

How to Merge the Masthead and the Tool Area

Applies To:

SAP NetWeaver 2004s (EP7)

Summary

This article is about customizing the SAP Enterprise Portal Masthead. The client may have his masthead image according to the company branding. But sometimes the image height is so large that the content area becomes smaller, because the tool area is also present below the masthead consuming more space. But if we look carefully, we can find some space that is being wasted in the tool area. As it contains only Collaboration Launch Pad and the Search functionality, if they are shifted up in the masthead then the issue can be solved. So, this may be a valid requirement of the client. But the problem is we cannot merge the Tool Area in the Masthead by out-of-the-box features provided by SAP Enterprise Portal.

Author: Mr. Sudip Majumder

Company: TCS – Tata Consultancy Services

Created on: 04 July 2007

Author Bio



Sudip Majumder is SAP certified - Development Consultant (SAP Netweaver 2004 - Portal & KMC). He is working in TCS since 2004.

Table of Contents

Problem Description	3
Solution Details.....	3
PAR File Extraction	3
Resource Bundle File Modification	5
Component-Profile Properties Modification	6
JSP File Modification	7
Masthead Deployment and iView Creation.....	14
Summary.....	14
Related Content.....	15
Disclaimer and Liability Notice.....	16

Problem Description

This article is about customizing the SAP Enterprise Portal Masthead. The client may have his masthead image according to the company branding. But sometimes the image height is so large that the content area becomes smaller, because the tool area is also present below the masthead consuming more space. But if we look carefully, we can find some space that is being wasted in the tool area. As it contains only Collaboration Launch Pad and the Search functionality, if they are shifted up in the masthead then the issue can be solved. So, this may be a valid requirement of the client. But the problem is we cannot merge the Tool Area in the Masthead by out-of-the-box features provided by SAP Enterprise Portal.

Solution Details

Standard SAP masthead (with tool area) looks like this –

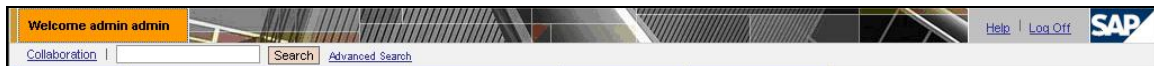


Figure 1 : SAP Standard Masthead (with separate tool area)

The requirement of the client is to put the search area and the Collaboration Launch Pad (i.e. the tool Area) in the masthead, just like the image below –



Figure 2 : Customized Masthead (with tool area merged with the masthead)

As this cannot be done by out-of-the-box features of SAP Enterprise Portal, we have to modify the SAP standard masthead component code, which is responsible for rendering the masthead. We need to incorporate the Tool Area component inside that.

Follow the steps mentioned below to perform the task –

PAR File Extraction

- In SAP Enterprise Portal navigate to *Systems Administration -> Support*.

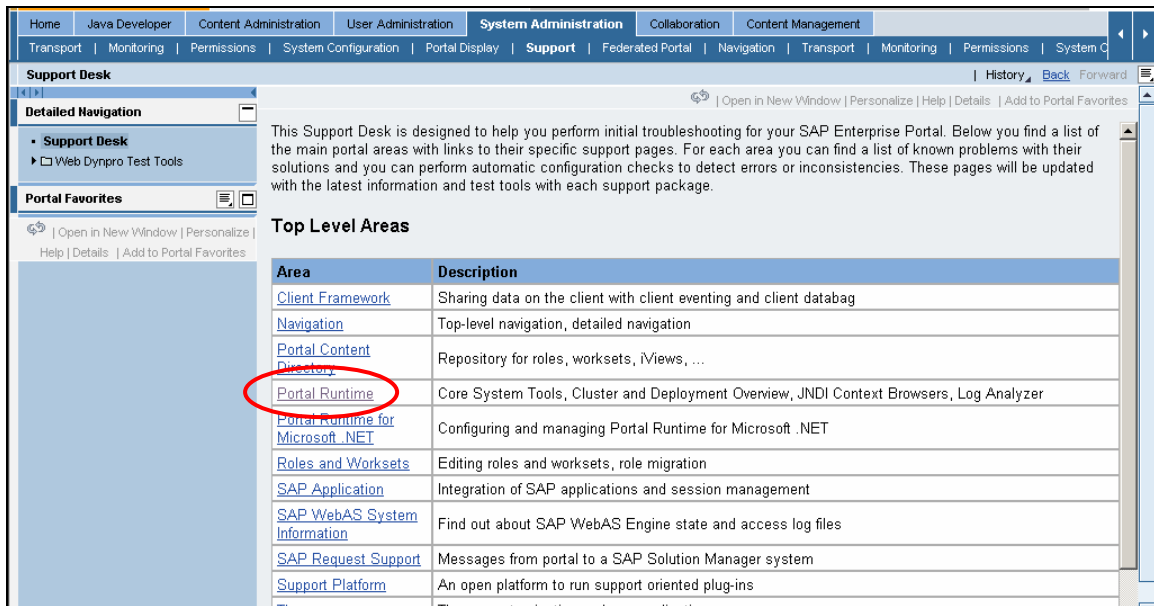


Figure 3 : Portal Support Desk

- In the portal inner page click “Portal Runtime”.

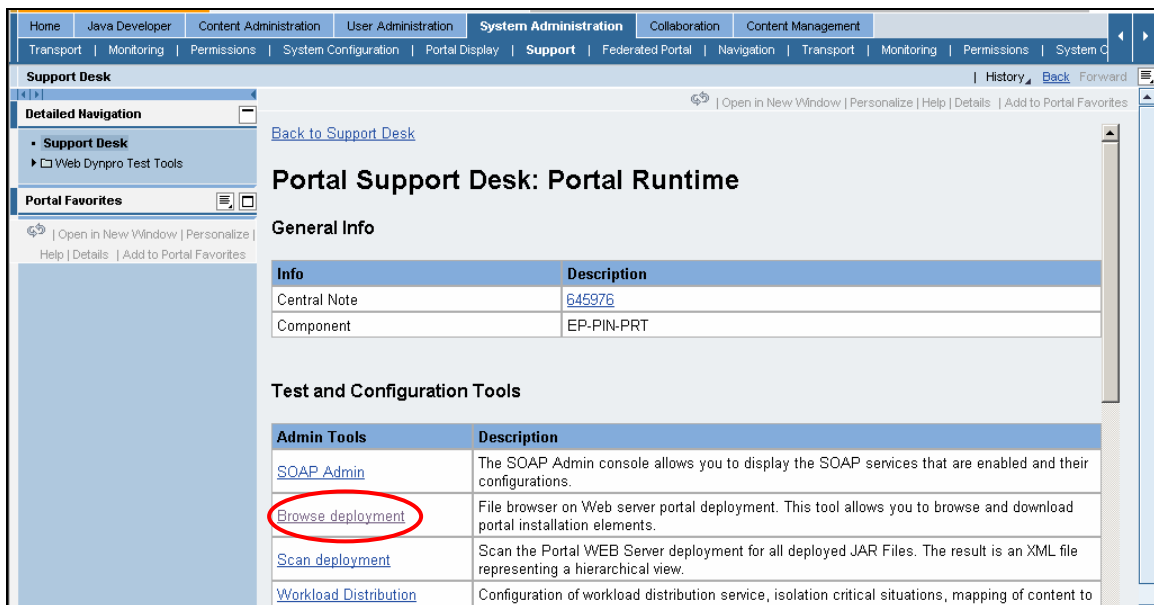


Figure 4 : Portal Support Desk > Portal Runtime

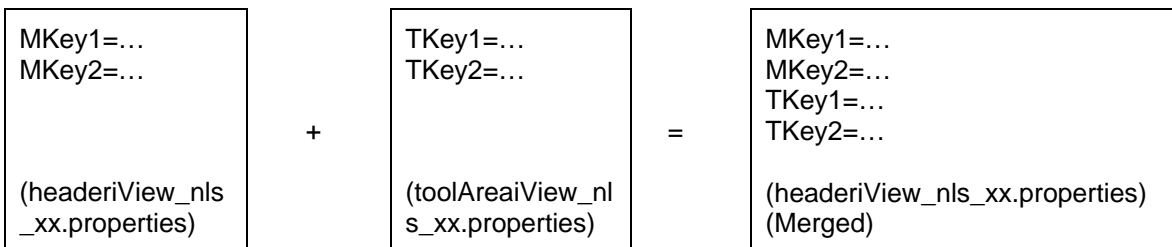
- Click “Browse Deployment”.
- Search for the “*com.sap.portal.navigation.masthead.par*” and download in your local PC.
- Search for the “*com.sap.portal.navigation.toolarea.par*” and download in your local PC (you need this component to get the code of the tool area components).
- Open NetWeaver Developer Studio (NWDS).
- Import the PAR file (*com.sap.portal.navigation.masthead.par*) and create a new project with a new name (we will use “*MyMastheadToolArea*”) (Provide a name that does not clash with the applications already deployed in the portal. Otherwise the existing application will be overwritten).

Resource Bundle File Modification

Both of the masthead component and the tool area component provide internationalization support i.e. link text, labels etc will be changed when user changes the language. Resource bundle files are used in each of the components to achieve this. If you open “*headeriView_nls_en.properties*” located in “*./dist/PORTAL-INF/private/classes*”, you will be able to some keys and its corresponding value. e.g.

```
LOG_OFF_CONFIRM_MSG=Are you sure you want to log off?
```

There are many files in that location named “*headeriView_nls_xx.properties*”, “xx” being the language name, e.g. for Deutsche, the file will be “*headeriView_nls_de.properties*”. All these keys (e.g. LOG_OFF_CONFIRM_MSG) are used in the component code to provide internationalization support. Tool Area is also a component and it also contains this kind of Resource Bundle files. If you create a project by importing “*com.sap.portal.navigation.toolarea.par*”, these files will be available in “*./dist/PORTAL-INF/private/classes*” location with the name as “*toolAreaiView_nls_xx.properties*”. As we are going to merge the tool area in the masthead we also need to merge the keys available in the “*toolAreaiView_nls_xx.properties*” files into the “*headeriView_nls_xx.properties*” files. i.e. copy the key-values in the “*toolAreaiView_nls_xx.properties*” file and paste them at the last of the corresponding “*headeriView_nls_xx.properties*” files as shown in the image below -



Component-Profile Properties Modification

We need to merge the component-profile properties of the tool area in to the masthead for smooth operation of the tool area component in the masthead. Follow the below mentioned steps to accomplish the task –

- In NWDS, Open the newly created project (*MyMastheadToolArea*).
- Open the *portalapp.xml* and navigate to the “Components” tab.

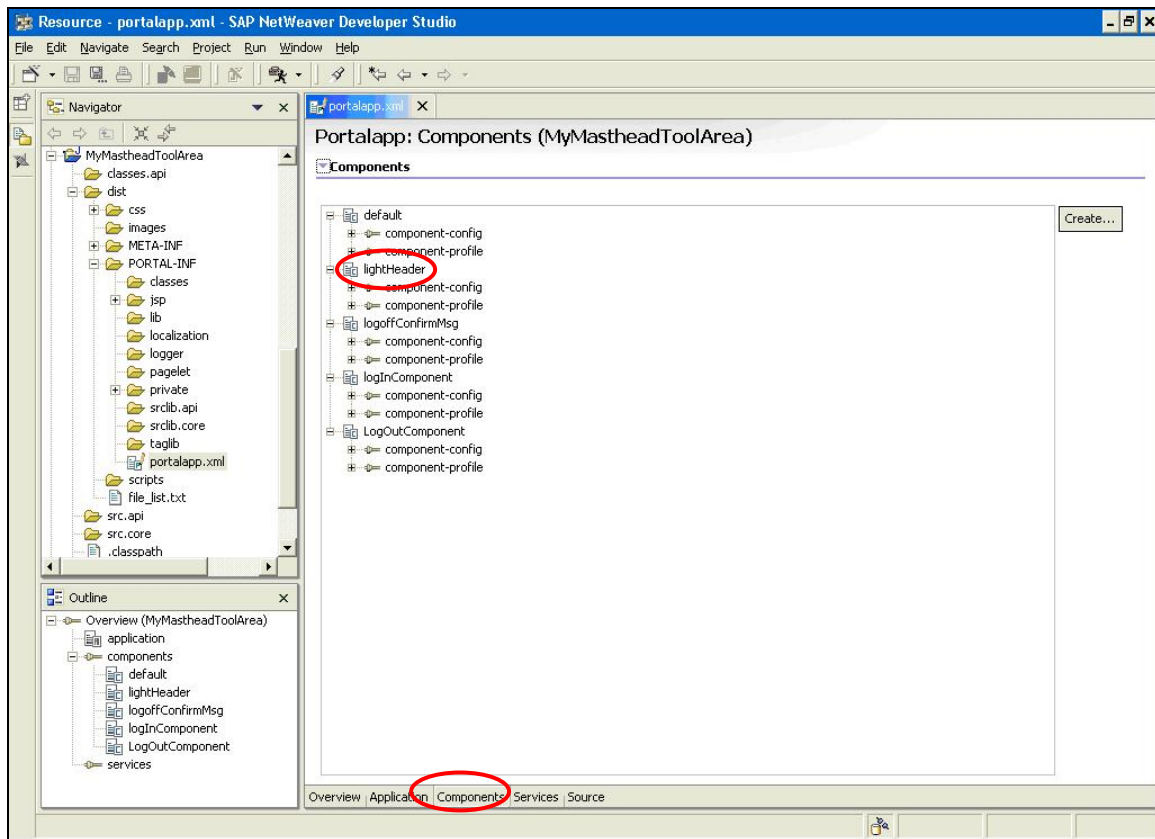


Figure 5 : portalapp.xml

- Add the following component-profile properties and sub-properties in the lightHeader component.

Property	Value	Sub-Property
SearchComponent	pcd:portal_content/every_user/general/eu_role/com.sap.km.home_ws/com.sap.km.hidden/com.sap.km.urlaccess/com.sap.km.basicsearch	description=LBL_SEARCH_COMPONENT plainDescription=KM Search Component personalization=none
SearchComponentParameters	layoutSetMode=exclusive&ResourceListType=com.sapportals.wcm.SearchResultList&SearchType=quick&QueryString=	description=LBL_SEARCH_PARAMETERS plainDescription=KM Search Component personalization=none
AdvancedSearchComponentParameters	layoutSetMode=exclusive&ResourceListType=com.sapportals.wcm.SearchResultList&SearchType=quick&Advanced=true&QueryString=	description=LBL_ENABLE_SRCH_ADV_PARAM plainDescription=KM Search Advanced Parameters personalization=none
EnableCLP	true	type= select[true,false]
ClpComponent	com.sap.netweaver.coll.shared.ui.CollaborationLaunchPad	
EnableRtcSessionManager	true	description=LBL_ENABLE_RTC_SESS_MANAGER plainDescription=Enable Real-Time Collaboration personalization=none type=select[true,false]
RtcSessionManagerComponent	com.sap.netweaver.coll.appl.ui rtc.SessionManager	description=LBL_RTC_SESS_MANAGER_COMP plainDescription=Real-Time Collaboration Component personalization=none

JSP File Modification

Now we are going to update the masthead JSP file. Before doing that I want to give you a brief introduction about the structure of the masthead. All the elements of the masthead (e.g. welcome area, brand image, link area and logo) are kept inside a HTML table. Each of the elements is kept in separate columns of that table, as shown below –

Welcome Area	Brand Image	Link Area	Logo
--------------	-------------	-----------	------

We will concentrate in the Link Area section. The link area column itself contains another table. Different links (e.g. Help, Log Off etc.) are kept inside different columns of the table as shown below –

Welcome Area	Brand Image	Link Area			Logo
		Help	Personalize	Log Off	

Our aim is to put the tool area components just below in the Link Area section. The masthead should look like this -

Welcome Area	Brand Image	Link Area				Logo
			Help	Personalize	Log Off	
		Collaboration Launch Pad	Search Text Box	Search Button	Advanced Search	

To implement this design, follow the below mentioned steps –

- In NWDS, open light_HeaderiView.jsp file.
- Locate the section where the general variables are declared.
- Create the following new variables that will get the value from the resource bundle file. You can use the following code –

```
final String ENABLE_CLP = "EnableCLP";
final String ENABLE_RTC_SESSION_MANAGER = "EnableRtcSessionManager";
final String SEARCH_NO_ENTRY_MESSAGE = "SEARCH_NO_ENTRY_MESSAGE";
final String SEARCH_INPUT_TOOLTIP = "SEARCH_INPUT_TOOLTIP";
final String SEARCH_BUTTON_TEXT = "SEARCH_BUTTON_TEXT";
final String SEARCH_BUTTON_TOOLTIP = "SEARCH_BUTTON_TOOLTIP";
final String SEARCH_ADVANCED_LINK_TEXT = "SEARCH_ADVANCED_LINK_TEXT";
final String SEARCH_ADVANCED_LINK_TOOLTIP = "SEARCH_ADVANCED_LINK_TOOLTIP";
final String CLP_LINK_TEXT = "CLP_LINK_TEXT";
final String CLP_LINK_TOOLTIP = "CLP_LINK_TOOLTIP";

String
searchNoEntryMessageStr=getNLSString(componentRequest,SEARCH_NO_ENTRY_MESSAGE);
String searchInputTooltipStr=getNLSString(componentRequest,SEARCH_INPUT_TOOLTIP);
String searchButtonTextStr= getNLSString(componentRequest,SEARCH_BUTTON_TEXT);
String
searchButtonTooltipStr=getNLSString(componentRequest,SEARCH_BUTTON_TOOLTIP);
String
searchAdvancedLinkTextStr=getNLSString(componentRequest,SEARCH_ADVANCED_LINK_TEXT
);
```



```
String searchAdvancedLinkTooltipStr = "";
String clpLinkTextStr=getNLSString(componentRequest, CLP_LINK_TEXT);
String clpLinkTooltipStr= getNLSString(componentRequest,CLP_LINK_TOOLTIP);
```

- Locate the section where the javascript functions are defined.
- Create the following javascript functions (necessary for the tool area component) –

```
function ToolAreaSearch_onReturnKey() {
    if(event.keyCode=="13") {
        openSearch();
        return false;
    }
    return true;
}
```

```
function ToolAreaSearch_onSpaceKey() {
    if(event.keyCode=="32") {
        openSearch();
        return false;
    }
    return true;
}
```

```
function openSearch()
{
    <%
    IPortalComponentURI componentURI=componentRequest.createPortalComponentURI ();
    String searchComponentName = profile.getProperty("SearchComponent");
    componentURI.setContextName(searchComponentName);
    String searchComponentURI = componentURI.toString();
    String searchComponentParams=profile.getProperty("SearchComponentParameters");
    String url = searchComponentURI + "?" + searchComponentParams;
    %>
    var QueryString=document.all("quickSearchField").value;
    if (QueryString==" " || QueryString.length==0){
        alert("<%=searchNoEntryMessageStr%>");
        return;
    }
    try {
        QueryString = encodeURI(QueryString);
    }
}
```

```

catch (e) {
}
var regexp_plus=/\+/g;
var regexp_quot=/\?/g;
var regexp_amp=/\&/g;
QueryString = QueryString.replace(regexp_plus,"%2B");
QueryString = QueryString.replace(regexp_quot,"%3F");
QueryString = QueryString.replace(regexp_amp, "%26");
var searchURL="<%=url%>" + QueryString;
window.open( searchURL, "_blank");
}

function openAdvancedSearch()
{
<%
    String advancedSearchComponentName = profile.getProperty("SearchComponent");
    componentURI.setContextName(advancedSearchComponentName);
    String advancedSearchComponentURI = componentURI.toString();
    String
advancedSearchComponentParams=profile.getProperty("AdvancedSearchComponentParamet
ers");
    String urlAdv = advancedSearchComponentURI + "?" +advancedSearchComponentParams;
%>
    var QueryString=document.all("quickSearchField").value;
    try {
        QueryString = encodeURI(QueryString);
    }
    catch (e) {
    }
    var regexp_plus=/\+/g;
    var regexp_quot=/\?/g;
    var regexp_amp=/\&/g;
    QueryString = QueryString.replace(regexp_plus,"%2B");
    QueryString = QueryString.replace(regexp_quot,"%3F");
    QueryString = QueryString.replace(regexp_amp, "%26");

    var advancedSearchURL="<%=urlAdv%>" + QueryString;
    window.open( advancedSearchURL, "_blank");
}

function openCollaborationLaunchPad()
{

```

<%

```
String clpComponentName = profile.getProperty("ClpComponent");
componentURI.setContextName(clpComponentName);
String clpComponentURI = componentURI.toString();
%>
```

```
clpParameter='height=510,width=322,location=No,menubar=No,resizable=Yes,scrollbar
s=Yes,status=No,titlebar=No,toolbar=No,left=' + (screen.width / 3) + ',top=180' ;
clp_wnd=window.open('<%=clpComponentURI%>', 'STATIC_CLP_WINDOW_TITLE',clpParameter
);
clp_wnd.focus();
}
```

- Locate the section where the link tables are created (as shown in the image below)

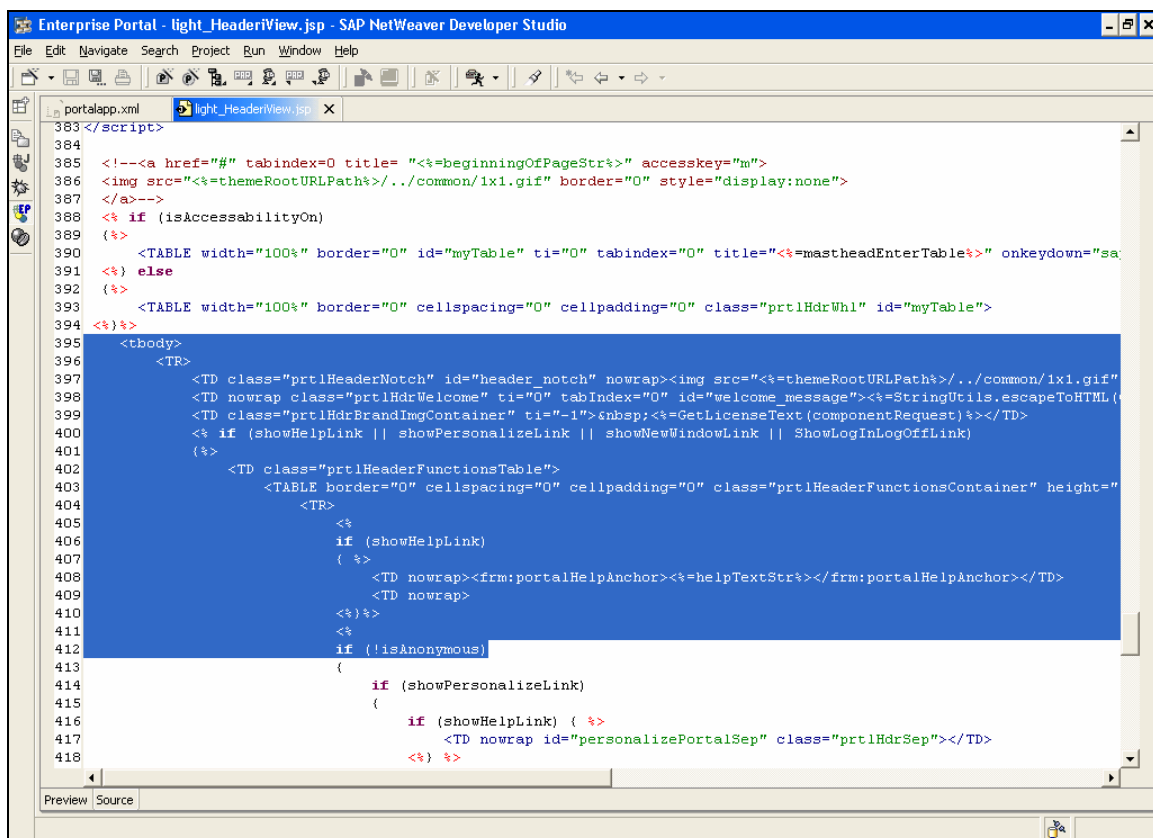


Figure 6 : light_HeaderView.jsp

- Just before the link table is ended, add another row for the tool area. Add the collaboration Launch Pad, Search Text Box, Search Button and the Advanced Search link. You can copy the code from the tool area and paste them here. You can also use the following code -

```

<TR>
<TD colspan="7">
<TABLE width="100%" border="0" cellspacing="0" cellpadding="0"
class="prtlToolAreaTaskTable">
<TR>
<%
if (isEnabled(profile, ENABLE_CLP)) {
%>
<TD width="15%"></TD>
<TD width="15%"></TD>

<td width="15%" height="30px" valign="center"><FONT SIZE="3" ><A href="#"
id="openCollaborationLaunchPad" align="center" class="prtlToolAreaTaskText"
ct="Link" onClick="openCollaborationLaunchPad(); return false;"
title="<%=clpLinkTooltipStr%>"><%=clpLinkTextStr%></A></FONT></TD>

<TD nowrap id="personalizePortalSep" class="prtlHdrSep"></TD>
<%} else
{
%>
<TD width="15%"></TD>
<TD width="15%"></TD>
<TD width="15%" height="30px" valign="center"></TD>
<%}%>

<td nowrap width="25%" height="30px" valign="center" align="right"><INPUT
type="text" class="prtlHdrToolAreaSearch" title="<%=searchInputTooltipStr%>"
size="20" value="" id="quickSearchField" name="QueryString"
onKeyDown="ToolAreaSearch_onReturnKey();" </td>

<td nowrap width="10%" height="30px" valign="center" align="right"><A href="#"
class="urBtnStd" onClick="openSearch();" onKeyDown="ToolAreaSearch_onSpaceKey();"
ct="Button" title="<%=searchButtonTooltipStr%>"><%=searchButtonTextStr%></A></td>

<td nowrap width="20%" height="30px" valign="center" align="left"><FONT SIZE="3"
color="red"><A href="#" class="prtlToolAreaTaskText" ct="Link"
onClick="openAdvancedSearch();"
title="<%=searchAdvancedLinkTooltipStr%>"><%=searchAdvancedLinkTextStr%></A></FON
T>
</td>
</TR>
</TABLE>
</TD>
</TR>
.

```

- Add code at the end of the masthead table for session management in the portal. You can use the following code -

```
String sessionManagerComponent=profile.getProperty("RtcSessionManagerComponent");
if(null == sessionManagerComponent)
{
sessionManagerComponent="com.sap.netweaver.coll.appl.ui.rtc.SessionManager";
}
componentURI.setContextName(sessionManagerComponent);
String sessionManagerComponentURI = componentURI.toString();
if(isEnabled(profile, ENABLE_RTC_SESSION_MANAGER))
{
%>
<td>
<IFRAME id="SessionManager" name="SessionManager" alt=""
SRC="<%=sessionManagerComponentURI + "?ENABLE=true"%>"
style='position:absolute;height:0;visibility:hidden;display:none'></IFRAME>
</td>
<%
}
else
{
%>
<td>
<IFRAME id="SessionManager" name="SessionManager" alt=""
SRC="<%=sessionManagerComponentURI + "?ENABLE=false"%>"
style='position:absolute;height:0;visibility:hidden;display:none'></IFRAME>
</td>
<%
}
```

Masthead Deployment and iView Creation

- Deploy the application *MyMastheadToolArea* using PAR file Export.
- Log on to the SAP Enterprise Portal where you have deployed the application.
- Now create an iView using Portal Component.
- Select your component (*MyMastheadToolArea*) and select *lightHeader*.
- Create the iView and see the preview.

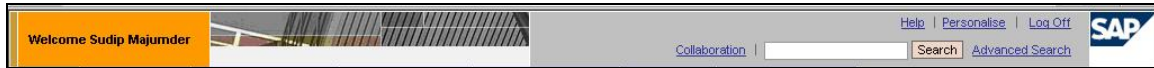


Figure 7 : Customized Masthead (with tool area merged with the masthead)

- The desired result is obtained. Add this iView to your framework page.
- In the framework page, remove the Tool Area, as the tool area is already integrated in the masthead.

Summary

Now we are able to merge the tool area components in to the masthead to implement client's requirement. As the tool area components are added as per the standard SAP code, the look and feel of these components can be easily customized from the theme editor, just like customizing the tool area.

Related Content

sdn.sap.com

help.sap.com

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.