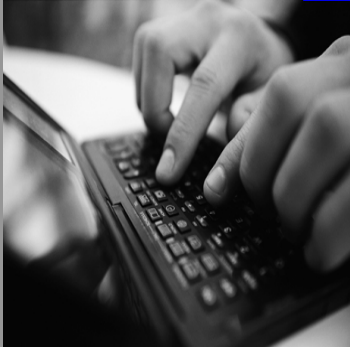


**How-to Guide
SAP NetWeaver 7.0**



How To... Create Section- Based Page Layouts for Web Page Composer

Version 1.10 – January 2008

**Applicable Releases:
SAP NetWeaver 7.0 ≥ SPS 13
User Productivity Enablement
Enterprise Knowledge Management**

© Copyright 2008 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®, 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.

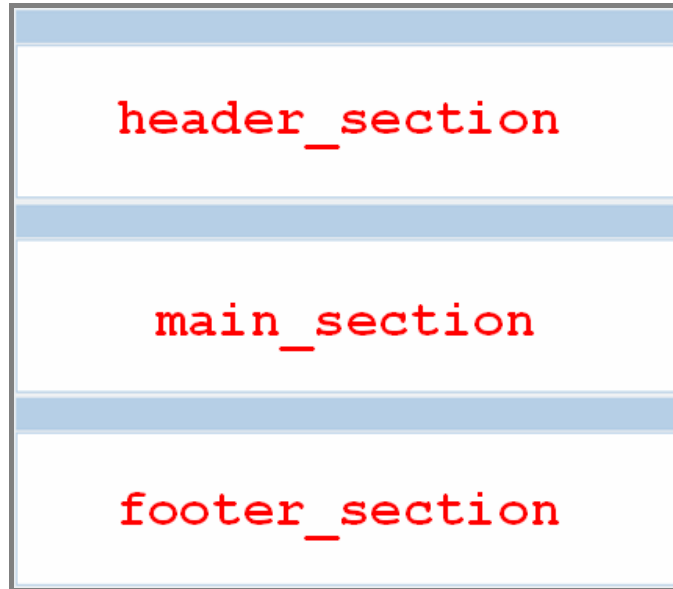
Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the 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.

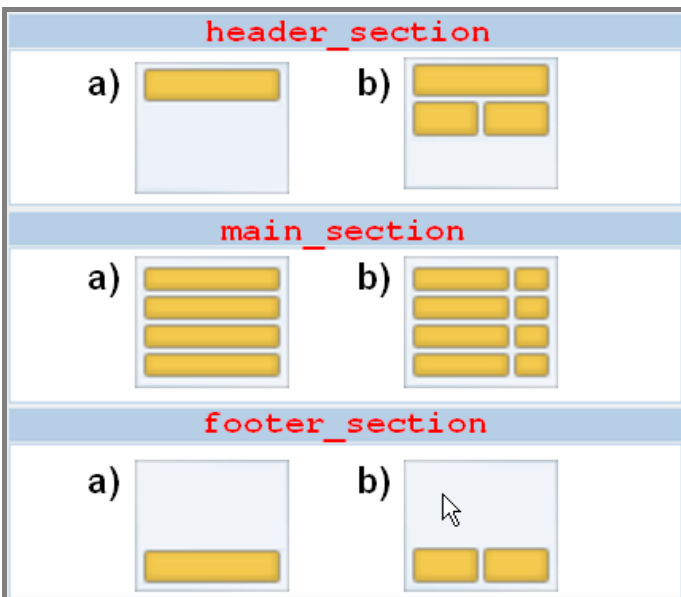
This guide explains how to create, deploy and configure a section-based page layout for Web Page Composer. In order to do so we use the following page layout called “Demo Page Layout” as an example. As indicated below the basic structure of our exemplary page layout consists of three horizontally arranged sections:



New section-based page layout: “Demo 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_section**” for your header area, “**main_section**” for your main area and “**footer_section**” for your footer area.

For every section of our “Demopage Layout” we will create two different sub-layouts:



Section-based page layouts offer you a lot of flexibility. In order to simplify and shorten this How-to Guide we limited the number of sub-layouts to two per section. Of course, it is possible to increase the number of sub-layouts per section if this meets your requirements.

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 or higher
- 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 layouts 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 main page layout and its sub-layouts as Java Server Pages (JSP files) 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

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

2.1.2 Define the Page Layout and the Section Layouts as Java Server Pages

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 cells of our layout (JSP file) we use the `<wpc:container>` tag with unique IDs (e.g. header) to define the containers for the sections.

- ❏ If you want to create CSS-based layouts or use style definitions in your layout definition, we recommend to keep them in a separate stylesheet which should be stored in the `/dist/css` folder of your project. This way your style definitions can be shared among all your layout definitions of this project by referencing this stylesheet in each JSP file, e.g.:

```
<%@ page import="com.sapportals.portal.prt.component.IPortalComponentResponse" %>
<%@ page import="com.sapportals.portal.prt.resource.IResource" %>

<% IPortalComponentResponse componentResponse =
  (IPortalComponentResponse)pageContext.getAttribute(javax.servlet.jsp.PageContext.RESPONSE);
  IResource yourCss = componentRequest.getResource(IResource.CSS, "css/your.css");
  componentResponse.include(componentRequest, yourCss); %>
```

1. In your project, create a JSP file in the folder `/dist/PORTAL-INF/jsp` and name it `"wpc_demopage_layout.jsp"`.
2. Create the main 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>

<table width="100%">
<tbody>
<tr>
<td>
<wpc:container id="header_section" />
</td>
</tr>

<tr>
<td>
<wpc:container id="main_section" />
</td>
</tr>

<tr>
<td>
<wpc:container id="footer_section" />
</td>
</tr>

</tbody>
</table>
</lyt:template>
```

Note: The **highlighted terms** assign unique names (IDs) to each section of the main page layout.

- Now create the sub-layouts (JSP files) for each section. In this demo project we create two alternative sub-layouts per section.

a) Sub-layouts for the header section:

wpc_header_section_layout1.jsp

```
<%@ 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>

<table width="100%">

<tbody>

<tr>
<td>
<wpc:container id="h1" />
</td>
</tr>

</tbody>
</table>

</lyt:template>
```

Layout preview:



wpc_header_section_layout2.jsp

```
<%@ 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>

<table width="100%">

<tbody>

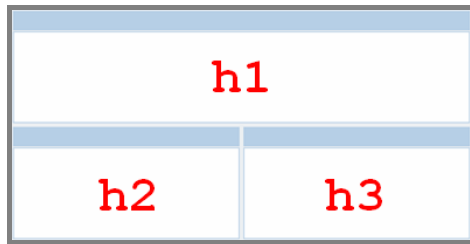
<tr>
<td colspan="2" width="100%">
<wpc:container id="h1" />
</td>
</tr>

<tr>
<td width="50%">
<wpc:container id="h2" />
</td>
<td width="50%">
<wpc:container id="h3" />
</td>
</tr>

</tbody>
</table>

</lyt:template>
```

Layout preview:



b) Sub-layouts for the main section:

wpc_main_section_layout1.jsp

```
<%@ 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>

<table width="100%">

<tbody>

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

<tr>
<td>
<wpc:container id="m2" />
</td>
</tr>

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

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

</tbody>
</table>
</lyt:template>
```

Layout preview:

m1
m2
m3
m4

wpc_main_section_layout2.jsp

```
<%@ 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>

<table width="100%">

<tbody>

<tr>

<td width="75%">
<wpc:container id="m1" />
</td>

<td width="25%">
<wpc:container id="m5" />
</td>

</tr>

<tr>

<td width="75%">
<wpc:container id="m2" />
</td>

<td width="25%">
<wpc:container id="m6" />
</td>

</tr>

<tr>

<td width="75%">
<wpc:container id="m3" />
</td>

<td width="25%">
<wpc:container id="m7" />
</td>
```



```

</tr>

<tr>
<td width="75%">
<wpc:container id="m4" />
</td>

<td width="25%">
<wpc:container id="m8" />
</td>

</tr>

</tbody>
</table>
</lyt:template>

```

Layout preview:

m1	m5
m2	m6
m3	m7
m4	m8

c) Sub-layouts for the footer section:

wpc_footer_section_layout1.jsp

```

<%@ 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>

<table width="100%">

<tbody>

<tr>
<td>
<wpc:container id="f1" />
</td>
</tr>

</tbody>
</table>
</lyt:template>

```

Layout preview:



wpc_footer_section_layout2.jsp

```
<%@ 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>

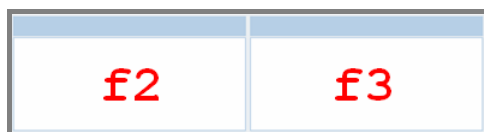
<table width="100%">

<tbody>

<tr>
<td width="50%">
<wpc:container id="f2" />
</td>
<td width="50%">
<wpc:container id="f3" />
</td>
</tr>

</tbody>
</table>
</lyt:template>
```

Layout preview:



2.1.3 Create a 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 our layout example we have to describe the properties of the main layout component and of each section component:

```
<?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_demopage_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="Demopage Layout"/>
      <property name="com.sap.portal.pcm.Description" value=""/>
    </component-profile>
  </component>
</components>
```

```

        <property name="com.sap.portal.reserved.layout.TagLibLayout"
value="/SERVICE/com.sap.portal.pagebuilder/taglib/layout.tld"/>
        <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_demopage_layout.jsp"/>
        <property name="AuthScheme" value="anonymous"/>
        <property name="EPCFLevel" value="0"/>
        <property name="com.sap.portal.reserved.layout.Cont1" value="header_section">
            <property name="title" value="Header Section"/>
            <property name="orientation" value="vertical"/>
        </property>
        <property name="com.sap.portal.reserved.layout.Cont2" value="main_section">
            <property name="title" value="Main Section"/>
            <property name="orientation" value="vertical"/>
        </property>
        <property name="com.sap.portal.reserved.layout.Cont3" value="footer_section">
            <property name="title" value="Footer Section"/>
            <property name="orientation" value="vertical"/>
        </property>
    </component-profile>
</component>

<component name="wpc_footer_section_layout1">
    <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="Footer Section Layout 1"/>
        <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"/>
        <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_footer_section_layout1.jsp"/>
        <property name="AuthScheme" value="anonymous"/>
        <property name="EPCFLevel" value="0"/>
        <property name="com.sap.portal.reserved.layout.Cont1" value="f1">
            <property name="title" value="f1"/>
            <property name="orientation" value="vertical"/>
        </property>
    </component-profile>
</component>

<component name="wpc_footer_section_layout2">
    <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="Footer Section Layout 2"/>
        <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"/>
        <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_footer_section_layout2.jsp"/>
        <property name="AuthScheme" value="anonymous"/>
        <property name="EPCFLevel" value="0"/>
        <property name="com.sap.portal.reserved.layout.Cont1" value="f2">
            <property name="title" value="f2"/>
            <property name="orientation" value="vertical"/>
        </property>
        <property name="com.sap.portal.reserved.layout.Cont2" value="f3">
            <property name="title" value="f3"/>
            <property name="orientation" value="vertical"/>
        </property>
    </component-profile>
</component>

```

```

<component name="wpc_header_section_layout1">
  <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="Header Section Layout 1"/>
    <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"/>
    <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_header_section_layout1.jsp"/>
    <property name="AuthScheme" value="anonymous"/>
    <property name="EPCFLevel" value="0"/>
    <property name="com.sap.portal.reserved.layout.Cont1" value="h1">
      <property name="title" value="h1"/>
      <property name="orientation" value="vertical"/>
    </property>
  </component-profile>
</component>

<component name="wpc_header_section_layout2">
  <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="Header Section Layout 2"/>
    <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"/>
    <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_header_section_layout2.jsp"/>
    <property name="AuthScheme" value="anonymous"/>
    <property name="EPCFLevel" value="0"/>
    <property name="com.sap.portal.reserved.layout.Cont1" value="h1">
      <property name="title" value="h1"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont2" value="h2">
      <property name="title" value="h2"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont3" value="h3">
      <property name="title" value="h3"/>
      <property name="orientation" value="vertical"/>
    </property>
  </component-profile>
</component>

<component name="wpc_main_section_layout1">
  <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="Main Section Layout 1"/>
    <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"/>
    <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_main_section_layout1.jsp"/>
  </component-profile>
</component>

```

```

<property name="AuthScheme" value="anonymous"/>
<property name="EPCFLevel" value="0"/>
<property name="com.sap.portal.reserved.layout.Cont1" value="m1">
  <property name="title" value="m1"/>
  <property name="orientation" value="vertical"/>
</property>
<property name="com.sap.portal.reserved.layout.Cont2" value="m2">
  <property name="title" value="m2"/>
  <property name="orientation" value="vertical"/>
</property>
<property name="com.sap.portal.reserved.layout.Cont3" value="m3">
  <property name="title" value="m3"/>
  <property name="orientation" value="vertical"/>
</property>
<property name="com.sap.portal.reserved.layout.Cont4" value="m4">
  <property name="title" value="m4"/>
  <property name="orientation" value="vertical"/>
</property>
</component-profile>
</component>

<component name="wpc_main_section_layout2">
  <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="Main Section Layout 2"/>
    <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"/>
    <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_main_section_layout2.jsp"/>
    <property name="AuthScheme" value="anonymous"/>
    <property name="EPCFLevel" value="0"/>
    <property name="com.sap.portal.reserved.layout.Cont1" value="m1">
      <property name="title" value="m1"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont2" value="m2">
      <property name="title" value="m2"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont3" value="m3">
      <property name="title" value="m3"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont4" value="m4">
      <property name="title" value="m4"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont5" value="m5">
      <property name="title" value="m5"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont6" value="m6">
      <property name="title" value="m6"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont7" value="m7">
      <property name="title" value="m7"/>
      <property name="orientation" value="vertical"/>
    </property>
    <property name="com.sap.portal.reserved.layout.Cont8" value="m8">
      <property name="title" value="m8"/>
      <property name="orientation" value="vertical"/>
    </property>
  </component-profile>
</component>
</components>

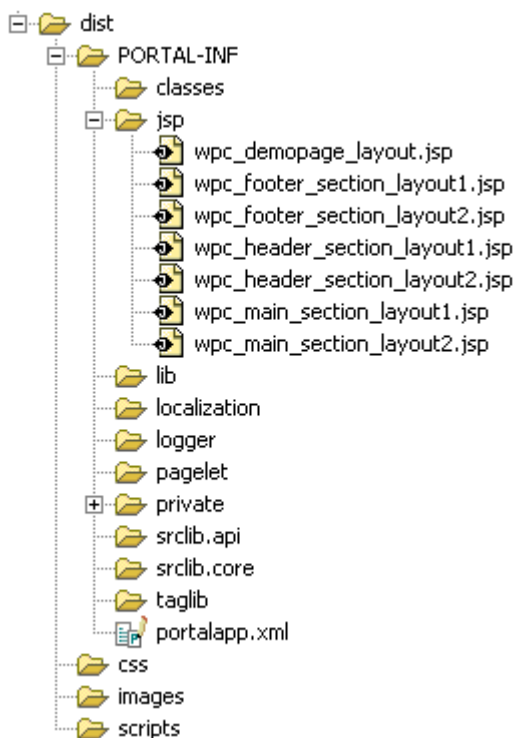
```

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

Important note:

1. For each new layout and all the corresponding section variants you have to declare new components in the `<components>` section of the `portalapp.xml` file.
2. The name of the JSP files within your project must match with the value of the corresponding `TemplateFile` property of your component profile in the `portalapp.xml` file.
3. The names of the sections and content areas (container IDs) in the JSP files must match with the values of the corresponding container properties within the component profile of the `portalapp.xml` file.

As a result of all the steps described above your project directory should contain the following files:



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

Use

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:

3. Use the secondary mouse button to choose the root folder *Portal Content*. In the context menu, choose *New* → *Folder*.
4. 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.

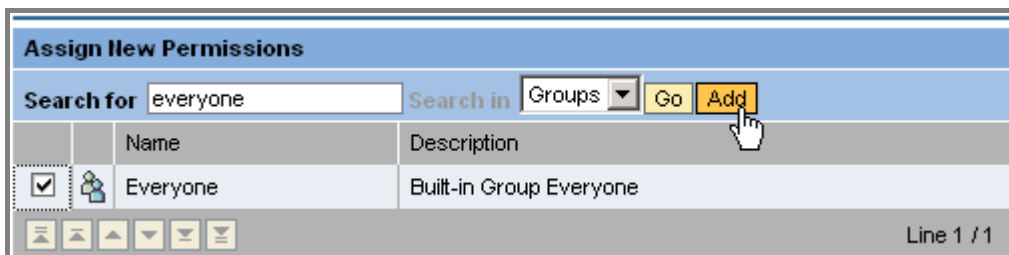
5. Use the secondary mouse button to choose the folder you just have created. In the context menu, choose *Open* → *Permissions*.
6. Now search in groups for "everyone".



Assign New Permissions

Search for: everyone Search in: Groups Go Add

7. Select the group `everyone` and click add.




Assign New Permissions

Search for: everyone Search in: Groups Go Add

	Name	Description
<input checked="" type="checkbox"/>	Everyone	Built-in Group Everyone

Line 1 / 1

- 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

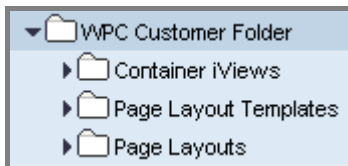
- To save the permissions, choose *Save*, than choose *Close* to close the Permission Editor.

2.2.3 Create Subfolders

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

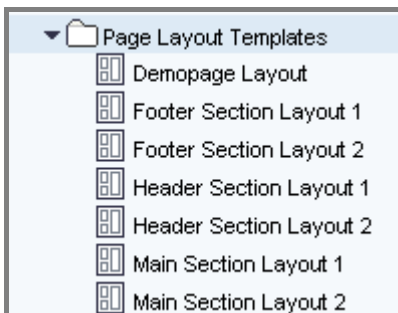
- Use the secondary mouse button to choose the new folder you previously created. In the context menu, choose *New → Folder*.
- Create three new subfolders and name the first one “*Container iViews*”, the second one “*Page Layout Templates*” and the third one “*Page Layouts*”.

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). For our example, we have to create seven new layout templates:



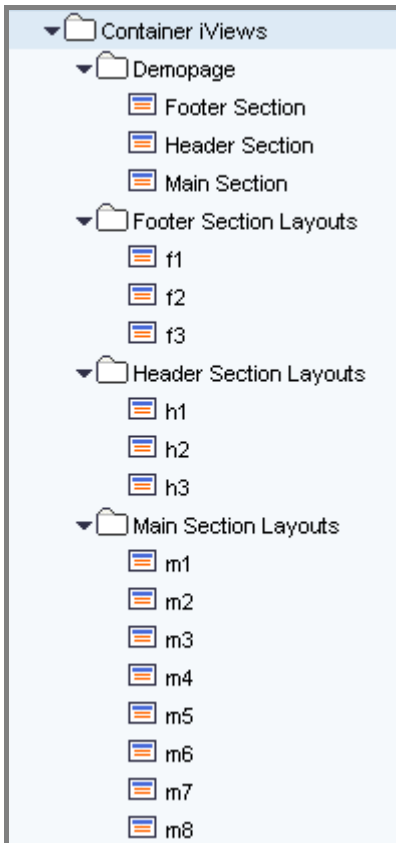
- Use the secondary mouse button to choose the folder *Portal Content → “Your_New_Folder” → Page Layout Templates*. In the context menu, choose *New → Layout*.
- Select the portal archive that you previously created in the SAP NetWeaver Developer Studio which contains the new layouts. Choose *Next*. The system displays all layouts that are defined in the `portalapp.xml` file as `<component>`.
- Select one of your new layouts and choose *Next*.
- Specify the default properties `Layout Name` and `Layout ID`.
- Check your entries and choose *Finish*.


6. Choose *Open the object for editing* and then choose *OK*. Check the `Object` is a `Template` property.
7. To save your layout template, choose *Save*.

Repeat these steps for each layout of your project.


2.2.5 Create Container iViews

The WPC container iViews are used to separate the page in different sections and hold the content on a specific area of these sections. Create a container iView for each section of your page layout (for our example: “Header Section”, “Main Section”, “Footer Section”) and for each content area within a section (for our example: “h1”, “h2”, “h3”, “m1”, etc.):



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*.

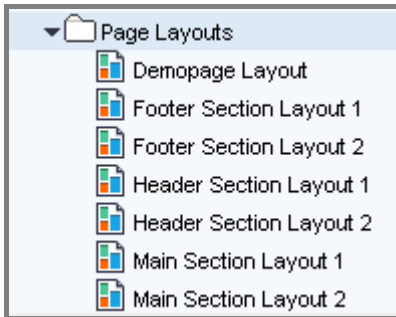
Repeat these steps for every container iView.

 For all the content areas (not the sections) you may re-use the existing container iViews which are already assigned to the default page layouts. **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 for the content areas.

You find the default container iViews for the content areas in the following folder: Portal Content → *Web Page Composer* → *Container iViews* → *WPC Default Containers*.

2.2.6 Create Page and Section Layouts and Assign the Container iViews

Create layouts (technically portal pages) for each layout template (page and sections):




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*.
3. Choose the *Default Page Template* template. Choose *Next*.
4. Select the layout template 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*.

Repeat these steps for the page layout and each section layout.


Assign the container iViews to the page layout and the section layouts in the following way:

Assign the three container iViews representing the sections to the page layout and the remaining container iViews representing the content areas to the section layouts.

 If you re-use existing container iViews for the section layouts 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 section layout should be logically equivalent.

1. Open the new layout (page or section) in the Portal Content Studio.
2. 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).
3. In the context menu, choose *Add iView to Page* → *Delta Link* while your target layout (portal page) is still open in the Portal Content Studio.
4. Switch to the *Page Layout* mode. Use *Drag&Drop* to move the container iView to the respective position in the WPC page layout.
5. Repeat steps 2 - 4 until all container iViews for this layout have been assigned.
6. Save your layout (*Save* button).

Repeat these steps for all page and section layouts.

 You can also assign other iViews to your WPC page layout that should be part of every page based on this layout. Typical examples are a constant footer (e.g. impressum) or the page header toolbar (RSS feed, print-optimized version, etc.) on each standard page template. To provide this toolbar also on your section-based layout you have to assign the *Page Header Links* iView once to your page layout. In this case it would make sense to define a separate header area in your *wpc_demopage_layout.jsp* file above the content sections.

The following screenshot shows how the container iViews representing the sections have been arranged for the main page layout:



Copy the ID of the new layouts into an empty text file:

To simplify the configuration of your new layouts in the configuration framework of KM we recommend storing the IDs of your layouts in a local text file.

7. Click on your new layouts in the folder *Portal Content* → “*Your_New_Folder*” → *Page Layouts*.
8. Mark and copy the ID of your layout in the *Quick Info* area (see screenshot below).
9. Paste the ID into an empty text file.

Repeat these steps for all of your new layouts (page and sections).

WPC Customer Folder

- Container iViews
- Page Layout Templates
- Page Layouts
 - Demopage Layout**
 - Footer Section Layout 1
 - Footer Section Layout 2
 - Header Section Layout 1
 - Header Section Layout 2
 - Main Section Layout 1
 - Main Section Layout 2

Quick Info

ID: `pcd:portal_content/wpc_customer_folder/page_layouts/demopage_layout`

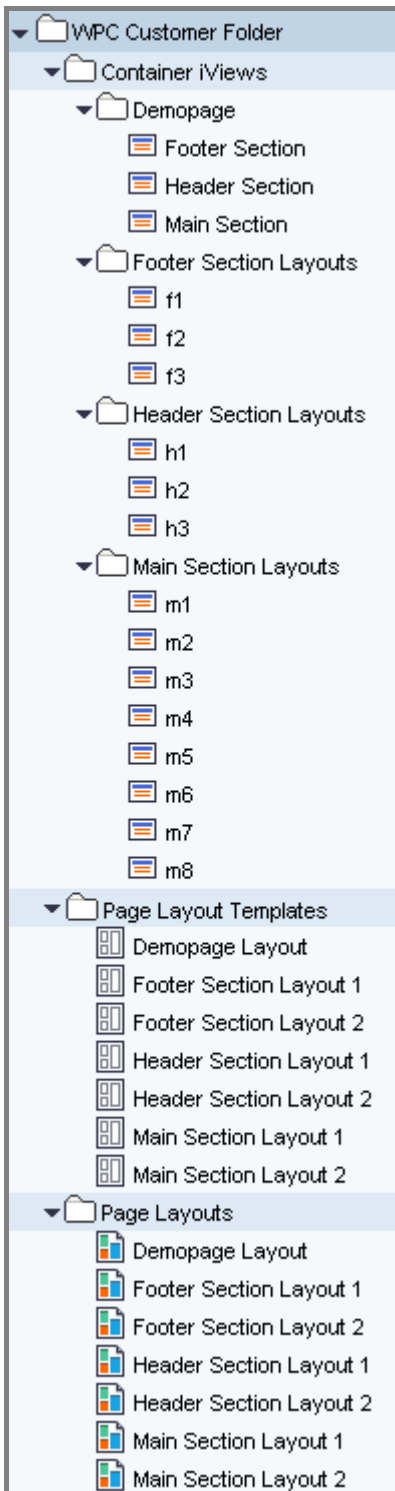
Description:

Permission: Owner, Full Control, Read and W (End User)

Undo
Cut
Copy
Paste

Result

The following screenshot shows all required layout objects (layout templates, container iViews, page and section layouts) that have been created for our example project:



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 Objects

For your page layout and each section layout create a new layout template configuration object.

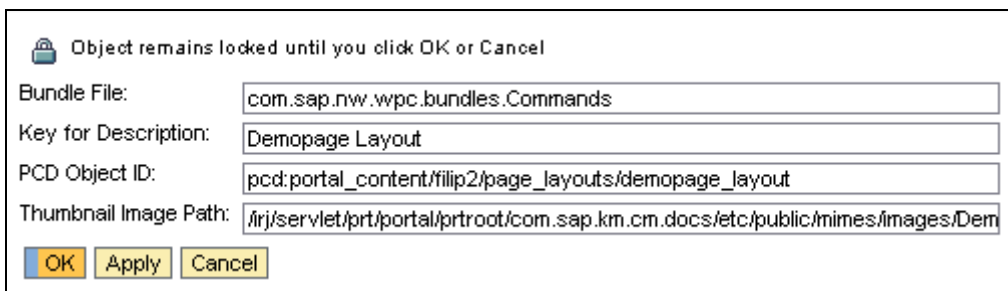
10. In the KM configuration, choose *Web Page Composer* → *Layout* → *Layout Templates* → *Layout Templates*.
11. Choose *New* to create a new layout template.
12. Specify a name and enter a key for description.
13. Enter the ID for the page layout that you copied from the Portal Content Studio in the *PCD Object ID* parameter. This step maps your PCD layout object to the corresponding WPC configuration object.
14. Specify the *Thumbnail Image Path* in order to see a preview icon of the 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/your_layout_image.gif"`

15. To save the new layout template, choose *OK*.
16. Repeat steps 2-6 for each section layout.

Here is a screenshot of the page layout config object:

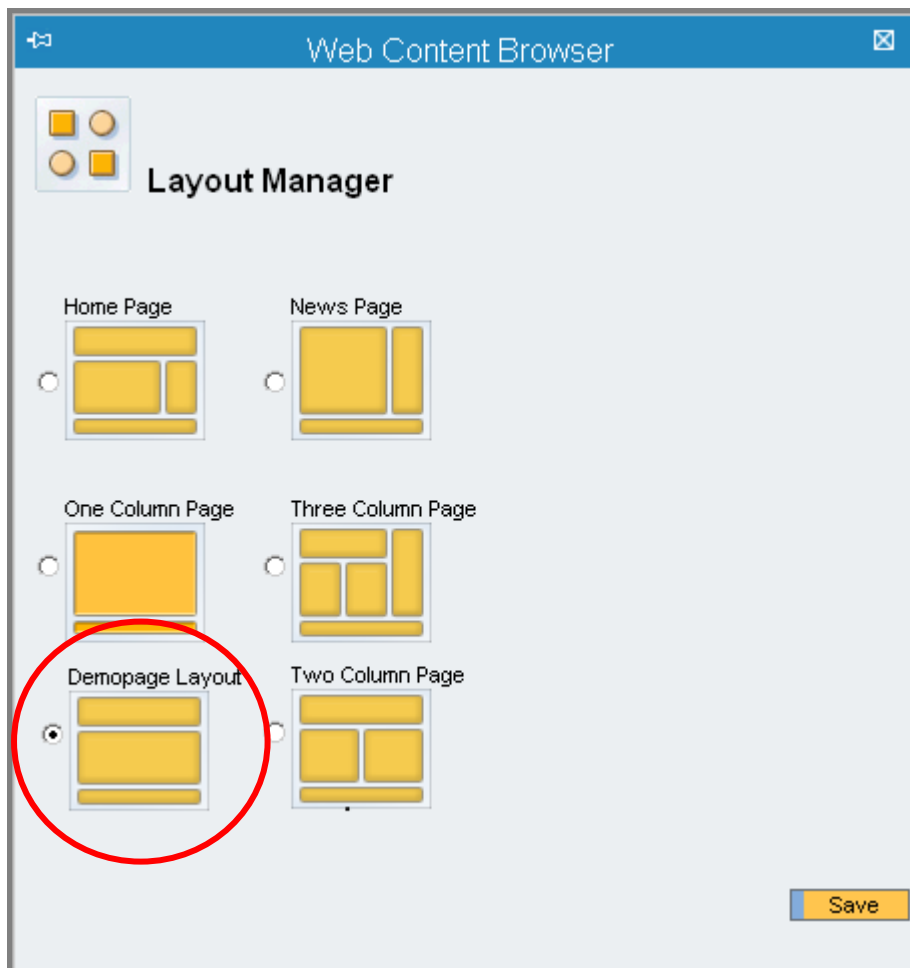


Note for preview images:

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



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



Finally you should have created the following layout template configuration objects:

Layout Templates
Define Layout Templates

Name: Objects per Page: ▾

<input type="checkbox"/>	Name	Bundle File	Key for Description
<input type="checkbox"/>	wpc_footer_section_layout1	com.sap.nw.wpc.bundles.Commands	Footer Section Layout 1
<input type="checkbox"/>	wpc_footer_section_layout2	com.sap.nw.wpc.bundles.Commands	Footer Section Layout 2
<input type="checkbox"/>	wpc_header_section_layout1	com.sap.nw.wpc.bundles.Commands	Header Section Layout 1
<input type="checkbox"/>	wpc_header_section_layout2	com.sap.nw.wpc.bundles.Commands	Header Section Layout 2
<input type="checkbox"/>	wpc_main_section_layout1	com.sap.nw.wpc.bundles.Commands	Main Section Layout 1
<input type="checkbox"/>	wpc_main_section_layout2	com.sap.nw.wpc.bundles.Commands	Main Section Layout 2
<input type="checkbox"/>	wpc_demopage_layout	com.sap.nw.wpc.bundles.Commands	Demopage Layout

2.3.2 Extend the existing Page Layout Group

To display the new page layout in the list of available layouts in the page editor (*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 (*wpc_demopage_layout*) to this layout group. Save your changes by choosing **OK**.

2.3.3 Create new Section Layout Groups

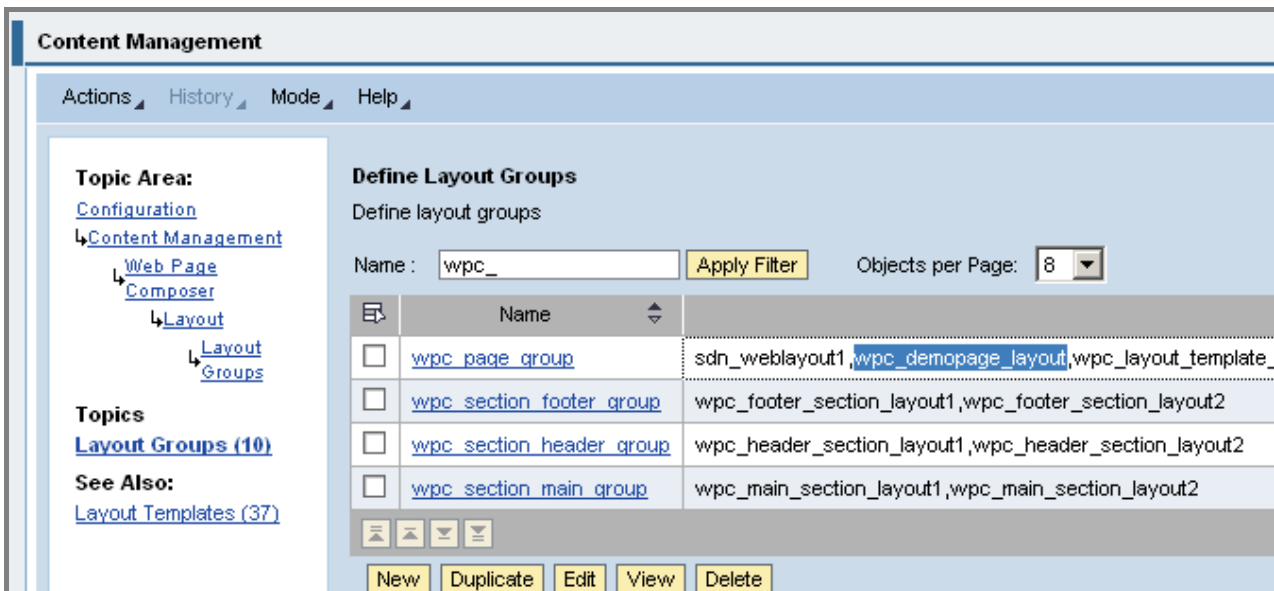
Create a new Layout Group for every section.

In our example: “*wpc_section_header_group*”, “*wpc_section_main_group*” and “*wpc_section_footer_group*”.

1. In the KM configuration, choose *Web Page Composer* → *Layout* → *Layout Groups* → *Layout Groups*.
2. Choose *New* to create a new layout group.
3. Specify a name and a description.
4. Select the section layouts which belong to this layout group.
5. Save the new Layout Group by choosing *OK*.

Repeat these steps for each section (header, main, footer).

For our example project you should have created the following layout groups:



2.3.4 Create new Resource Types for Sections

Create a new resource type for each section (header, main, footer).

1. In the KM configuration, choose *Global Services* → *Resource Types* → *Resource Types*.
2. Create a new resource type.
3. Enter a Name and a Resource Type ID. Use the prefix **wpc_** for these parameters.
4. Activate the parameters *Reserved for Application* and *Is for a Collection*.

Repeat these steps for each section.

This screenshot shows the new resource types for the three sections of our example project:

ResourceTypes								
Name : <input type="text" value="section"/>		<input type="button" value="Apply Filter"/>		Objects per Page: <input type="text" value="8"/>				
<input type="checkbox"/>	Name	Is for a Collection	Reserved for Application	Bundle File	Custom Properties	Icon	Key for Description	Resource Type ID
<input type="checkbox"/>	wpc_footer_section	✓	✓				WPC Section Footer	wpc_footer_section
<input type="checkbox"/>	wpc_header_section	✓	✓				WPC Section Header	wpc_header_section
<input type="checkbox"/>	wpc_main_section	✓	✓				WPC Section Main	wpc_main_section

2.3.5 Create new Web Resource Types for Sections

Create a new *Web Resource Type* for each section (header, main, footer).

1. In the KM configuration, choose *Web Page Composer* → *Web Content Types* → *Web Resource Type*.
2. Click *New* to create a new Web Resource
3. Configure the following parameters and save the new Web resource type:

Allowed Parent Types	<i>wpc_page</i> .
Allowed Layouts	Select the corresponding section layout group.
Default Layout	Select a section layout belonging to the just selected section layout group. This section layout will be set as the default section layout.
Resource Type	Select the corresponding resource type for this section
Toolbar	Select <i>section</i> .
Class Auto Create	<code>com.sap.nw.wpc.km.service.compoundprovider.CollectionAutoCreate</code>
Description	Enter a description for this section.

Repeat these steps for each section (header, main, footer).

This is a screenshot of the header section Web resource type:

View "wpc_header_section"

Published: Yes No Not set
 Requires Framework: Yes No Not set
 Allowed child types:

Name
Not set

Page 1 / 1

Allowed parent types:

Name
wpc_page

Page 1 / 1

Allowed repositories:

Name
Not set

Page 1 / 1

Allowed Layouts: [wpc_section_header_group](#)
 Default Layout: [wpc_header_section_layout1](#)
 Resource Type: * [wpc_header_section](#)
 Toolbar: [section](#)
 Allowed RIDS:
 Class Auto Create: [com.sap.nw.wpc.km.service.compoundprovider.CollectionAutoCreate](#)
 Description: [Header Page Section](#)
 Factory implementation:

Finally, you should have created the following Web resource types:

Web Resource Type
 Define web resource types


Name: Objects per Page:

Name	Allowed parent types	Allowed Layouts	Default Layout	Resource Type	Toolbar
wpc_footer_section	wpc_page	wpc_section_footer_group	wpc_footer_section_layout1	wpc_footer_section	section
wpc_header_section	wpc_page	wpc_section_header_group	wpc_header_section_layout1	wpc_header_section	section
wpc_main_section	wpc_page	wpc_section_main_group	wpc_main_section_layout1	wpc_main_section	section

2.3.6 Create Container Configuration Objects

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

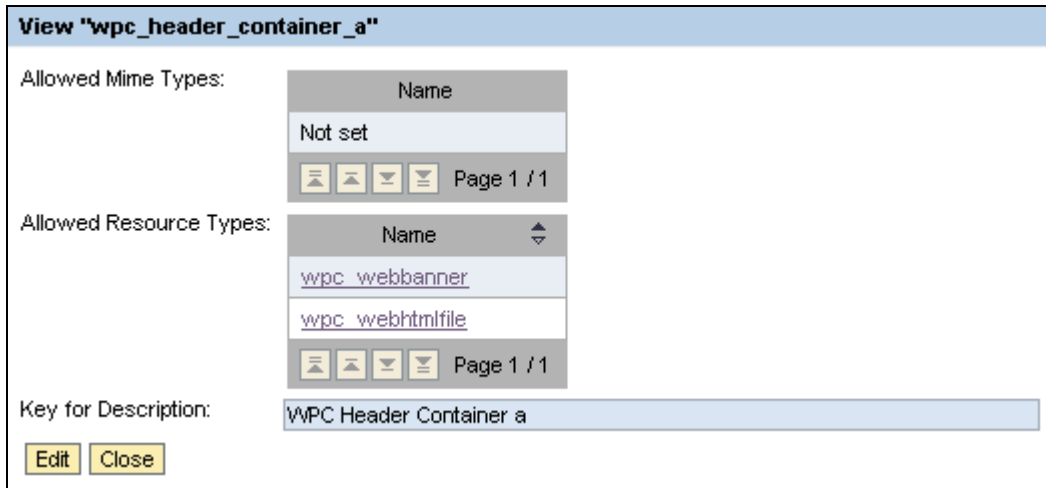
Create containers for content areas:

 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 content areas (container iViews).

1. In the KM configuration, navigate to *Web Page Composer* → *Containers* → *Containers*.
2. Choose *New* to create a container configuration object.
3. Allow the appropriate resource types and MIME types according to your rules.
4. Save the new container configuration object by choosing *OK*.

Repeat these steps for every container representing a content area.

Screenshot for one content area of the header:

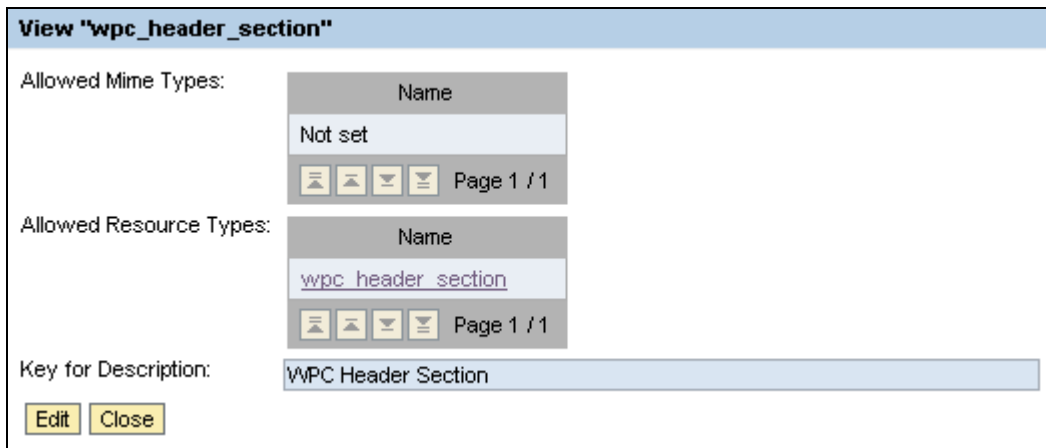


Create containers for each section:

5. Choose *New* to create a container configuration object.
6. Assign the corresponding KM resource type for this section.
7. Save the new container configuration object by choosing *OK*.


Repeat these steps for every container representing a section.

Screenshot for the header section of our example project:



By now you should have created all these container configuration objects:

Containers			
Define Containers			
Name :	<input type="text"/>	<input type="button" value="Apply Filter"/>	Objects per Page: <input type="text" value="All"/>
<input type="checkbox"/>	Name	Allowed Mime Types	Allowed Resource Types
<input type="checkbox"/>	wpc_footer_container_a		wpc_webarticle,wpc_webbanner,wpc_webhtmlfile
<input type="checkbox"/>	wpc_footer_container_b		wpc_webfeature,wpc_weblinklist
<input type="checkbox"/>	wpc_footer_section		wpc_footer_section
<input type="checkbox"/>	wpc_header_container_a		wpc_webbanner,wpc_webhtmlfile,wpc_webminiarticle
<input type="checkbox"/>	wpc_header_container_b		wpc_webfeature,wpc_weblinklist
<input type="checkbox"/>	wpc_header_section		wpc_header_section
<input type="checkbox"/>	wpc_main_container_a		wpc_webarticle,wpc_webfeature,wpc_webhtmlfile
<input type="checkbox"/>	wpc_main_container_b		wpc_weblinklist
<input type="checkbox"/>	wpc_main_section		wpc_main_section

 Note the names of your new container configuration objects or copy them to a text file. You will need these names in the next section to map them to the corresponding container iViews.

Result

You have worked through all configuration steps to integrate a section-based page layout and its sub-layouts into 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 for all new containers. For all existing containers which you are re-using as content areas 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* and open the folder in which the container 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.


To test your section-based layout of the demo project use the *Change Page Layout* command in the page editor and switch to the *Demopage Layout*. Now you'll also be able to change the layout of each section using the *Change Container Layout* command (e.g. the main section):

 **Header Page Section**




Drop content here
[Banner, HTML File]

 **Main Page Section**


- Edit
- Change Container Layout** 
- Cache Properties
- Check Links
- Information

Drop content here
[Article, Teaser, HTML File]

Drop content here
[Article, Teaser, HTML File]




Drop content here
[Article, Teaser, HTML File]



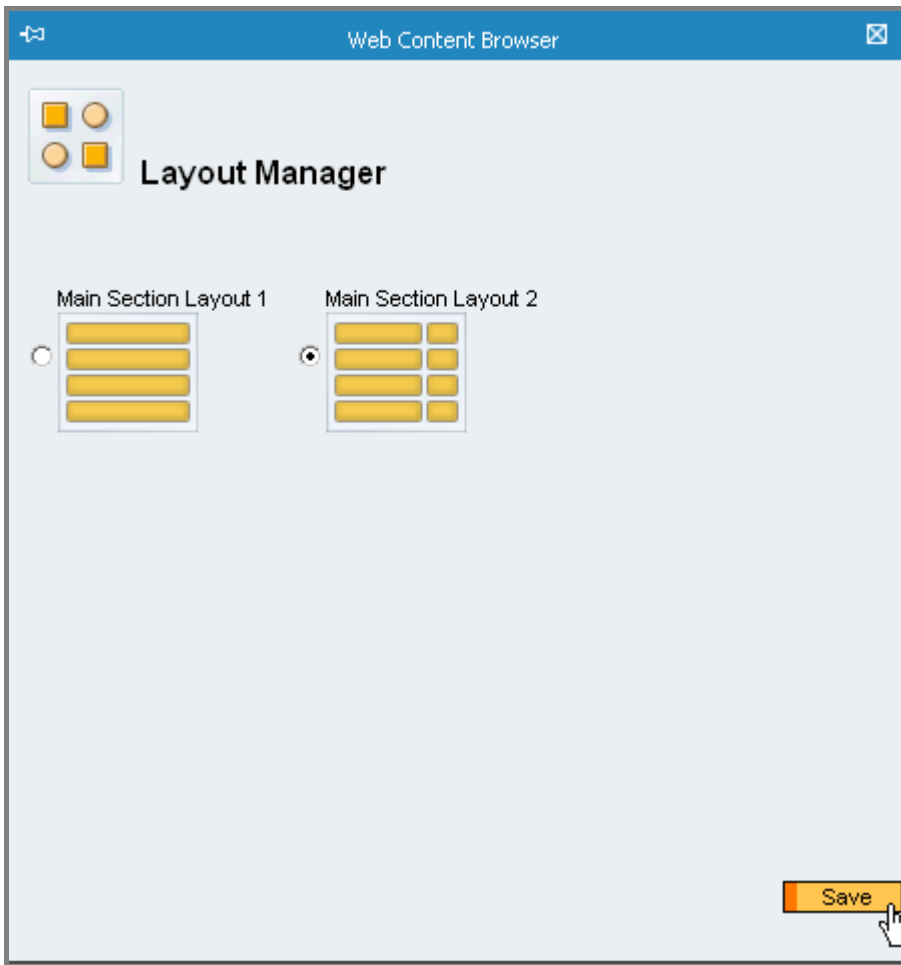
Drop content here
[Article, Teaser, HTML File]

 **Footer Page Section**



Drop content here
[Article, Banner, HTML File]

You may choose a different sub-layout for the main section:



Now your page layout reflects the selected sub-layout of the main section:

Preview Change Page Layout Publish Refresh Assignment Admin Cache Properties Check Links Information Finish & Unlock

Header Page Section

Drop content here
[Banner, HTML File]

Main Page Section

Drop content here [Article, Teaser, HTML File]	Drop content here [[Link List]]
Drop content here [Article, Teaser, HTML File]	Drop content here [[Link List]]
Drop content here [Article, Teaser, HTML File]	Drop content here [[Link List]]
Drop content here [Article, Teaser, HTML File]	Drop content here [[Link List]]

Footer Page Section

Drop content here
[Article, Banner, HTML File]

www.sdn.sap.com/irj/sdn/howtoguides