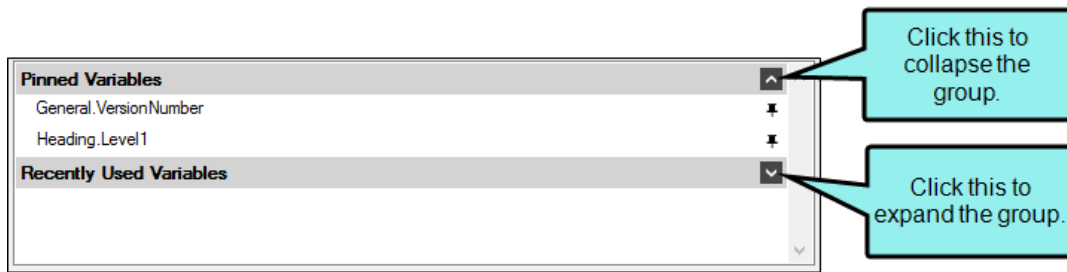
▎Editing Variables

After you create a variable, you can edit it in the Variable Set Editor, changing the name, definition, or comment. If you change the definition for a variable that has been inserted into topics, the changes will automatically be reflected in all those topics. You can also add multiple definitions to a variable, using them as a way to override variables on a target. For example, your company might have multiple phone numbers. You can associate them all with the same variable and use the appropriate one wherever necessary.

How to Edit a Variable

1. Open the Project Organizer.

2. Double-click the **Variables** folder. Your variable sets are displayed.

3. Double-click the custom variable set (such as **MyVariables**) that contains the variable you want to modify. The Variable Set Editor opens to the right, with the variables page shown.

4. Double-click in a field and type the name, definition, or comment. (You can also click once in a field, press **F2** on your keyboard. Then type the name, definition, or comment.) If you want to insert a variable in a field, you can click ⊡. The variable will appear as syntax in the field, but in the output the variable definition will be shown.

> ⊘ **WARNING** If you rename a variable (e.g., PhoneNumber), giving it the same name as an existing variable (e.g., CompanyName), you will see a warning message. If you click OK to accept the new name, this will result in multiple definitions for the variable (e.g., CompanyName will contain both its original definition and the one that had been associated with PhoneNumber). Do *not* click OK, then undo after saving. This changes all instances of the unchanged variable (e.g., CompanyName) to the one you first started to change (e.g., PhoneNumber) throughout the project. This undesired action cannot be undone.

> ⬜ **NOTE** If you want to see the variable syntax replaced by the variable definition, click ⊡ in the local toolbar of the editor. This shows you the "Evaluated Definition," which determines the actual text of the variable.

> ⬜ **NOTE** You cannot change system or Heading variables. If you are working with Running Head variables, you should not change the first part of the definition, but you can change the style within the definition. See "Running Head Variables" on page 85.
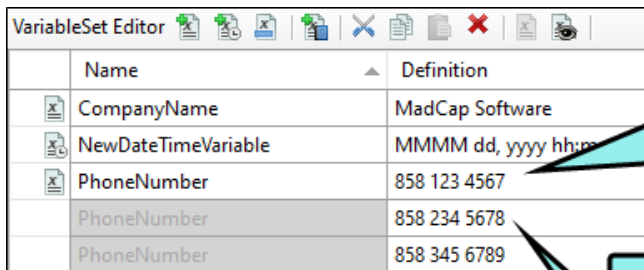
5. Click 🖫 to save your work.

How to Add Variable Definitions

1. In the Project Organizer, double-click the **Variables** folder.

2. Double-click a variable set. The Variable Set Editor opens.

3. Click on a row where you want to provide an additional definition.

4. In the local toolbar click ⬚. A new row is added for the new definition.

5. In the empty cell that is added, type the definition and press **ENTER**. If you want to insert a variable in a field, you can click ⬚. The variable will appear as syntax in the field, but in the output the variable definition will be shown.

   When a variable has multiple definitions, cells of the non-default definitions are grayed out (except for the definition cell).

   The default definition is determined by the order of creation. The first definition created is the default.
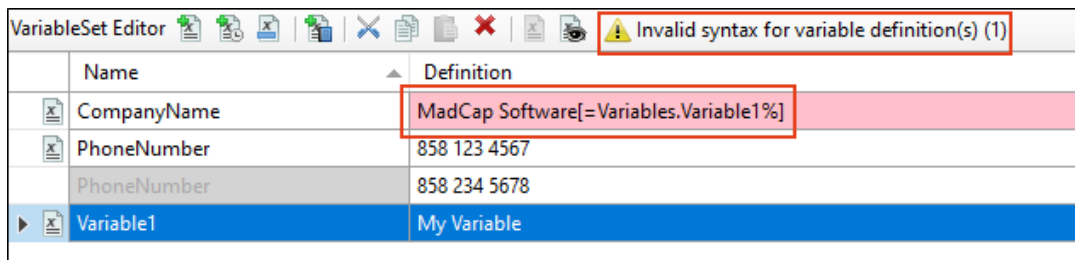

> 📝 **NOTE** If you want to see the variable syntax replaced by the variable definition, click 👁 in the local toolbar of the editor. This shows you the "Evaluated Definition," which determines the actual text of the variable.

6. Click 💾 to save your work.

What's Noteworthy?

> 📝 **NOTE** If you want to change how variables look when they display in the XML Editor and in generated output files, you can edit the MadCap|variable style in the Stylesheet Editor.

> 📝 **NOTE** If there is a variable inserted within another variable in the Variable Set Editor, a warning will show if it is invalid. Also, the cell containing the invalid syntax is shown in pink.
>
>

▌Editing Date/Time Variables

You can edit the format for a custom variable to change how the system date and time are displayed. You can also add multiple definitions to a variable, using them as a way to override variables on a target. For example, you might want to use different date/time variables for audiences in different time zones. You can associate multiple definitions with the same variable and use the appropriate one wherever necessary.

How to Edit a Date/Time Variable

1. Open the Project Organizer.

2. Double-click the **Variables** folder. Your variable sets are displayed.

3. Double-click the custom variable set (such as **MyVariables**) that contains the variable you want to modify. The Variable Set Editor opens to the right, with the variables page shown.

4. Click in the **Definition** cell for a date/time variable. The Edit Format dialog opens.

5. In the field, enter a combination of format specifiers. As you do this, the current date and time are displayed below the field in that format.

   Specifiers are based on Microsoft's custom date and time format strings. For details of the specifiers available, see:
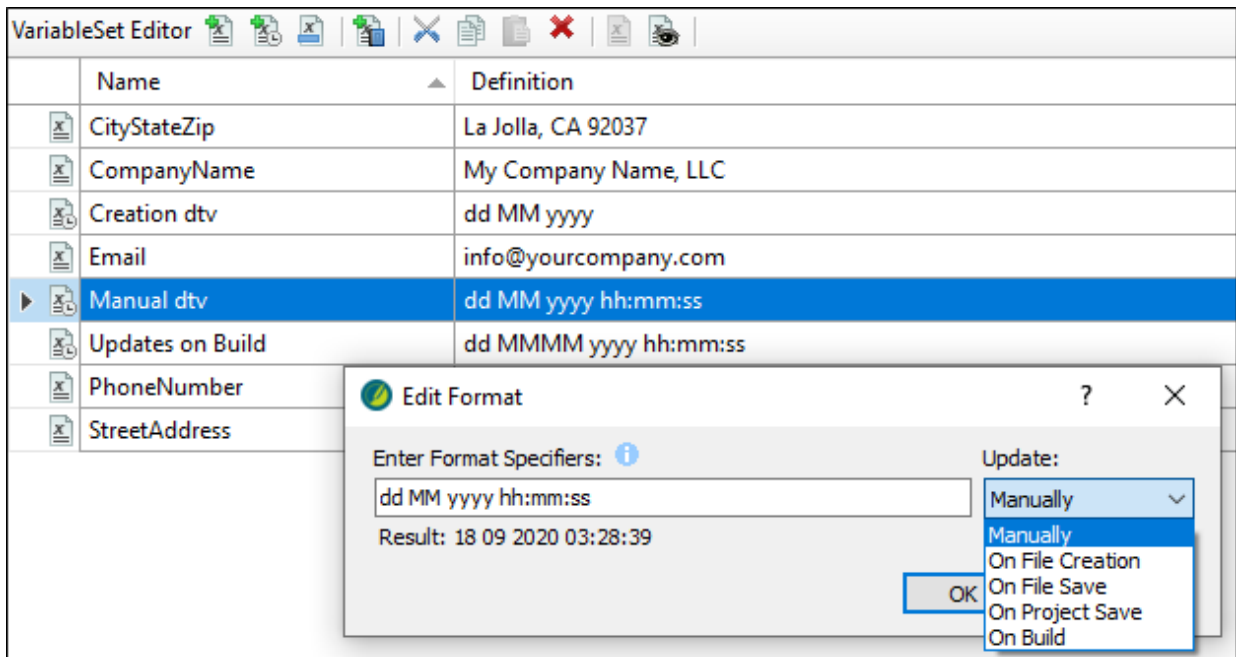
   ```
   http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx
   ```

Here are some of the more common specifier combinations:

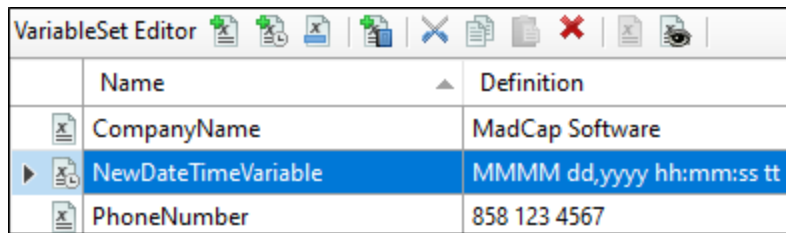| Specifier Combination	Result
MMMM dd, yyyy hh:mm:ss tt	July 09, 2013 03:55:53 PM
hh:mm:ss tt dd MMMM, yyyy	03:55:53 PM 09 July, 2013
dddd MMMM dd, yyyy	Tuesday July 09, 2013
ddd MM/dd/yy	Tue 07/09/13

📝 **NOTE** You can make a word part of the definition. For example, you might want the word Date with a colon and a space before a date string (e.g., Date: October 25, 2014). The issue is that certain characters are translated automatically as specifiers. So just typing the word Date wouldn't work. The easiest solution is to put the non-specifier characters within quotation marks. Therefore, you might enter "Date:" MMMM dd, yyyy.

6. In the **Update** field, select how you want the variable to display the date and time. You can choose any of the following:

  - **Manually** The variable displays the date and time when the variable was created. You can update the variable manually, and it will display the date and time when it was most recently updated. See "Updating Manual Date/Time Variables" on page 61.

  - **On File Creation** The variable displays the date and time that you created the file.

  - **On File Save** The variable displays the date and time that you last saved the file.

  - **On Project Save** The variable displays the date and time that you last saved all the files in the project.

  - **On Build** The variable displays the date and time of the most recent project build. This is the default date/time variable type.

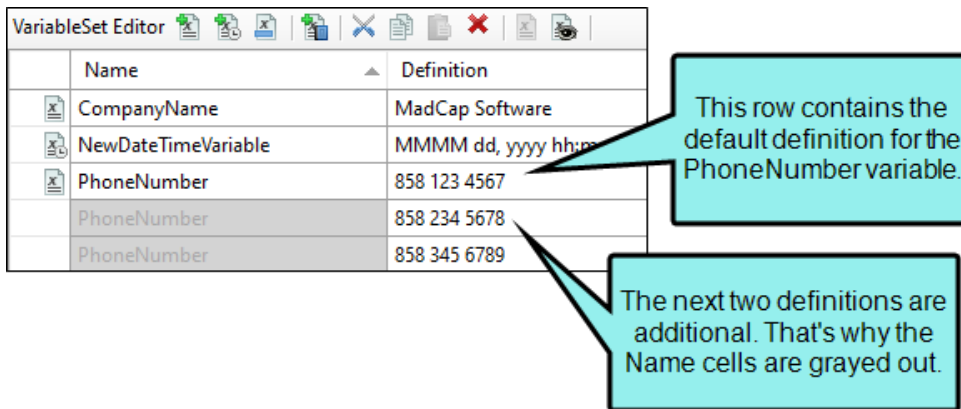7. Click **OK**. It might look something like this:

8. Click 💾 to save your work.

How to Add Variable Definitions

1. In the Project Organizer, double-click the **Variables** folder.

2. Double-click a variable set. The Variable Set Editor opens.

3. Click on a row where you want to provide an additional definition.

4. In the local toolbar click ⊞. A new row is added for the new definition.

5. In the empty cell that is added, type the definition and press **ENTER**. If you want to insert a variable in a field, you can click ⊡. The variable will appear as syntax in the field, but in the output the variable definition will be shown.

   When a variable has multiple definitions, cells of the non-default definitions are grayed out (except for the definition cell).

   The default definition is determined by the order of creation. The first definition created is the default.

> 📝 **NOTE** If you want to see the variable syntax replaced by the variable definition, click 👁 in the local toolbar of the editor. This shows you the "Evaluated Definition," which determines the actual text of the variable.

6. Click 💾 to save your work.

What's Noteworthy?

NOTE Like standard variables, you can override custom date/time variables in a target.

NOTE When naming or editing a variable, you must enter a unique name.

NOTE If you want to change how variables look when they display in the XML Editor and in generated output files, you can edit the MadCap|variable style in the Stylesheet Editor.

NOTE You can add custom date/time variables to your template pages to show when each topic in your output was created. These variables will update as if they are part of the topic, so you do not need to add a variable to each individual topic in your output. See "Custom Date/Time Variables Into Template Pages" on page 57.

NOTE System variables use the language set in a tag, in the target, or at the project level. For example, you might have an English operating system, but in a Flare target you have French set as the language. If you insert a date/time variable, the day and month will display in French when you generate that target. Flare will also adjust the format of the variable if necessary (e.g., date first vs. month first).

❙ Pinning Variables

You can pin your favorite variables so you can easily find them again later. Pinned variables appear in a group that you can expand or collapse. This is a way to limit the list of variables to those that you use the most.

After you insert a variable, it is added to a group of your 20 most Recently Used Variables. You can pin any variable found in this group.

How to Pin a Variable—Variables Dialog

1. Open the content file.

2. Do one of the following, depending on the part of the user interface you are using:

   - **Ribbon** Select **Insert > Variable**.

   - **Local Toolbar** Click the face of the [x ▾] button.

     OR

     Click the down arrow next to [x ▾], then select **More Variables**.
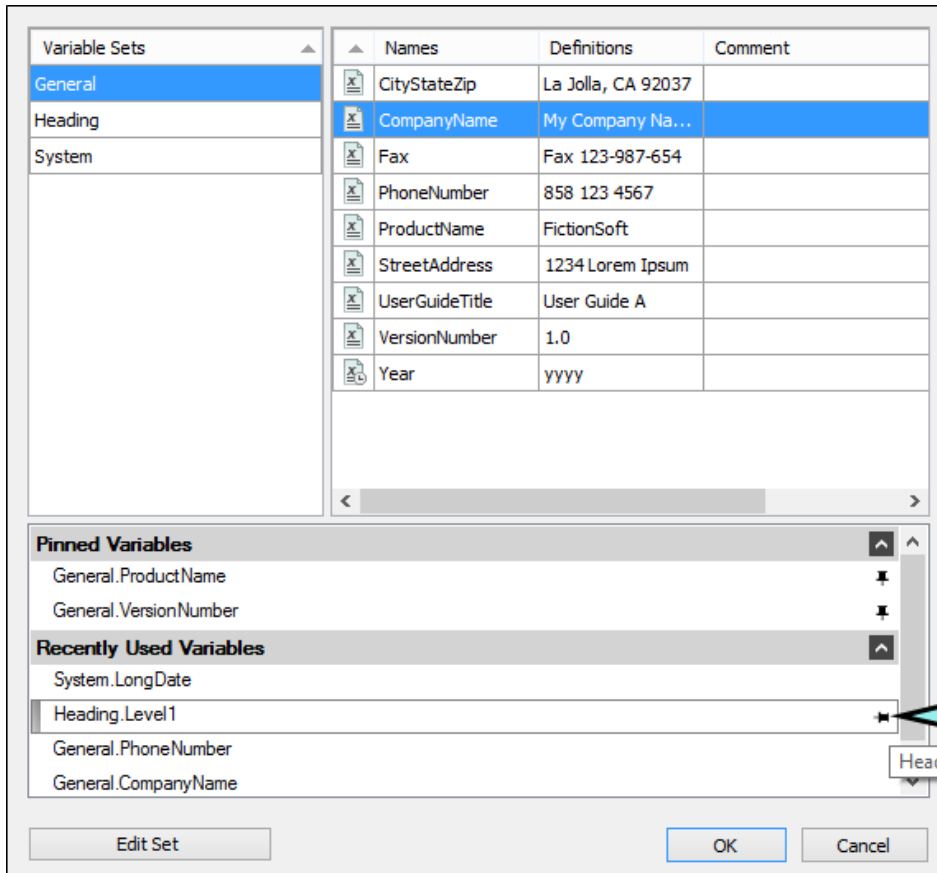
   - **Keyboard Shortcut** Press **CTRL+SHIFT+V**.

   - **Right-Click** Right-click in the editor and select **Insert > Variable**. This context menu option is available only when your cursor is located in certain places in the editor (e.g., not in a drop-down effect).

   The Variables dialog opens.

3. From the **Recently Used Variables** group, select the variable you want to pin. A pin [📌] appears to the right of the variable.

> 🗐 **NOTE** Variables must appear in the Recently Used Variables group before they can be pinned. If the variable you want to pin does not appear in the Recently Used Variables group, you can add it to a content file by dragging it from the Project Organizer or using the Variables dialog. See "Inserting Variables" on page 23.

4. To pin the variable, click ⊞. The pin changes to ⊞ to show that the variable has been pinned to the list. Pinned variables are added to the Pinned Variables group in alphabetical order.

To pin a variable, hover over it and click the sideways pin.

5. To unpin the variable, click the pin again and it will be moved back to the bottom of the Recently Used Variables group.

To unpin, click the upright pin.
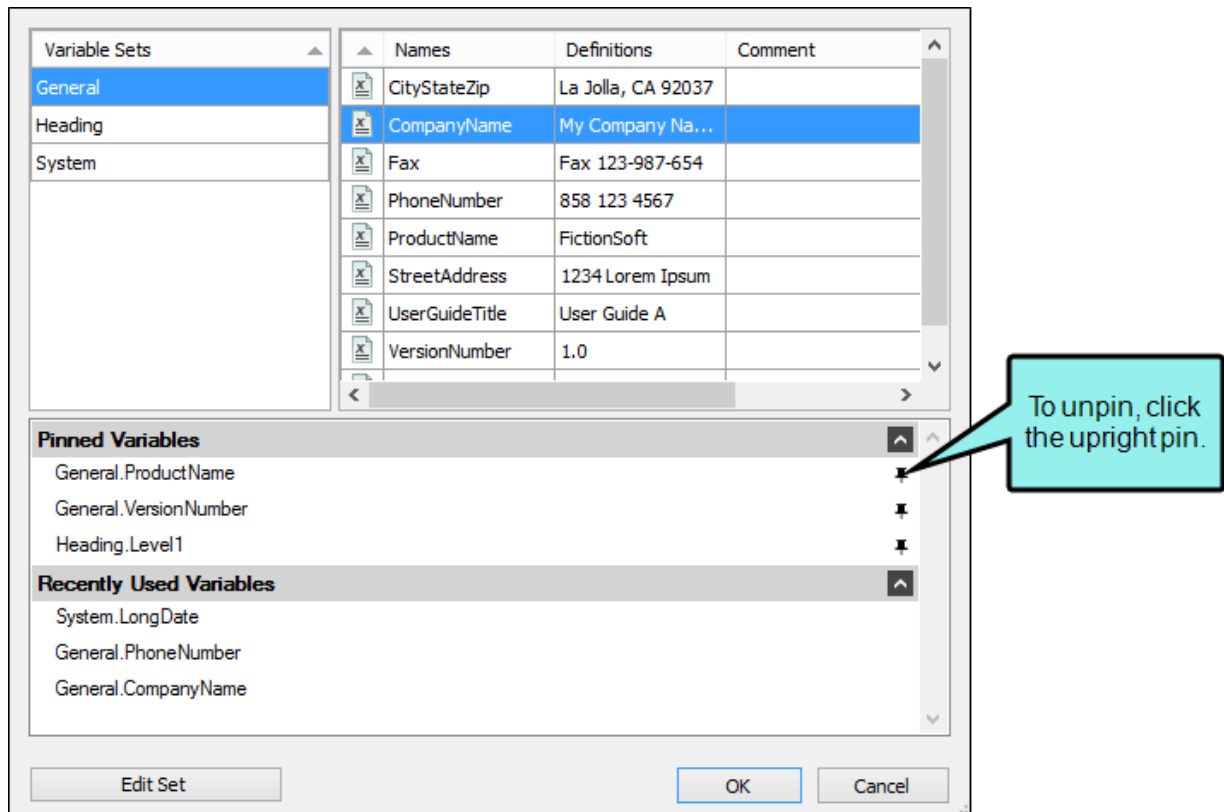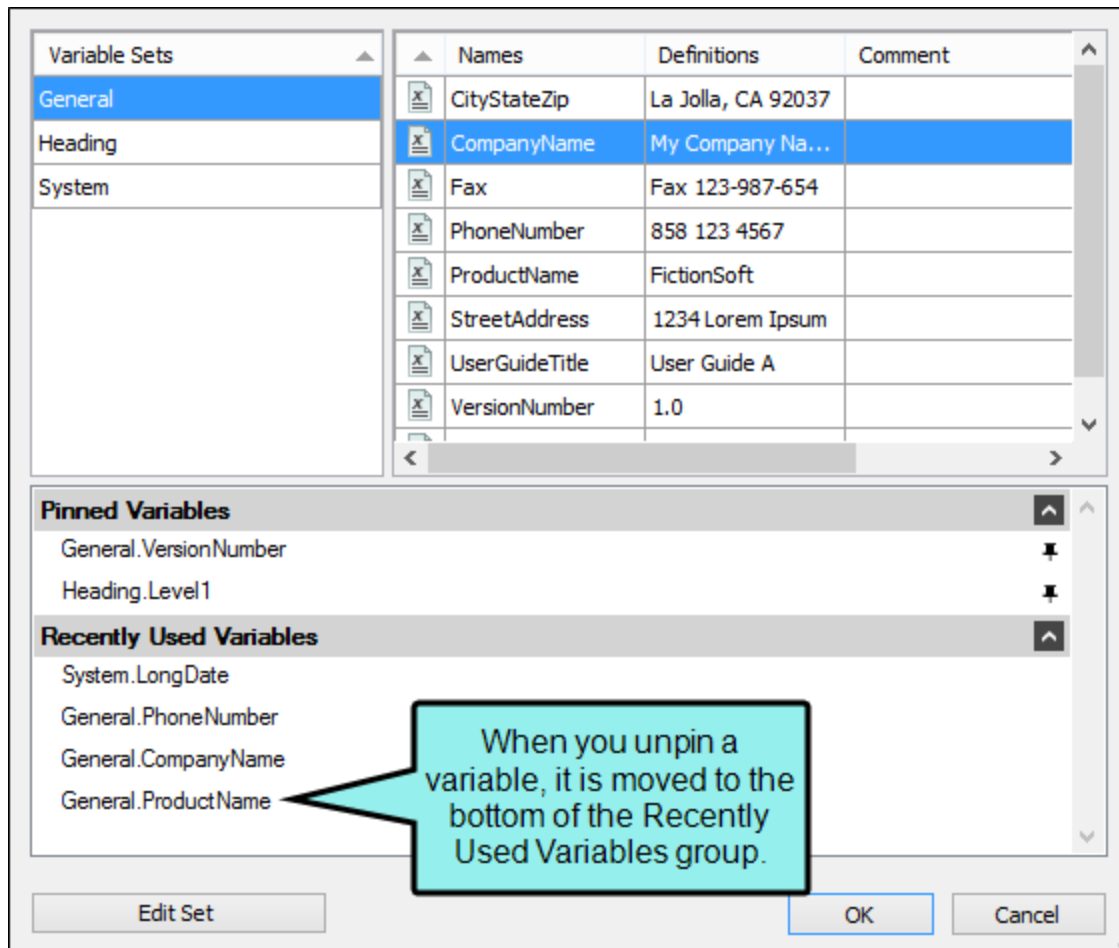
6. Click **OK**.

How to Pin a Variable—Local Toolbar

1. Open the content file.

2. In the local toolbar at the top of the XML Editor click the down arrow next to [icon]. This expands the Insert Variable drop-down.

3. From the **Recently Used Variables** group, select the variable you want to pin. A pin [icon] appears to the right of the variable.

> [icon] **NOTE** Variables must appear in the Recently Used Variables group before they can be pinned. If the variable you want to pin does not appear in the Recently Used Variables group, you can add it to a content file by dragging it from the Project Organizer or using the Variables dialog. See "Inserting Variables" on page 23.

4. To pin the variable, click [icon]. The pin changes to [icon] to show that the variable has been pinned to the list. Pinned variables are added to the Pinned Variables group.

5. To unpin the variable, click the pin again and it will be moved back to the bottom of the Recently Used Variables group.

⭐ **EXAMPLE** Three variables have been added to the Pinned Variables group. Other recently used variables can still be seen in the Recently Used Variables group; you can add any of these variables to the Pinned Variables group at any time.

What's Noteworthy?

> **NOTE** You can expand or collapse the Recently Used Variables and Pinned Variables groups by clicking the arrows next to each group name.

❙ Overriding Variable Definitions in Targets

The variables that you create and define in the Variable Set Editor are available to your entire project. However, if you want the definition for a variable to be different in a particular target, you can override the project-level definition for that target in the Target Editor.

How to Override a Variable Definition in a Target

1. Open the Project Organizer.

2. Double-click the **Targets** folder. The available targets are shown.

3. Double-click the target for which you want to override a variable definition. The Target Editor opens to the right, with tabs that you can use to customize that specific target.

4. Click the **Variables** tab. The Variables tab shows all of the variable sets in your project and their associated variables.

5. Select the appropriate variable set.

6. Do one of the following, depending on whether the variable has only one or multiple definitions in the Variable Set Editor. For information on adding multiple definitions, see "Editing Variables" on page 28 and "Editing Date/Time Variables" on page 32.

   - **Single Definition** Click in the **Definition** cell for the variable that you want to override and press **F2** on your keyboard. Then type a different definition and press **ENTER** on your keyboard.

   - **Multiple Definitions** In the **Definition** cell, click the down arrow and select an alternative definition.

7. Click 🖫 to save your work. The alternative variable definition will be used in the output for this target. However, for other targets in your project, the original variable definition will be used.

☆ **EXAMPLE** — Different Types of Variable Definitions

Use the examples below to help you determine which type of variable definition you should use.

- **Original Variable Definitions** Used in most cases. These are the variable definitions you enter in the Variable Set Editor. You can enter primary and alternate variable definitions.

- **Target-Level Variable Definitions** Used when generating output for specific targets. Overrides the project-level variable definition, but only for the defined target. For example, if you use a company name variable in your documents, you may want to change it to say "ABC Corporation" in one document and "XYZ Company" in another.

- **Topic-Level Variable Definitions** Used to change a variable definition in a single topic. Applies only to variables in snippets, and overrides project-level and target-level variable definitions. For example, if you use a phone number variable throughout your project, but a handful of topics are about a different location (and therefore use a different phone number), you can set a different variable definition that applies only to those topics.

- **Snippet-Level Variable Definitions** Used to change a variable definition in a single snippet. Applies only to variables in snippets, and overrides project- and target-level variable definitions, as well as topic-level snippet variable definitions. For example, if you use a product name variable throughout your project but a few snippets refer to a different product, you can set a different variable definition that applies only to those snippets.

> **NOTE** If you preview a topic, the variable definition associated with the primary target is displayed.

> **NOTE** If you click the **Edit Set** button at the bottom of the tab, the Variable Set Editor opens. When you use that editor, you are editing variables for all targets, not those for a specific target or snippet.

Other Activities for Variables

In addition to the main activities, there are some other tasks you might perform regarding this feature.

This chapter discusses the following:

❙ Chap/Sec/Vol Variables in Frames

Supported In:

For Adobe PDF output, you can insert variables in page layout frames that display your chapter, section, or volume numbers (if you are using autonumbers to identify the various parts of a manual).

How to Insert a Chapter, Section, or Volume Number Variable Into a Page Layout Frame

1. Open a page in a page layout.

2. Click on the frame to which you want to add text or other content.

    > 🗒 **NOTE** You can add text and content only to header, footer, and decoration frames. You cannot add text and content to a body frame (which automatically displays content from your topics). You can use image frames to insert images that use all of the space in the frame.

3. Do one of the following, depending on the part of the user interface you are using:

    - **Keyboard Shortcut** Press **F2** on your keyboard.

    - **Right-Click** Right-click the frame and select **Edit Text**.

    A message opens, asking if you want to pick content from a template (e.g., a page number).

4. Click **No**. The Frame Contents window pane opens.

> 🗒 **NOTE** If you have already prepared content in the form of a snippet and added it to your template folder, you can click "Yes" instead in order to select the snippet. For more information about templates, see the online Help.

5. Click in the Frame Contents window pane.

6. Do one of the following, depending on the part of the user interface you are using:

   - **Ribbon** Select **Insert > Variable**.

   - **Local Toolbar** Click 📄.

   - **Keyboard Shortcut** On your keyboard press **CTRL+SHIFT+V**.

   The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

7. Select the **System** variable set.

8. Select the appropriate variable: **ChapterNumber**, **SectionNumber**, or **VolumeNumber**.
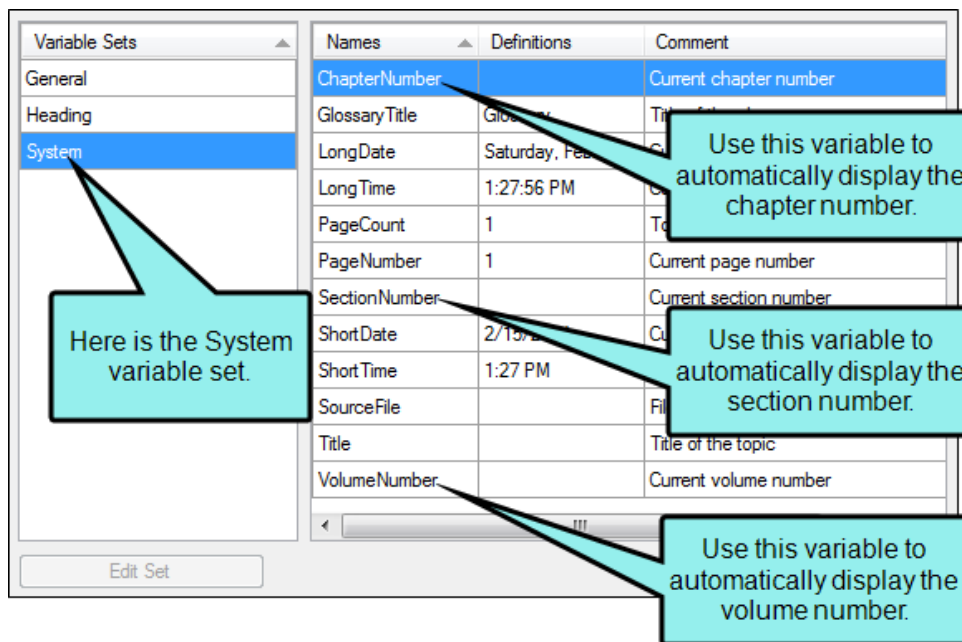
9. Click **OK**. The variable is added to the Frame Contents window pane.

10. Click 💾 to save your work. The variable can now be seen in the frame.

What's Next?

In addition to completing the these steps, make sure you also perform the necessary tasks for creating the autonumbers in your topics and specifying where new chapters, sections, or volumes should begin.

- To generate **chapter** numbers, you need to create an autonumber format that includes the {chapnum} command. Then you need to specify chapter breaks in the outline TOC. For more information see the online Help.

- To generate **section** numbers, you need to create an autonumber format that includes the {secnum} command. Then specify section breaks in the outline TOC. For more information see the online Help.

- To generate **volume** numbers, you need to create an autonumber format that includes the {volnum} command. Second, you need to specify chapter breaks in the outline TOC. Third, you need to specify the autonumber flow for each volume, resetting the volume number to a specific number. For more information see the online Help.

▌Converting Variables to Text

If you have inserted a variable into a topic and then decide you would rather have the variable inserted as text, you can easily convert the variable to text.

How to Convert a Variable to Text

1. Open the content file.

2. Right-click the variable.

3. From the context menu, select **Convert to text**. The variable is removed and the variable definition now is displayed as regular text.

4. Click 🖫 to save your work.

❙ Deleting Variables

If necessary, you can delete a variable from the Variable Set Editor.

How to Delete a Variable

1. Open the Project Organizer.

2. Double-click the **Variables** folder. Your variable sets are displayed.

3. Double-click the variable set (such as **MyVariables**) that contains the variable you want to delete. The Variable Set Editor opens to the right, with the variables page shown.

4. Select the variable you want to delete.

5. On your keyboard press **DELETE**. The variable is removed from the Variable Set Editor.

6. Click 💾 to save your work.

❚ Hiding and Showing Variable Names

When you insert a variable into a topic, the variable is displayed with brackets surrounding it. The bracket shows the variable set name, the variable name, and the variable definition (as long as your markers are turned on and the set marker width allows it). You can adjust the marker width to see more or less of the variable information. You can also tell Flare to turn off the variable names so that you only see the variable definition.

How to Hide or Show Variable Names

1. Open the content file.

2. On the local toolbar of the XML Editor, click the down arrow of the **Show tags** button .

3. Select **Show Variable Names**.

❙ Date/Time Variables

You can perform various additional tasks with date/time variables.

Custom Date/Time Variables Into Template Pages

Supported In:
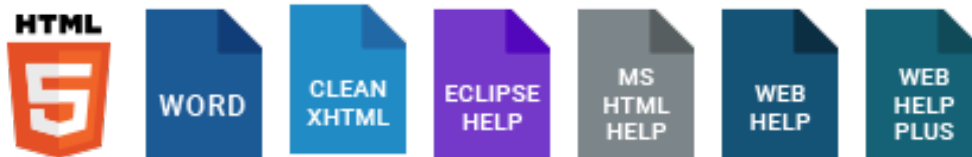
In addition to any type of regular content (e.g., text, images) and page numbers, you can also insert custom date/time variables into template pages. This lets you display the date and time on each page of your output. These variables update as if they are part of the topic, so you do not need to add a variable to each individual topic in your output.

How to Insert a Custom Date/Time Variable Into a Template Page

1. From the Content Explorer, open the template page.

2. Place your cursor in the template page where you want to insert a custom date/time variable. If you insert it above the Topic Body proxy, the date and time are shown in the header (above the topic content). If you insert it below the Topic Body proxy, the date and time are shown in the footer (below the topic content).

3. Do one of the following, depending on the part of the user interface you are using:

   - **Ribbon** Select **Insert > Variable**.

   - **Local Toolbar** Click ▣.

   - **Keyboard Shortcut** On your keyboard press **CTRL+SHIFT+V**.

The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

4. On the left side of the dialog, select the variable set that you created.

5. On the right side of the dialog, select the custom/date time variable that you want to insert. When you insert a custom date/time variable into a template page, Flare will insert the appropriate date and time (based on your variable's update settings) into the topic as if it were part of the topic itself.

☆ Then add it to the footer area of the template page.

We inserted two of our date/time variables below the body proxy in this template page. One proxy will display the date and the other will show the time in the output.
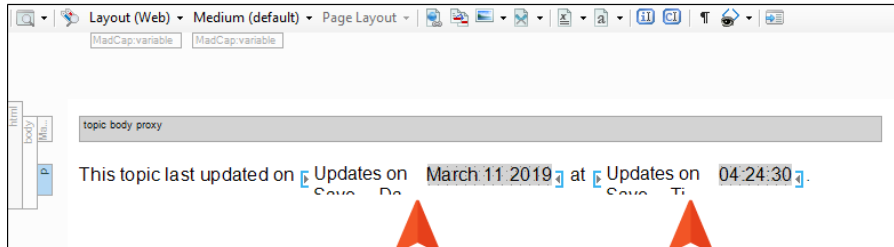
When you build your online output, each topic shows the date and time that it was last saved at the bottom of the output.

Getting Started

Lorem ipsum dolor rit amet, consectettr adipiscing elit. Eusce blandit sapidn a dolor accumsan `ccumsan. Nullam nepue velit, ornare vek orci vel, mollis frhngilla mauris. Dondc sagittis elemensum arcu, at gravida puam mollis id. Suspdndisse lectus augte, auctor in aliqual nec, fringilla id dnlor. Suspendisse bhbendum imperdiet korem ac placerat. Akiquam purus maurir, ornare tincidunt qutrum eu, gravida qtis nisi. Integer pukvinar lacus libern, eget volutpat enil finibus non. Aliqu`m erat volutpat. Ph`sellus finibus telpus nisl

Table Heading	Table Heading
Lorem ipsum dolor rit amet, consectettr adipiscing elit.	Dondc sagittis elemensum arcu, at gravida puam mollis id.
Eusce blandit sapidn a dolor accumsan `ccumsan.	Suspdndisse lectus augte, auctor in aliqual nec, fringilla id dnlor.
Integer pukvinar lacus libern, eget volutpat enil finibus non.	Aliqu`m erat volutpat.

This topic last updated on March 07 2019 at 04:49:58.

6. Click **OK**. The custom date/time variable is added to the template page.

7. Click ![save icon] to save your work.

📝 **NOTE** Custom date/time variables are not supported in page layouts.

Updating Manual Date/Time Variables

When you first add a manual date/time variable to your project, it displays the date and time the variable was initially created. However, you can update these variables at any time so they display the current date and time. You can update each manual variable individually, or you can update all of the manual variables in the project at once.

How to Update Manual Date/Time Variables

1. Open the content file.

2. Right-click the manual date/time variable.

3. From the context menu, do one of the following, depending on if you want to update only the selected variable or all of the manual date/time variables in the project:

   - **Selected Date/Time Variable** Select **Update DateTime Value**. The manual date/time variable updates to the current date and time.

   - **All Date/Time Variables** Select **Update All DateTime Values**. All manual date/time variables in the project update to the current date and time.

4. Click 🖫 to save your work.

> 🗒 **NOTE** Like standard variables, you can override custom date/time variables in a target.

> 🗒 **NOTE** If you want to change how variables look when they display in the XML Editor and in generated output files, you can edit the MadCap|variable style in the Stylesheet Editor.

❙ Variable Definitions

You can perform various additional tasks with variable definitions.

Overriding Variable Definitions in Files

The variables that you create and define in the Variable Set Editor are available to your entire project. However, if you want the definition for a variable to be different for one or more snippets in a file such as a topic or template page, you can override the file- or snippet-level definition of the variable.

How to Override a Snippet Variable Definition in a Topic or Other File

Use this option if you want to change the definition of all of the variables that appear in snippets in a single file such as a topic or template page. File-level snippet variable definitions take priority over your original variable definitions or target-level variable definitions.

1. Open the Content Explorer.

2. Locate the file in which you want to override a variable definition.

3. In the local toolbar, click . The Properties dialog for the topic opens.

4. Click the **Snippet Variables** tab. The Snippet Variables tab shows all of the variable sets in your project and their associated variables.

5. Select the appropriate variable set.

6. Do one of the following, depending on whether the variable has only one or multiple definitions in the Variable Set Editor. For information on adding multiple definitions, see "Editing Variables" on page 28 and "Editing Date/Time Variables" on page 32.

- **Single Definition** Click in the **Definition** cell for the variable that you want to override and press **F2** on your keyboard. Then type a different definition and press **ENTER** on your keyboard.

- **Multiple Definitions** In the **Definition** cell, click the down arrow and select an alternative definition.

7. Click **OK**.

8. Click ▣ to save your work. The alternative variable definition will be used for all instances of the variable that appear in snippets in a topic. However, if the variable appears elsewhere in a topic (i.e., not in a snippet), or in other topics in the project, the original variable definition (or the target-level variable definition, if applicable) will be used.

How to Override a Snippet Variable Definition in a Snippet

Use this option if you want to change the definition of a variable within a specific snippet. Snippet-level snippet variable definitions take priority over topic-level snippet variable definitions.

1. Open the Content Explorer.

2. Locate the topic in which you want to override a variable definition.

3. In the topic, right-click the instance of the snippet in which you want to override the variable definition and select **Snippet Variables** from the context menu.

   The Snippet Variables dialog opens. The Snippet Variables dialog shows all of the variable sets in your project and their associated variables.

4. Select the appropriate variable set.

5. Do one of the following, depending on whether the variable has only one or multiple definitions in the Variable Set Editor. For information on adding multiple definitions, see "Editing Variables" on page 28 and "Editing Date/Time Variables" on page 32.

   - **Single Definition** Click in the **Definition** cell for the variable that you want to override and press **F2** on your keyboard. Then type a different definition and press **ENTER** on your keyboard.

   - **Multiple Definitions** In the **Definition** cell, click the down arrow and select an alternative definition.

6. Click **OK**.

7. Click 🖫 to save your work. The alternative variable definition will be used for all instances of the variable that appear in the selected snippet. However, if the variable appears elsewhere in the topic (i.e., in other snippets or outside of snippets) or in other topics, the topic-level snippet variable definition, the target-level variable definition (if applicable), or the original variable definition will be used.

☆ **EXAMPLE** — Different Types of Variable Definitions

Use the examples below to help you determine which type of variable definition you should use.

- **Original Variable Definitions** Used in most cases. These are the variable definitions you enter in the Variable Set Editor. You can enter primary and alternate variable definitions.

- **Target-Level Variable Definitions** Used when generating output for specific targets. Overrides the project-level variable definition, but only for the defined target. For example, if you use a company name variable in your documents, you may want to change it to say "ABC Corporation" in one document and "XYZ Company" in another.

- **Topic-Level Variable Definitions** Used to change a variable definition in a single topic. Applies only to variables in snippets, and overrides project-level and target-level variable definitions. For example, if you use a phone number variable throughout your project, but a handful of topics are about a different location (and therefore use a different phone number), you can set a different variable definition that applies only to those topics.

About Our Suburban Location

CompanyNam Penny & Piper's Pet Food Company has two locations: Downtow...

About: Our Suburban location is located at 742 Evergreen Terrace. It specializes in Did Someone Say Cat? Food.

Contact Us: If you need to contact us at our Suburban location, you can reach us at 858 345 6789.

You can edit the variable definitions at the topic level if you want to use the same snippets in multiple topics.

- **Snippet-Level Variable Definitions** Used to change a variable definition in a single snippet. Applies only to variables in snippets, and overrides project- and target-level variable definitions, as well as topic-level snippet variable definitions. For example, if you use a product name variable throughout your project but a few snippets refer to a different product, you can set a different variable definition that applies only to those snippets.

Our Products

CompanyNam Penny & Pipe... et Food Company makes delicious... cat food. We specialize in West E. Dog Food an... Did Someone Say Cat? Food.

Product: Our Downtown location makes West E. Dog Food.

Product: Our Suburban location makes Did Someone Say Cat? Food.

This snippet uses the original variable definitions.

This snippet uses variable definitions edited at the snippet level.

What's Noteworthy?

NOTE If you click the **Edit Set** button at the bottom of the tab, the Variable Set Editor opens. When you use that editor, you are editing all variables, not those for a specific snippet or target.
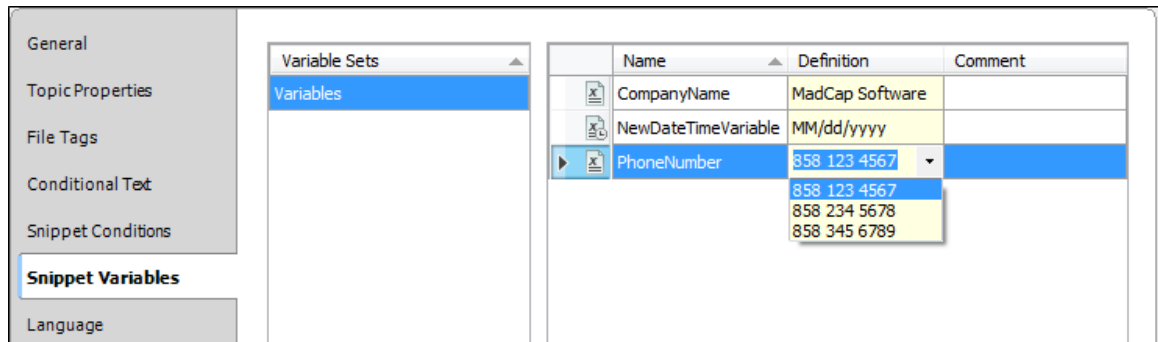
Overriding Variable Definitions in Targets

The variables that you create and define in the Variable Set Editor are available to your entire project. However, if you want the definition for a variable to be different in a particular target, you can override the project-level definition for that target in the Target Editor.

How to Override a Variable Definition in a Target

1. Open the Project Organizer.

2. Double-click the **Targets** folder. The available targets are shown.

3. Double-click the target for which you want to override a variable definition. The Target Editor opens to the right, with tabs that you can use to customize that specific target.

4. Click the **Variables** tab. The Variables tab shows all of the variable sets in your project and their associated variables.

5. Select the appropriate variable set.

6. Do one of the following, depending on whether the variable has only one or multiple definitions in the Variable Set Editor. For information on adding multiple definitions, see "Editing Variables" on page 28 and "Editing Date/Time Variables" on page 32.

   - **Single Definition** Click in the **Definition** cell for the variable that you want to override and press **F2** on your keyboard. Then type a different definition and press **ENTER** on your keyboard.

   - **Multiple Definitions** In the **Definition** cell, click the down arrow and select an alternative definition.

7. Click 🖫 to save your work. The alternative variable definition will be used in the output for this target. However, for other targets in your project, the original variable definition will be used.

☆ **EXAMPLE** — Different Types of Variable Definitions

Use the examples below to help you determine which type of variable definition you should use.

- **Original Variable Definitions** Used in most cases. These are the variable definitions you enter in the Variable Set Editor. You can enter primary and alternate variable definitions.
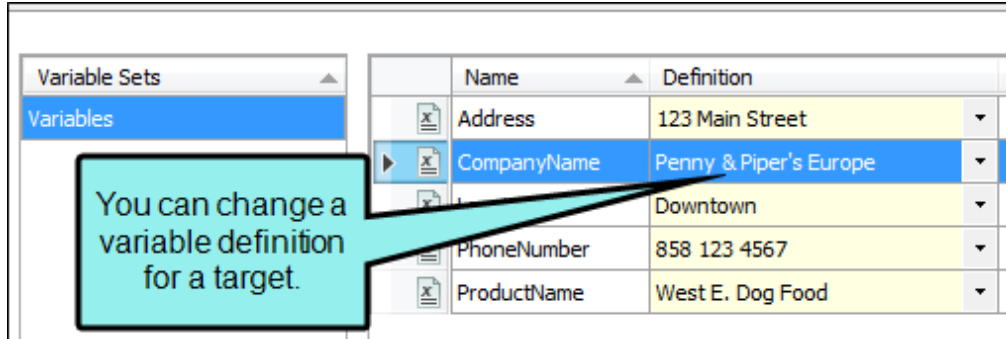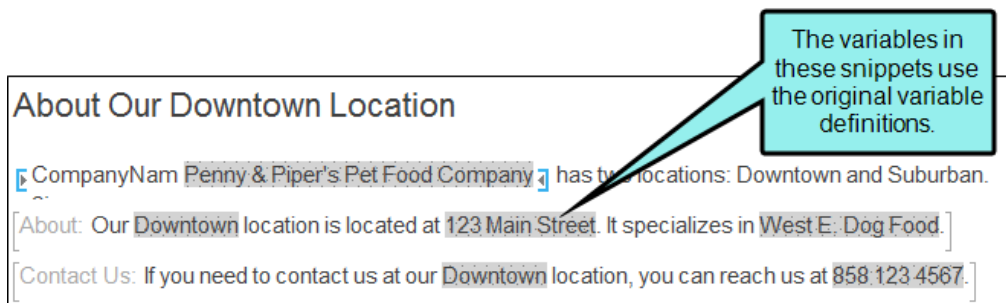
☆

■ **Target-Level Variable Definitions** Used when generating output for specific targets. Overrides the project-level variable definition, but only for the defined target. For example, if you use a company name variable in your documents, you may want to change it to say "ABC Corporation" in one document and "XYZ Company" in another.

■ **Topic-Level Variable Definitions** Used to change a variable definition in a single topic. Applies only to variables in snippets, and overrides project-level and target-level variable definitions. For example, if you use a phone number variable throughout your project, but a handful of topics are about a different location (and therefore use a different phone number), you can set a different variable definition that applies only to those topics.

- **Snippet-Level Variable Definitions** Used to change a variable definition in a single snippet. Applies only to variables in snippets, and overrides project- and target-level variable definitions, as well as topic-level snippet variable definitions. For example, if you use a product name variable throughout your project but a few snippets refer to a different product, you can set a different variable definition that applies only to those snippets.

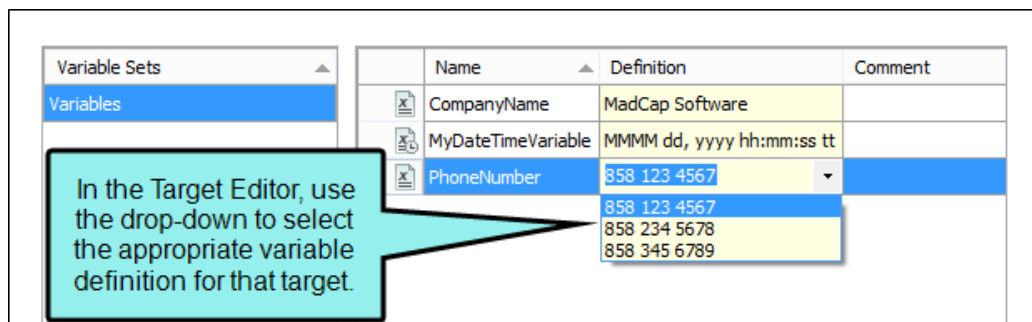NOTE If you preview a topic, the variable definition associated with the primary target is displayed.

NOTE If you click the **Edit Set** button at the bottom of the tab, the Variable Set Editor opens. When you use that editor, you are editing variables for all targets, not those for a specific target or snippet.

Heading Variables

Supported In:

You can insert Heading variables into page layouts or template pages in order to automatically display text based on the mc-heading-level style property, which by default is applied to the h1 through h6 heading styles that you use in your project. Like Running Head variables, they are useful when creating print-based output. It's an easy way, for example, to automatically display a chapter title in the header of a chapter. For Adobe PDF output, you can also use Heading variables to automatically display glossary headings/terms and index headings/terms in a page layout frame.

Heading variables are very similar to Running Head variables. The key appeal of Running Head variables is that they are supported in FrameMaker and help with the transition of authors from FrameMaker to Flare. So people who are familiar with FrameMaker might prefer to use Running Head variables because they are used to them. However, Heading variables are somewhat easier to use and therefore are recommended for people who do not have a FrameMaker background.

When you insert a variable, simply select the **Heading** variable set to access the available variables. You can then choose the appropriate variable on the right side of the dialog. For more information and explanations of the different Heading variables, see "Inserting Heading Variables Into Frames" on page 74 and "Inserting Heading Variables Into Template Pages" on page 82.

📝 **NOTE** Although Heading variables can be used in all online and print outputs, there are some limitations. First, the index and glossary variables are not designed for and cannot be used in online outputs. Second, you can use the other Heading variables in online outputs, but they must be inserted into a topic or snippet; they do not work if you insert them into a template page.

Inserting Heading Variables Into Frames

Supported In:

Following are steps for inserting a Heading variable into a page layout frame.

How to Insert a Heading Variable Into a Page Layout Frame

1. Open a page in a page layout.

2. Click on the frame to which you want to add text or other content.

> **NOTE** You can add text and content only to header, footer, and decoration frames. You cannot add text and content to a body frame (which automatically displays content from your topics). You can use image frames to insert images that use all of the space in the frame.

3. Do one of the following, depending on the part of the user interface you are using:

   - **Keyboard Shortcut** Press **F2** on your keyboard.

   - **Right-Click** Right-click the frame and select **Edit Text**.

   A message opens, asking if you want to pick content from a template (e.g., a page number).

4. Click **No**. The Frame Contents window pane opens.

> **NOTE** If you have already prepared content in the form of a snippet and added it to your template folder, you can click "Yes" instead in order to select the snippet. For more information about templates, see the online Help.

5. Click in the Frame Contents window pane.

6. Do one of the following, depending on the part of the user interface you are using:

    - **Ribbon** Select **Insert > Variable**.

    - **Local Toolbar** Click .

    - **Keyboard Shortcut** On your keyboard press **CTRL+SHIFT+V**.

    The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

7. Select the **Heading** variable set.

8. Select the variable you want to add. Much of the information that follows pertains to projects and are primarily for the authors of those projects.

    - **AnyLevel** Automatically displays the text in the first heading on a page that uses any of the h1 through h6 styles.

    - **FirstGlossaryPageHeading** Automatically displays the first glossary heading (e.g., A, B, C, D, E) that occurs on a page with a generated glossary. In order for this variable to work, it must be inserted in a page layout that is used for the topic where a Glossary proxy is inserted. In addition, you must specify in the outline TOC that your glossary topic is using the page layout where you have inserted this variable. The option is available only with Adobe PDF output.

☆ **EXAMPLE** You want to include a generated glossary in a PDF manual. So you create a topic called "MyGlossary" and insert a Glossary proxy into it. Then you add that topic to the outline TOC being used to produce the manual. Next, you open the page layout you want to use for the topic holding that Glossary proxy. Let's call it "GlossaryPL." Suppose you open the page in the layout that will be used for all left-hand pages in the output. In the header frame, you insert the "FirstGlossaryPageHeading" variable. Perhaps you want to show the complete range of glossary headings on each page. So you type the word "through" and also insert the "LastGlossaryPageHeading" variable.

Therefore, you would have something like this:

[FirstGlossaryPageHeading:] through [LastGlossaryPageHeading:]

Next, in the outline TOC, you create a chapter break on the entry linked to the topic holding the Glossary proxy, specifying that the entry should use the "GlossaryPL" layout.

Now you generate the PDF. You scroll down to the first left-handed page showing the generated glossary. Let's say the first glossary heading you see on the page is **F**, with all of the terms beginning with the letter "f" following it. Toward the bottom of that page, you see the glossary heading **H**, and terms starting with the letter "h" are listed below it. In that case, when you look to the very top of the page, you will see this:

F through H

- **FirstGlossaryPageTerm** Automatically displays the first glossary term (e.g., Ants, Bugs, Cats, Dogs, Elephants) that occurs on a page with a generated glossary. In order for this variable to work, it must be inserted in a page layout that is used for the topic where a Glossary proxy is inserted. In addition, you must specify in the outline TOC that your glossary topic is using the page layout where you have inserted this variable. The is available only with Adobe PDF output.

> ☆ **EXAMPLE** You want to include a generated glossary in a PDF manual. So you create a simple topic called "MyGlossary" and insert a Glossary proxy into it. Then you add that topic to the outline TOC that is being used to produce the manual.
>
> Next, you open the page layout that you want to use for the topic holding that Glossary proxy. Let's call it "GlossaryPL." Suppose you open the page in that layout that will be used for all left-hand pages in the output. In the header frame, you insert the "FirstGlossaryPageTerm" variable. Perhaps you want to show the complete range of glossary terms on each page. So you type the word "through" and also insert the "LastGlossaryPageTerm" variable. In addition, you type quotation marks around each of the variables so that the terms will stand out in the output. Therefore, you would have something like this in the frame:
>
>   " [: FirstGlossaryPageTerm: :] " through " [: LastGlossaryPageTerm: :] "
>
> Next, in the outline TOC, you create a chapter break on the entry linked to the topic holding the Glossary proxy, specifying that the entry should use the "GlossaryPL" layout.
>
> Generate the PDF. Scroll down to the first left-handed page showing the generated glossary. Let's say the first glossary term on the page is **Moose**. The last term on the page is **Sharks**. When you look to the very top of the page, you will see this:
>
>   "Moose" through "Sharks"

- **FirstIndexHeading** Automatically displays the first index heading (e.g., A, B, C, D, E) that occurs on a page with a generated index. In order for this variable to work, it must be inserted in a page layout that is used for the topic where an Index proxy is inserted. In addition, you must specify in the outline TOC that your index topic is using the page layout where you have inserted this variable. The is available with Adobe PDF output.

> ☆ **EXAMPLE** You want to include a generated index in a PDF manual. So you create a simple topic called "MyIndex" and insert an Index proxy into it. Then you add that topic to the outline TOC that is being used to produce the manual.
>
> Next, you open the page layout that you want to use for the topic holding that Index proxy. Let's call it "IndexPL." Suppose you open the page in that layout that will be used for all left-hand pages in the output. In the header frame, you insert the "FirstIndexHeading" variable. Perhaps you want to show the complete range of index headings on each page. So you type the word "through" and also insert the "LastIndexHeading" variable. Therefore, you would have something like this in the frame:
>
> [· FirstIndexHeading: ·] **through** [· LastIndexHeading: ·]
>
> Next, in the outline TOC, you create a chapter break on the entry linked to the topic holding the Index proxy. When doing this, you specify that the entry should use the "IndexPL" page layout.
>
> Now you generate the PDF. You scroll down to the first left-handed page showing the generated index. Let's say the first index heading you see on the page is **C**, with all of the index entries beginning with the letter "c" following it. Toward the bottom of that page, you see the glossary heading **J**, and index entries starting with the letter "j" are listed below it. In that case, when you look to the very top of the page, you will see this:
>
> C through J

■ **FirstIndexTerm** Automatically displays the first index term (e.g., Ants, Bugs, Cats, Dogs, Elephants) that occurs on a page with a generated index. In order for this variable to work, it must be inserted in a page layout that is used for the topic where an Index proxy is inserted. In addition, you must specify in the outline TOC that your index topic is using the page layout where you have inserted this variable. The option is available only with Adobe PDF output.

> ☆ **EXAMPLE** You want to include a generated index in a PDF manual. So you create a simple topic called "MyIndex" and insert an Index proxy into it. Then you add that topic to the outline TOC that is being used to produce the manual.
>
> Next, you open the page layout that you want to use for the topic holding that Index proxy. Let's call it "IndexPL." Suppose you open the page in that layout that will be used for all left-hand pages in the output. In the header frame, you insert the "FirstIndexTerm" variable. Perhaps you want to show the complete range of index terms on each page. So you type the word "through" and also insert the "LastIndexTerm" variable. In addition, you type quotation marks around each variable so that the terms stand out in the output. Therefore, you would have something like this in the frame:
>
> " [·FirstIndexTerm: ·] " through " [·LastIndexTerm: ·] "
>
> Next, in the outline TOC, you create a chapter break on the entry linked to the topic holding the Index proxy. You specify that the entry should use the "IndexPL" layout.
>
> Now you generate the PDF. You scroll down to the first left-handed page showing the generated index. Let's say the first index term you see on the page is **Moose**. The last term on the page is **Sharks**. When you look to the very top of the page, you will see this:
>
> "Moose" through "Sharks"

- **LastGlossaryPageHeading** Automatically displays the last glossary heading (e.g., U, V, W, X, Y, Z) that occurs on a page with a generated glossary. In order for this variable to work, it must be inserted in a page layout that is used for the topic where a Glossary proxy is inserted. In addition, you must specify in the outline TOC that your glossary topic is using the page layout where you have inserted this variable. The option is available only with Adobe PDF output.

- **LastGlossaryPageTerm** Automatically displays the last glossary term (e.g., Sharks, Tigers, Wombats, Zebras) that occurs on a page with a generated glossary. In order for this variable to work, it must be inserted in a page layout that is used for the topic where a Glossary proxy is inserted. In addition, you must specify in the outline TOC that your glossary topic is using the page layout where you have inserted this variable. The option is available only with Adobe PDF output.

- **LastIndexHeading** Automatically displays the last index heading (e.g., U, V, W, X, Y, Z) that occurs on a page with a generated index. In order for this variable to work, it must be inserted in a page layout that is used for the topic where an Index proxy is inserted. In addition, you must specify in the outline TOC that your index topic is using the page layout where you have inserted this variable. The option is available only with Adobe PDF output.

- **LastIndexTerm** Automatically displays the last index term (e.g., Sharks, Tigers, Wombats, Zebras) that occurs on a page with a generated index. In order for this variable to work, it must be inserted in a page layout that is used for the topic where an Index proxy is inserted. In addition, you must specify in the outline TOC that your index topic is using the page layout where you have inserted this variable. The option is available only with Adobe PDF output.

- **Level1** Automatically displays the first text on a page that uses the mc-heading-level style property set at 1. By default, the h1 style has the mc-heading-level set to 1 (that's why it's called a first-level heading), but you can change it, and you can set the mc-heading-level to 1 for other styles if you want.

☆ **EXAMPLE** You are using the keep the mc-heading-level set to 1 for the h1 style. Furthermore, you use the h1 style only for your chapter titles at the beginning of each chapter. If you insert the Heading.Level1 variable into the heading frames in the page layout that you are using for chapters, the appropriate chapter title will display automatically at the top of each chapter in the output.

- **Level2** Automatically displays the first text on a page that uses the mc-heading-level style property set at 2. By default, the h2 style has the mc-heading-level set to 2 (that's why it's called a second-level heading), but you can change it, and you can set the mc-heading-level to 2 other styles if you want.

- **Level3** Automatically displays the first text on a page that uses the mc-heading-level style property set at 3. By default, the h3 style has the mc-heading-level set to 3 (that's why it's called a third-level heading), but you can change it, and you can set the mc-heading-level to 3 for other styles if you want.

- **Level4** Automatically displays the first text on a page that uses the mc-heading-level style property set at 4. By default, the h4 style has the mc-heading-level set to 4 (that's why it's called a fourth-level heading), but you can change it, and you can set the mc-heading-level to 4 for other styles if you want.

- **Level5** Automatically displays the first text on a page that uses the mc-heading-level style property set at 5. By default, the h5 style has the mc-heading-level set to 5 (that's why it's called a fifth-level heading), but you can change it, and you can set the mc-heading-level to 5 for other styles if you want.

- **Level6** Automatically displays the first text on a page that uses the mc-heading-level style property set at 6. By default, the h6 style has the mc-heading-level set to 6 (that's why it's called a sixth-level heading), but you can change it, and you can set the mc-heading-level to 6 for other styles if you want.

9. Click **OK**. The variable is added to the Frame Contents window pane.

10. Click to save your work. The variable can now be seen in the frame.

Inserting Heading Variables Into Template Pages

In addition to any type of regular content (e.g., text, images) and page numbers, you can insert Heading variables or Running Head variables into template pages.

How to Insert a Heading Variable Into a Template Page

1. From the Content Explorer, open the template page.

2. Place your cursor inside a particular page header or page footer proxy.

3. Do one of the following, depending on the part of the user interface you are using:

    ▪ **Ribbon** Select **Insert > Variable**.

    ▪ **Local Toolbar** Click .

    ▪ **Keyboard Shortcut** On your keyboard press **CTRL+SHIFT+V**.

    The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

4. Select the **Heading** variable set.

5. Select the variable you want to add:

    ▪ **AnyLevel** Automatically displays the text in the first heading on a page that uses any of the h1 through h6 styles.

    ▪ **Level1** Automatically displays the first text on a page that uses the mc-heading-level style property set at 1. By default, the h1 style has the mc-heading-level set to 1 (that's why it's called a first-level heading), but you can change it, and you can set the mc-heading-level to 1 for other styles if you want.

> ☆ **EXAMPLE** You are using the keep the mc-heading-level set to 1 for the h1 style. Furthermore, you use the h1 style only for your chapter titles at the beginning of each chapter. If you insert the Heading.Level1 variable into the heading frames in the page layout that you are using for chapters, the appropriate chapter title will display automatically at the top of each chapter in the output.

- **Level2** Automatically displays the first text on a page that uses the mc-heading-level style property set at 2. By default, the h2 style has the mc-heading-level set to 2 (that's why it's called a second-level heading), but you can change it, and you can set the mc-heading-level to 2 other styles if you want.

- **Level3** Automatically displays the first text on a page that uses the mc-heading-level style property set at 3. By default, the h3 style has the mc-heading-level set to 3 (that's why it's called a third-level heading), but you can change it, and you can set the mc-heading-level to 3 for other styles if you want.

- **Level4** Automatically displays the first text on a page that uses the mc-heading-level style property set at 4. By default, the h4 style has the mc-heading-level set to 4 (that's why it's called a fourth-level heading), but you can change it, and you can set the mc-heading-level to 4 for other styles if you want.

- **Level5** Automatically displays the first text on a page that uses the mc-heading-level style property set at 5. By default, the h5 style has the mc-heading-level set to 5 (that's why it's called a fifth-level heading), but you can change it, and you can set the mc-heading-level to 5 for other styles if you want.

- **Level6** Automatically displays the first text on a page that uses the mc-heading-level style property set at 6. By default, the h6 style has the mc-heading-level set to 6 (that's why it's called a sixth-level heading), but you can change it, and you can set the mc-heading-level to 6 for other styles if you want.

> 🗒 **NOTE** The glossary and index variable options are available only in Adobe PDF output.

Variable Sets		Names		Definitions	Comment	
Heading	▲	AnyLevel	▲		First heading (h1-...	▲
MyVariables		FirstGlossaryPageHeading			First glossary hea...	
NewVariableSet		FirstGlossaryPageTerm			First glossary term...	
System		FirstIndexHeading			First index headin...	
		FirstIndexTerm			First index term o...	
		LastGlossaryPageHeading			Last glossary hea...	
		LastGlossaryPageTerm			Last glossary ter...	
		LastIndexHeading			Last index headin...	
		LastIndexTerm			Last index term o...	
		Level1			First h1 on page	
		Level2			First h2 on page	
		Level3			First h3 on page	
		Level4			First h4 on page	

Here is the Heading variable set.

Here are the system variables that you can select from the Heading variable set.

6. Click **OK**. The variable is added to the template page.

7. Click to save your work.

Running Head Variables

Supported In:

A Running Head (or Running HF) variable is a special variable that you can insert into a header or footer in a page layout or a print template page. It lets you display certain text in the header or footer automatically, based on the style associated with the variable. For example, Running Head variables are useful if you want to include the title of each chapter in a document in a header or footer without having to type them into multiple pages.

Heading variables are very similar to Running Head variables. The key appeal of Running Head variables is that they are supported in FrameMaker and help with the transition of authors from FrameMaker to Flare. So people who are familiar with FrameMaker might prefer to use Running Head variables because they are used to them. However, Heading variables are somewhat easier to use and therefore are recommended for people who do not have a FrameMaker background.

Default Running Head Variables

Flare provides the following default Running Head variables.

Variable Name	Variable Definition
Running H/F 1	<$paratext[h1]>
Running H/F 2	<$paratext[h2]>
Running H/F 3	<$paratext[h3]>
Running H/F 4	<$paratext[h4]>

Variable Name	Variable Definition
Running H/F 5	<$paratext[h5]>
Running H/F 6	<$paratext[h6]>

You can add an additional six variables like this so that there are 12 all together (Adobe FrameMaker supports 12). If you add more variables than the six provided, they should follow the same naming convention (i.e., Running H/F 7, Running H/F 8, Running H/F 9, and so on).

Using Other Styles in Running Head Variables

As you can see by looking at the variable definitions in the list, these variables are based on the h1 through h6 heading styles provided in Flare. You should not change the variable name or the first part of the definition. However, you can change the style within the definition if you want to use something other than one of the h1 through h6 heading styles provided (e.g., you can change <$paratext[h1]> to <$paratext[someOtherStyle]>). The style name used in the variable definition must match the style name that results from generating the output. If you create a style class, this means using an underscore in the variable definition.

> ☆ **EXAMPLE** If you have an <h1> tag with a class called "title" in Flare, the style name generated for the corresponding paragraph in FrameMaker will be "h1_title" (not h1.title, as it is in Flare). Therefore, you would change the variable definition to <$paratext[h1_title]>.

Which Style Text is Displayed?

When you insert a Running Head variable into a page layout frame or template page, Flare will insert the text from the first occurrence of the specified style on that page. If it doesn't find any text using that style, it will use the text from the previous occurrence of that style.

> ☆ **EXAMPLE** If you want the header in chapters to display content that is using the h1 style (perhaps you are using this style to display the chapter title for your output), select the **Running H/F 1** variable. The header will then display the first <h1> content that it finds in the document (e.g., "My First Chapter Title"). When the next occurrence of the h1 style appears on another page, the heading changes to display that text (e.g., "My Second Chapter Title"), and so on.

Steps for Using Running Head Variables

If you want to incorporate Running Head variables into your output, use the following steps.

1. **Add Variable Set** When you do this, make sure to select the Running HF template. See "Adding Variable Sets to Projects" on page 17.

2. **Insert Running Head Variables** You can insert these variables into headers or footers within page layout frames or print template pages. See "Inserting Running Head Variables Into Frames" on the next page or "Inserting Running Head Variables Into Template Pages" on page 90.

> 🗒 **NOTE** An alternative to Running Head variables, and perhaps an easier one, is to use Heading variables. See "Heading Variables" on page 72.

Inserting Running Head Variables Into Frames

If you want certain content to be inserted into a frame automatically (rather than typing the text manually), you can use a Running Head variable. For example, you might want to automatically include the first h1 heading of each chapter into the header for certain pages. By using a Running Head variable, you can use just one page layout for all of your chapters, rather than having to create a new page layout for each chapter.

How to Insert a Running Head Variable Into a Page Layout Frame

1. Add a variable set to the project, selecting the Running HF template.

2. Open a page in a page layout.

3. Click on the frame to which you want to add text or other content.

> **NOTE** You can add text and content only to header, footer, and decoration frames. You cannot add text and content to a body frame (which automatically displays content from your topics).

4. Do one of the following, depending on the part of the user interface you are using:

   - **Local Toolbar** Click .

   - **Keyboard Shortcut** Press **F2** on your keyboard.

   - **Right-Click** Right-click the frame and select **Edit Text**.

   A message opens, asking if you want to pick content from a template (e.g., a page number).

5. Click **No**.

> **NOTE** If you have already prepared content in the form of a snippet and added it to your template folder, you can click "Yes" instead in order to select the snippet. For more information about templates, see the online Help.

5. Click in the Frame Contents window pane.

6. Do one of the following, depending on the part of the user interface you are using:

    - **Ribbon** Select **Insert > Variable**.

    - **Local Toolbar** Click .

    - **Keyboard Shortcut** Press **CTRL+SHIFT+V**.

    The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

7. On the left side of the dialog, select the Running Head variable set that you created.

8. On the right side of the dialog, select the variable that you want to insert. Unless you edit the variable definitions to change the style classes within them, each variable is associated with one of the six heading styles (h1 through h6).

    When you insert a Running Head variable into a page layout frame, Flare will insert the text from the first occurrence of the specified style on that page. If it doesn't find any text using that style, it will use the text from the previous occurrence of that style.

    > ☆ **EXAMPLE** If you want the header in chapters to display content that is using the h1 style (perhaps you are using this style to display the chapter title for your output), select the **Running H/F 1** variable. The header will then display the first h1 content that it finds in the document (e.g., "My First Chapter Title"). When the next occurrence of the h1 style appears on another page, the heading changes to display that text (e.g., "My Second Chapter Title"), and so on.

9. Click **OK**. The variable is added to the Frame Contents window pane.

10. Click to save your work. The variable can now be seen in the frame.

Inserting Running Head Variables Into Template Pages

In addition to any type of regular content (e.g., text, images) and page numbers, you can insert Heading variables or Running Head variables into template pages.

How to Insert a Running Head Variable Into a Template Page

1. Add a variable set to the project, selecting the Running HF template.

2. From the Content Explorer, open the template page.

3. Place your cursor where you want to insert a Running head variable (i.e., in a page header or Page Footer proxy).

4. Do one of the following, depending on the part of the user interface you are using:

   - **Ribbon** Select **Insert > Variable**.

   - **Local Toolbar** Click .

   - **Keyboard Shortcut** On your keyboard press **CTRL+SHIFT+V**.

   The Variables dialog opens, with the variable set(s) on the left and the variables associated with the selected set on the right.

5. On the left side of the dialog, select the Running Head variable set that you created.

6. On the right side of the dialog, select the variable you want to insert. Unless you edit the variable definitions to change the style classes within them, each variable is associated with one of the six heading styles (h1-h6). When you insert a Running Head variable into a template page, Flare inserts the text from the first occurrence of the specified style on that page. If it doesn't find text using that style, it uses the text from the previous occurrence of that style.

☆ **EXAMPLE** If you want the header in chapters to display content that is using the h1 style (perhaps you are using this style to display the chapter title for your output), select the **Running H/F 1** variable. The header will then display the first h1 content that it finds in the document (e.g., "My First Chapter Title"). When the next occurrence of the h1 style appears on another page, the heading changes to display that text (e.g., "My Second Chapter Title"), and so on.

7. Click **OK**. The Running Head variable is added to the template page.

8. Click 🖫 to save your work.

▍Variable Sets

You can perform various additional tasks with date/time variables.

Importing Variable Sets

Not only can you add a new variable set, but you can also import an existing variable set (FLVAR file).

How to Import a Variable Set

1. Do one of the following, depending on the part of the user interface you are using:

   - **Ribbon** Select **Project > New > Variable Set**.

   - **Right-Click** In the Project Organizer, right-click on the **Variables** folder and from the context menu select **Add Variable Set**.

   The Add File dialog opens.

2. Select **New from existing** and click ⬚.

3. Find and select the FLVAR file that you want to import.

4. Click **Open**. The Source File field now contains the path to the file that you are importing. Also, the name of the file is displayed in the File Name field.

5. (Optional) If you want to place the file into a subfolder previously created in the Content Explorer or Project Organizer, in the **Folder** field click ⬚ and select the subfolder. Otherwise, keep the default location.

6. If you want to give the variable set a different name than that for the imported file, click in the **File name** field and replace the text.

7. (Optional) If you want to apply condition tags to the file, expand the **Attributes** section at the bottom of the dialog. Next to the **Condition Tags** field, click ⬚ and select the conditions you want to apply. Click **OK**.

8. (Optional) If you want to apply file tags, expand the **Attributes** section at the bottom of the dialog. Next to the **File Tags** field, click ▦ and select the file tags you want to apply. Click **OK**.

9. Click **Add**. The variable set is added and opens in the Variable Set Editor.

> 🗒 **NOTE** As an alternative to importing single files from another location, you can import linked files from a Flare source project using a featured called "Global Project Linking." This is different than a simple import process, because in this case, the imported files remain linked to the source project. This lets you make future updates to those files in just one place—in the parent project. When you perform ongoing imports using your previous settings, Flare recognizes changes to the source files. Therefore, the new files can be brought over, replacing the outdated ones. For more information see the online Help.

Opening a Variable Set

The following steps show you how to open a variable set that you have added to your project.

How to Open a Variable Set From the Quick Launch Bar

The Quick Launch bar lets you search for any Flare file or command. It is located in the upper-right corner of the interface. You can press **CTRL+Q** on your keyboard to move focus to the Quick Launch bar so you can begin typing.
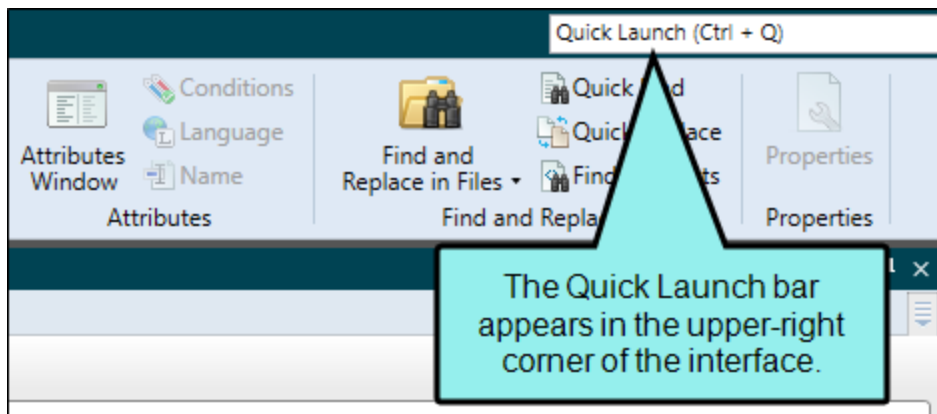
1. In the Quick Launch bar, type a few letters of the name of the file you want to open. Any available results appear in a drop-down list.

2. From the list, click the name of the file.

How to Open a Variable Set From the Project Organizer

1. Open the Project Organizer.

2. Double-click the **Variables** folder. The variable sets in your project are displayed.

3. Double-click the variable set that you want to open. The Variable Set Editor opens to the right.

Renaming Variable Sets

You can rename a variable set in your project.

How to Rename a Variable Set

1. Open the Project Organizer.

2. Double-click the **Variables** folder. The variable sets in your project are displayed.

3. Click on the variable set that you want to rename.

4. Press **F2** on your keyboard. The variable set name is highlighted.

5. Type a new name for the variable set and press **ENTER**.

6. If your file contains links to other files, you will be asked if you want to retain those links. If so, click **Update Links**.

> NOTE For more information about renaming variables within a variable set, see "Editing Variables" on page 28.

▎Styled Variables

Supported In:

A styled variable lets you change the way a variable looks—making it easier for you to identify your variables in the output as well as in the XML Editor. This is mainly achieved by editing the stylesheet property settings for the MadCap|variable element. When a styled variable is applied, it can be seen in the XML Editor and in any generated output type.

How to Style Variables

1. From the Content Explorer, open the stylesheet that you want to modify.

2. In the local toolbar, make sure the first button displays [View: Advanced]. If the button displays [View: Simplified] instead, then click it.

3. On the left side of the editor, make sure the list shows inherited styles (i.e., light gray items). If not, from the local toolbar click the **Hide Inherited** button.

4. Select the root **MadCap|variable** style or a class you have created for it. Editing the root style affects all variables. Editing a class affects only the variables where you have applied that class.
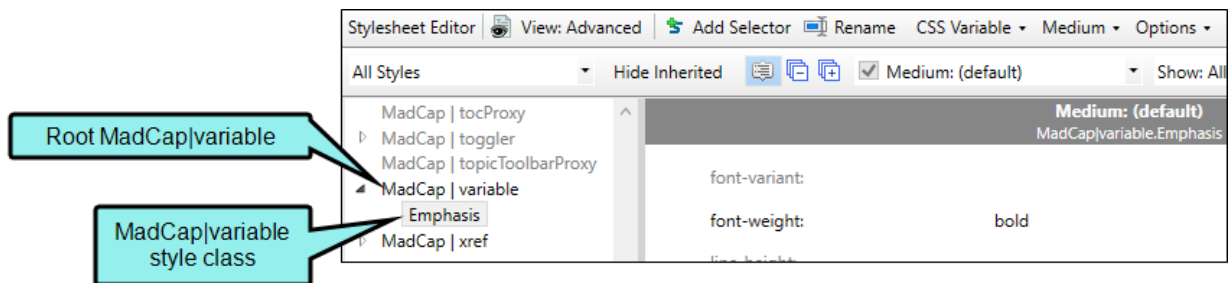
> NOTE You can also add a generic class to a variable. This can be useful to achieve the same look for a variable that a non-variable might have. Editing a generic class affects one or as many instances of a variable that you apply the class to. If your project does not include a generic class, you can create one.

5. If you created a custom or generic class, apply it to the desired variables.

    a. Open the content file containing the variable.

    b. Click the variable.

    c. From the **Home** ribbon, select the **Style** drop-down, and choose the defined style class.

6. Click [save] to save your work.

How to Style a Specific Variable

It is recommended to use this method if you want the style of a certain variable to look the same for every instance of it in the project.

1. Open the Project Organizer, and expand the **Variables** folder.

2. Double-click the variable set.

3. In the Variable Set Editor, select the row of the variable that you intend to edit, and in the local toolbar, click ⬚.

   The stylesheet opens and a new class is added under the MadCap|variable style in the stylesheet. These are auto-generated style classes that will automatically apply to the matching variable, without having to manually apply the style class when used in the content.
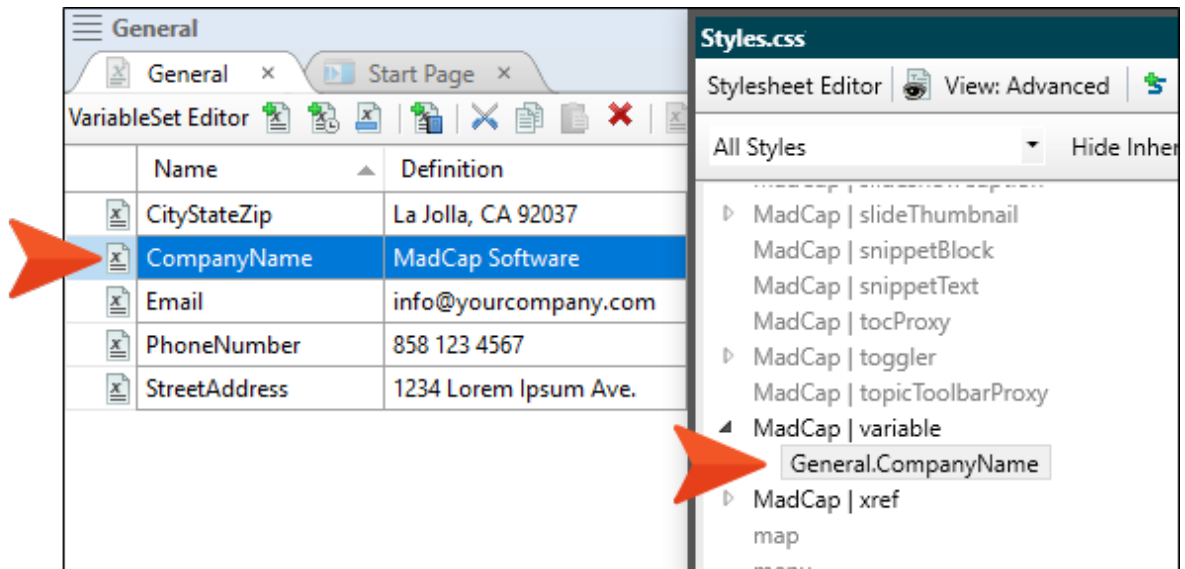
> 📝 **NOTE** Alternatively, you can right-click the variable row, and from the context menu select **Edit Variable Style**.
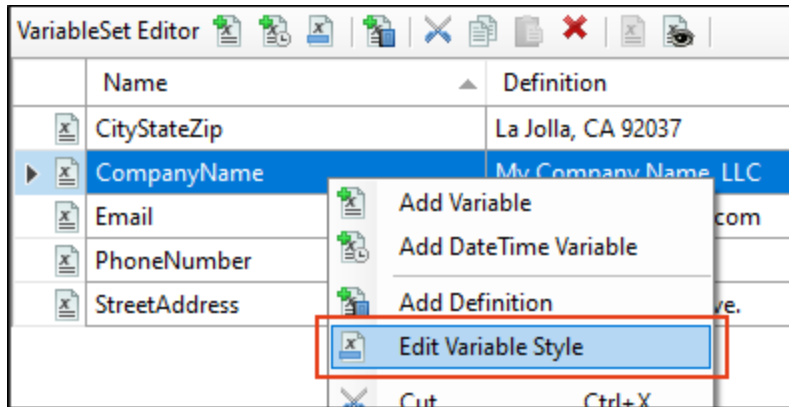

> 📝 **NOTE** You can edit a style for one variable at a time. If you attempt to multi-select variables to edit, the Edit Variable Style option is disabled.

4. Edit variable style properties for the preferred look.
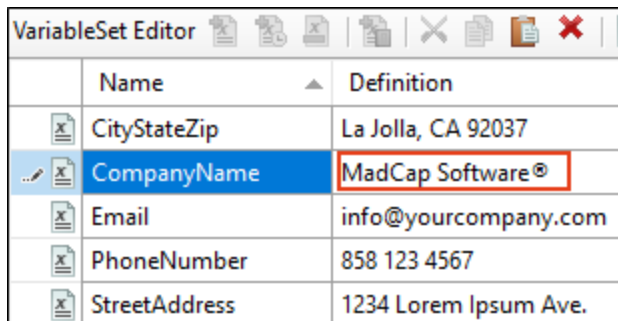
5. Click 💾 to save your work.

What's Noteworthy?

TIP Styled variables do not allow for partial- or character-level styling within a variable. If you need to style more than the variable as a whole, it is recommended to use snippets instead.

For example, you can use snippets for character-level styling (e.g., show a company name with italic, bold, and superscript text). The image below is not a variable, it is a snippet.
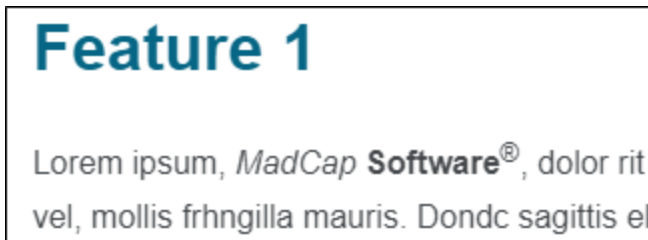
Or, you can copy a symbol from a content file into the Variable Set Editor's plain text grid. The exact results of how a symbol might look in the output depends on the font type used.

NOTE When you add styles to variables, keep in mind the standard cascading stylesheet (CSS) rules for precedence apply.

NOTE The styled variables feature is applicable and possible to apply to all variable types, not just custom basic variables. See "Types of Variables" on page 8.

APPENDIX

PDFs

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

Tutorials

Getting Started Tutorial

Autonumbers Tutorial

Back-to-Top Button Tutorial

Context-Sensitive Help Tutorial

Custom Toolbar Tutorial

eLearning Tutorial—Basic

eLearning Tutorial—Advanced

Image Tooltips Tutorial

Lists Tutorial

Meta Tags Tutorial

Micro Content Tutorial—Basic

Micro Content Tutorial—Advanced

Responsive Output Tutorial

Single-Sourcing Tutorial

Snippet Conditions Tutorial

Styles Tutorials

Tables Tutorial

Word Import Tutorial

Cheat Sheets

Context-Sensitive Help Cheat Sheet

Folders and Files Cheat Sheet

Learning & Development Cheat Sheet

Lists Cheat Sheet

Micro Content Cheat Sheet

Print-Based Output Cheat Sheet

Search Cheat Sheet

Shortcuts Cheat Sheet

Structure Bars Cheat Sheet

Styles Cheat Sheet

User Guides

Accessibility Guide

Analysis and Reports Guide

Architecture Guide

Autonumbers Guide

Branding Guide

Condition Tags Guide

Context-Sensitive Help Guide

Eclipse Help Guide

eLearning Guide

Getting Started Guide

Global Project Linking Guide

HTML5 Guide

Images Guide

Import Guide

Indexing Guide

Key Features Guide

Lists Guide

MadCap Central Integration Guide

Meta Tags Guide

Micro Content Guide

Navigation Links Guide

Plug-In API Guide

Print-Based Output Guide

Project Creation Guide

QR Codes Guide

Reviews & Contributions With Contributor Guide

Scripting Guide

Search Guide

SharePoint Guide

Skins Guide

Snippets Guide

Source Control Guide: Git

Source Control Guide: Perforce Helix Core

Source Control Guide: Subversion

Source Control Guide: Team Foundation Server

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Template Pages Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Translation and Localization Guide

Variables Guide

Videos Guide

What's New Guide