



Part II: How To... Customize Look and Feel using Ajax Framework L- Shape API

Ajax Framework Features

Applicable Releases:

SAP NetWeaver 7.3 SP2

IT Practice:

User Productivity Enablement

IT Capability:

Running an Enterprise Portal

Version 1.0

February 2011



© Copyright 2011 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, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated 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, SAP NetWeaver, 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.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Document Version **Description**

1.00	First official release of this guide
------	--------------------------------------

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons





Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1.	Business Scenario	1
2.	Background Information	1
3.	Prerequisites	1
4.	Step-by-Step Procedure	2
4.1	Tutorial Setup	2
4.2	Changes in the Content Area	3
4.3	History, Favorites and Tabset Menus.....	4
4.3.1	Modify Portal Component	4
4.3.2	Create CSS menu.css file.....	5
4.3.3	Add the History, Favorites and Tabset's menus in the HTML Layout	5
4.3.4	Create JavaScript menu.js file	7
4.3.5	Deploying Portal Application	11
4.3.6	Configure Custom Framework Page in your Portal	11
5.	Related Links	14
6.	Appendix	15

1. Business Scenario

In the previous guide <http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/70bfe58d-1ed9-2d10-42b6-821d62777d6f> you learned how to create your own framework page using the Navigation capabilities of the L-Shape API (LSAPI). In this guide you will learn additional features of the Ajax Framework that will enable you to access and control the navigation history, favorites and tabsets.

2. Background Information

The L-Shape API (LSAPI) enables you to access and control the navigation history, favorites and tabsets:

- **The History API** enables you to access and control the navigation history and provides a set of methods to access and manipulate history entries. To access the methods of the History API you must use the following prefix:

```
EPCM.getSAPTop().LSAPI.getHistoryPlugin()
```

- **The Favorites API** enables you to add, delete or move favorites entries from one location to another. To access the methods of the Favorites API you must use the following prefix:

```
EPCM.getSAPTop().LSAPI.getFavoritesPlugin()
```

- **The Tabset API** enables you get and change the tabsets. To access the methods of the Tabset API you must use the following prefix:

```
EPCM.getSAPTop().LSAPI.getTabsetPlugin()
```

3. Prerequisites

The following is a list of all you need for developing your customer framework page.

- SAP NetWeaver 7.3 SP2 and NWDS.

Knowledge

- You have a basic knowledge of configuring SAP Portal Desktop
- You have completed Part I: How to... Customize Look & Feel using Ajax Framework L-Shape – Navigation capabilities <http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/70bfe58d-1ed9-2d10-42b6-821d62777d6f>

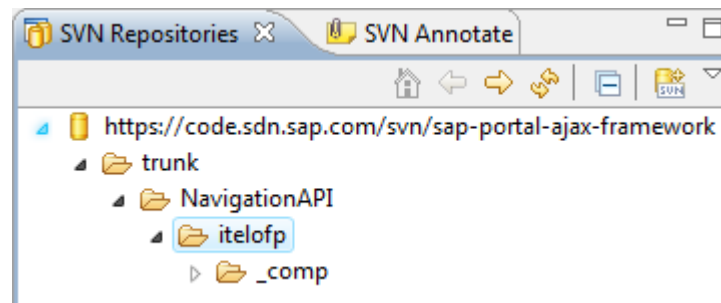
4. Step-by-Step Procedure

Upon completion of this document, you will be able to manipulate history navigation, favorites and tabsets using L-Shape API

4.1 Tutorial Setup

If you completed the previous guide <http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/70bfe58d-1ed9-2d10-42b6-821d62777d6f>, you can skip the steps of the Tutorial setup, otherwise you can get the code from the Code Exchange as explained in these steps:

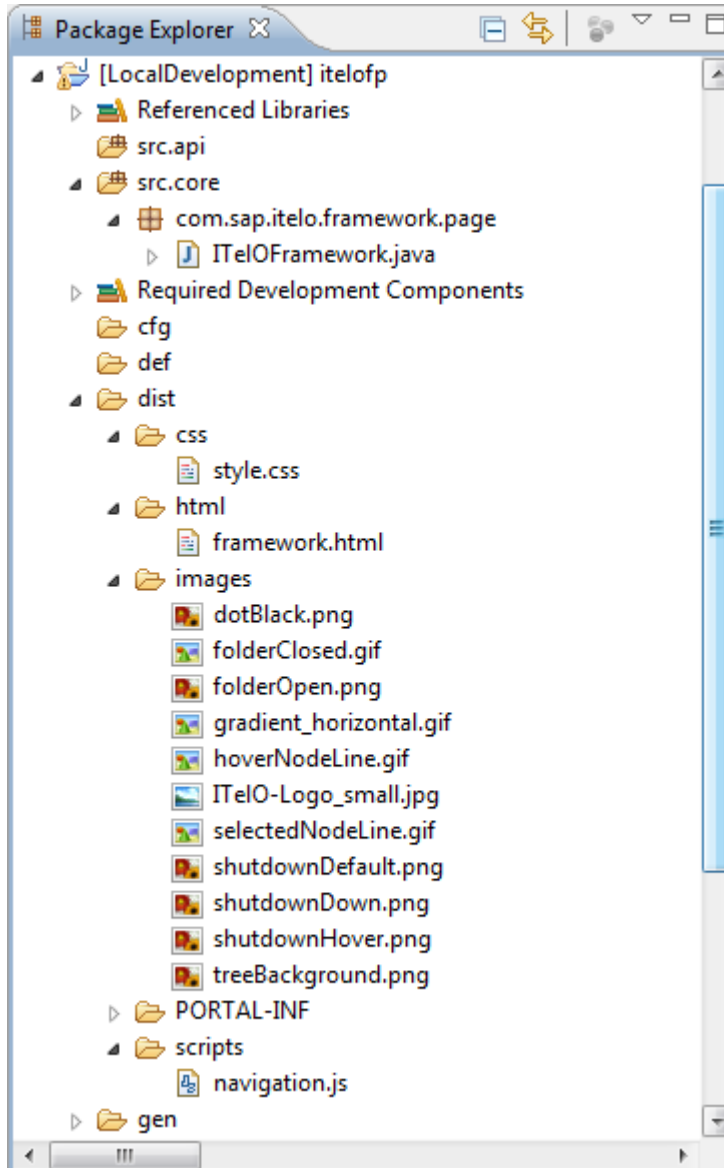
1. Connect SAP NetWeaver Developer Studio with the Code Exchange subversion repository <https://code.sdn.sap.com/svn/sap-portal-ajax-framework> and download the source code that is located in the *NavigationAPI* folder that contains the Portal DC required to continue with this tutorial



Note

For more information about how to connect the SAP NetWeaver Developer Studio with the code repository of Ajax Framework SDN code exchange project, please visit this [weblog](#)

2. Copy the content of the *itelofp* folder into the JDI workspace of the SAP NetWeaver Developer Studio under LocalDevelopment → DCs folder
3. Drill into LocalDevelopment → MyComponents
4. Right click on the *itelofp* component and select *Sync / Create Project* → *Create Project*
5. Follow the wizard and click Finish
6. Switch to the Enterprise Portal perspective, your project should look like the image below

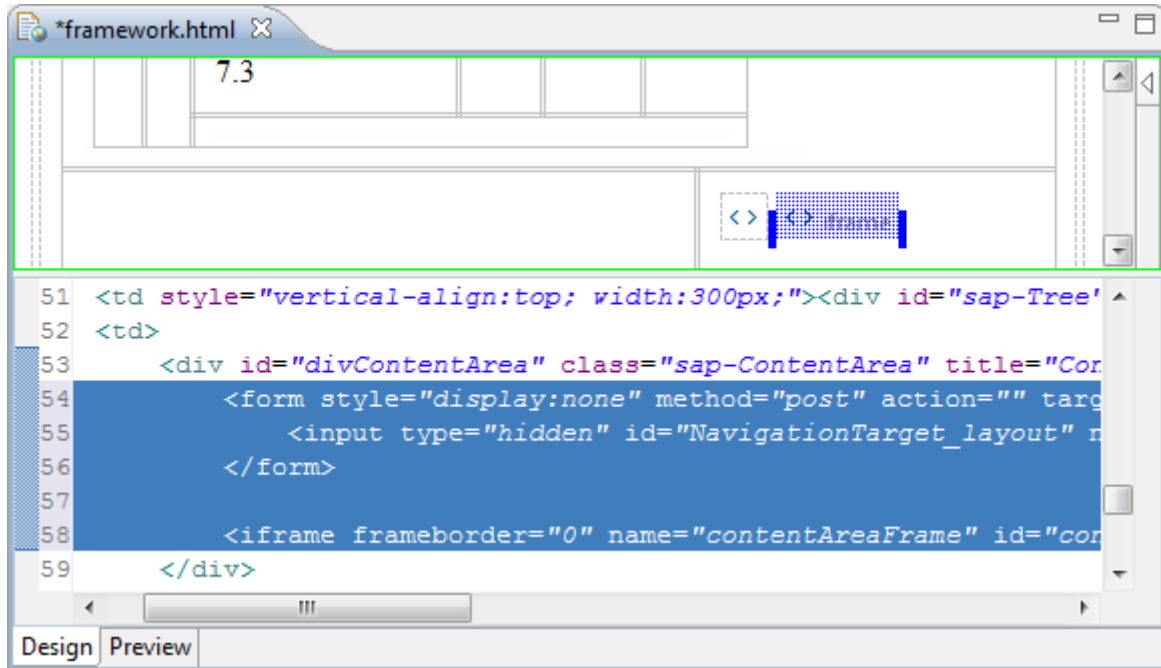


4.2 Changes in the Content Area

One of the improvements made in SAP NetWeaver 7.3 SP2 and higher versions include the initialization of the Ajax Framework with the JavaScript method:

`EPCM.getSAPTop().LSAPI.AFPPugin.service.initAjaxFrameworkPage()`. This means the use of the `iframe` and `form` tags, on the HTML document, are not longer needed. Please check [Appendix A – framework.html](#) for more information

1. Go to `dist` → `html` and open `framework.html`
2. Locate the `iframe` tag and remove the code shown in the below image



3. Add the following code in the body tag

```

<body
onload="EPCM.getSAPTop().LSAPI.AFPPlugin.service.initAjaxFrameworkPage();">

```

4. Save the changes

4.3 History, Favorites and Tabset Menus

4.3.1 Modify Portal Component

1. Open IteloPage.java located in src.core → com.sap.itelo.framework.page → IteloPage.java
2. Paste the following code into the portal component's doContent() method. This code attaches the new JavaScript and the CSS files needed to display the History, Favorites and Tabset's menus to the portal component.

```

//Menu elements
IResource menuCSS = request.getResource(IResource.CSS, "css/menu.css");
response.include(request, menuCSS);
IResource menuJs = request.getResource(IResource.SCRIPT,
"scripts/menu.js");
response.include(request, menuJs);

```

3. Save the changes

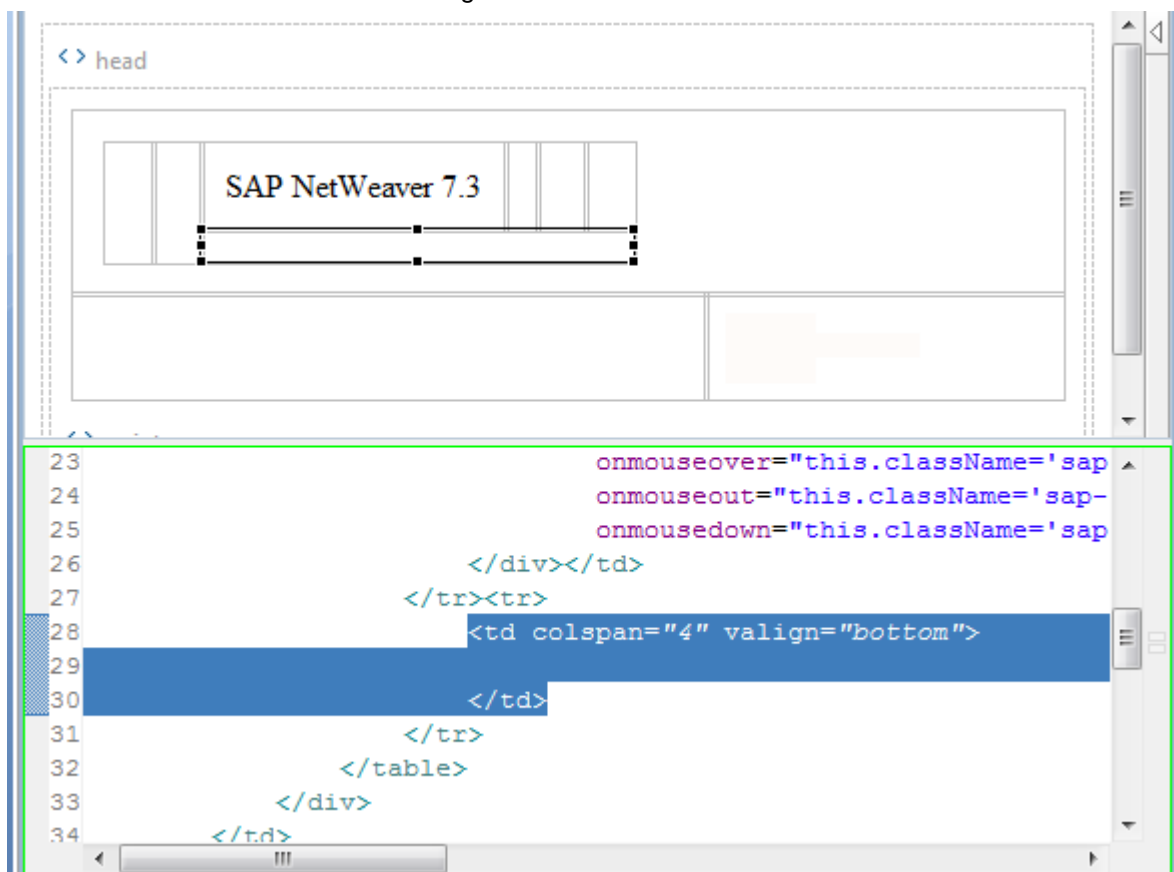
4.3.2 Create CSS menu.css file

1. Go to the *ITeloFramework* → *dist* → *css* folder
2. Select New → Other → Web → CSS from the context menu
3. Enter *menu* in the name field and click Finish
4. Open *meny.css* and copy the source code found in the [appendix B](#)
5. The CSS classes shown in the below image are part of the classes available in *menu.css*



4.3.3 Add the History, Favorites and Tabset's menus in the HTML Layout

1. Locate the row shown in the below image



- Copy and paste the following HTML code between the <TD> tags to add History, Menu and Tabset's menu

 **Note**

There are 2 elements associated with each menu

- Link element that generates dynamically the menu onclick event
- DIV element that contains the items of the menu

```
<table border="0" cellpadding="0" cellspacing="0" >
  <tr><td><ul id="sddm">
    <li><a href="#" onmouseover="mopen('history') "
onmouseout="mclose()" >History</a>
      <div id="history" onmouseover="mclose()" "
onmouseout="mclose()" "></div>
    </li>
  </ul></td>
  <td><ul id="sddm">
    <li><a href="#" onmouseover="mopen('favorites') "
onmouseout="mclose()" >Favorites</a>
      <div id="favorites" onmouseover="mclose()" "
onmouseout="mclose()" "></div>
    </li>
  </ul></td>
  <td><ul id="sddm">
    <li><a href="#" onmouseover="mopen('tabsets') "
onmouseout="mclose()" >Tabsets</a>
      <div id="tabsets" onmouseover="mclose()" "
onmouseout="mclose()" "></div>
    </li></ul></td></tr>
</table>
```

- In the HTML code, add a JavaScript code with the following code. This is to register a notification function each time users switch Tabsets by calling `registerOnTabsetSwitch`. The notification function `refreshMenu` will be implemented later in this guide

```
<script type="text/javascript">
EPCM.getSAPTop().LSAPI.getTabsetPlugin().registerOnTabsetSwitch (refreshM
enu);
initTree();
</script>
```

 **Important**

`EPCM.getSAPTop().LSAPI.getTabsetPlugin().registerOnTabsetSwitch` registers a notification function to call upon tabset switch.

- Save the changes

4.3.4 Create JavaScript menu.js file

Please check [Appendix C – menu.js](#) for more details

1. Go to the *ITeloFramework* → *dist* → *scripts* folder
2. Select *New* → *Other* → *JavaScript* → *JavaScript Source File* from the context menu
3. Enter *menu* in the *Name* field and click *Finish*
4. Declare the following global variables

```
var timeout = 500;
var closetimer = 0;
var ddmenuitem = 0;
```

5. Add `refreshHTMLContainer` function. For simplicity purposes the items from the menu are removed before populating the new list. You should consider a better way to keep the menus up to date

```
function refreshHTMLContainer(container) {
    // Remove Children
    if (container.hasChildNodes()) {
        while (container.childNodes.length >= 1) {
            container.removeChild(container.firstChild);
        }
    }
}
```

6. Add `createMenuItem` function. This function will add a menu item in the corresponding menu.

```
function createMenuItem(container, name, link) {
    var title = document.createElement("a");
    title.innerHTML = name;
    title.href = "javascript:void(0)";
    title.onclick = link;
    container.appendChild(title);
}
```

7. Add `mopen` (id) function. This function handles the visibility of the menus

```
function mopen(id) {
    // cancel close timer
    mcancelclosetime();

    if (id == 'history') {
        getHistoryMenu();
    } else if (id == 'favorites') {
        getFavoriteMenu();
    } else {
        getTabsetMenu();
    }
    // close old layer
    if (ddmenuitem)
        ddmenuitem.style.visibility = 'hidden';
    // get new layer and show it
    ddmenuitem = document.getElementById(id);
    ddmenuitem.style.visibility = 'visible';
}
```

8. Add `getHistoryMenu` function

```
function getHistoryMenu () {
    var historyMenu = document.getElementById("history");
    refreshHTMLContainer(historyMenu);

    // Get History entries
    var historyArray =
    EPCM.getSAPTop().LSAPI.getHistoryPlugin().getEntries();
    var funcArr = [];
    for ( var i = 0; i < historyArray.length; i++) {
        funcArr.push(function(val) {
            return function() {
                EPCM.doNavigate(val);
            }
        })(historyArray[i].getUrl())

        createMenuItem(historyMenu, historyArray[i].title,
            funcArr.shift())
    }
}
```

 **Important**

`EPCM.getSAPTop().LSAPI.getHistoryPlugin().getEntries()` returns all history entries as an array of `HistoryEntry` objects

These are the properties and methods of the `HistoryEntry` object that are used in this example

- `historyEntry.title` Represents the title of the navigation node
- `historyEntry.getUrl()` Returns the ID (URL) of the navigation node

9. Add `getFavoriteMenu` function

```
function getFavoriteMenu() {
    var favoriteMenu = document.getElementById("favorites");
    // Get Favorite Provider entries
    EPCM.getSAPTop().LSAPI.getFavoritesPlugin().getFavoritesProviders
    (drawFavorites, favoriteMenu);
}
```

 **Important**

`EPCM.getSAPTop().LSAPI.getFavoritesPlugin().getFavoritesProviders(callback, arguments)` returns an array of favorites providers assigned to the current user

- param `callback` - A notification function to call with the favorites providers
- param `arguments` - A JavaScript object passed to the callback function

10. Add the callback function `drawFavorites` that is responsible for rendering the Favorite's menu

```
function drawFavorites(favoriteProviders, favoriteMenu){
    // Removing Children
    refreshHTMLContainer(favoriteMenu);

    var selectedNode =
        EPCM.getSAPTop().LSAPI.AFPPlugin.model.getCurrentLaunchNode();
    var title = selectedNode.getTitle();
    var url = selectedNode.getName();

    var funcArr = [];
    funcArr.push(function(nodeURL, title, callback, arguments) {
        return function() {
            EPCM.getSAPTop().LSAPI.getFavoritesPlugin().addFavorite
                (nodeURL, title, callback, arguments);
        }
    })(url, title, notify, title))

    createMenuItem(favoriteMenu, "Add to Favorites", funcArr.shift())

    for ( var i = 0; i < favoriteProviders.length; i++) {

        var favorites = favoriteProviders[i].getFavorites();
        for ( var j = 0; j < favorites.length; j++) {
            funcArr.push(function(val) {
                return function() {
                    EPCM.doNavigate(val);
                }
            })(favorites[j].getURL())

            createMenuItem(favoriteMenu, favorites[j].getTitle(),
                funcArr.shift());
        }
    }
}
```

Important

`EPCM.getSAPTop().LSAPI.AFPPlugin.model.getCurrentLaunchNode()` returns A `NavigationNode` object representing the currently active navigation node

`EPCM.getSAPTop().LSAPI.getFavoritesPlugin().addFavorite(nodeURL, title, callback, arguments)` adds a favorite for the current user

- param `nodeURL` - The navigation node URL
- param `title` - the title of the favorite
- param `callback`- A notification function to call when this favorite is selected
- param `arguments` - A JavaScript object passed to the callback function

`favoriteProviders[i].getFavorites()` Returns an array of favorites assigned to this Favorites Provider

11. Create the callback function notify, which is passed as a parameter in the addFavorite function

```
function notify(title){
    alert("The favorite "+title+" has been added successfully");
}
```

12. Add the function getTabsetMenu

 **Note**

Tabsets are subsets of entry points which allow users to see only a specific set of entry points in the top-level navigation. For more information on how to enable/disable the tabsets feature, please visit [Creating Tabsets](#)

```
function getTabsetMenu () {
    var tabsetsMenu = document.getElementById("tabsets");
    refreshHTMLContainer(tabsetsMenu);

    // Get Tabset entries
    var funcArr = [];
    var tabsetArray =
        EPCM.getSAPTop().LSAPI.getTabsetPlugin().getTabsets();
    for ( var i = 0; i < TabsetArray.length; i++) {
        funcArr.push(function (val) {
            return function () {
                EPCM.getSAPTop().LSAPI.getTabsetPlugin().
                    setCurrentTabset(val);
            }
        })(tabsetArray[i].id)

        createMenuItem(tabsetsMenu, tabsetArray[i].name, funcArr.shift());
    }
}
```

 **Important**

EPCM.getSAPTop().LSAPI.getTabsetPlugin().setCurrentTabset(tabsetId)
sets the tabset as current

EPCM.getSAPTop().LSAPI.getTabsetPlugin().getTabsets() returns all available
tabsets as an array of Tabset objects

These are the properties and methods of the Tabset object that are used in this example

- tabsetEntry.name Represents the name of the tabset
- tabsetEntry.id Returns the ID of the tabset

13. Create a function to refresh the navigation tree when users switch Tabsets

```
function refreshMenu() {
    var navigationContainer = document.getElementById("sap-Tree");
    refreshHTMLContainer(navigationContainer);
    initTree();
}
```

14. Complete the menu.js with the following functions that handle the menu visibility

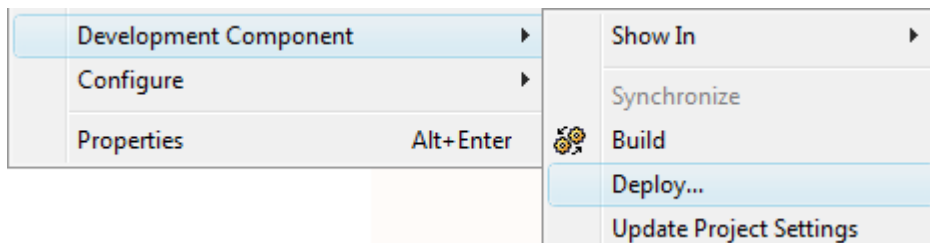
```
// close showed layer
function mclose() {
    if(ddmenuitem) ddmenuitem.style.visibility = 'hidden';
}
// go close timer
function mclosetime() {
    closetimer = window.setTimeout(mclose, timeout);
}
// cancel close timer
function mcancelclosetime(){
    if(closetimer) {
        window.clearTimeout(closetimer);
        closetimer = null;
    }
}
// close layer when click-out
document.onclick = mclose;
```

15. Save the changes

4.3.5 Deploying Portal Application

Once you have updated the portal application, you need to deploy it to the portal

1. Choose your Portal Application Development component
2. Right click and choose *Development Component* → *Build* from the context menu
3. Right click and choose *Development Component* → *Deploy* from the context menu



4.3.6 Configure Custom Framework Page in your Portal

If you completed the previous guide <http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/70bfe58d-1ed9-2d10-42b6-821d62777d6f>, you can skip this configuration step

1. Open a web browser and go to <http://<host>:<port>/irj/portal/itelo>

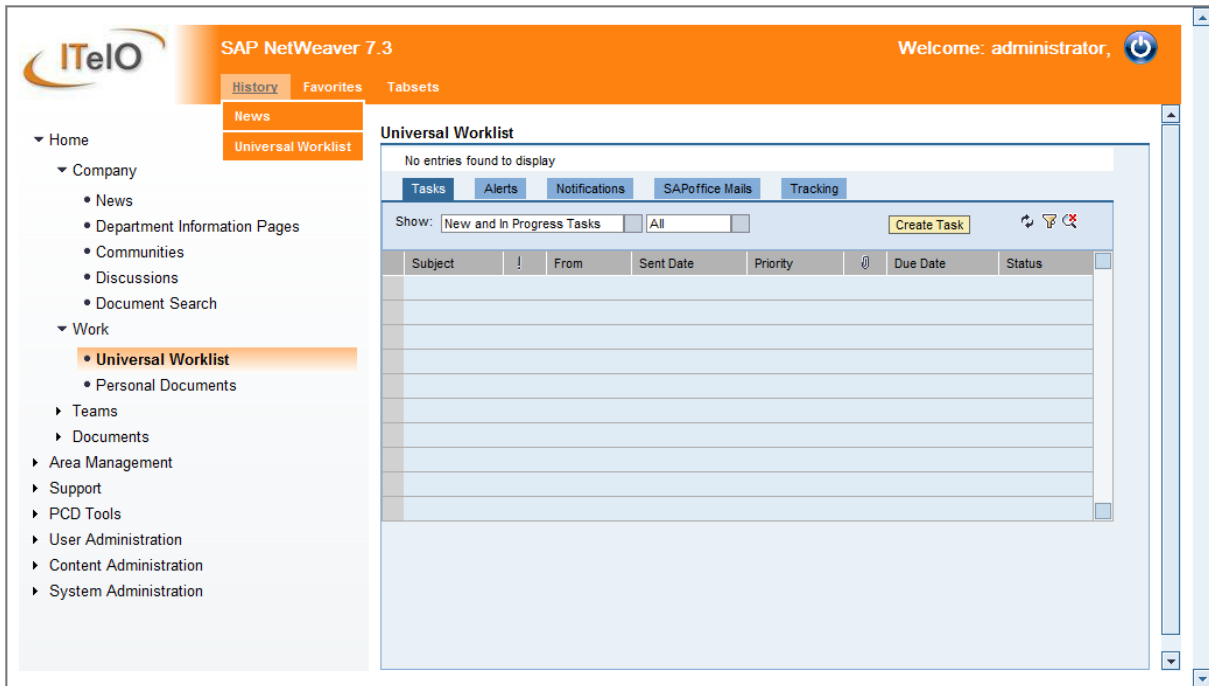
Otherwise please follow step 4.2 *From the Portal* found in the previous guide

<http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/70bfe58d-1ed9-2d10-42b6-821d62777d6f>, This step includes

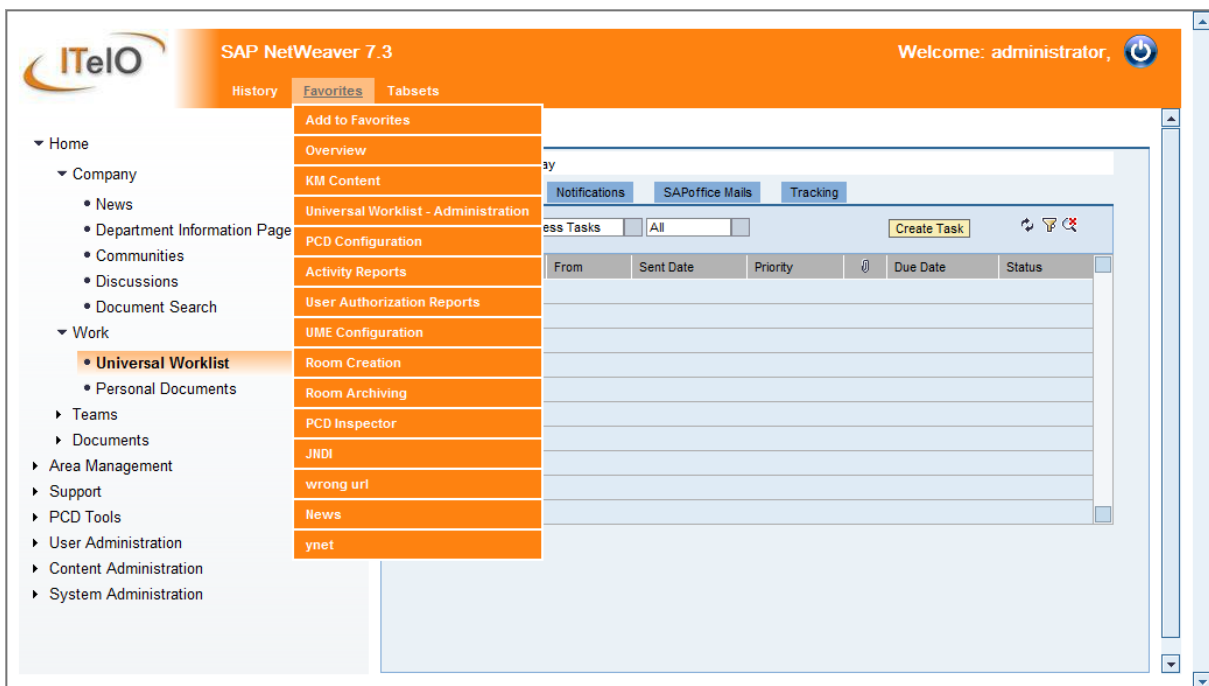
1. Creating iView
2. Creating new Framework Page
3. Creating new Desktop
4. Creating new Desktop rule

Your portal should look like the image below

History Menu



Favorites menu



Tabsets menu

The screenshot displays the SAP NetWeaver 7.3 user interface. At the top, the header bar includes the ITelO logo, the text 'SAP NetWeaver 7.3', and a user greeting 'Welcome: administrator,' with a power icon. Below the header, there are navigation tabs for 'History', 'Favorites', and 'Tabsets'. The 'Tabsets' menu is open, showing options for 'All', 'Administration', and 'Home'. The 'Home' option is selected, leading to a sub-menu with tabs for 'Alerts', 'Notifications', 'SAPoffice Mails', and 'Tracking'. The 'Alerts' tab is active, displaying a 'Show:' dropdown set to 'New and In Progress Tasks' and a 'Create Task' button. Below this is a table with the following columns: Subject, From, Sent Date, Priority, Due Date, and Status. The table is currently empty.

5. Related Links

[Part I: How To Customize Look & Feel using Ajax Framework L-Shape API](#)

6. Appendix

Appendix A – framework.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
onload="EPCM.getSAPTop().LSAPI.AFPPlugin.service.initAjaxFrameworkPage();"

<div class="sap-Framework">
  <table style="width:100%; height:100%" border="0" cellpadding="0"
cellspacing="0">
    <tr>
      <td colspan="2">
        <div class="sap-pageFooter">
          <table border="0" cellpadding="0" cellspacing="0">
            <tr>
              <td rowspan="2"><div class="sap-logoImage"></div></td>
              <td rowspan="2"><div class="sap-logogradient"></div></td>
              <td><div class="sap-welcomeMessage">SAP NetWeaver
7.3</div></td>
              <td style="width: 100%"></td>
              <td><div id="welcomeMessage" class="sap-
welcomeMessage"></div></td>
              <td><div class="sap-logOff"
onclick="LSAPI.sessionPlugin.logoff()"
onmouseover="this.className='sap-logOff sap-logOff-
hover'"
onmouseout="this.className='sap-logOff'"
onmousedown="this.className='sap-logOff sap-logOff-
down'">
                </div></td>
            </tr><tr>
              <td colspan="4" valign="bottom">
                <table border="0" cellpadding="0" cellspacing="0" >
                  <tr><td><ul id="sddm">
                    <li><a href="#" onmouseover="mopen('history')"
onmouseout="mcloseTime()">History</a>
                    <div id="history"
onmouseover="mcancelcloseTime()" onmouseout="mcloseTime()"></div>
                    </li>
                  </ul></td>
                  <td><ul id="sddm">
                    <li><a href="#" onmouseover="mopen('favorites')"
onmouseout="mcloseTime()">Favorites</a>
                    <div id="favorites"
onmouseover="mcancelcloseTime()" onmouseout="mcloseTime()"></div>
                    </li>
                  </ul></td>
                  <td><ul id="sddm">
                    <li><a href="#" onmouseover="mopen('tabsets')"
onmouseout="mcloseTime()">Tabsets</a>
                    <div id="tabsets"
onmouseover="mcancelcloseTime()" onmouseout="mcloseTime()"></div>

```

```

        </li></ul></td></tr>
    </table>
</td>
</tr>
</table>
</div>
</td>
</tr>

<tr class="sap-mainNavigationTR">
    <td class="sap-MenuColumn" ><div id="sap-Tree" class="sap-
Tree"></div></td>
    <td class="sap-ContentColumn">
        <div id="divContentArea" title="Content Area - To enter press
Tab">
            </div>
        </td>
    </tr>
</table>
</div>

<script type="text/javascript">
    EPCM.getSAPTop().LSAPI.getTabsetPlugin().registerOnTabsetSwitch(refreshM
enu);
    initTree();
</script>

</body>
</html>

```

Appendix B – menu.css

Menu.css source code

```

@CHARSET "UTF-8";
/* --- menu styles ---
#sddm {
    margin: 0;
    padding: 0;
    z-index: 30
}

#sddm li {
    margin: 0;
    padding: 0;
    list-style: none;
    float: left;
    font: bold 11px arial
}

#sddm li a {
    display: block;
    margin: 0 1px 0 0;
    padding: 4px 10px;
    width: 60px;
    background: #fe8210;
    color: #FFF;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
}

```

```
#sddm li a:hover {
  background: #febd81;
  color: #6b6b6b;
  text-decoration: underline;
}

#sddm div {
  position: absolute;
  visibility: hidden;
  margin: 0;
  padding: 0;
  background: #fe8210;
  border: 1px solid #ffffff
}

#sddm div a {
  position: relative;
  display: block;
  margin: 0;
  padding: 5px 10px;
  width: auto;
  white-space: nowrap;
  text-align: left;
  text-decoration: none;
  background: #fe8210;
  color: #ffffff;
  font: 11px arial;
  border: 1px solid #ffffff;
  font-weight: bold;
}

#sddm div a:hover {
  background: #febd81;
  color: #6b6b6b;
  text-decoration: underline;
}
```

Go [back](#)

Appendix C – menu.js

```
var timeout = 500;
var closetimer = 0;
var ddmenuitem = 0;

function mopen(id) {
  // cancel close timer
  mcancelclosetime();

  if (id == 'history') {
    // Get History elements form History API
    getHistoryMenu();
  } else if (id == 'favorites'){
    getFavoriteMenu();
  } else {
    getTabsetMenu();
  }
}
```

```

// close old layer
if (ddmenuitem)
    ddmenuitem.style.visibility = 'hidden';

// get new layer and show it
ddmenuitem = document.getElementById(id);
ddmenuitem.style.visibility = 'visible';
}

function getHistoryMenu() {
    var historyMenu = document.getElementById("history");
    refreshHTMLContainer(historyMenu);

    // Get History entries
    var historyArray =
EPCM.getSAPTop().LSAPI.getHistoryPlugin().getEntries();
    var funcArr = [];
    for ( var i = 0; i < historyArray.length; i++) {
        funcArr.push(function(val) {
            return function() {
                EPCM.doNavigate(val);
            }
        })(historyArray[i].getUrl())

        createMenuItem(historyMenu, historyArray[i].title, funcArr.shift())
    }
}

function getFavoriteMenu() {
    var favoriteMenu = document.getElementById("favorites");
    // Get Favorite Provider entries
    EPCM.getSAPTop().LSAPI.getFavoritesPlugin().getFavoritesProviders(drawFavorites, favoriteMenu);
}

function drawFavorites(favoriteProviders, favoriteMenu){
    refreshHTMLContainer(favoriteMenu);

    var selectedNode =
EPCM.getSAPTop().LSAPI.AFPPlugin.model.getCurrentLaunchNode();
    var title = selectedNode.getTitle();
    var url = selectedNode.getName();

    var funcArr = [];
    funcArr.push(function(nodeURL,title,callback,arguments) {
        return function() {
            EPCM.getSAPTop().LSAPI.getFavoritesPlugin().
                addFavorite(nodeURL,title,callback,arguments);
        }
    })(url,title,notify,title))

    createMenuItem(favoriteMenu,"Add to Favorites", funcArr.shift())

    for ( var i = 0; i < favoriteProviders.length; i++) {

        var favorites = favoriteProviders[i].getFavorites();
        for ( var j = 0; j < favorites.length; j++) {
            funcArr.push(function(val) {
                return function() {

```

```

        EPCM.doNavigate(val);
    }
    }(favorites[j].getURL()))

    createMenuItem(favoriteMenu, favorites[j].getTitle(),
funcArr.shift());

    }
}

function notify(title){
    alert("The favorite "+title+" has been added successfully");
}

function getTabsetMenu (){
    var tabsetsMenu = document.getElementById("tabsets");
    refreshHTMLContainer(tabsetsMenu);

    // Get Tabset entries
    var funcArr = [];
    var TabsetArray = EPCM.getSAPTop().LSAPI.getTabsetPlugin().getTabsets();
    for ( var i = 0; i < TabsetArray.length; i++) {
        funcArr.push(function(val) {
            return function() {
                EPCM.getSAPTop().LSAPI.getTabsetPlugin().setCurrentTabset(val);
            }
        })(TabsetArray[i].id)

        createMenuItem(tabsetsMenu, TabsetArray[i].name, funcArr.shift());
    }
}

function refreshMenu(){
    var navigationContainer = document.getElementById("sap-Tree");
    refreshHTMLContainer(navigationContainer);
    initTree();
}

//Remove elements from a container
function refreshHTMLContainer(container){
    // Remove Children
    if (container.hasChildNodes()) {
        while (container.childNodes.length >= 1) {
            container.removeChild(container.firstChild);
        }
    }
}

function createMenuItem(container, name, link){
    var title = document.createElement("a");
    title.innerHTML = name;
    title.href = "javascript:void(0)";
    title.onclick = link;
    container.appendChild(title);
}

// close showed layer
function mclose() {
    if (ddmenuitem)

```



```
        ddmenuitem.style.visibility = 'hidden';
    }

    // go close timer
    function mclosetime() {
        closetimer = window.setTimeout(mclose, timeout);
    }
    // cancel close timer
    function mcancelclosetime() {
        if (closetimer) {
            window.clearTimeout(closetimer);
            closetimer = null;
        }
    }
    // close layer when click-out
    document.onclick = mclose;
```

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