Leveraging the Power of Web Content Manager Within a Portal Theme

Sarah Hall *WebSphere Portal Lab Services (SEAL) Team | IBM*
Thomas Hurek *WebSphere Portal Lab Services (SEAL) Team | IBM*

6/25/2015

Purpose

More and more companies using Digital Experience software want to store their theme artifacts (CSS, JavaScript (JS) and images) in IBM Web Content Manager (WCM). Storing these items in WCM allows customers to manage the theme artifacts without having to change the theme structure itself. Customers often do not use WebDav, but instead opt to create a WebSphere Application EAR containing separate WAR files for their static and dynamic theme content. Whenever theme files in these WARs are changed, the theme EAR must be redeployed. If the artifacts are stored in WCM, theme redeployment can be avoided. There are also other caching and syndication benefits to be gained from this approach. The purpose of this document is to highlight lessons learned from developing several themes for enterprise customers using WCM.

Benefits of Using WCM in the Theme

There are many reasons for using WCM in the theme especially for customers who already leverage WCM. Below is a list of some of the major benefits:

- Customers develop sites collaboratively between their line-of-business areas and their IT areas. Putting theme artifacts in WCM creates a natural division of responsibilities between these areas. Line-of-business areas can change/create theme artifacts without always having to involve the IT team.

- Important WCM features such as workflow and project creation can be leveraged. Theme artifacts can be pushed through a workflow or included as part of a project (including preview). That allows a business user to preview and approve or reject a set of changes.

- If the theme is an EAR based theme changing static content requires that the theme be redeployed. If theme artifacts are in WCM, the theme (either EAR or WebDav based) does not have to be redeployed.

- It is easy to access theme artifacts which are stored in WCM and image or style changes will show up in the site immediately.

- Instead of having to deploy changes to all environments, changes to theme artifacts can be syndicated.

- Versioning in WCM can be leveraged to undo a change or see when and how a file was changed in the past.

- Caching benefits in WCM apply to theme artifacts.

Limitations of Using WCM in the Theme

There are a few limitations for using WCM in the theme. Below is a list of some of the limitations:

- Anytime WCM is being used it is important that caching be setup properly. If caching is not used or is setup incorrectly, performance will degrade significantly.

- Generally the theme stands alone as a web application deployed to WebSphere Application Server (WAS). If WCM is used in the theme, there is a dependency on the WCM libraries for the theme to render correctly. In addition, there are certain parts of the theme (such as dynamic content spots) which cannot be stored in WCM.

Ways to Access WCM Artifacts from the Theme

The theme artifacts can be stored a variety of ways in WCM. For example, they can be stored as WCM components or content. If components are used, these files have to be downloaded, changed and uploaded when the files are changed. If content is used, changes can be made directly in the WCM authoring UI. The following list summarizes the main ways WCM content or components can be included in the theme.

- Directly using the component or content reference URL. See the following link for details on how to build a URL to WCM artifacts: http://www-01.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_config_delivery_servlet.dita?lang=en

- Via AJAX using the component or content reference URL. See the following link for details on how to build a URL to WCM artifacts: http://www-01.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_config_delivery_servlet.dita?lang=en

- Using the WCM JSP tag library. See the following link to the WCM JSP Tag library documentation: http://www-

01.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_reference_wcm-jsp-tags.dita?lang=en

- Programmatically using the WCM API. See the following link to the WCM API documentation: http://www-01.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_dev_api.dita?lang=en

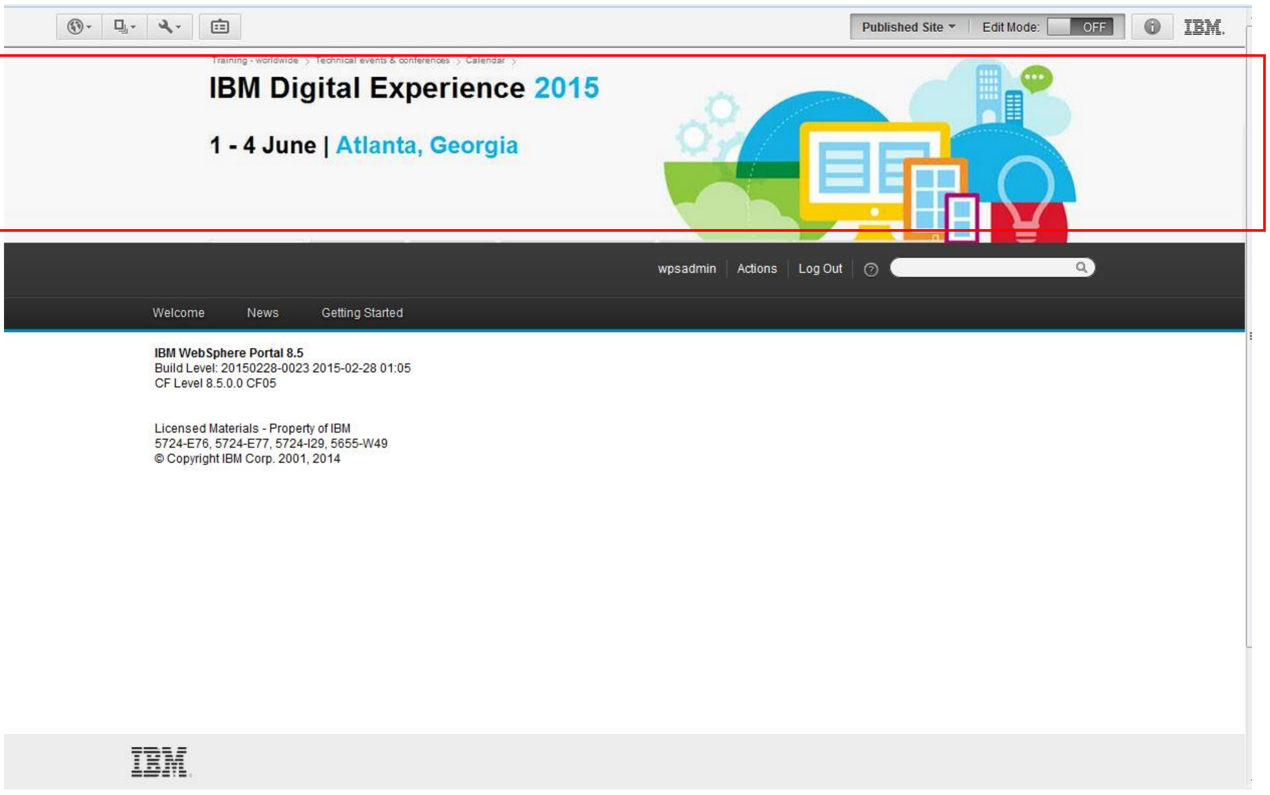
There are no limitations in terms of what markup the included WCM artifacts produce and what WCM components they use to produce the markup. As an example the included component could be a Menu that renders a set of link content items in a certain Site Area – giving the business users so a possibility to easily adjust the links in e.g. the footer.

Another use case could be to render a flyout menu in the theme with links that are produced as JSON list via a navigator component in WCM.

Below are simple step-by-step examples of how theme artifacts can be retrieved from WCM. These examples are intended to help anyone trying to incorporate WCM into their theme to get started. The examples in this document use an EAR based theme, but static content could also reside in WebDav on the Portal server. The directory structures would be the same.

Simple Image Example

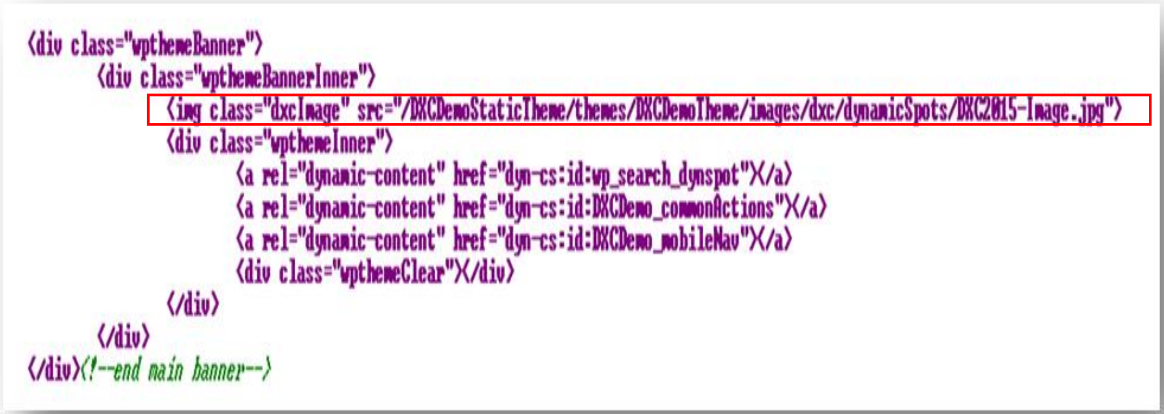
Suppose the site looks like below and a new banner image is required.


Published Site
Edit Mode: OFF
IBM.
Training - worldwide > Technical events & conferences > Calendar
IBM Digital Experience 2015
1 - 4 June | Atlanta, Georgia
wpsadmin
Actions
Log Out
Welcome
News
Getting Started
IBM

IBM WebSphere Portal 8.5
Build Level: 20150228-0023 2015-02-28 01:05
CF Level 8.5.0.0 CF05

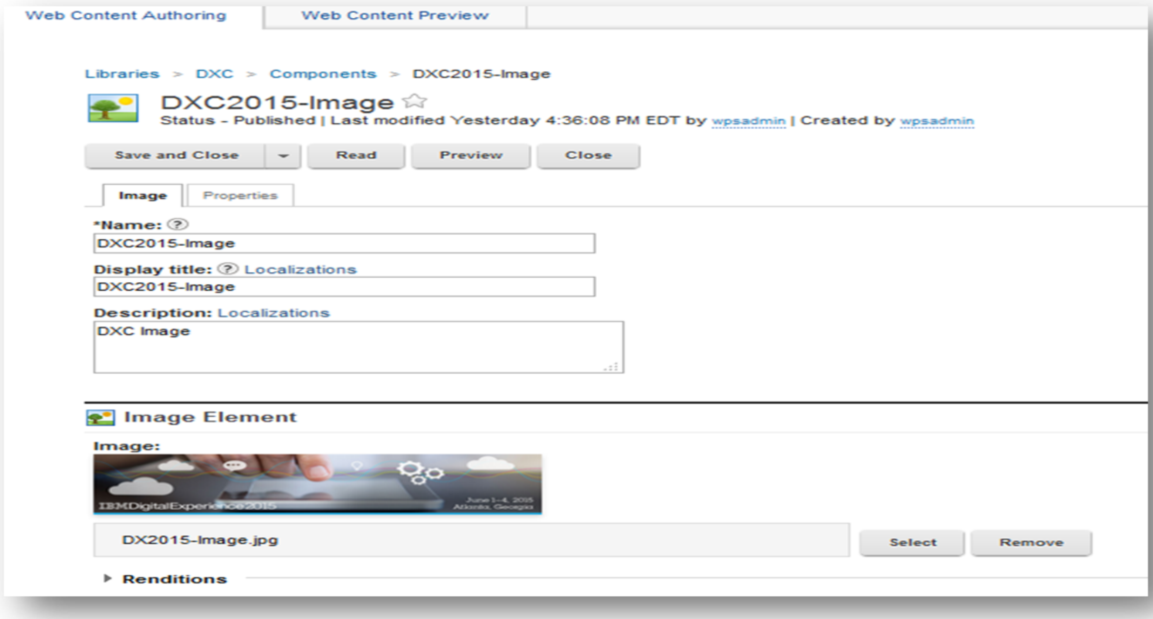
Licensed Materials - Property of IBM
5724-E76, 5724-E77, 5724-I29, 5655-W49
© Copyright IBM Corp. 2001, 2014

Let’s pull a new image into the theme from WCM. The figure below shows the usual theme code for pulling in an image from a static directory in the theme. The markup below which pulls in the image is contained in the theme template (theme.html or theme_en.html) file.



```
<div class="wpthemeBanner">
        <div class="wpthemeBannerInner">
                <img class="dxcImage" src="/DXCDemoStaticTheme/themes/DXCDemoTheme/images/dxc/dynamicSpots/DXC2015-Image.jpg">
                <div class="wpthemeInner">
                        <a rel="dynamic-content" href="dyn-cs:id:wp_search_dynspot"></a>
                        <a rel="dynamic-content" href="dyn-cs:id:DXCDemo_commonActions"></a>
                        <a rel="dynamic-content" href="dyn-cs:id:DXCDemo_mobileNav"></a>
                        <div class="wpthemeClear"></div>
                </div>
        </div>
</div><!--end main banner-->
```

To pull the image in from WCM, the first step would be to create a WCM image component in the WCM authoring UI. In this case, the image component contains an image called DX2015-Image.jpg



DXC2015-Image

Status - Published | Last modified Yesterday 4:36:08 PM EDT by wpsadmin | Created by wpsadmin

*Name: DXC2015-Image

Display title: Localizations DXC2015-Image

Description: Localizations DXC Image

Image: DX2015-Image.jpg

Once the image is created in WCM, the next thing needed is a way to reference the image. Instead of manually building the URL as documented in the Infocenter we will use the URL of the image from the authoring UI. In this example, Firebug is used to get the image reference URL. From the WCM UI, right click on the image and select "Inspect Element with Firebug". Firebug will open and display the HTML <img> tag. Copy the "src=" URL from the HTML.

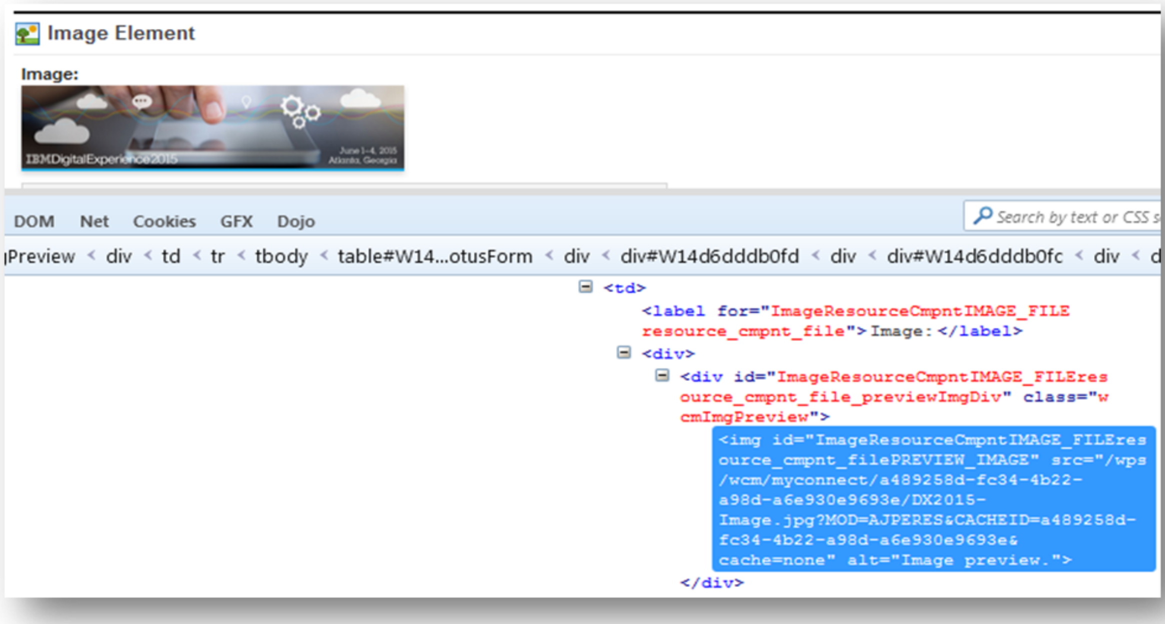
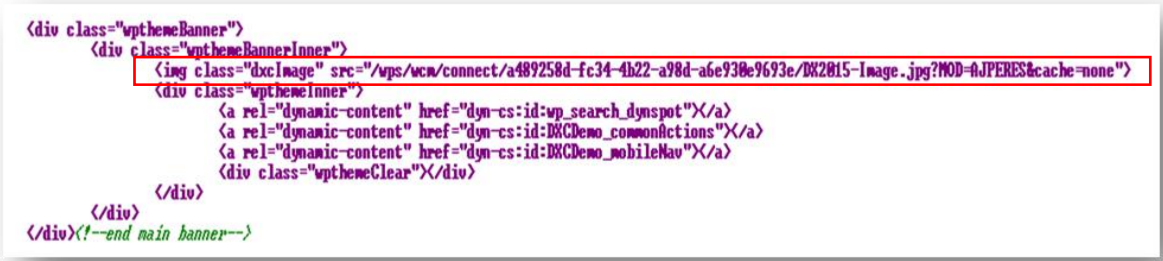


Image Element
Image:
DOM
Net
Cookies
GFX
Dojo
Search by text or CSS s
Preview < div < td < tr < tbody < table#W14...otusForm < div < div#W14d6dddb0fd < div < div#W14d6dddb0fc < div < d

```
<td>
    <label for="ImageResourceCmpntIMAGE_FILE
    resource_cmpnt_file">Image:</label>
    <div>
        <div id="ImageResourceCmpntIMAGE_FILEres
        ource_cmpnt_file_previewImgDiv" class="w
        cmImgPreview">
            <img id="ImageResourceCmpntIMAGE_FILEres
            ource_cmpnt_filePREVIEW_IMAGE" src="/wps
            /wcm/myconnect/a489258d-fc34-4b22-
            a98d-a6e930e9693e/DX2015-
            Image.jpg?MOD=AJPERES&CACHEID=a489258d-
            fc34-4b22-a98d-a6e930e9693e&
            cache=none" alt="Image preview.">
        </div>
```

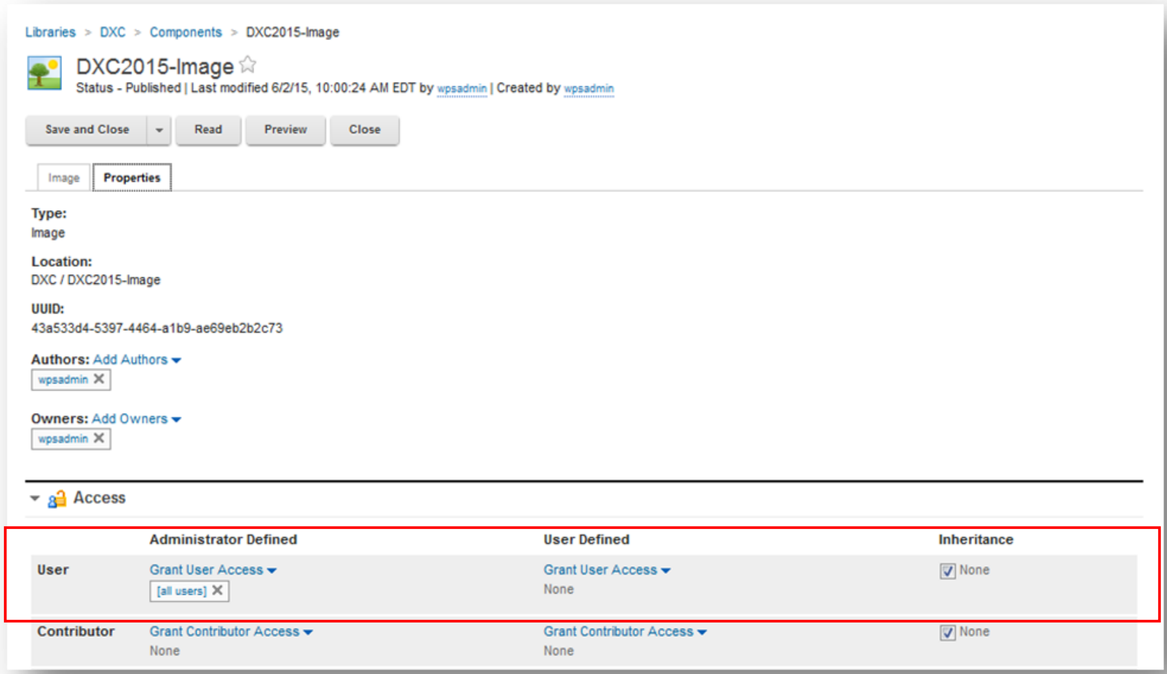
Use the copied URL to replace the “src=” portion of the theme template HTML <img> tag.



```
<div class="wpthemeBanner">
        <div class="wpthemeBannerInner">
                <img class="dxcImage" src="/wps/wcm/connect/a489258d-fc34-4b22-a98d-a6e930e9693e/DX2015-Image.jpg?MOD=AJPERES&cache=none">
                <div class="wpthemeInner">
                        <a rel="dynamic-content" href="dyn-cs:id:wp_search_dynspot"></a>
                        <a rel="dynamic-content" href="dyn-cs:id:DXCDemo_commonActions"></a>
                        <a rel="dynamic-content" href="dyn-cs:id:DXCDemo_mobileNav"></a>
                        <div class="wpthemeClear"></div>
                </div>
        </div>
</div><!--end main banner-->
```

/connect vs /myconnect

It is important to use /connect in the HTML <img> tag so that the image will be retrieved regardless of whether or not the user is anonymous (unauthenticated) or authenticated. If /myconnect is used, the image will not render for anonymous users. Make sure to set the correct access when creating the WCM image component.



Libraries > DXC > Components > DXC2015-Image

DXC2015-Image

Status - Published | Last modified 6/2/15, 10:00:24 AM EDT by wpsadmin | Created by wpsadmin

Type:
Image

Location:
DXC / DXC2015-Image

UUID:
43a533d4-5397-4464-a1b9-ae69eb2b2c73

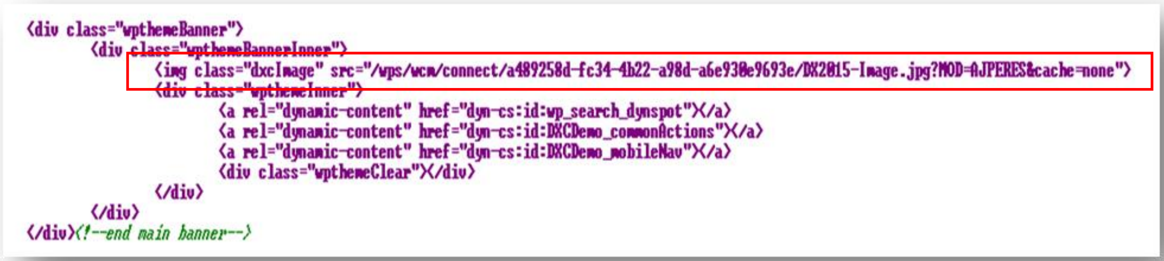
Authors: Add Authors
wpsadmin

Owners: Add Owners
wpsadmin

Access

	Administrator Defined	User Defined	Inheritance
User	Grant User Access [all users]	Grant User Access None	✓ None
Contributor	Grant Contributor Access None	Grant Contributor Access None	✓ None

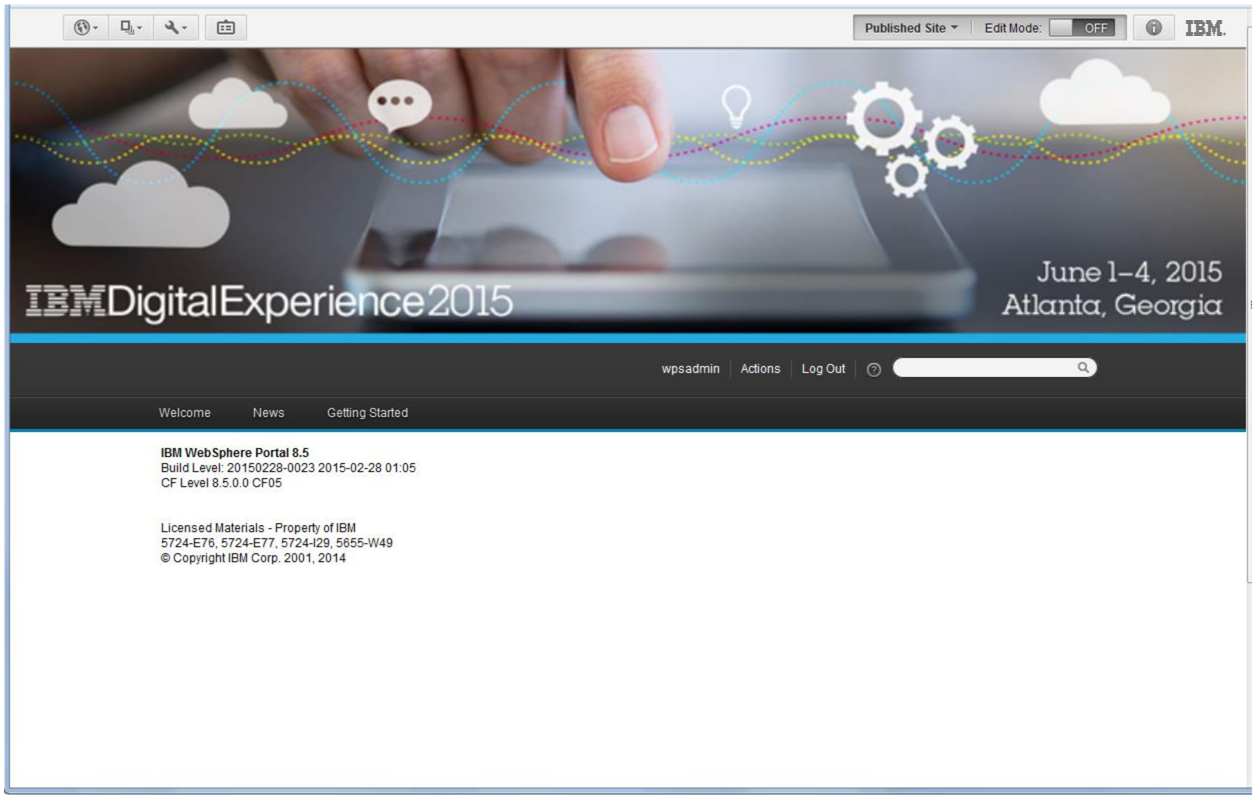
/myconnect	authenticated users only
/connect	anonymous and authenticated users



```
<div class="wpthemeBanner">
        <div class="wpthemeBannerInner">
                <img class="dxcImage" src="/wps/wcm/connect/a489258d-fc34-4b22-a98d-a6e930e9693e/DX2015-Image.jpg?MOD=AJPERES&cache=none">
                <div class="wpthemeInner">
                        <a rel="dynamic-content" href="dyn-cs:id:wp_search_dynspot"></a>
                        <a rel="dynamic-content" href="dyn-cs:id:DXCDemo_commonActions"></a>
                        <a rel="dynamic-content" href="dyn-cs:id:DXCDemo_mobileNav"></a>
                        <div class="wpthemeClear"></div>
                </div>
        </div>
</div><!--end main banner-->
```

Resulting Site

The site will look like below once the image is changed in the DXC2015-Image component in WCM.

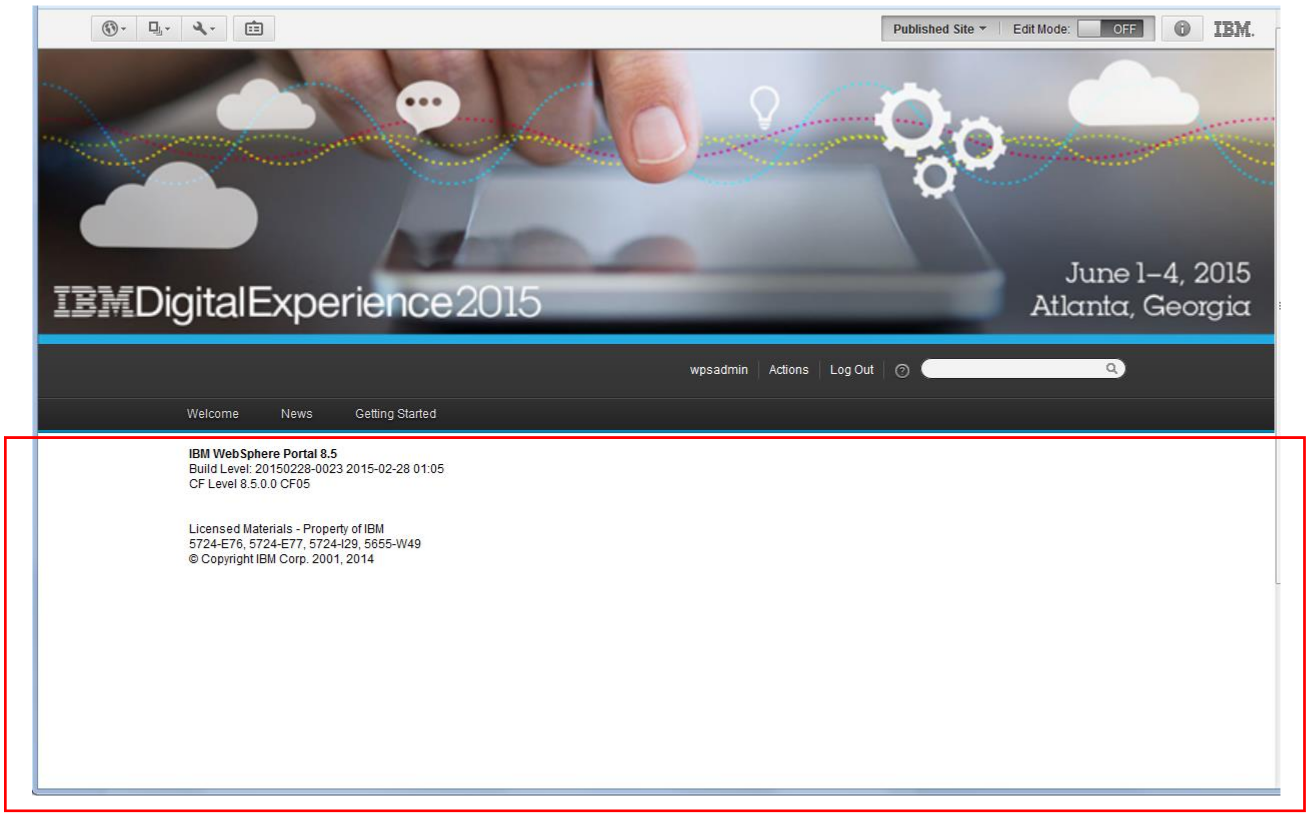


IBM WebSphere Portal 8.5
Build Level: 20150228-0023 2015-02-28 01:05
CF Level 8.5.0.0 CF05

Licensed Materials - Property of IBM
5724-E76, 5724-E77, 5724-I29, 5655-W49
© Copyright IBM Corp. 2001, 2014

CSS Example

The next example shows how to pull CSS files from WCM into the theme. In this example multiple CSS files will be pulled in together in a WCM HTML component. Suppose the site looks like below and a new color is required for the main content background.



IBM WebSphere Portal 8.5
Build Level: 20150228-0023 2015-02-28 01:05
CF Level 8.5.0.0 CF05

Licensed Materials - Property of IBM
5724-E76, 5724-E77, 5724-I29, 5655-W49
© Copyright IBM Corp. 2001, 2014

The initial dxcdemomain.css stylesheet which controls the main content color looks like this:

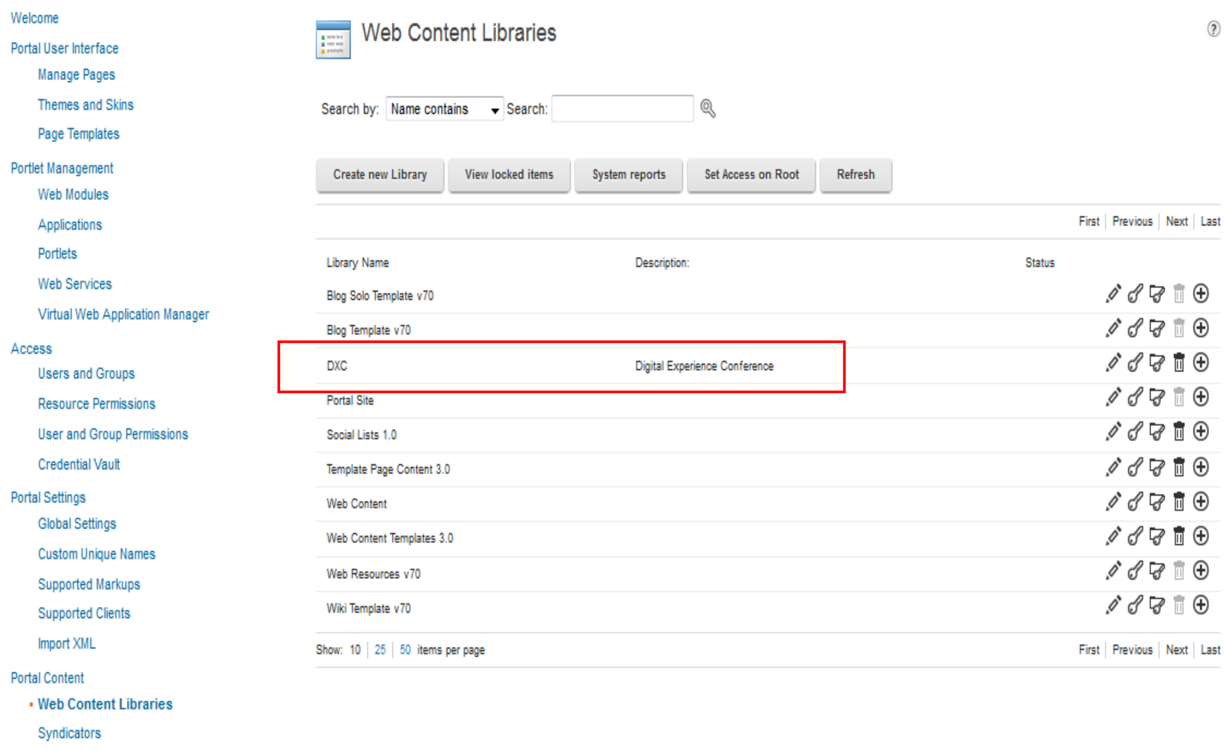
```
.wpthemeMainContent {
    background: white;
    min-height: 400px;
}
```

There is a requirement to change the background color to "powderblue".

```
.wpthemeMainContent {
    background: powderblue;
    min-height: 400px;
}
```

Create a WCM library

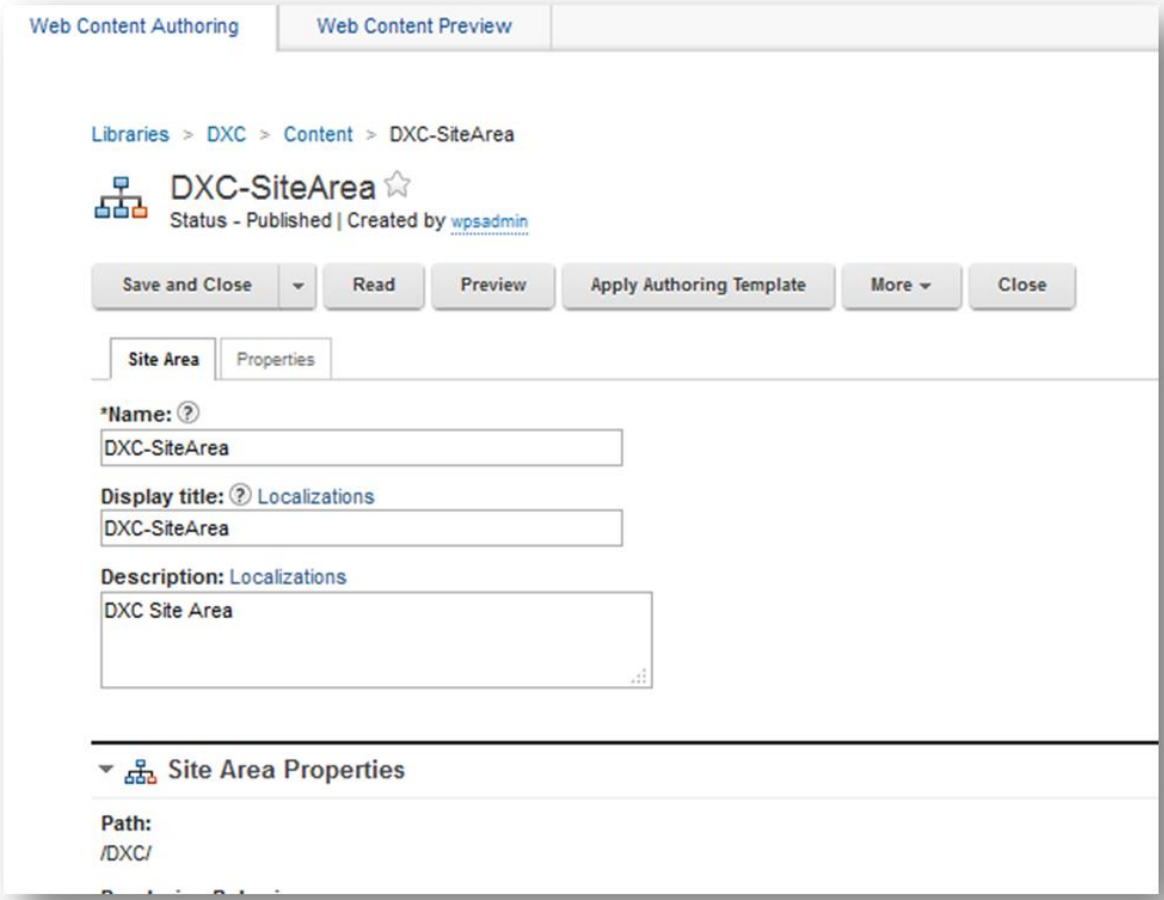
The first step in our sample is to create a WCM library in Portal administration under “Administration->Portal Content->Web Content Libraries”. In the example below, the DXC library is new in WCM. It is possible to use any existing library as well.



Library Name	Description:	Status
Blog Solo Template v70		
Blog Template v70		
DXC	Digital Experience Conference	
Portal Site		
Social Lists 1.0		
Template Page Content 3.0		
Web Content		
Web Content Templates 3.0		
Web Resources v70		
Wiki Template v70		

Create a Site Area

It is necessary to create a WCM site area that will be used as reference for the WCM component. This site area will also be used in the next example.



DXC-SiteArea

Status - Published | Created by wpsadmin

*Name: DXC-SiteArea

Display title: Localizations DXC-SiteArea

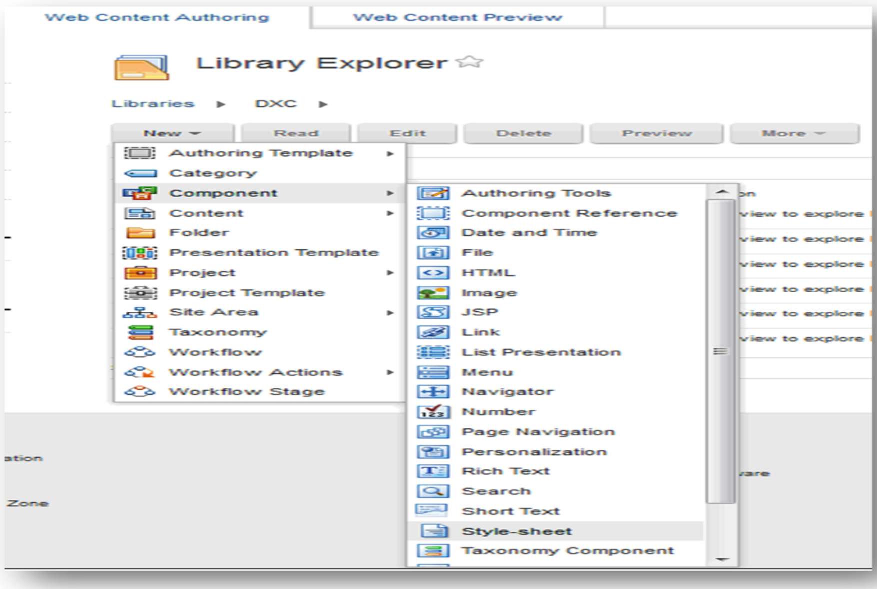
Description: Localizations DXC Site Area

Site Area Properties

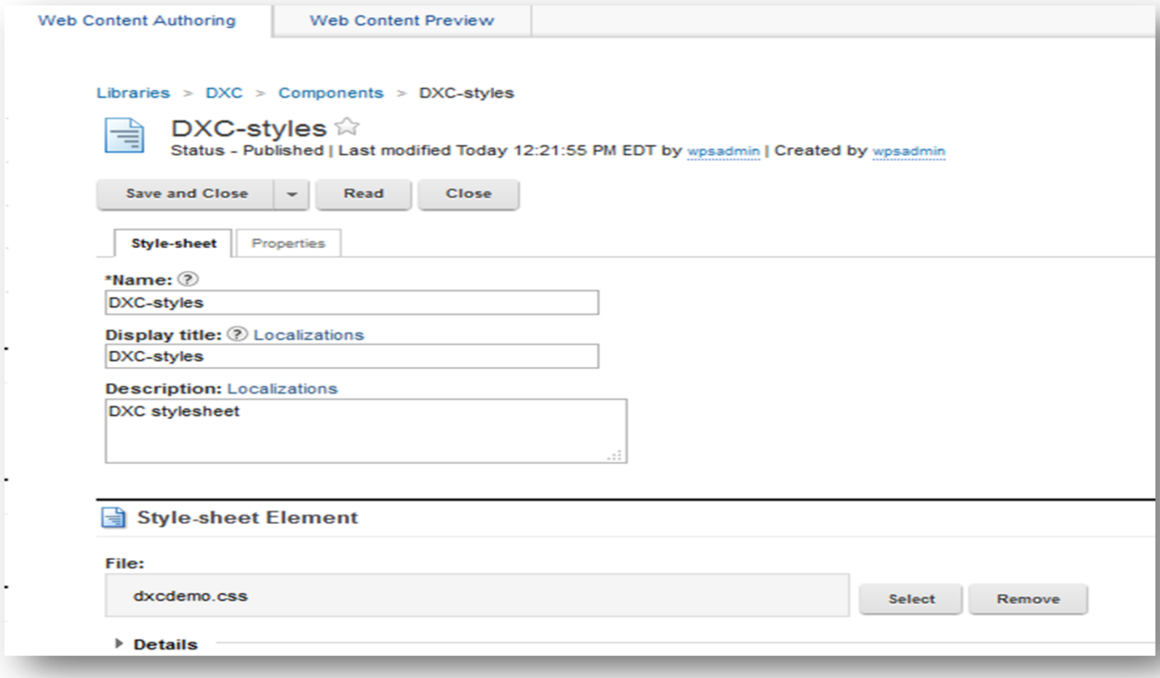
Path: /DXC/

Create a Stylesheet component for the CSS file(s)

In the WCM UI create a style sheet component for each stylesheet that will be included in the HTML component. In this example, two stylesheets will be included: dxcdemo.css (DXC-styles) and dxcdemomain.css (DXC-stylesMain).



For example:



DXC-styles

Status - Published | Last modified Today 12:21:55 PM EDT by wpsadmin | Created by wpsadmin

*Name: DXC-styles

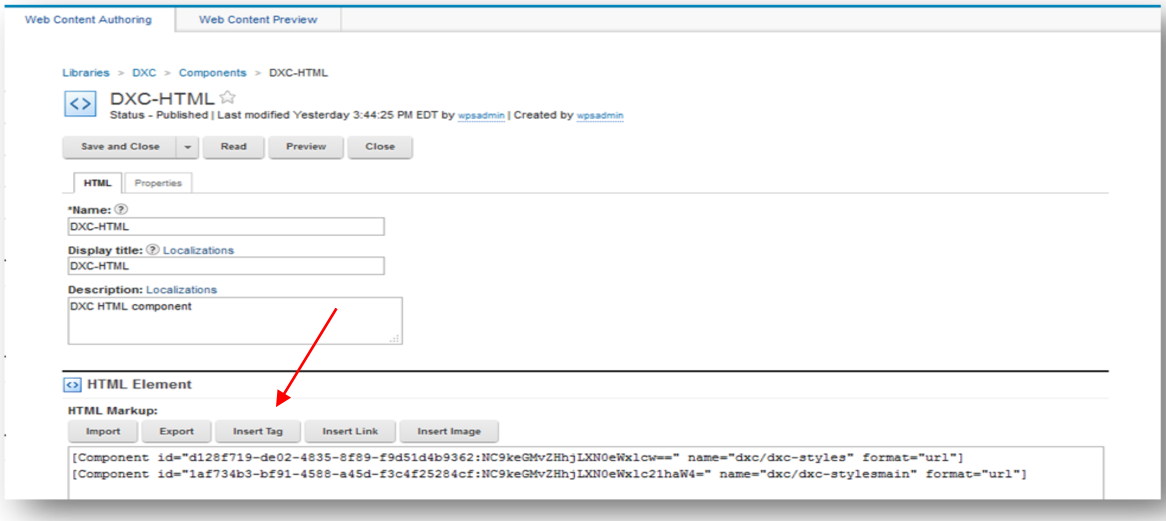
Display title: Localizations DXC-styles

Description: Localizations DXC stylesheet

File: dxcdemo.css

Create an HTML component

Create a WCM HTML component which will reference the stylesheets. Select the “Insert Tag” button.



Status - Published | Last modified Yesterday 3:44:25 PM EDT by wpsadmin | Created by wpsadmin

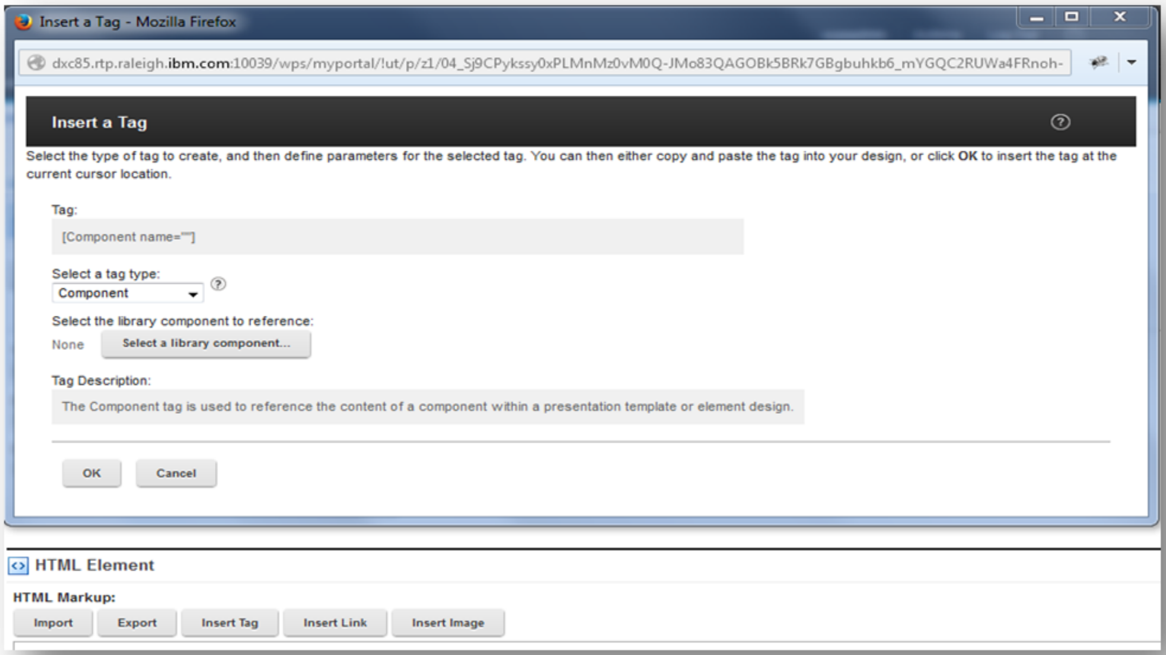
*Name: DXC-HTML

Display title: Localizations DXC-HTML

Description: Localizations DXC HTML component

```
[Component id="d128f719-de02-4835-8f89-f9d51d4b9362:NC9keGMvZHhjLXN0eWxlcw==" name="dxc/dxc-styles" format="url"]
[Component id="1af734b3-bf91-4588-a45d-f3c4f25284cf:NC9keGMvZHhjLXN0eWxlc21haW4=" name="dxc/dxc-stylesmain" format="url"]
```

The “Insert a Tag” dialog will open.


Insert a Tag - Mozilla Firefox
dxc85.rtp.raleigh.ibm.com:10039/wps/myportal/!ut/p/z1/04_Sj9CPykssy0xPLMnMz0vM0Q-JMo83QAGOBk5BRk7GBgbuhkb6_mYGQC2RUWa4FRnoh-
Insert a Tag
Select a library component...
OK
Cancel
HTML Element
HTML Markup:
Import
Export
Insert Tag
Insert Link
Insert Image

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

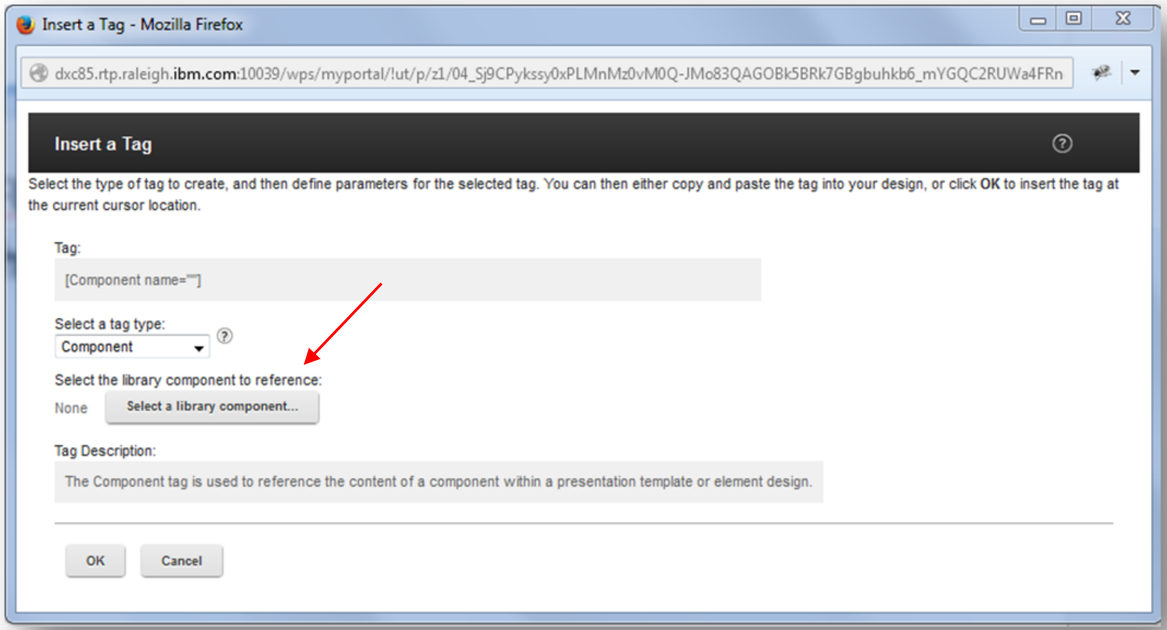
Tag: [Component name=""]

Select a tag type: Component

Select the library component to reference: None

Tag Description: The Component tag is used to reference the content of a component within a presentation template or element design.

Set the “Select a tag type:” to “Component” and select the “Select a library component…” button.


Insert a Tag - Mozilla Firefox

dxc85.rtp.raleigh.ibm.com:10039/wps/myportal/!ut/p/z1/04_Sj9CPykssy0xPLMnMz0vM0Q-JMo83QAGOBk5BRk7GBgbuhkb6_mYGQC2RUWa4FRn

Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

[Component name=""]

Select a tag type: Component

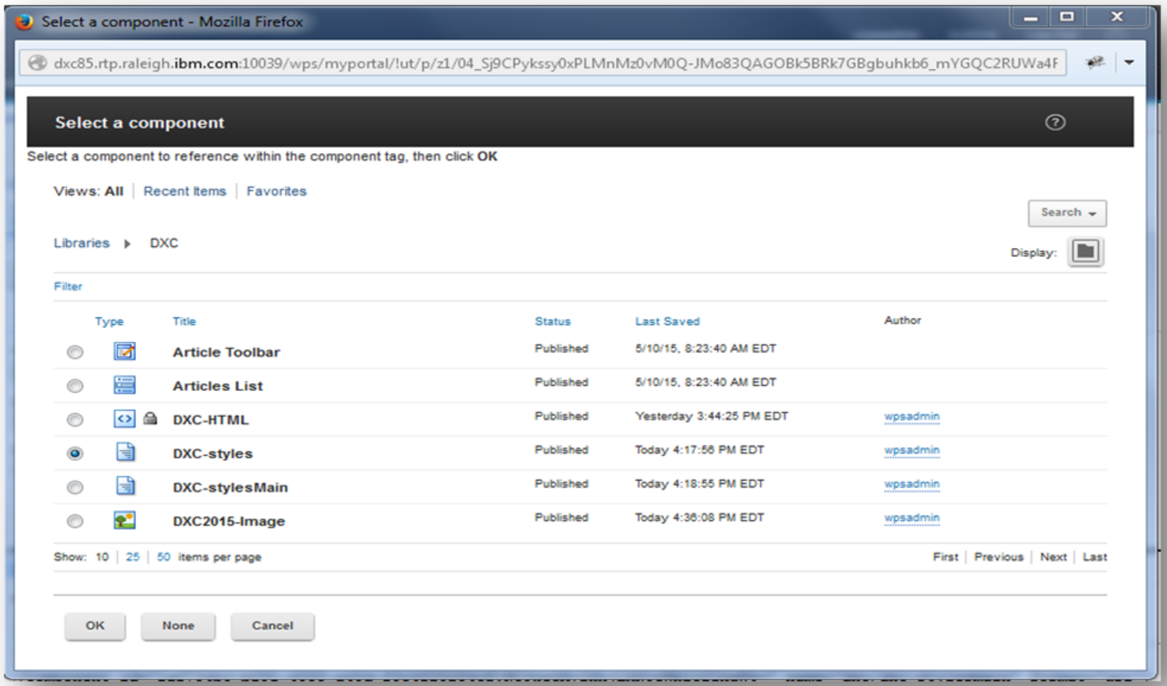
Select the library component to reference:

None

Tag Description:

The Component tag is used to reference the content of a component within a presentation template or element design.

Select the CSS component created earlier.


Select a component - Mozilla Firefox

dxc85.rtp.raleigh.ibm.com:10039/wps/myportal/!ut/p/z1/04_Sj9CPykssy0xPLMnMz0vM0Q-JMo83QAGOBk5BRk7GBgbuhkb6_mYGQC2RUWa4F

Select a component

Select a component to reference within the component tag, then click **OK**

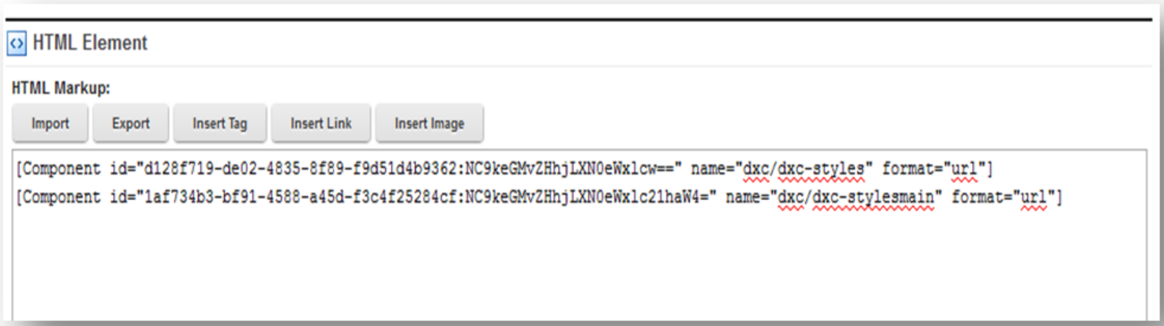
Views: All | Recent Items | Favorites

Libraries ▸ DXC

Type	Title	Status	Last Saved	Author
○	Article Toolbar	Published	5/10/15, 8:23:40 AM EDT	
○	Articles List	Published	5/10/15, 8:23:40 AM EDT	
○	DXC-HTML	Published	Yesterday 3:44:25 PM EDT	wpsadmin
●	DXC-styles	Published	Today 4:17:56 PM EDT	wpsadmin
○	DXC-stylesMain	Published	Today 4:18:55 PM EDT	wpsadmin
○	DXC2015-Image	Published	Today 4:36:08 PM EDT	wpsadmin

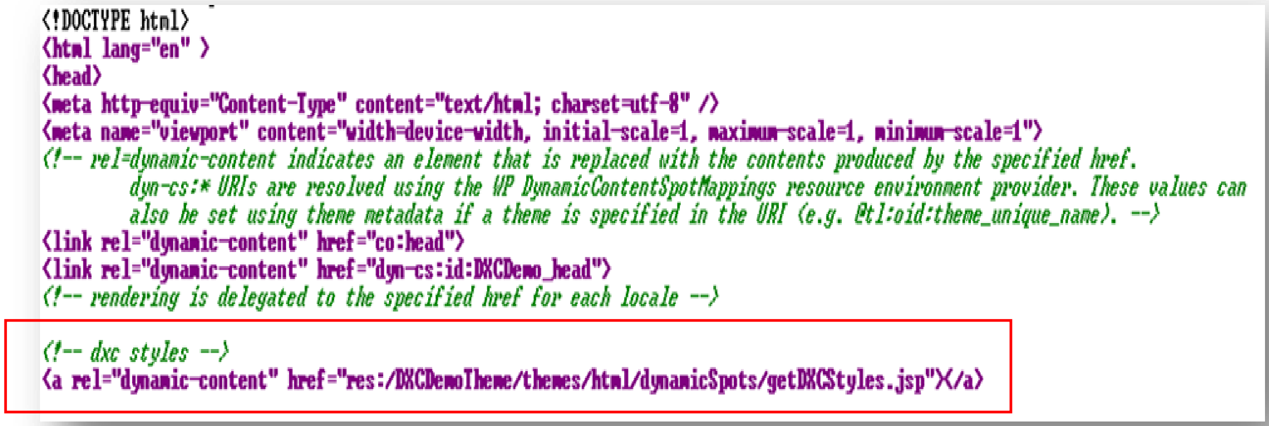
Show: 10 | 25 | 50 items per page

Repeat the process described above for the second stylesheet and the HTML component will be updated to include both of the stylesheets.



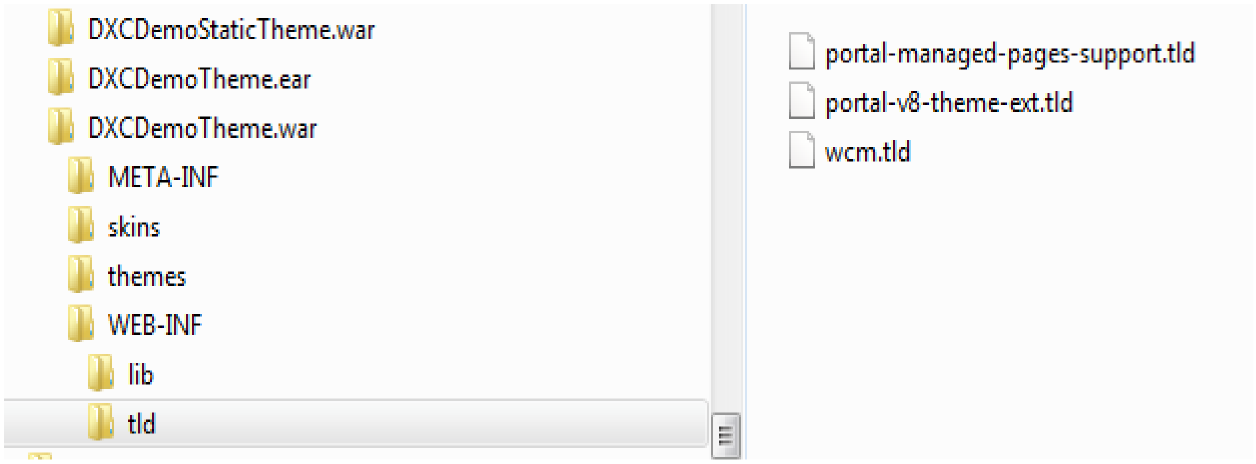
```
[Component id="d128f719-de02-4835-8f89-f9d51d4b9362:NC9keGMvZHhjLXN0eWxlcw==" name="dxc/dxc-styles" format="url"]
[Component id="1af734b3-bf91-4588-a45d-f3c4f25284cf:NC9keGMvZHhjLXN0eWxlc21haW4=" name="dxc/dxc-stylesmain" format="url"]
```

Create a dynamic content spot in the theme template (theme.html) file which references a JSP. In this example the getDXCStyles.jsp is used.



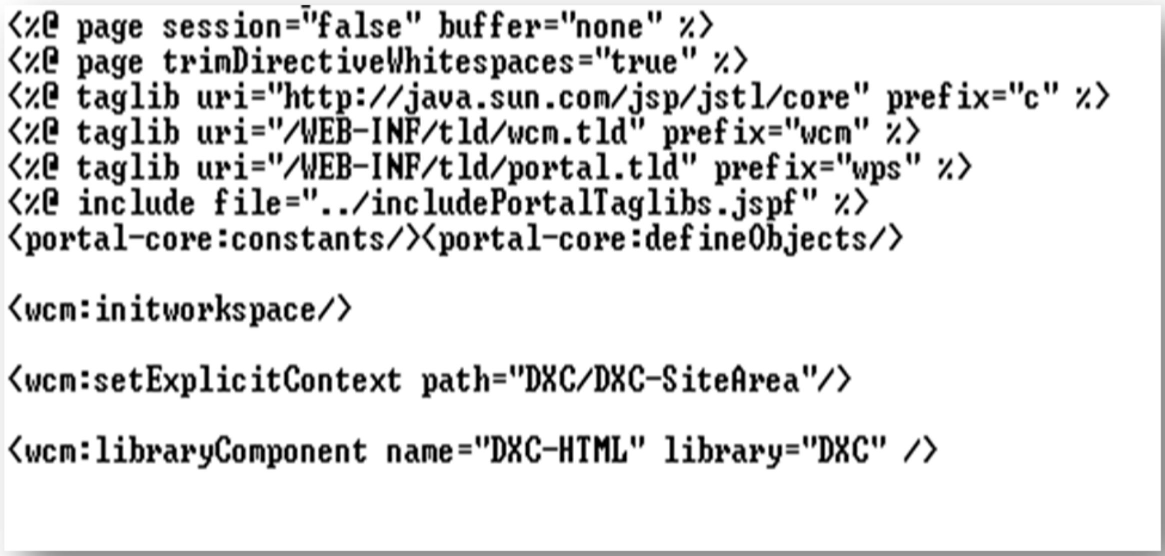
```
<!DOCTYPE html>
<html lang="en" >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1">
<!-- rel=dynamic-content indicates an element that is replaced with the contents produced by the specified href.
        dyn-cs:* URIs are resolved using the WP DynamicContentSpotMappings resource environment provider. These values can
        also be set using theme metadata if a theme is specified in the URI (e.g. @tl:oid:theme_unique_name). -->
<link rel="dynamic-content" href="co:head">
<link rel="dynamic-content" href="dyn-cs:id:DXCDemo_head">
<!-- rendering is delegated to the specified href for each locale -->

<!-- dxc styles -->
<a rel="dynamic-content" href="res:/DXCDemoTheme/themes/html/dynamicSpots/getDXCStyles.jsp"></a>
```

Include the WCM JSP tag library in the theme. This library can be obtained from the Portal server install directory: /installedApps/<node name>/wcm.ear/ilwwcm.war/WEB-INF/tld. The theme directories will look like:



Use the WEB Content JSP tags to reference the HTML component in the JSP. The path must reference either content or a site area. In this example, it makes sense to use the site area created earlier.

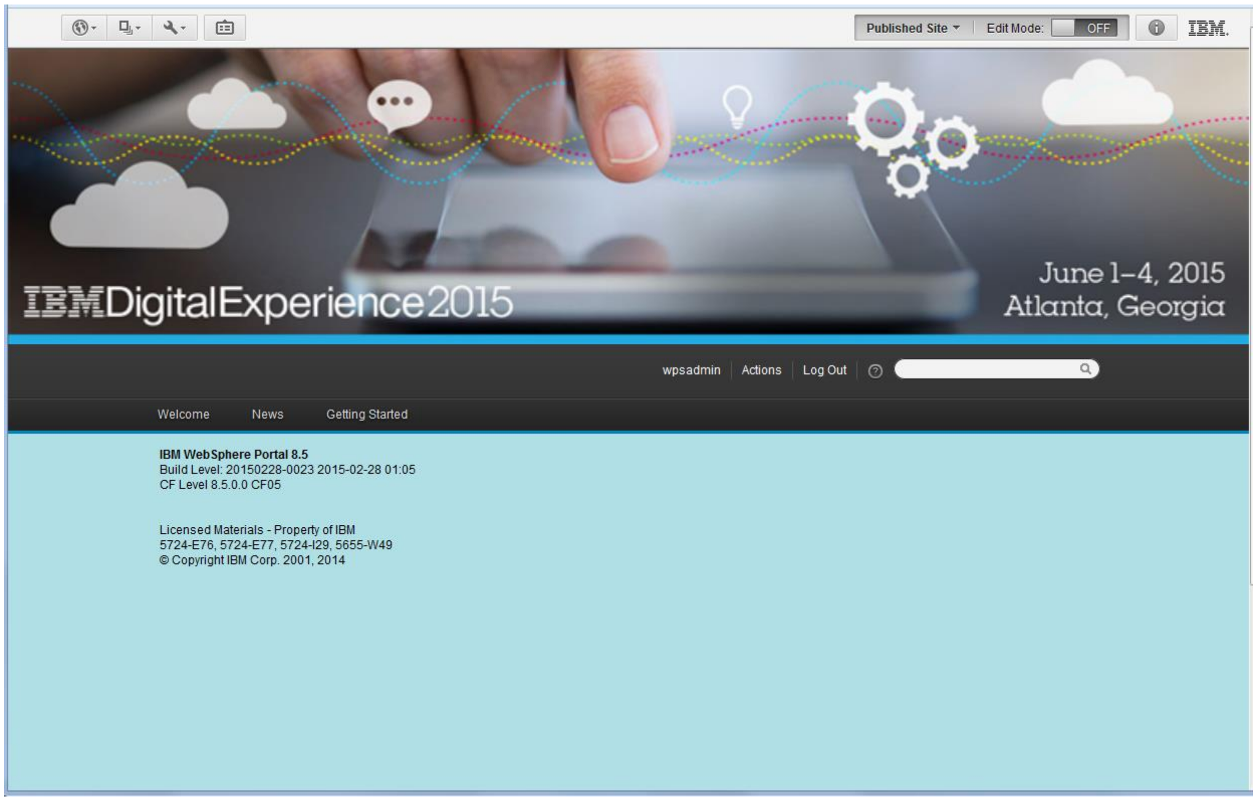
getDXCStyles.jsp



```
<%@ page session="false" buffer="none" %>
<%@ page trimDirectiveWhitespaces="true" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="/WEB-INF/tld/wcm.tld" prefix="wcm" %>
<%@ taglib uri="/WEB-INF/tld/portal.tld" prefix="wps" %>
<%@ include file="../includePortalTaglibs.jspf" %>
<portal-core:constants/><portal-core:defineObjects/>

<wcm:initworkspace/>

<wcm:setExplicitContext path="DXC/DXC-SiteArea"/>

<wcm:libraryComponent name="DXC-HTML" library="DXC" />
```

Resulting Site

The site will look like below once the background color is changed in the DXC-stylesMain component in WCM.



IBM WebSphere Portal 8.5
Build Level: 20150228-0023 2015-02-28 01:05
CF Level 8.5.0.0 CF05

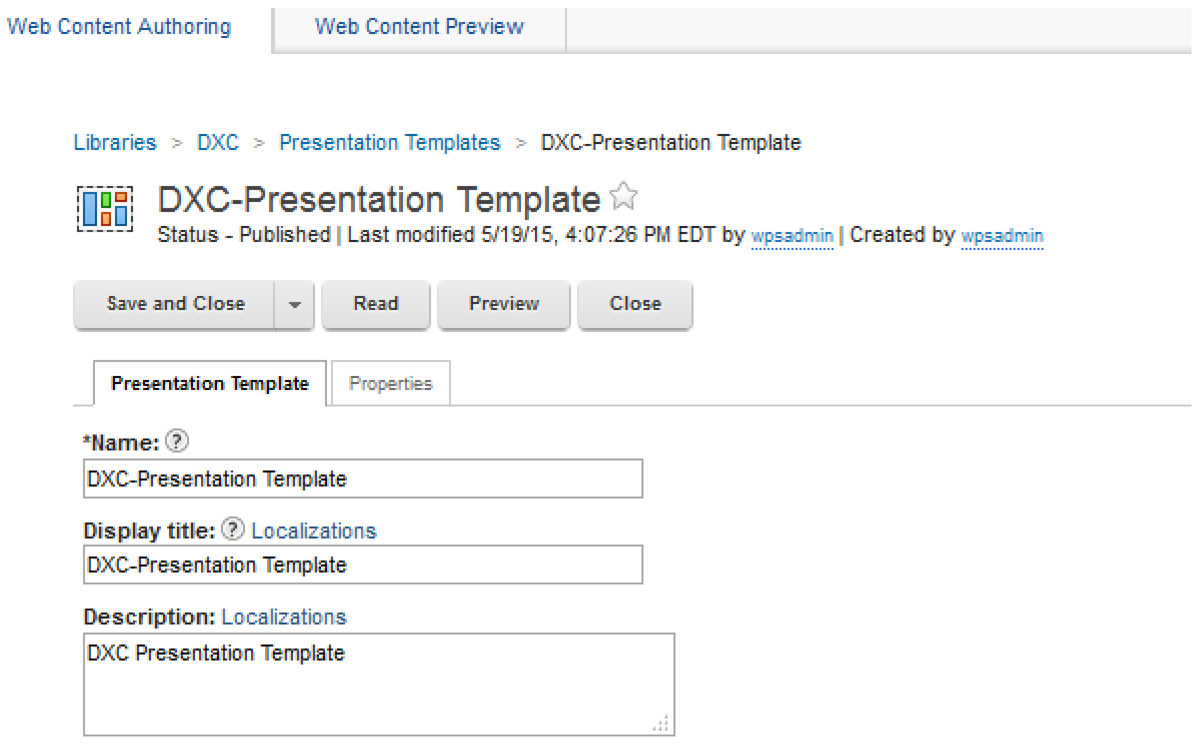
Licensed Materials - Property of IBM
5724-E76, 5724-E77, 5724-I29, 5655-W49
© Copyright IBM Corp. 2001, 2014

Footer Example

The next example shows how to pull WCM content of type rich text into the theme footer. In difference to the earlier samples we are using content vs. components. This can reduce the amount of components needed for a theme and enable a user to make changes to multiple fields in the content in one artifact. This is a similar approach to the Script Portlet.

Create a Presentation Template

The DXC site area was created in the previous example, so the first step in the footer example is to create a WCM presentation template. In WCM, the presentation template determines how content is being rendered.



DXC-Presentation Template

Status - Published | Last modified 5/19/15, 4:07:26 PM EDT by wpsadmin | Created by wpsadmin

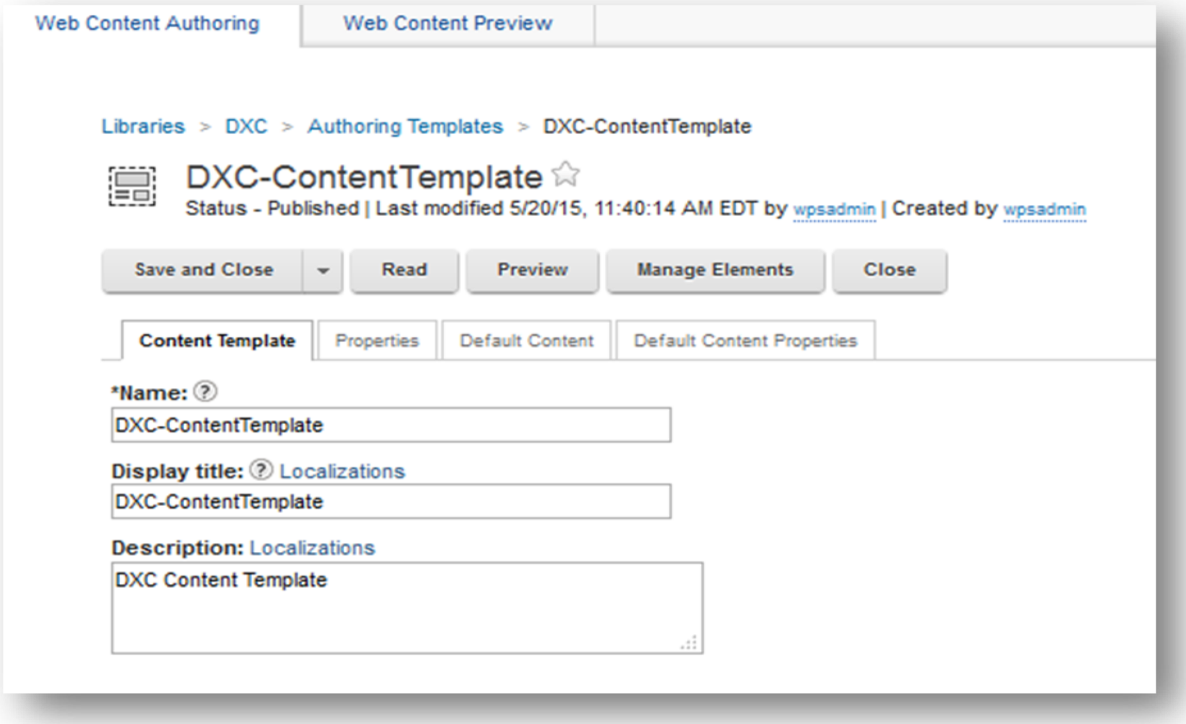
*Name: DXC-Presentation Template

Display title: Localizations DXC-Presentation Template

Description: Localizations DXC Presentation Template

Create a Content (Authoring) Template and associate it with the Presentation Template

The next step is to create a WCM content or authoring template. Authoring templates identify what fields and elements can be part of the site.



Libraries > DXC > Authoring Templates > DXC-ContentTemplate

DXC-ContentTemplate

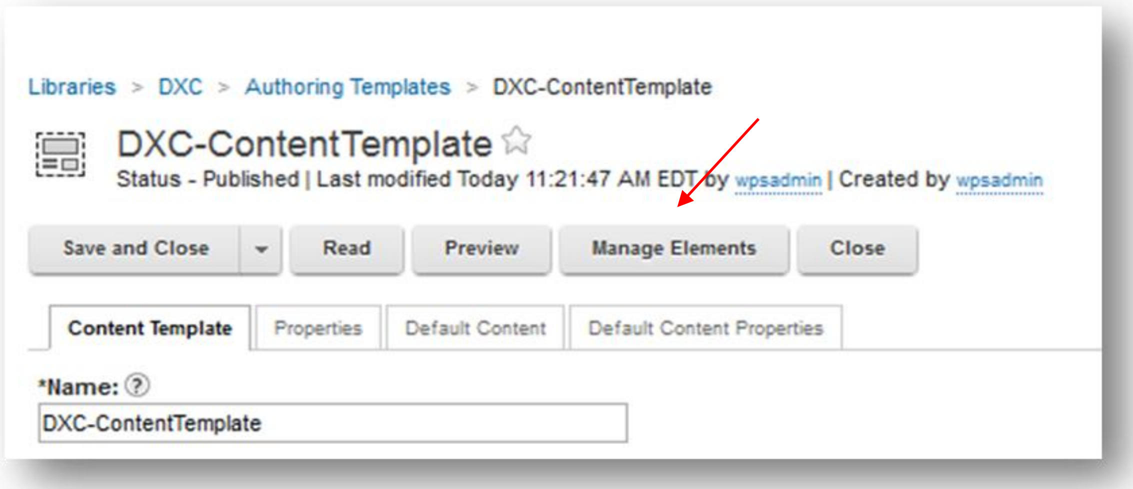
Status - Published | Last modified 5/20/15, 11:40:14 AM EDT by wpsadmin | Created by wpsadmin

*Name: DXC-ContentTemplate

Display title: Localizations DXC-ContentTemplate

Description: Localizations DXC Content Template

Select Manage Elements and add a Rich Text element.



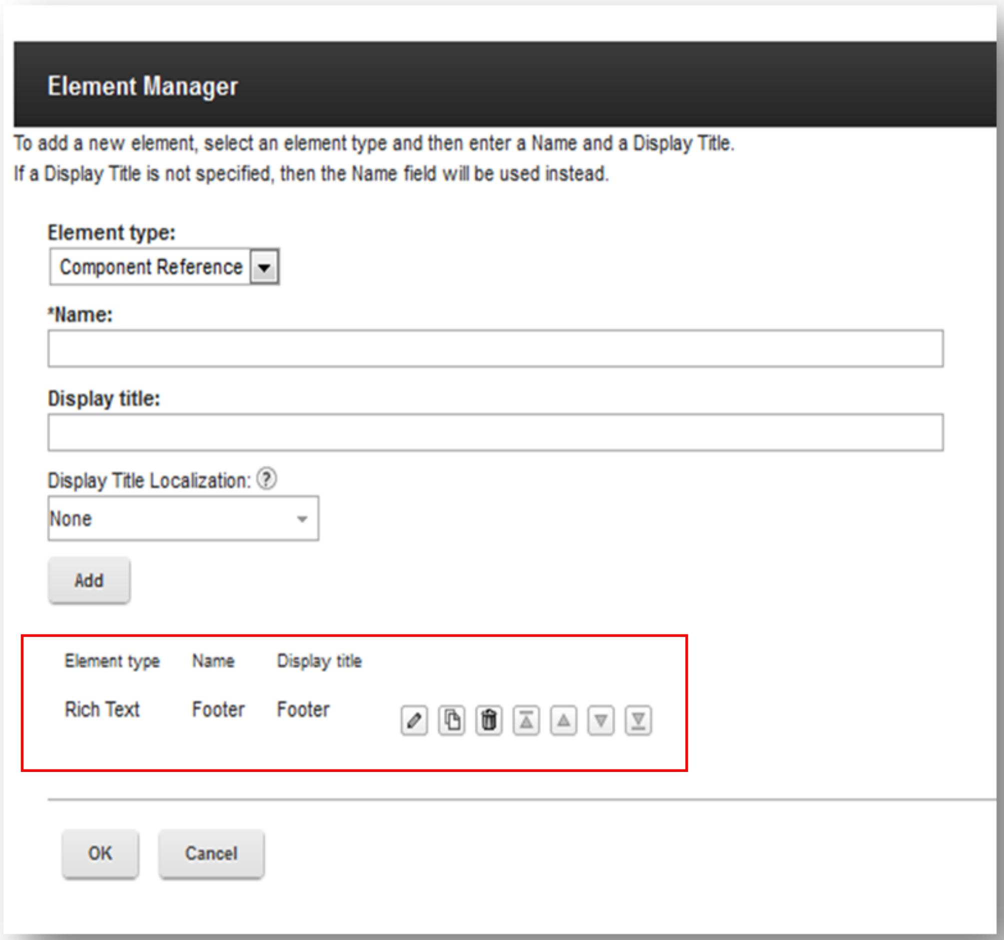
Libraries > DXC > Authoring Templates > DXC-ContentTemplate

DXC-ContentTemplate

Status - Published | Last modified Today 11:21:47 AM EDT by wpsadmin | Created by wpsadmin

*Name: DXC-ContentTemplate

The “Element Manager” dialog will open. Add a “Rich Text” element type and give it a name of “Footer”.



Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

Element type: Component Reference

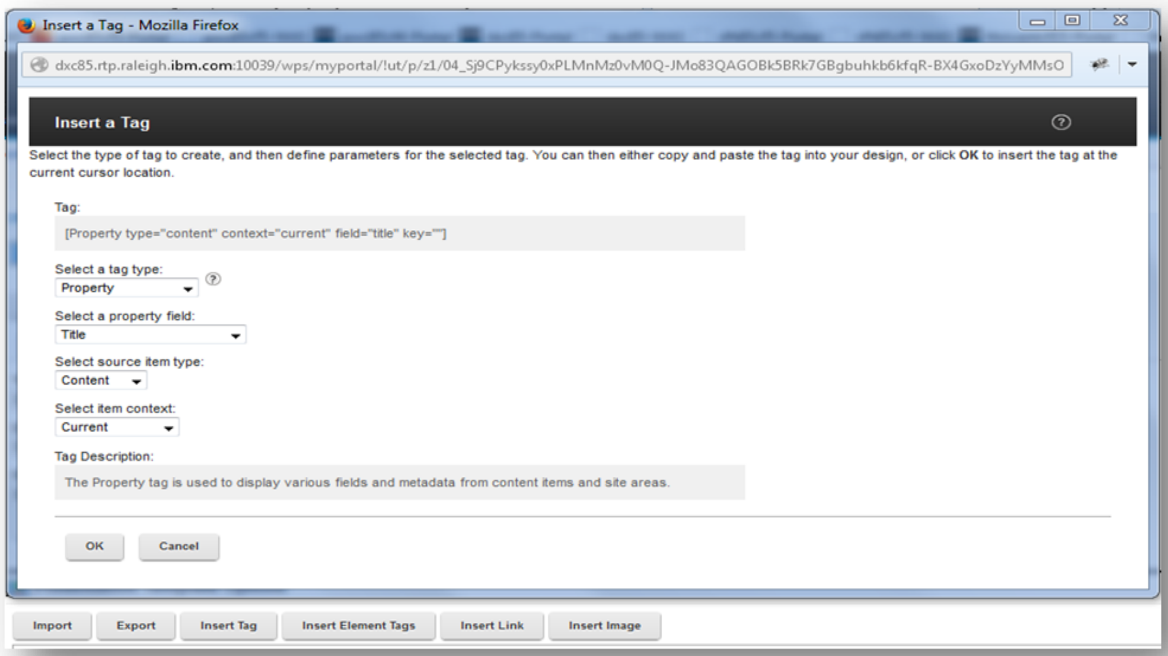
*Name:

Display title:

Display Title Localization: None

Element type	Name	Display title
Rich Text	Footer	Footer

Revisit the Presentation Template and select the “Insert Tag” button and the Insert a Tag dialog shown below will open.



Insert a Tag - Mozilla Firefox

dxc85.rtp.raleigh.ibm.com:10039/wps/myportal/!ut/p/z1/04_Sj9CPykssy0xPLMnMz0vM0Q-JMo83QAGOBk5BRk7GBgbuhkb6kfqR-BX4GxoDzYyMMsO

Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

```
[Property type="content" context="current" field="title" key=""]
```

Select a tag type: Property

Select a property field: Title

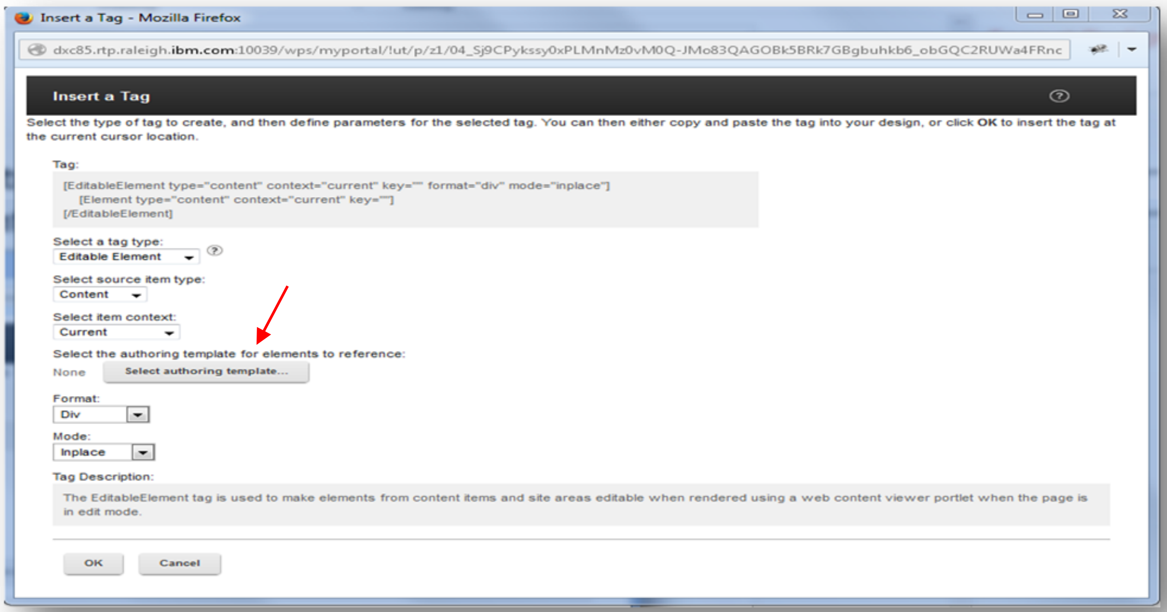
Select source item type: Content

Select item context: Current

Tag Description:

The Property tag is used to display various fields and metadata from content items and site areas.

Set the “Select a tag type:” to “Editable Element” and select the “Select authoring template…” button.



Insert a Tag - Mozilla Firefox

dxc85.rtp.raleigh.ibm.com:10039/wps/myportal/!ut/p/z1/04_Sj9CPykssy0xPLMnMz0vM0Q-JMo83QAGOBk5BRk7GBgbuhkb6_obGQC2RUWa4FRnc

Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

```
[EditableElement type="content" context="current" key="" format="div" mode="inplace"]
    [Element type="content" context="current" key=""]
[/EditableElement]
```

Select a tag type: Editable Element

Select source item type: Content

Select item context: Current

Select the authoring template for elements to reference:

None

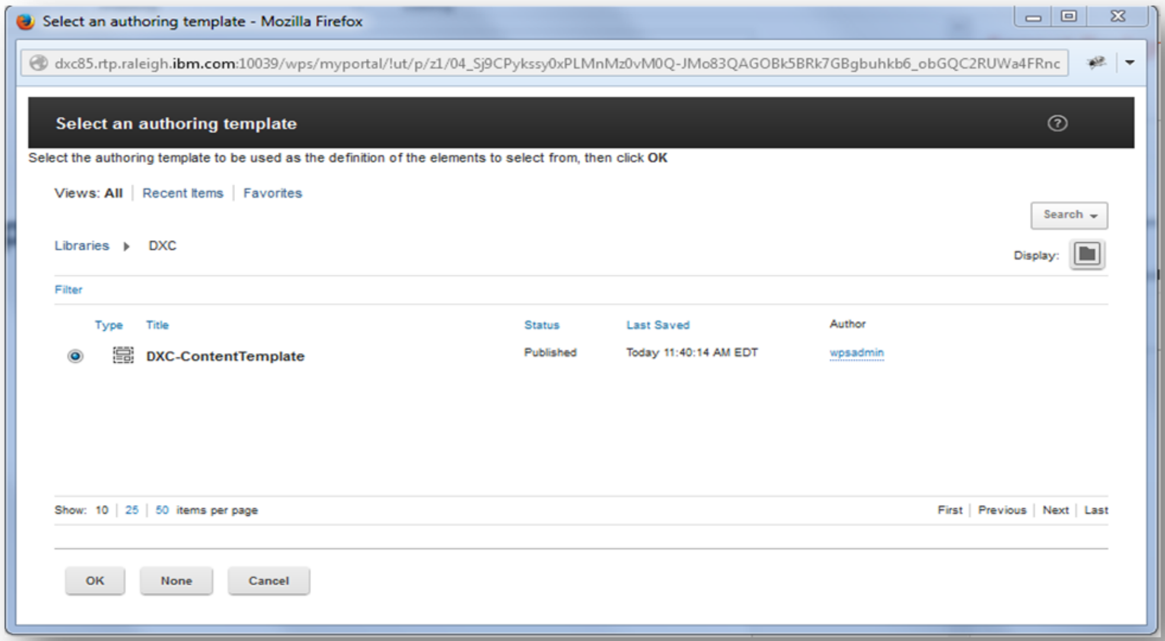
Format: Div

Mode: Inplace

Tag Description:

The EditableElement tag is used to make elements from content items and site areas editable when rendered using a web content viewer portlet when the page is in edit mode.

Select the Authoring Template.


Select an authoring template - Mozilla Firefox
dxc85.rtp.raleigh.ibm.com:10039/wps/myportal/!ut/p/z1/04_Sj9CPykssy0xPLMnMz0vM0Q-JMo83QAGOBk5BRk7GBgbuhkb6_obGQC2RUWa4FRnc
Select an authoring template

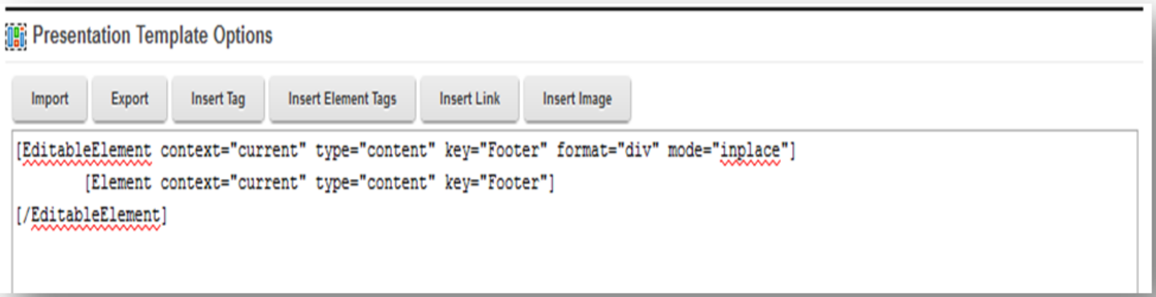
Select the authoring template to be used as the definition of the elements to select from, then click **OK**

Views: All | Recent Items | Favorites
Search
Libraries ▸ DXC
Display:
Filter

	Type	Title	Status	Last Saved	Author
●		DXC-ContentTemplate	Published	Today 11:40:14 AM EDT	wpsadmin

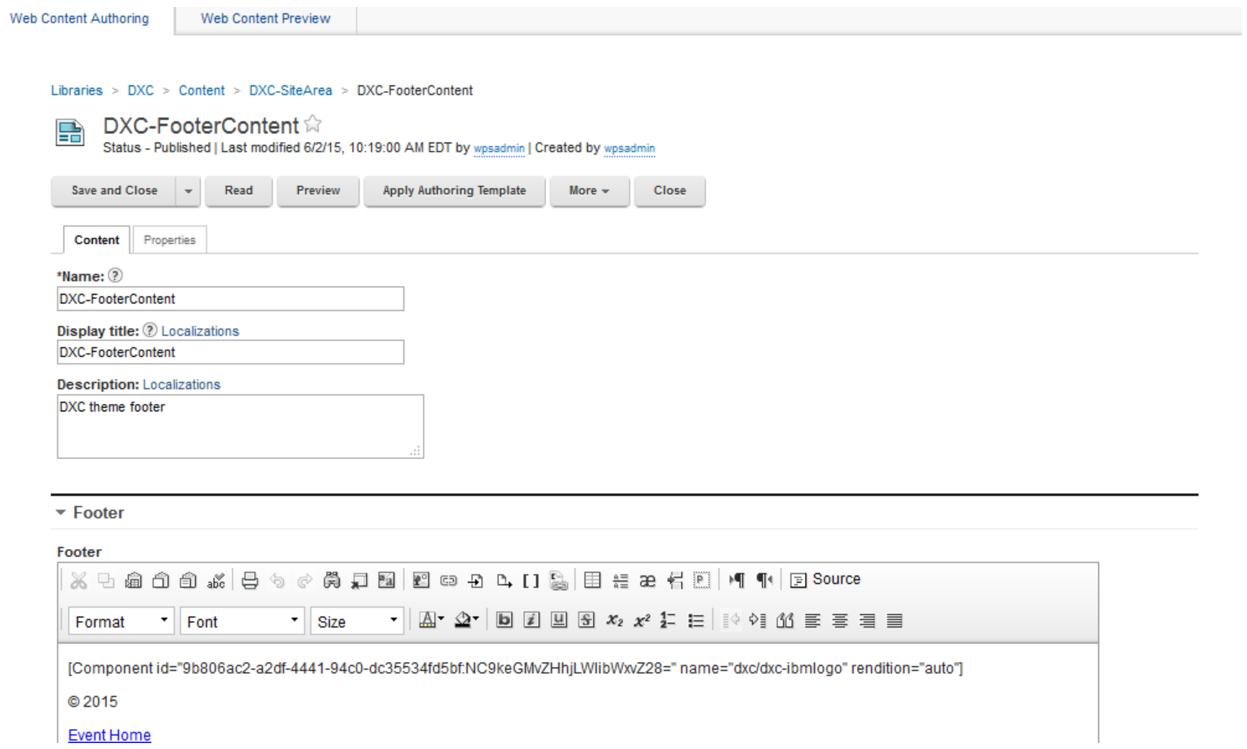
Show: 10 | 25 | 50 items per page
First | Previous | Next | Last
OK
None
Cancel

The Presentation Template will be updated.



```
[EditableElement context="current" type="content" key="Footer" format="div" mode="inplace"]
        [Element context="current" type="content" key="Footer"]
[/EditableElement]
```

Create content and use the Footer (Rich Text) area to define the theme footer.



Libraries > DXC > Content > DXC-SiteArea > DXC-FooterContent

DXC-FooterContent

Status - Published | Last modified 6/2/15, 10:19:00 AM EDT by wpsadmin | Created by wpsadmin

*Name: DXC-FooterContent

Display title: Localizations DXC-FooterContent

Description: Localizations DXC theme footer

Footer

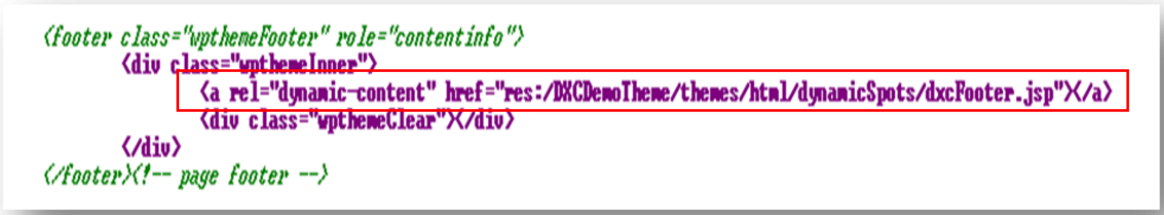
Footer

[Component id="9b806ac2-a2df-4441-94c0-dc35534fd5bf:NC9keGMvZHhjLWlibWxvZ28=" name="dxc/dxc-ibmlogo" rendition="auto"]

© 2015

Event Home

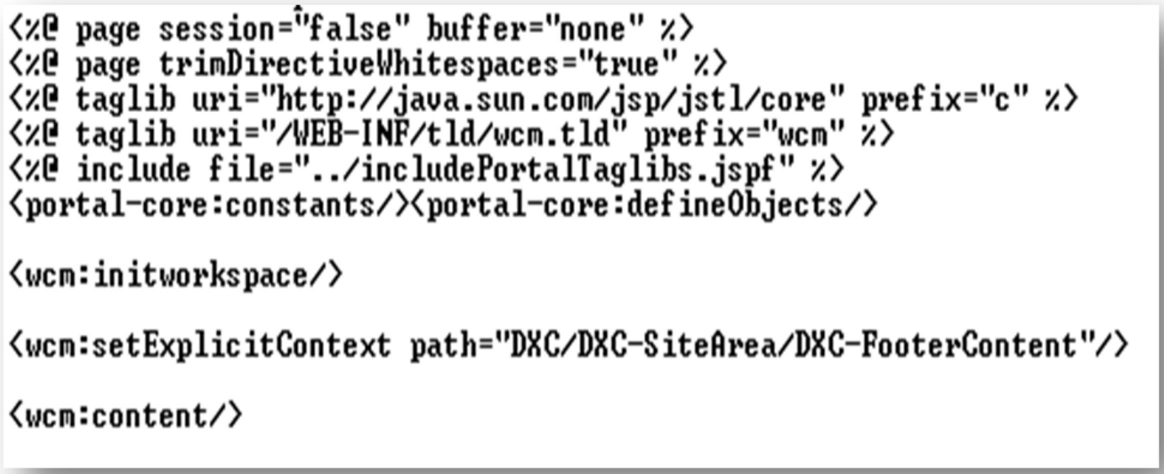
Create a dynamic content spot in the theme template (theme.html) file which references a JSP.



```
<footer class="wpthemeFooter" role="contentinfo">
        <div class="wpthemeInner">
                <a rel="dynamic-content" href="res:/DXCDemoTheme/themes/html/dynamicSpots/dxcFooter.jsp"></a>
                <div class="wpthemeClear"></div>
        </div>
</footer><!-- page footer -->
```

Use the WEB Content JSP tags to reference the content in the JSP.

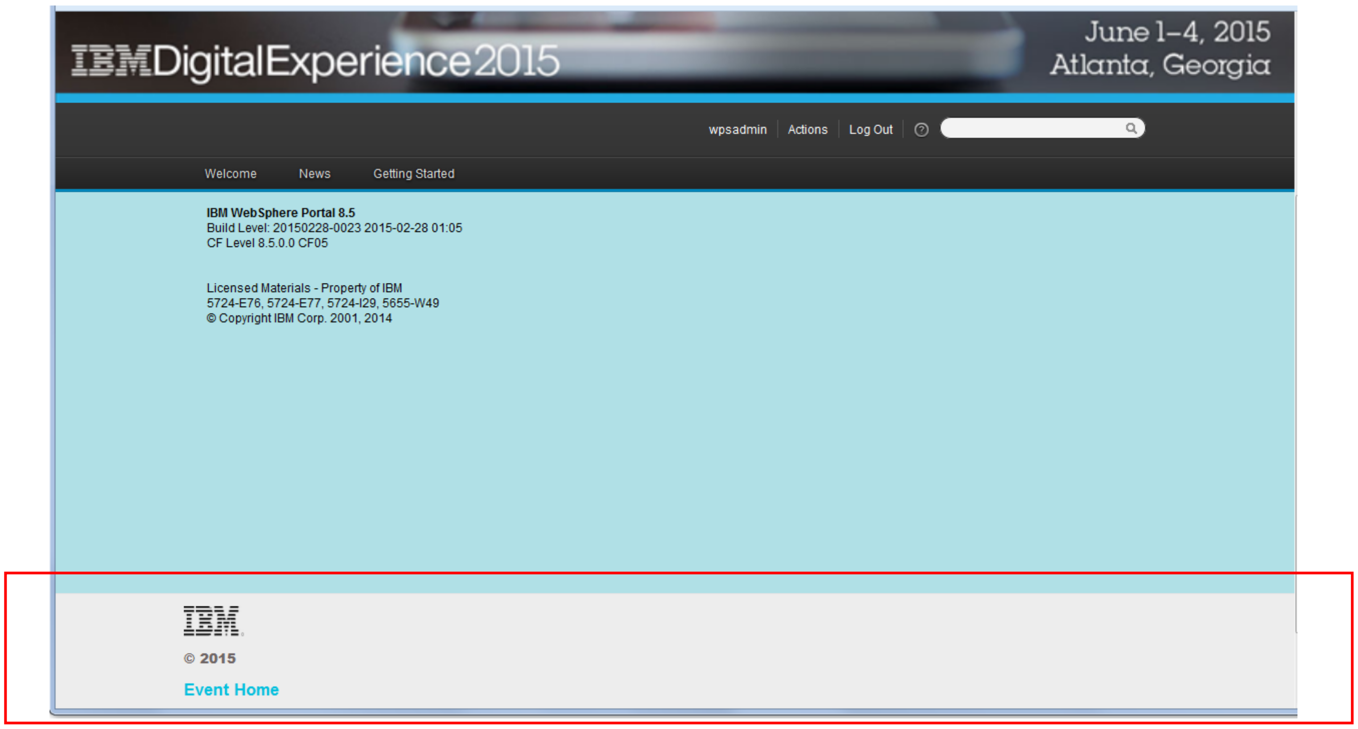
dxcFooter.jsp



```
<%@ page session="false" buffer="none" %>
<%@ page trimDirectiveWhitespaces="true" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="/WEB-INF/tld/wcm.tld" prefix="wcm" %>
<%@ include file="../includePortalTaglibs.jspf" %>
<portal-core:constants/><portal-core:defineObjects/>

<wcm:initworkspace/>

<wcm:setExplicitContext path="DXC/DXC-SiteArea/DXC-FooterContent"/>

<wcm:content/>
```

Resulting Site

The site will look like below once the dxcFooter.jsp runs.



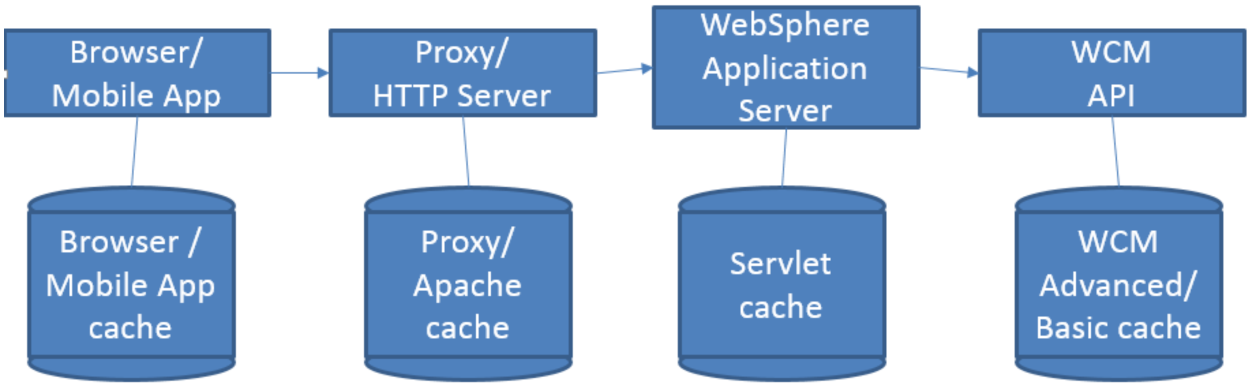
IBM WebSphere Portal 8.5
Build Level: 20150228-0023 2015-02-28 01:05
CF Level 8.5.0.0 CF05

Licensed Materials - Property of IBM
5724-E76, 5724-E77, 5724-I29, 5655-W49
© Copyright IBM Corp. 2001, 2014

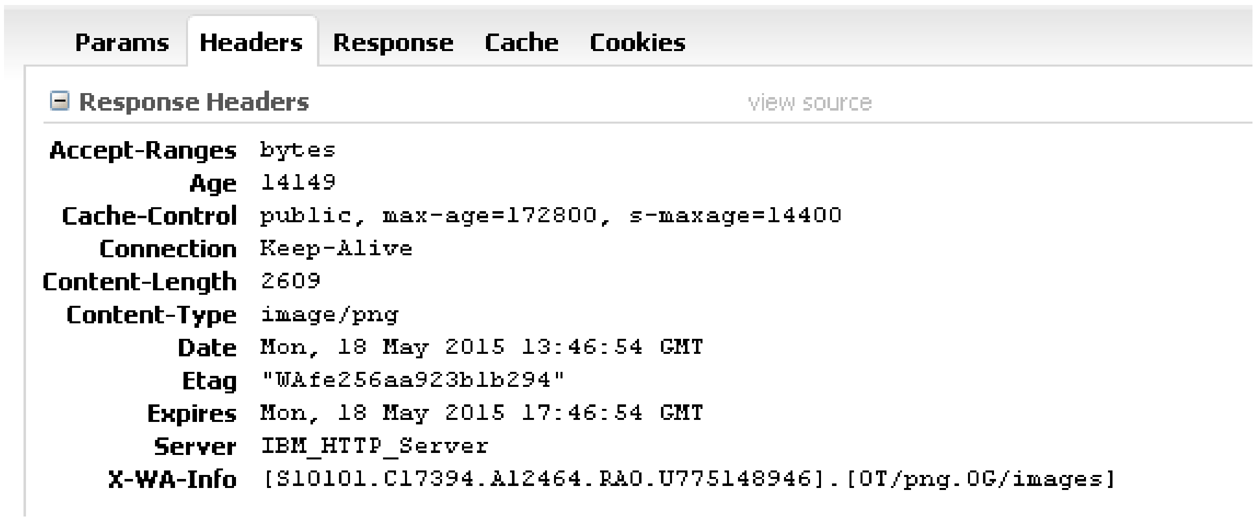
IBM
© 2015
Event Home

Performance

It is important to set the proper caching when using WCM in the theme to avoid constantly generating the content. Especially triggering the WCM API from every execution should be avoided if possible. The graph below shows the caches that can be leveraged when rendering WCM content.


Browser/ Mobile App
Proxy/ HTTP Server
WebSphere Application Server
WCM API
Browser / Mobile App cache
Proxy/ Apache cache
Servlet cache
WCM Advanced/ Basic cache

The Browser and Proxy / HTTP Server will cache the markup for URLs that have a cache header that allows caching. The relevant headers are Cache-Control and the Expires header.



```
 Accept-Ranges  bytes
           Age  14149
 Cache-Control  public, max-age=172800, s-maxage=14400
    Connection  Keep-Alive
Content-Length  2609
  Content-Type  image/png
          Date  Mon, 18 May 2015 13:46:54 GMT
          Etag  "WAfe256aa923b1b294"
       Expires  Mon, 18 May 2015 17:46:54 GMT
        Server  IBM_HTTP_Server
   X-WA-Info  [S10101.C17394.A12464.RA0.U775148946].[OT/png.OG/images]
```

The headers can be influenced by the WCM configuration as well as by using HTTP server features to modify the headers. The servlet cache stores static resources and WCM servlet calls while the Advanced and Basic WCM cache are effective the API level when leveraging the WCM java or JSP tag API.

The details on how to implement the tuning are covered within the Portal performance tuning guide: http://www-10.lotus.com/ldd/portalwiki.nsf/dx/IBM_WebSphere_Portal_V_8.5_Performance_Tuning_Guide

API Performance

When leveraging the WCM java or servlet API we recommend to consider the following:

- Creating a WCM workspace via API is expensive. Therefore try to cache the output of the JSP via Dynacache or other Java means. When creating the cache key think about points of variability though – e.g. you might want to render different content for a mobile device.

- When creating your own JSPs try to avoid creating a session – especially for anonymous users to reduce the overhead of servlet sessions if possible.

- Unless access checks for the current user are really required (e.g. because different theme content should be rendered based on the logged in user) try to bypass the access control checks with run as system.

Additional Information

- **IBM Digital Experience wiki Developing Themes for WebSphere Portal** http://www-10.lotus.com/ldd/portalwiki.nsf/xpViewCategories.xsp?lookupName=Developing%20Themes%20for%20WebSphere%20Portal
- **IBM Knowledge Center - Developing themes and skins**

  http://www-01.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_themes.dita
- **IBM Knowledge Center - Web Content Management API**

  http://www-01.ibm.com/support/knowledgecenter/SSYJ99_8.5.0/wcm/wcm_dev_api.dita