www.protecmedia.com

IterWeb Web Development

Velocity Macros & Catalog Macros

May 2017

Editorial Business Unit

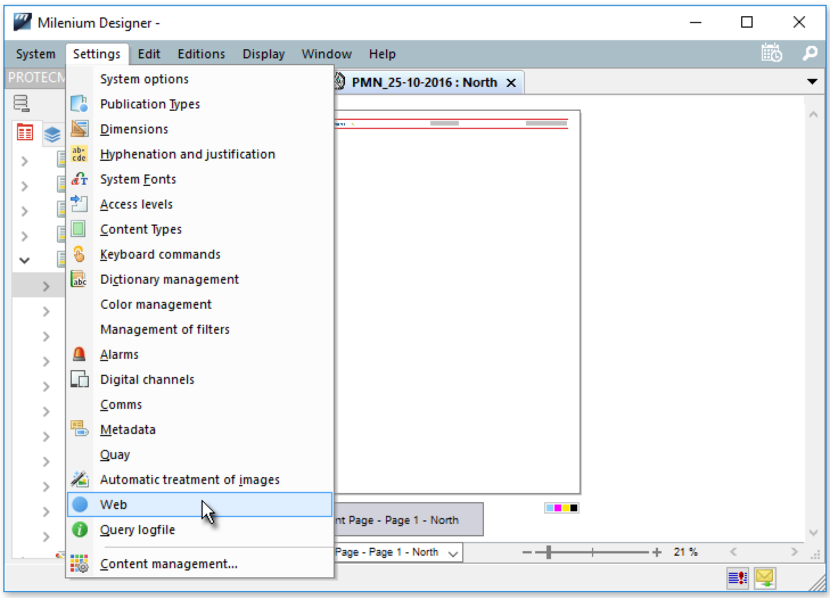
2 VELOCITY MACROS AND ITERWEB GENERIC FUNCTIONS (CATALOG)

In this part of the course, we will talk about the creation of macros, their use and some useful macros that are already included in any catalog theme.

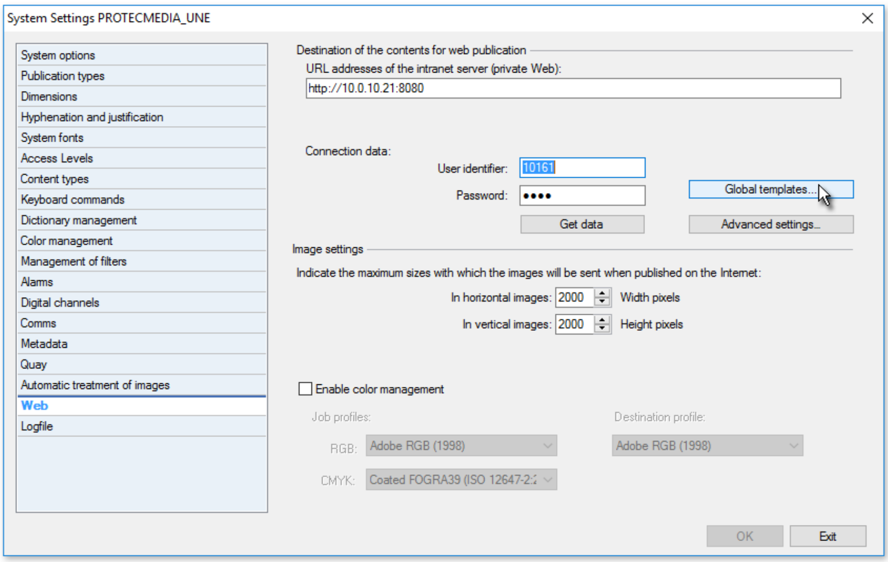
Macros are reusable Velocity code portions that can be used in the Teaser Viewer, Content Viewer, and Velocity Container portlet templates. At the time of creation of this catalog, the macros are Global for all sites on the same Milenium server, regardless of whether they are separated into organizations.

2.1 CREATION OF GLOBAL TEMPLATE

In order to create a Global Templates (the files in which the macros are stored) in Milenium click on the **Settings** option of top menu and then select **Web**.



In the displayed dialog box click on **Global templates...**



Destination of the contents for web publication

URL addresses of the intranet server (private Web): http://10.0.10.21:8080

Connection data:

User identifier: 10161

Password: ••••

Image settings

Indicate the maximum sizes with which the images will be sent when published on the Internet:

In horizontal images: 2000 Width pixels

In vertical images: 2000 Height pixels

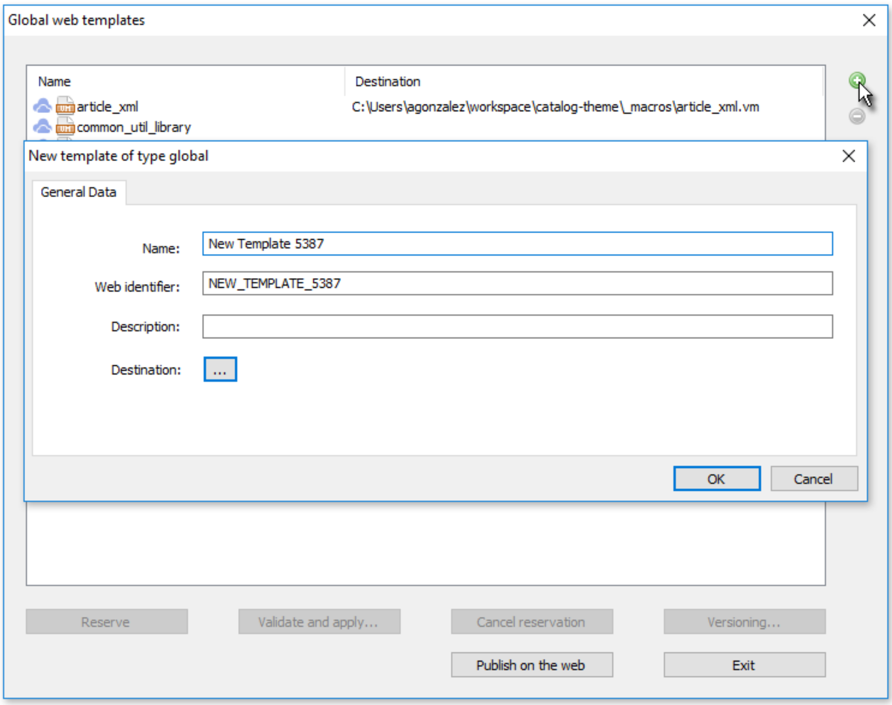
□ Enable color management

Job profiles: RGB: Adobe RGB (1998) CMYK: Coated FOGRA39 (ISO 12647-2:

Destination profile: Adobe RGB (1998)

A new dialog box will appear with the list of the global templates.

We can add the global template by clicking on the **(+)** icon, which is located on the right side of the dialog box. The name and the identifier of the new global template will be requested.



Global web templates

Name	Destination
article_xml	C:\Users\agonzalez\workspace\catalog-theme_macros\article_xml.vm
common_util_library	

New template of type global

General Data

Name: New Template 5387

Web identifier: NEW_TEMPLATE_5387

Description:

Destination:

When the **OK** button is clicked, an empty dialog box is displayed where we can create new macro. Once the macro is created, select our Global Templates file and click on **Validate and apply** button (You can see the modification in PREVIEW environment without Applying changes in the theme).

2.2 GLOBAL TEMPLATES MODIFICATION IN MILENIUM

The modification process is very similar to the creation process. To do this follow the previous steps until you reach the list of global templates. In the list, select the file that you want to modify and click on the **Reserve** button. Then click on the **blue eye** icon on the right side of this dialog box.

Once the modification is finished, select our Global Templates file and click on **Validate and Apply** button (You can see the modification in PREVIEW environment without Applying changes in the theme).

IMPORTANT NOTE:

We recommend **NOT** modifying any global templates of the catalog themes (article_xml, commom_util_library, constants, order_text, project) because when updating a theme by an error these file usually are overwrite by those of the catalog.

The appropriate way is to create a new global template for our project with a specific name for the customer.

2.3 CREATING A MACRO

A macro is just a piece of code that is repeated throughout a project. When used within the templates it is possible to access all types of content and all the functions that we could have in a normal template. (For example, recover the instrumental article from a section, displaying a list of images …).

Macros in iterWeb use velocity code (For more info: http://velocity.apache.org/engine/devel/translations/user-guide_es.html).

Here is an example of a basic macro:

```
## Macro recovering title for TEASERS

#macro (printHello)

  Hello!!! My Velocity Code goes here.

#end
```

Now we analyze it in parts:

1- **#macro (printHello)**

Starting and setting the name of the Macro. To define a macro, always use **#macro**, the first parameter, between parentheses is the name that the macro will have at the time of calling it (in this case **printHello**). It is very important that the macro name **does NOT** carry the $ symbol.

2**-** **Hello!!! My Velocity Code goes here.**

This is the code of the macro, in this case all you do is print the Hello string!!!

3- **#end**

End of macro.

2.4 USEFUL MACROS OF THE CATALOG THEMES

2.4.1 MACROS OF ARTICLE_XML FILE

- **#printAssignedSectionID()**
  This macro is used to recover the id of the current section of the article that we are viewing. It is very useful for retrieving variables from section or for retrieving instrumental articles from section.

- **#printAssignedSection**
  This macro returns the name of the section to which an article is assigned.

- **#printURLAssignedSection**
  This macro returns the URL of the section to which an article is assigned.

- **#printAcUrl**
  This macro is used in the teaser templates to recover the detail URL of an article.

  An example of use would be as follows:

  ```
  <a href="#printAcUrl"><h2 class="headline font-1 small bold fx-
  hover">#getTeaserTitle</h2></a>
  ```

- **#getAcAssignedCatExtended($hmResult $VocabularyName)**
  This macro is used to recover a hashmap that contains the metadata that is assigned to an article in a particular vocabulary.

  It is very useful to recover instrumental articles associated with a metadata or display a list with authors of an article.

  An example of use would be as follows:

  ```
  #set($hmAutores = {})
  #getAcAssignedCatExtended($hmAutores "Author")
  #if($hmAutores.size() > 0)
          #set($hmAutor = $hmAutores.get(0))
          $hmAutor.name
  #end
  ```

- **#getAcAssignedCatMain ($hmResult $VocabularyName)**
  This macro is used to recover a hashmap that contains the metadata that is assigned to an article and furthermore **defined as the main** of a particular vocabulary.

  It is very useful to recover instrumental articles associated with a metadata or display a headline containing the name of this metadata.

  An example of use would be as follows:

  ```
  #set($hmTopics = {})
  #getAcAssignedCatMain($hmTopics "Topic")
  #if($hmTopics.size() > 0)
          #set($hmTopic = $hmTopics.get(0))
          <a href="#metadataUrl()$hmTopic.friendlyname">$hmTopic.name</a>
  #end
  ```

- **#printDateFrom ($dateFormat $checkToday $hourFormat)**
  This macro recovers what is known as "Article Date" (for more info query a Milenium consultant). In order to do this receives as parameter the format of the date, the parameter checktoday that checks if the article is of today and if so it shows the time instead of the date and the time format. When creating a template for the catalog it is necessary that the date and time format (if used) are constants defined in the global template constants (See section **modifiable constants of the catalog themes**).

2.4.2 MACROS OF THE PROJECT FILE

- **#getTeaserTitle**

  This macro recovers the headline of an article for a teaser and checks to see if has TeaserHeadline, Headline_Web and Headline in that order of preference.

- **#getFullContentTitleGeneral**

  This macro recovers the headline of a content viewer and checks to see if has Headline_Web and Headline in that order of preference.

- **#hasMultimedia($templateType)**

  This macro checks the number of Multimedia elements an article has. This macro is useful to change the html when an article has no images (for example to display more text in a teaser when it does not have images or to make that the text occupies the 100% of the width). It receives as parameter a string of text that can be "teaser" or "full". If it receives the teaser, it will check TeaserImage and Image. If it receives full, it will only check Image.

  The following contents types are checked:

  TeaserImage, Image, Multimedia, Jwplayer, Vimeo, Youtube.

  Returns the **NUMBER OF MULTIMEDIA ELEMENTS**.

  An example of use would be the following:

  ```
  #set ($hasImages = "#hasMultimedia('teaser')")
  #if($hasImages <= 0)
          #set($templateClass = "no-media")
  #end
  ```

- **#getTeaserText($maxSize)**

  This macro returns the text of a cropped article to the number marked by the maxSize variable. It does not split the words in half and by default adds ellipses to the text.

  An example of use would be the following:

  ```
  #getTeaserText(380)
  ```

- **#getMultimediaStandard($minImagesGallery $templateType $encuadre $widthMultimedia $heightMultimedia $cropHD $cropSizes)**

  This macro displays a multimedia or gallery element. It receive as parameters the minimum number of images to display the gallery, the type of template that can be "teaser" or "full", the framing or cropping with which you want to display the images and the width and height of Multimedia that must match the dimensions of the frame (only affects multimedia elements in non-responsive themes). "cropHD" is the high definition cropping, "cropSizes" sizes parameter of the cropping, is given by the width of the template in which the image will be displayed.

This template **DOES NOT MIX** different types of content and the priority when displaying content types is defined by the macro itself.

An example of use would be the following:

#getMultimediaStandard(2,"full","crop_768x432", "1000" , "562", "66%")

- **#getMultimediaVideoGallery($widthMultimedia $heightMultimedia)**
  This macro displays a gallery with all the multimedia elements of the article. It receives as parameters the width and height of the multimedia (only affects the multimedia elements in NOT responsive themes) and recovers the types of content Multimedia, Jwplayer, Vimeo and Youtube.

  An example of use would be the following:

  #getMultimediaVideoGallery("100" , "541")

- **#getMultimediaMixed($templateType $encuadre $widthMultimedia $heightMultimedia $cropSizes)**
  This macro displays a multimedia or gallery element. It receive as parameter the type of template that can be "teaser" or "full", the framing or cropping with which you want to display the images and the width and height of multimedia that must match the dimensions of the frame (only affects multimedia elements in NON-responsive themes). "CropSizes" sizes parameter of the croppings, is given by the width of the template in which the image will be displayed. This template **MIX** different types of content and order when displaying content types and is defined by the macro itself.

  #getMultimediaMixed("full","crop_900x506", "1000" , "562", "66%")

- **#getMultimediaIcon($encuadre $tamanoIcono $valign $halign $frameHighRes $frameName $disableLink $cropSizes)**
  This macro is used to recover an image along with the icon indicating whether it has a gallery or a multimedia. It receives as parameters the Milenium framing or cropping, the size of the icon that can be "small", "medium" or "big", the vertical alignment that can be "left", "right" or "center" and the horizontal alignment can be "top", "bottom" or "middle".

  The parameters "frameHighRes" and "frameName" are the names of the cropping that will be used for high resolution images, "disableLink" tells us whether to have link to the detail content. "cropSizes" parameter is given by the width of the template in which the image is to be displaying.

  An example of use would be the following:

  #getMultimediaIcon("crop_900x506","small","bottom","left", "66%")

2.4.3 MACROS OF THE ORDER TEXT FILE

- **#getTextFlow($cropName $widthMultimedia $heightMultimedia)**
  This macro returns multiple content types sorted in the order given in Milenium.

  It receives as parameters the name of the Milenium framing or cropping. If more than one image is found then the gallery is made up in the text flow.

  The width and the height of the multimedia elements only affects NON-responsive themes.

  By default, it displays the following types of content:

  Text, Lead, Image_Text, HTML_Text, Quote, Youtube_Text, Multimedia_Text, Vimeo_Text, Intext.

2.4.4 Product macros

- **#renderDisqusComments()**
  This macro returns the number of comments that an article has with the format defined in IterAdmin.

2.5 MODIFIABLES CONSTANTS OF CATALOG THEME

In the global template global **Constants** we find several constants that modify the behavior of all the catalog templates that use them:

```
## Customize the base URL of the metadatalist
#macro (metadataUrl)#*
 *#/metadatalist/-/meta/#*
*##end

## Customize the format of dates in templates using the short date format.
#macro (dateMaskShort)#*
 *#dd MMM yyyy#*
*##end

## Customize the format of dates in templates that use the long date format
#macro (dateMaskLong)#*
 *#dd MMM yyyy / HH:mm#*
*##end
```

```
## Macro that returns literals to use in templates, allows us to create custom literals for different themes. An example of its use is the translation of these literals into different languages; just enter in the file and replace the literals in a language without having to change the templates.

#macro (getGlobalString $stringName)#*
 *##if           ($stringName == "voteString")#*
        *#Vote#*
 *##end#*
*##end
```

This file will grow as required.

IMPORTANT NOTE:

It is necessary to remember that all the sites that are under the same Milenium server will be affected regardless of whether they belong to different organizations.

2.6 PRACTICAL EXERCISE OF CREATING AND USING MACROS

2.6.1 TIME

3 Hours

2.6.2 DESCRIPTION

EXERCISE 1:

This exercise consists of creating several macros for a teaser template defined by the student.

First, you create a macro that will not receive any parameters and will only print the string "**World**"

Then you create a macro that will receive a text string as a parameter, this macro will print this string and then will call the previous macro.

After you create a teaser template that calls the second macro and will pass the **Hello** string. The expected result is that the "Hello World" string is printed.

Then you have to modify the teaser template again so the result would be saved in a velocity variable instead of directly printing the string. Then you have to verify that the variable is not empty and print its value on screen.

EXERCISE 2:

This exercise consists of creating a teaser template and a full content template using the macros defined for the catalog themes.

The teaser template should display the following:

- Date of the article in short format
- Section to which the article belongs with link to that section
- Headline of the article in teaser format
- Article image that display a multimedia or gallery icon
- Article text in teaser format

The fullcontent template should display the following:

- Date of the article in long format
- Name of author of the article
- Headline of the article in full content format
- Multimedia gallery that mixes all types of image and multimedia content
- Article text flow

The aspect achieved with CSS will be highly valuated.