

**How-to Guide  
SAP NetWeaver 7.0**



# **How To... Create Page Layouts for Web Page Composer**

**Version 1.00 – November 2007**

**Applicable Releases:  
SAP NetWeaver 7.0  
User Productivity Enablement  
Enterprise Knowledge Management**

© Copyright 2007 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

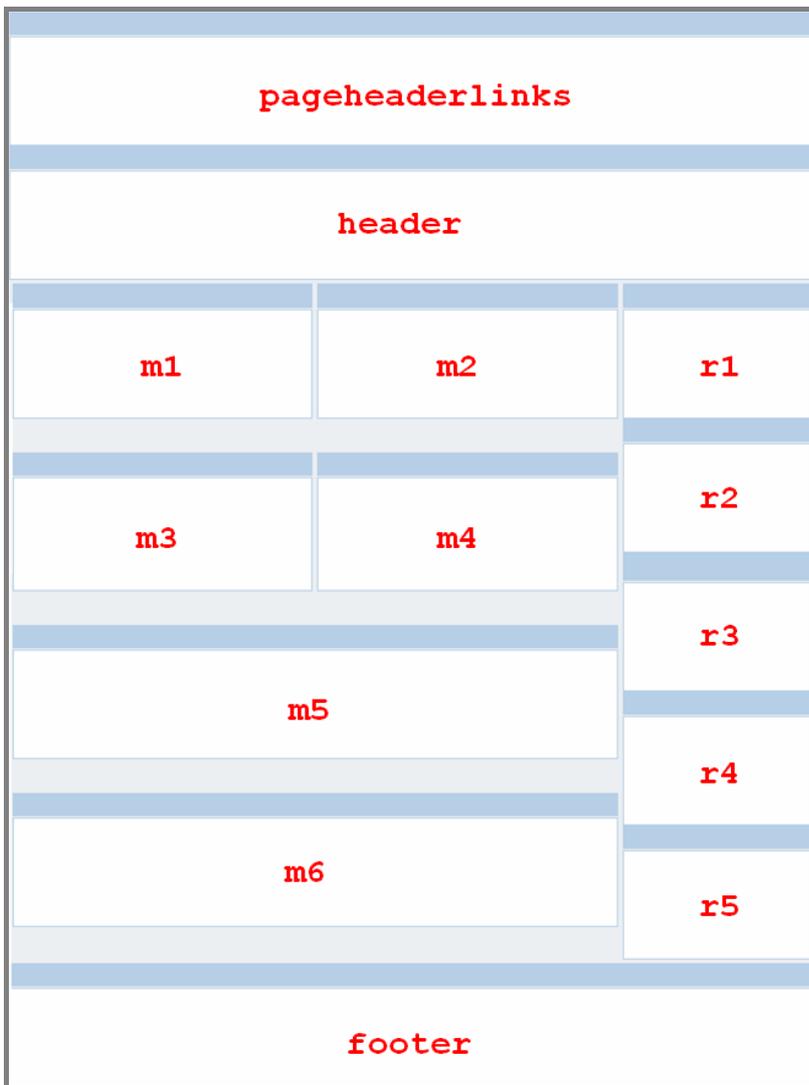
Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

# 1 Scenario

## 1.1 Overview

If the default page layouts for Web Page Composer (WPC) do not exactly meet your requirements, you can create your own page layouts and deploy them to your portal.

This guide explains how to create, deploy and configure a new page layout for Web Page Composer. In order to do so we use the following page layout called "Topic Page Layout" as an example:



**Remark:**

The "pageheaderlinks" container is reserved for the standard WPC iView that contains the page toolbar commands (e.g. RSS feed, Add to favorites).

Example page layout: "Topic Page Layout"

It is essential to assign a unique name (ID) to each content area (WPC terminology: container) of the page layout. Thus, choose names that you can easily associate with the corresponding container. For example choose "header" for your header area, "footer" for your footer area, the prefix "r" for areas on the right-hand side and so on.

## 1.2 Prerequisites

Before you go on please ensure that the following prerequisites are fulfilled:

- SAP NetWeaver 7.0 (usage type Enterprise Portal) with support package stack 13
- Web Page Composer add-on (installation see SAP Note 1080110)
- You have the Content Administrator and the System Administrator Role
- You have the WPC Editor Role (required for testing)
- SAP NetWeaver Developer Studio (corresponding version for your SAP NetWeaver Portal)

## 2 Procedure

You have to complete the following steps in order to create a new page layout for Web Page Composer:

1. Build the page layout(s) in SAP NetWeaver Developer Studio
2. Create the layout objects in the Portal Content Studio
3. Update the Web Page Composer configuration
4. Link containers to container iViews

### 2.1 Building the page layout in SAP NetWeaver Developer Studio

To build a new page layout for Web Page Composer, you create a new project in the SAP NetWeaver Developer Studio and define the page layout as Java Server Page (JSP file) using the tag library for Web Page Composer layouts.

**Remark:** Of course, you may create more than one page layout per project and deploy these layouts together as one PAR file.

## Procedure

### 2.1.1 Create a Project in SAP NetWeaver Developer Studio

1. Start SAP NetWeaver Developer Studio.
2. Create a *New Project* using the *Portal Application* template.

### 2.1.2 Define the Page Layout as Java Server Page

The exemplary layout used in this how-to guide is based on standard HTML tables to keep our example as simple as possible. Because of increased flexibility and advanced design options we recommend to use mainly CSS-based layouts instead. Within the table cells of our JSP file we use the `<wpc:container>` tag with unique IDs (e.g. header) to define the designated containers for web content.

1. In your project, create a JSP file in the folder `/dist/PORTAL-INF/jsp` and name it `"wpc_topic_page_layout.jsp"`.
2. Create the exemplary page layout by using the taglib directive to specify the required libraries (TagLibLayout, WpcTagLibLayout) and defining the detailed layout as Java Server Page:

```
<%@ taglib uri="prt:taglib:com.sap.portal.reserved.layout.WpcTagLibLayout" prefix="wpc"%>
<%@ taglib uri="prt:taglib:com.sap.portal.reserved.layout.TagLibLayout" prefix="lyt" %>
<lyt:template>

<div id="wpc_casheader"><wpc:container id="pageheaderlinks" /></div>
<div><wpc:container id="header" /></div>

<table width="100%">
<tbody valign="top">

<tr>
<td width="38%">
<wpc:container id="m1" />
</td>

<td width="38%">
<wpc:container id="m2" />
</td>

<td rowspan="4" width="24%">
<div><wpc:container id="r1" /></div>
<div><wpc:container id="r2" /></div>
```

```

<div><wpc:container id="r3" /></div>
<div><wpc:container id="r4" /></div>
<div><wpc:container id="r5" /></div>
</td>
</tr>

<tr>
<td width="38%">
<wpc:container id="m3" />
</td>

<td width="38%">
<wpc:container id="m4" />
</td>
</tr>

<tr>
<td colspan="2" width="76%">
<wpc:container id="m5" />
</td>
</tr>

<tr>
<td colspan="2" width="76%">
<wpc:container id="m6" />
</td>
</tr>

</tbody>
</table>

<div><wpc:container id="footer" /></div>
</lyt:template>

```

**Note:** The **highlighted terms** assign unique names (IDs) to each content area (container) of the page layout. (See [layout view at the beginning](#) of this How-To Guide)

### 2.1.3 Update the portalapp.xml File

The `portalapp.xml` file is the *deployment descriptor* for portal applications. In the case of our layout project this file describes all the layout components and their properties.

1. Open the `portalapp.xml` file of your project for editing (`/dist/PORTAL-INF` folder).
2. Use this file to configure your new layout components and their properties for all containers. For our layout example we only need to describe one new component called `"wpc_topic_page_layout"`:

```

<?xml version="1.0" encoding="UTF-8"?>
<application>

  <application-config>
    <property name="SharingReference"
value="com.sap.nw.wpc.page.templates,com.sap.portal.pagebuilder"/>
  </application-config>

  <components>

    <component name="wpc_topic_page_layout">
      <component-config>
        <property name="ClassName" value="com.sapportals.portal.pb.layout.PageLayout"/>
        <property name="ResourceBundleName" value="pagebuilder_nls"/>
        <property name="SecurityZone" value="com.sap.portal/no_safety"/>
      </component-config>

      <component-profile>
        <property name="ComponentType" value="com.sapportals.portal.layout"/>
        <property name="com.sap.portal.pcm.Title" value="Topic Page Layout"/>
        <property name="com.sap.portal.pcm.Description" value=""/>
        <property name="com.sap.portal.reserved.layout.TagLibLayout"
value="/SERVICE/com.sap.portal.pagebuilder/taglib/layout.tld"/>
      </component-profile>
    </component>
  </components>
</application>

```

```

    <property name="com.sap.portal.reserved.layout.WpcTagLibLayout"
value="/SERVICE/com.sap.nw.wpc.page.templates/taglib/wpclayout.tld"/>
    <property name="com.sap.portal.reserved.layout.TemplateFile"
value="wpc_topic_page_layout.jsp"/>
    <property name="AuthScheme" value="anonymous"/>
    <property name="EPCFLevel" value="0"/>

    <property name="com.sap.portal.reserved.layout.Cont1" value="pageheaderlinks">
    <property name="title" value="pageheaderlinks"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont2" value="header">
    <property name="title" value="header"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont3" value="m1">
    <property name="title" value="m1"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont4" value="m2">
    <property name="title" value="m2"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont5" value="m3">
    <property name="title" value="m3"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont6" value="m4">
    <property name="title" value="m4"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont7" value="m5">
    <property name="title" value="m5"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont8" value="m6">
    <property name="title" value="m6"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont9" value="r1">
    <property name="title" value="r1"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont10" value="r2">
    <property name="title" value="r2"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont11" value="r3">
    <property name="title" value="r3"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont12" value="r4">
    <property name="title" value="r4"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont13" value="r5">
    <property name="title" value="r5"/>
    <property name="orientation" value="vertical"/>
    </property>

    <property name="com.sap.portal.reserved.layout.Cont14" value="footer">
    <property name="title" value="footer"/>
    <property name="orientation" value="vertical"/>
    </property>

```

```
        </component-profile>
    </component>

</components>
<services/>
</application>
```

### Important note:

1. For each new layout you have to declare a new component in the `<components>` section of the `portalapp.xml` file.
2. The name of the JSP file within your project must match with the value of the `TemplateFile` property of your component profile in the `portalapp.xml` file (here: `"wpc_topic_page_layout.jsp"`).
3. The names of the content areas (container IDs) in the JSP file must match with the values of the corresponding container properties within the component profile of the `portalapp.xml` file. (Here: `"pageheaderlinks"`, `"header"`, `"m1"` and so on)

## 2.1.4 Export the Project and Upload it to the Portal

1. Use the Export as PAR File command to build and pack your layout project.
2. Upload the PAR file to your SAP NetWeaver portal: Logon to your portal as System Administrator, choose *System Administration* → *Support* → *Portal Runtime* → *Administration Console*. Select the PAR file in the input field below *Upload Archive* and hit the *Upload* button.

## Result

You have created your new page layout(s), built and packed your layout project using SAP NetWeaver Developer Studio and uploaded the resulting PAR file to your SAP NetWeaver portal.

Now proceed with the steps described in section [Creating the layout objects in the Portal Content Studio](#).

## 2.2 Creating the layout objects in the Portal Content Studio

To be able to use a new page layout for Web Page Composer, you must create the following objects in the *Portal Content Directory* using the *Portal Content Studio*:

- Page Layout Template
- Container iViews
- Page Layout

### Procedure

Logon to your SAP NetWeaver Portal as Content Administrator and navigate to the *Portal Content Studio: Content Administration* → *Portal Content*.

#### 2.2.1 Create a New Folder

In the Portal Content Directory, create a new folder for your custom Web Page Composer content:

1. Use the secondary mouse button to choose the root folder *Portal Content*. In the context menu, choose *New* → *Folder*.
2. Specify the default properties `Folder Name` and `Folder ID` (in our example: *WPC Customer Folder*). Check your entries and choose *Finish*.

#### 2.2.2 Set the Folder Permissions

To make your new page layout usable at runtime it is essential to change the permissions of your folder. Thus, all child objects in this folder will automatically inherit these permissions.

1. Use the secondary mouse button to choose the folder you just have created. In the context menu, choose *Open* → *Permissions*.
2. Now search in groups for “everyone”.



3. Select the group *everyone* and click add.



4. For the *everyone* group set `None` as Administrator and activate the End User check box.

Assigned Permissions					
Remove					
	Name	Administrator	End User	Role Assigner	Description
<input type="checkbox"/>	 Everyone	None	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Built-in Group Everyone

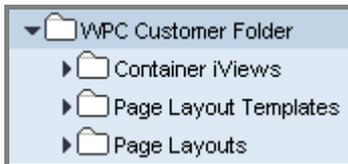
- To save the permissions, choose *Save*, than choose *Close* to close the Permission Editor.

### 2.2.3 Create Subfolders

In this new folder create subfolders for storing Container iViews, Page Layout Templates and Page Layouts.

- Use the secondary mouse button to choose the new folder you previously created. In the context menu, choose *New* → *Folder*.
- Create three new subfolders and name the first one “*Container iViews*”, the second one “*Page Layout Templates*” and the third one “*Page Layouts*”.

By now you have created the following folder structure:

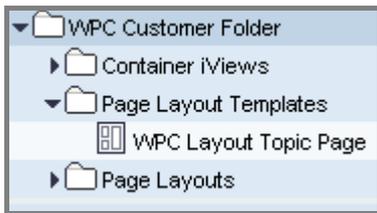


### 2.2.4 Create Layout Templates

In the Portal Content Studio, create a new layout template for each JSP layout file that is part of your layout project (PAR file). In our example, we create only one new layout template.

- Use the secondary mouse button to choose the folder *Portal Content* → “*Your\_New\_Folder*” → *Page Layout Templates*. In the context menu, choose *New* → *Layout*.
- Select the portal archive that you previously created in the SAP NetWeaver Developer Studio and that contains the new page layout. Choose *Next*. The system displays all page layouts that are defined in the `portalapp.xml` file as `<component>`.
- Select your new page layout. Choose *Next*.
- Specify the default properties `Layout Name` and `Layout ID`.  
`Layout Name`: “*WPC Layout Topic Page*”  
`Layout ID`: “*wpc\_layout\_topic\_page*”
- Check your entries and choose *Finish*.
- Choose *Open the object for editing* and then choose *OK*. Check the `Object` is a `Template` property.
- To save your layout template, choose *Save*.

By now you should see the following PCD objects:



## 2.2.5 Create Container iViews (optional)

The WPC container iViews are used to hold the content on a specific area of a page. You have two options:

- Create new container iViews from scratch

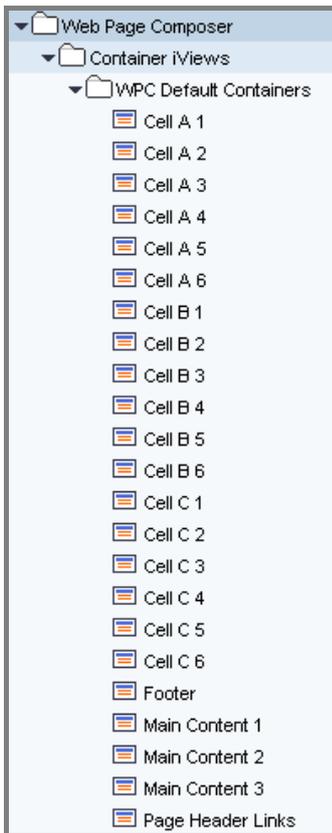
For every new container iView perform the following steps:

1. Navigate to the folder *Portal Content* → “*Your\_New\_Folder*” → *Container iViews*.
2. Use the secondary mouse button to click on the folder. Choose *New* → *iView* from the context menu.
3. Choose *iView Template*. Choose *Next*.
4. Select the *WPC Container template* iView template. Choose *Next*.
5. Specify the default properties *Name* and *iView ID*. Choose *Next*.  
 Enter a name that it is easy to associate with the corresponding page area.
6. Check your entries and choose *Finish*.
7. The following step is necessary for a successful indexing process: Choose *Open the object for editing* and then choose *OK*. Switch to the *Availability* property category. In the *Supported User Agents* property, make sure that no entry is selected.
8. To save the iView, choose *Save*.

- Re-use the existing container iViews

You can reuse the existing container iViews which are assigned to the page layouts that are already available. **Reusing existing container iViews has the advantage that it enables you to keep the assigned content of each container within a page even after changing the page layout via the page editor.** Thus, we strongly recommend to use this option.

You find the default container iViews in the following folder: *Portal Content* → *Web Page Composer* → *Container iViews* → *WPC Default Containers*. (See following Screenshot)

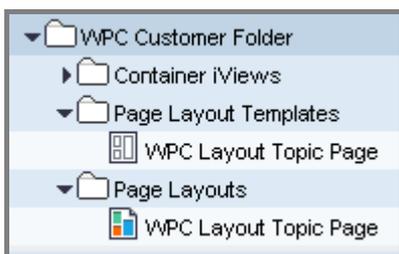


## 2.2.6 Create a Page Layout and Assign the Container iViews

Create a WPC page layout (technically a portal page) for each layout template.

1. Use the secondary mouse button to choose the folder *Portal Content* → “*Your\_New\_Folder*” → *Page Layouts*. In the context menu, choose *New* → *Page*.
2. Specify the default properties Name and ID and choose *Next*.  
 Layout Name: “*WPC Layout Topic Page*”  
 Layout ID: “*wpc\_layout\_topic\_page*”
3. Choose the *Default Page Template* template. Choose *Next*.
4. Select the layout template you created previously and choose *Add*. Choose *Next*. Check your entries and choose *Finish*.
5. Choose *Open the object for editing* and then choose *OK*.
6. Go to the *Availability* property category in the property editor. In the *Supported User Agents* property, make sure that no entry is selected.
7. To save your entries, choose *Save*.

By now you should see the following objects:



Assign the container iViews to the WPC page layout (portal page).

 If you re-use the existing container iViews like in this example it is essential to arrange these in a reasonable way: For example, if a container is already used on an existing page layout the position on the new page layout should be logically equivalent.

1. Use the secondary mouse button to select one of the iViews in the folder *Container iViews* (either your custom folder with the new container iViews or the standard *WPC Default Container* folder).
2. In the context menu, choose *Add iView to Page* → *Delta Link* while your target page layout (portal page) is still open in the Portal Content Studio.
3. Repeat these steps until all container iViews have been assigned to your new WPC page layout (portal page).
4. If you want to provide the page header toolbar (RSS feed, print-optimized version, etc.) on each page you also have to assign the *Page Header Links* iView once to your WPC page layout.

 You can also assign other iViews to your WPC page layout that should be part of every page based on this layout. For example this enables very useful scenarios like the integration of a company-wide footer or header that can't be modified in the page editor but is managed centrally.

In the following screenshot you see the iViews which have been assigned to our WPC page layout by now (the default *Page Header Links* iView and 14 container iViews):

	Type	Name	ID	Visible	Fixed
<input type="checkbox"/>		Page Header Links	pageheader	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Main Content 1	wpc_iview_main1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Footer	wpc_iview_footer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell A 1	wpc_iview_a1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell A 2	wpc_iview_a2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell A 3	wpc_iview_a3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell A 4	wpc_iview_a4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell B 1	wpc_iview_b1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell B 2	wpc_iview_b2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell C 1	wpc_iview_c1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell C 2	wpc_iview_c2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell C 3	wpc_iviews_c3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell C 4	wpc_iviews_c4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		Cell C 5	wpc_iview_c5	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5. Switch to the *Page Layout* mode. Use *Drag&Drop* to move the container iViews to the respective positions in the WPC page layout.

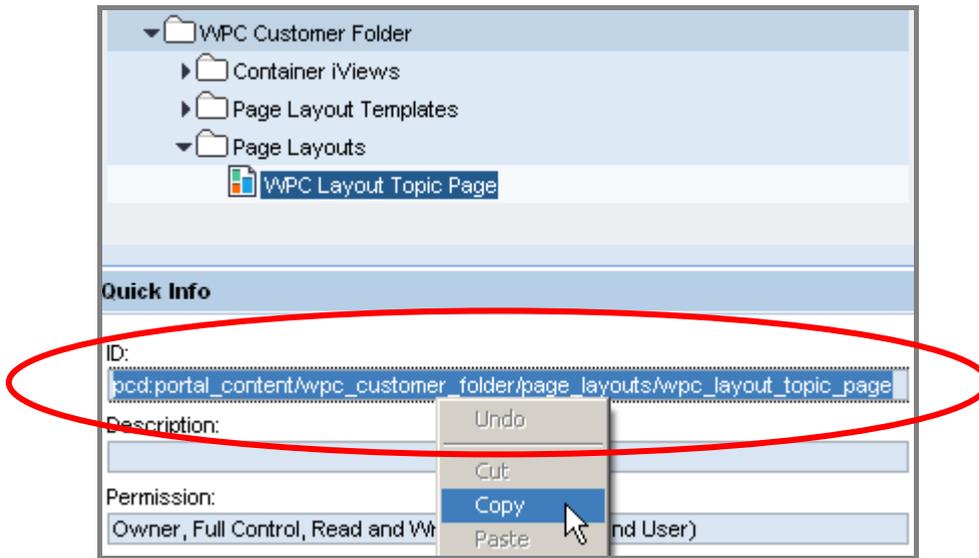
In the following screenshot you can see how the iViews have been arranged for our example page layout:



6. To save the page, choose *Save*.

Copy the ID of the new page layout to your clipboard or into a text editor.

1. Click on your new Page Layout in the folder *Portal Content* → “*Your\_New\_Folder*” → *Page Layouts*.
2. Mark and copy the ID of your WPC page layout in the *Quick Info* area (see following screenshot).



## Result

You have created all the required layout objects (layout template, container iViews, page layout) in the Portal Content Studio. You can now go on with the section [Updating the Web Page Composer configuration](#).

## 2.3 Updating the Web Page Composer configuration

### Use

You have to update the configuration settings for the layout objects of Web Page Composer accordingly.

### Procedure

Logon to the portal as System Administrator and launch the configuration user interface for Knowledge Management: *System Administration* → *System Configuration* → *Knowledge Management* → *Content Management*.

#### 2.3.1 Create Layout Template Configuration Object(s)

For each page layout create a new layout template configuration object.

1. In the KM configuration, choose *Web Page Composer* → *Layout* → *Layout Templates* → *Layout Templates*.
2. Choose *New* to create a new layout template.
3. Specify a name and enter a key for description.

New "Layout Templates"	
Name: *	wpc_layout_template_topic_page
Bundle File:	com.sap.nw.wpc.bundles.Commands
Key for Description:	Topic Page

4. Enter the ID for the page layout that you copied from the Portal Content Studio in the `PCD Object ID` parameter. This step maps your PCD layout object to the corresponding WPC configuration object.
5. Specify the `Thumbnail Image Path` in order to see a preview icon of the page layout in the Layout Manager of the WPC (see note below).

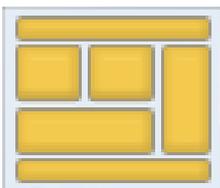
Thumbnail Image Path in our example:

`/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/etc/public/mimes/images/topic_page_weblayout.gif`

6. To save the new layout template, choose *OK*.

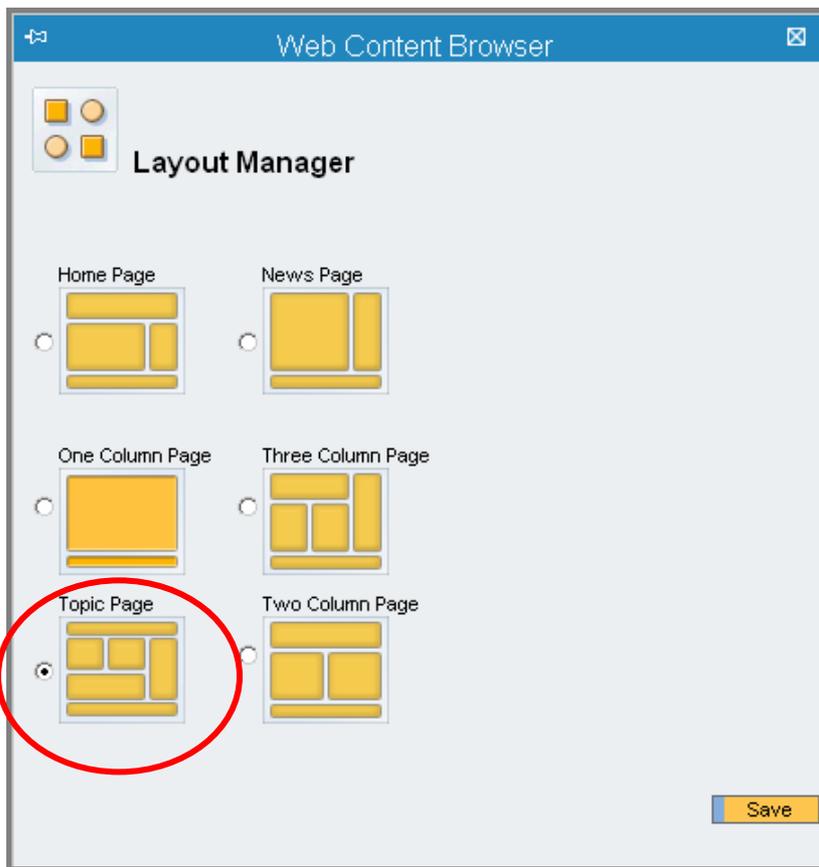
#### Note for preview images:

1. Create a preview image for your page layout in the Graphics Interchange Format (GIF) with the following dimensions: 70px to 60px.
2. Upload your preview image (GIF file) to the following folder of the *etc* repository: `/etc/public/mimes/images`.



Exemplary preview image: "topic\_page\_weblayout.gif"

3. After uploading the image and specifying the `Thumbnail Image Path` the appropriate icon will be displayed in the Layout Manager of Web Page Composer (see screenshot below).



### 2.3.2 Extend the existing Layout Group

To display the new page layout(s) in the list of available layouts in the page editor of Web Page Composer (*change page layout* command), you must assign your new page layout to the default layout group for web pages. This group contains all available layouts for web pages.

1. In the KM configuration, choose *Web Page Composer* → *Layout* → *Layout Groups* → *Layout Groups*.
2. Open the *wpc\_page\_group* configuration object for editing and add your new page layout(s) to this layout group. Save your changes by choosing **OK**.

### 2.3.3 Create Container Configuration Objects

Create a container configuration object for each new container iView (in other words: for each container iView that doesn't already exist) and assign the resource types and MIME types that you want to allow as valid content type in this container (rule-based design).

 If in various page areas exactly the same resource types and MIME types should be allowed it is sufficient to create one container configuration object representing all the homogeneous areas (container iViews).

1. In the KM configuration, navigate to *Web Page Composer* → *Containers* → *Containers*.
2. Choose *New* to create a container configuration object for each area (container iView) of your page layout and allow the appropriate resource types and MIME types according to your rules.
3. Note the names of your new container configuration objects or copy them to a text editor. You will need these names to map them to the corresponding container iViews.

**Important remark:** Don't create new container configuration objects if you re-use the existing containers.

## Result

You have worked through all configuration steps to integrate a new page layout in the layout manager of Web Page Composer. Continue with the section [Linking Containers to Container iViews](#).

## 2.4 Linking Containers to Container iViews

### Use

Finally, you have to map the container configuration objects to the corresponding container iViews in the Portal Content Directory.

**Important remark:** This is only necessary if you created additional container iViews and container configuration objects. If you re-use the existing container iViews and container configuration objects for your new page layout(s) you can skip this step.

### Procedure

Open each container iView in the Portal Content Directory and specify the name of the corresponding container configuration object as *Container Type* property (property category: Web Page Composer).

#### 2.4.1 Specify the Container Type of the Container iViews

1. Logon to your portal as Content Administrator and navigate to the Portal Content Studio: *Content Administration* → *Portal Content*.
2. In the tree view, choose *Portal Content* → *Web Page Composer* → *Container iViews*. Now open the folder in which the iViews for your layout template are stored.
3. Open each container iView by using the secondary mouse button to select the iView and choose *Open* → *Object* from the context menu.
4. Use the property filter by choosing the property category *Web Page Composer*.
5. Enter the name of the corresponding container configuration object from the Web Page Composer configuration as *Container Type* property.  
 **Again:** If one container configuration object is valid for multiple container iViews (same MIME and resource types allowed) it is sufficient to map this configuration object to multiple container iViews.
6. Save your changed iView properties.

## 3 Result

Now you can create new web pages based on your new page layout(s). You can switch to your new page layout(s) by using the *Change Page Layout* command in the page editor menu. This opens the layout manager and displays all available layouts for web pages. Your new page layout(s) should be offered there.

[www.sdn.sap.com/irj/sdn/howtoguides](http://www.sdn.sap.com/irj/sdn/howtoguides)