

Cascade-Specific Velocity

Tools and API's Unique to Cascade CMS

Built-in Variables	<pre> \$currentPageSiteName - returns a string \$currentPagePath - returns a string \$currentPage - returns an object </pre>
<u>Locator Tool</u> (8 methods)	
<p>Invoked with <code>\$_locate()</code> or <code>\$_locate</code> followed by the asset type in camelcase. Asset type does not need to be bolded.</p>	<pre> \$_locate() \$_locateBlock() \$_locateLinkable() \$_locateFile() \$_locatePage() \$_locateFolder() \$_locateSymlink() \$_locateFormat() \$_locateReference() </pre>
Parameters Using the explicit asset method (e.g., <code>\$_locatePage</code>), the Path to the asset is required along with a Sitename. When the asset is located in the site from which it is called, the Sitename is optional. For the non-explicit method, the asset type is also required.	<pre> \$_locate[Asset Type](PATH, SITENAME) \$_locate[Asset Type](PATH) </pre> <p><u>Example</u> <code>\$_locatePage("path/to/page", "sitename")</code></p> <p><u>Non-explicit method</u> <code>\$_locate(PATH, TYPE, SITENAME)</code></p>
<u>Property Tool</u> (10 methods)	
<p>The <code>.outputProperties()</code> method displays the properties and methods of an object.</p> <p>The result is a list with properties (methods) on the left and the output of each method listed to the right. For example, the method "link" returns a string.</p>	<p><u>Example</u> <code>\$_PropertyTool.outputProperties(\$currentPage)</code></p> <p><u>Sample Output Snippet</u> Properties: - link: String - metadata: Metadata - parentFolder: Folder</p>
<p><code>.isEmpty()</code> returns true if the object is null, is an empty string, or contains only whitespace characters, otherwise false.</p>	<pre> \$_PropertyTool.isEmpty(\$currentPage.link) </pre>
<p><code>.isNull()</code> returns true if the object is null, otherwise false.</p>	<pre> #set(\$page = \$_locatePage("index")) \$_PropertyTool.isNull(\$page) </pre>

Cascade-Specific Velocity

Tools and API's Unique to Cascade CMS

Query API (27 methods)	
<p>To initiate a query, create a query object with the <i>query()</i> constructor and assign it to a variable.</p> <p><u>After</u> all filtering and search options (below) have been specified, you must execute the query to capture the results.</p>	<p><u>Creating a Query Object</u></p> <pre>#set(\$queryObject = \$_.query())</pre> <p><u>Executing a Query*</u></p> <pre>#set(\$results = \$queryObject.execute())</pre>
<p>Queries are done either by metadata set or content type. When the Site is not specified, the query looks only within the current site.</p> <p>Once a site-specific asset type is specified, one can search for assets based on that type within other sites by using the <i>.siteName()</i> method.</p> <p>To search across all sites, use the <i>.searchAcrossAllSites()</i> function.</p>	<pre>\$queryObject.byMetadataSet("Default")</pre> <pre>\$queryObject.byContentType("Event")</pre> <pre>\$queryObject.byContentType("site://Calendar/Event")</pre> <pre>\$queryObject.siteName("Art Museum")</pre> <pre>\$queryObject.byContentType("site://Calendar/Event".siteName("Art Museum"))</pre> <p><i>Looks for Events in the Art Museum site (only) that are generated with the Event ContentType located in the Calendar site.</i></p> <pre>\$queryObject.searchAcrossAllSites()</pre>
<p>By default, queries include non-publishable assets but these can be excluded using "false". Unindexed assets are included by "false".</p> <p>For MetadataSet queries, all assets based on a particular Set are included by default. You can explicitly exclude assets by using the <i>.include</i> method and setting the boolean to <i>false</i>.</p> <p>Limit the number of returned assets with the <i>.maxResults()</i> method. When not specified, the default value is 100. There is a hard limit of 2000 maximum.</p>	<pre>\$queryObject.publishableOnly(BOOLEAN)</pre> <pre>\$queryObject.indexableOnly(BOOLEAN)</pre> <pre>\$queryObject.includeFolders(BOOLEAN)</pre> <pre>\$queryObject.maxResults(25)</pre>
<p>The <i>.sortBy()</i> method uses a metadata field (e.g., title, startDate, last modified date) or name or path.</p> <p>The <i>.sortDirection()</i> can also be set.</p>	<pre>\$queryObject.sortBy("displayName")</pre> <pre>\$queryObject.sortDirection("asc") or</pre> <pre>\$queryObject.sortDirection("desc")</pre>

* Using the Java dot notation, filter and sort methods can be chained for cleaner, more concise, coding.

Cascade-Specific Velocity

Tools and API's Unique to Cascade CMS

Cascade PageAPI Syntax Examples

1) Information outside (and usually prior to) the Data Definition XML.

Metadata Set Options

Title - Title generally is used to display in the web browser tab at the top of a page and in the URL.

Display Name - Used to provide a longer label on the page proper.

```
#set($title = $currentPage.metadata.title)
```

```
#set($displayName = $currentPage.metadata.displayName)
```

```
#set($summary = $currentPage.metadata.summary)
```

```
#set($teaser = $currentPage.metadata.teaser)
```

```
#set($keywords = $currentPage.metadata.keywords)
```

```
#set($description = $currentPage.metadata.description)
```

```
#set($author = $currentPage.metadata.author)
```

```
#set($startDate = $currentPage.metadata.startDate)
```

```
#set($endDate = $currentPage.metadata.endDate)
```

the expiration folder is not viewable.

```
#set($reviewDate = $currentPage.metadata.reviewDate)
```

Dynamic/Custom Metadata Fields

```
#set($customName = $currentpage.metadata.getDynamicField("FieldName").value or .values)
```

Syntax for other commonly used properties of the Page asset type

```
#set($label = $currentPage.label) ## .label will return a Display Name if it exists; a Title if it exists and the DN is empty. If neither DN, nor Title, exist, it returns the page name.
```

```
#set($systemName = $currentPage.name)
```

```
#set($url = $currentPage.link)
```

```
#set($path = $currentPage.path) ## recall $currentPagePath is a builtin variable.
```

```
#set($assetType = $currentPage.assetType)
```

```
#set($parentFolderName = $currentPage.parentFolder)
```

Tags

```
#set($tags = $currentPage.tags) .... returns an array.
```

Cascade-Specific Velocity

Tools and API's Unique to Cascade CMS

2) Structured Data (i.e., content from a Data Definition)

Brief sample XML file:

```
<system-data-structure>
  <mainContent>
</mainContent>
  <otherContentNode>
</otherContentNode>
</system-data-structure>
```

Example of selecting a data definition group with an identifier of "mainContent".

```
#set($mainNode = $currentPage.getStructuredDataNode("mainContent"))
```

After the first level nodes, switch to .getChild or .getChildren syntax:

Text Field: #set(\$txt = \$_EscapeTool.xml(\$mainNode.getChild("txt").textValue))

Text Area: #set(\$txtArea = \$_EscapeTool.xml(\$mainNode.getChild("txtArea").textValue))

WYSIWYG: #set(\$wysiwyg = \$mainNode.getChild("wysiwyg").textValue)

Radio Button: #set(\$radioBtn = \$_EscapeTool.xml(\$mainNode.getChild("radioBtn").textValue))

Drop Down: #set(\$dropDown = \$_EscapeTool.xml(\$mainNode.getChild("dropDown").textValue))

Single Checkbox - check to see if contains the one checked value

```
#set($singleCheckBox = $mainNode.getChild("singleCheckBx").textValues) ##plural
#if($singleCheckBox.contains("yes"))
  The box was selected (i.e., checked)
#else
  The box was not selected.
#end
```

Multiple Checkbox - output the checked values

```
#set($multiCheckBx = $mainNode.getChild("multiCheckBx").textValues)
```

- List all values selected: \$_DisplayTool.list(\$multiCheckBx, ", ")
- Loop through the values: #foreach (\$checked in \$multiCheckBx) \$checked #end

Multi-Select - output the selected values

```
#set($multiSelect = $mainNode.getChild("multiSelect").textValues)
```

- List all values selected: \$_DisplayTool.list(\$multiSelect, ", ")

- Loop through the values: #foreach (\$selected in \$multiSelect) \$selected #end

Cascade-Specific Velocity

Tools and API's Unique to Cascade CMS

Get Current Date

```
#set ($currentDate = $_DateTool.getDate())  
Modified $_DateTool.difference($lastModified, $currentDate).hours hours ago
```

Calendar - output a string: (10-12-2019)

```
#set($calendar = $mainNode.getChild("calendar").textValue)  
## convert to date data type and format  
#set ($date = $_DateTool.toDate("MM-dd-yyyy", $calendar))
```

Date Time - outputs data type string timestamp

```
#set($dateTimeField = $mainNode.getChild("dateTime").textValue)  
## convert to date data type and format as needed  
#set($dateTime = $_DateTool.getDate($mainNode.getChild("dateTime").textValue))  
#set($time = $_DateTool.format("h:mm a", $_DateTool.getDate($mainNode .getChild  
("dateTime").textValue)))  
#set($date = $_DateTool.format("full_date", $dateTime))  
## for comparing dates  
#set($currentDate = $_DateTool.getDate())  
#set($currentDateNumber = $_NumberTool.toNumber($currentDate))  
#set($dateTimeNum = $_NumberTool.toNumber($dateTimeField))
```

Page Chooser (same for block/file) Note requirement for .asset

```
#set($pageSelector = $mainNode.getChild("pageChooser").asset)  
#set($title = $pageSelector.metadata.title)  
#set($data = $pageSelector.getStructuredDataNode("mainContent node for this page"))
```

File chooser

```
#set($fileSelector = $mainNode.getChild("fileChooser").asset)
```

Block chooser

```
#set($profileBlock = $mainNode.getChild("blockChooser").asset)  
#set($image = $profileBlock.getStructuredDataNode("image").asset.link)  
#set($origin = $profileBlock.metadata.getDynamicField("origin").value)
```