



How-To...Create an Instant Navigation Launcher Portal Application using Navigation APIs with Ajax Framework

Applicable Releases:

SAP NetWeaver 7.0 EhP2 and above

IT Practice:

User Productivity Enablement

IT Scenario:

Running an Enterprise Portal

Version 1.0

July 2010



© Copyright 2010 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 expressively 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	Creating a Portal Application Project	2
4.2	Creating an AbstractPortalComponent	3
4.3	Writing JSP.....	4
4.4	Attach JSP and JavaScript files to Portal Component.....	5
4.5	Creating UI and calling JavaScripts in JSP.....	5
4.6	Deploy the application.....	9
4.7	Integrate and test the application	11

1. Business Scenario

This tutorial walks you through the steps necessary to implement a basic navigation launch utility using the Navigation API.

You create a navigation iView with an input text box. In this box, you can enter the navigation node which you want to launch. The text box has an auto-complete capability and displays a drop-down list of available navigation nodes. Once you press Enter, the selected node is launched.

2. Background Information

SAP NetWeaver 7.0 EhP1 introduced the Ajax Framework and it is available for use with SAP NetWeaver 7.0 EhP2. This leads to harmonization of UI across SAP Products.

The Ajax Framework Page (AFP) is based on the Ajax technology that provides improved performance, dynamic navigation and enhanced user experience. The Ajax technology determines that most of the page's functionality takes place on the client side, for example, navigation hierarchy is cached on the client side.

AFP provides the JavaScript client-side Navigation API that exposes the navigation hierarchy and provides functionality to control navigation in the Portal. You can use the Navigation API to trigger and manipulate the navigation iViews and/or regular portal navigation.

3. Prerequisites

The following are the prerequisites that must be in place before using this How-To-Guide.

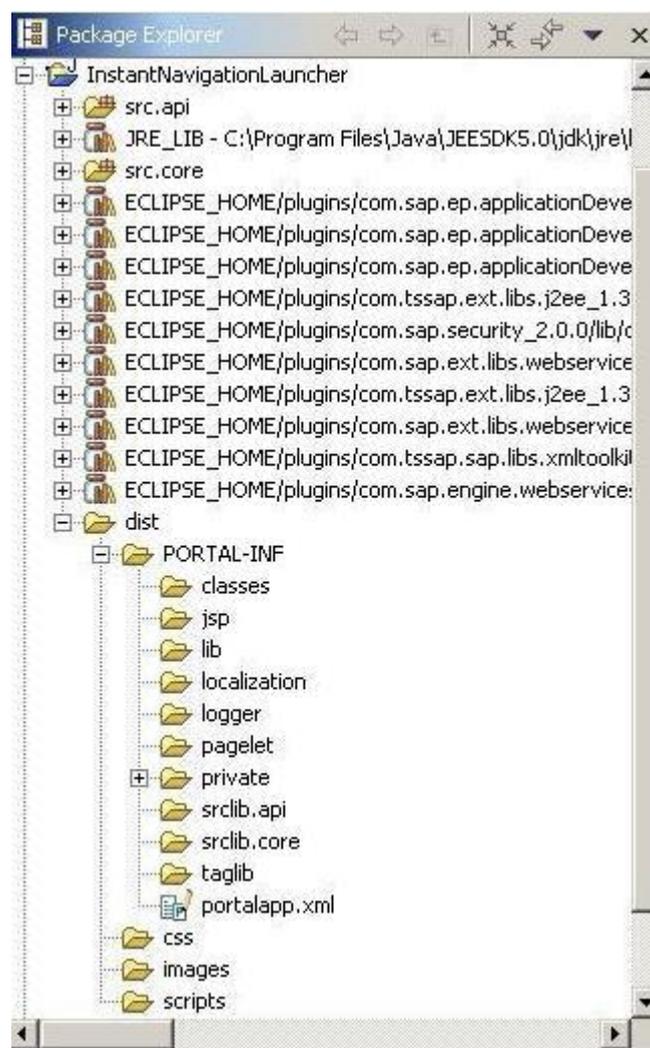
- SAP NetWeaver 7.0 EhP2 Java Server and NetWeaver Developer Studio
- Experience in JavaScript programming
- Familiarity with DOM. More information: <http://www.w3.org/DOM>
- Familiarity with portal application development.
- More information: Running an Enterprise Portal:
http://help.sap.com/saphelp_nw70ehp1/helpdata/EN/19/4554426dd13555e10000000a1550b0/frameset.htm in the SAP NetWeaver Developer's Guide
- Refer to SAP Note 1166135 (Central note for Ajax Framework Page Navigation). This note includes known issues and limitations

4. Step-by-Step Procedure

In this guide, we will cover the steps to develop the instant navigation launcher application, deploy and integrate the application to Portal.

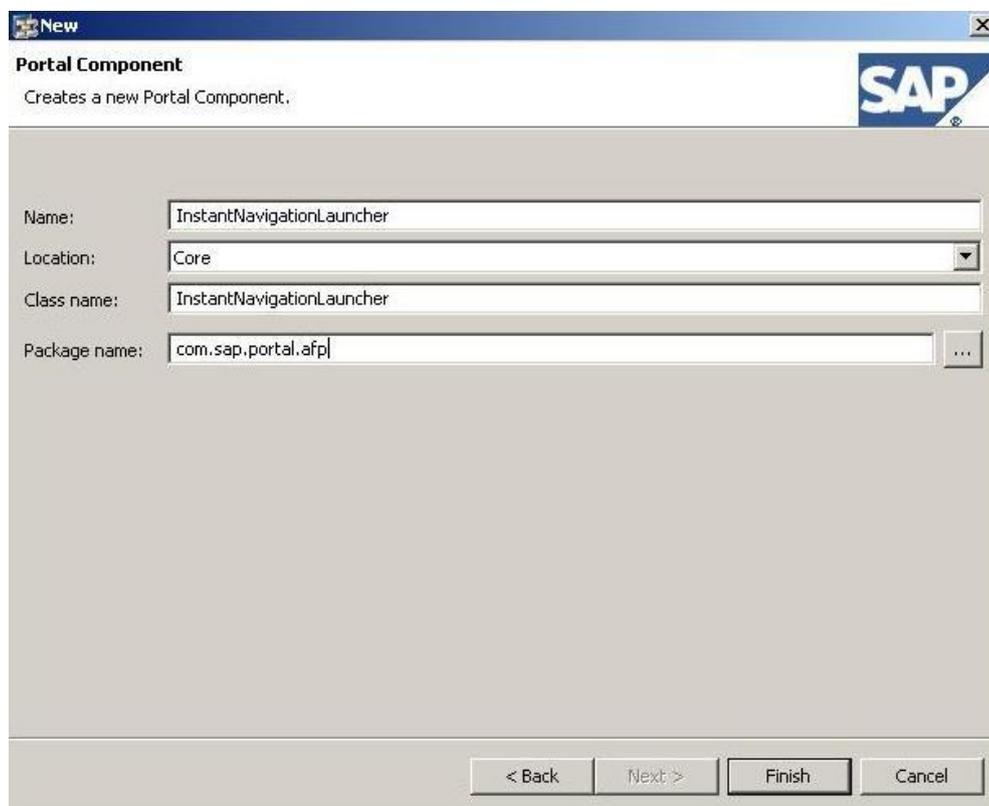
4.1 Creating a Portal Application Project

1. In SAP NetWeaver Developer Studio, open the Enterprise Portal perspective.
2. From the File menu, select New → Other. The New window is displayed.
3. Select Portal Application, and then Create a Portal Application Project. Click Next.
4. In the Project name textbox, enter a name for the project, eg: InstantNavigationLauncher. Click Finish.
5. The wizard creates the following. Directory structure for the project, as shown below:



4.2 Creating an AbstractPortalComponent

1. From the File menu, select New → Other. The New window is displayed.
2. Select Portal Application, and then Create a New Portal Application Object. Click Next.
3. Select the project i.e. InstantNavigationLauncher to which you want to add the portal component. Click Next.
4. Select Portal Component → AbstractPortalComponent. Click Next.
5. Enter the following fields:
 - Name: Name of component, which is defined by the name attribute of the <component> element for this component in the portalapp.xml. **Eg. InstantNavigationLauncher**. By default, this is also the name of the Java class file for this component.
 - Location: The location of the new Java class file for this component, which determines whether other applications can reference the classes. Generally, the choice is only meaningful when defining services and the interfaces to make available to those applications the reference the service. You can specify one of the following. In this example we will select the option **Core**.
 - Core (default): Places the file in the src.core project directory, and makes it private.
 - API: Places the file in the src.api, and makes it public.
 - Class name: The Java class file for this component.
 - Package: In this example, we use `com.sap.portal.afp`



New

Portal Component
Creates a new Portal Component.

Name: InstantNavigationLauncher

Location: Core

Class name: InstantNavigationLauncher

Package name: com.sap.portal.afp

< Back Next > Finish Cancel

6. The wizard does the following:

- Creates a class in the src.core folder called InstantNavigationLauncher that extends AbstractPortalComponent. A blank doContent method is also created, which is the only method that must be implemented.
- Creates an entry for the component in the application's portalapp.xml file, as shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <application-config/>
  <components>
    <component name="InstantNavigationLauncher">
      <component-config>
        <property name="ClassName" value="com.sap.portal.afp.InstantNavigationLauncher"/>
      </component-config>
      <component-profile/>
    </component>
  </components>
  <services/>
</application>
```

4.3 Writing JSP

1. Go to dist → PORTAL-INF → jsp subfolder
2. Right click the jsp subfolder and select New → Other
3. Go to Simple → File. Click Next



4. Enter the File Name as InstantNavigationLauncher.jsp
5. Click Finish.

4.4 Attach JSP and JavaScript files to Portal Component

1. Paste the following code into the portal component's doContent() method. This code attaches the JSP file to the portal component.
2. Organize import using CTRL + SHIFT + O

```
public void doContent(IPortalComponentRequest request,
    IPortalComponentResponse response)
{
    IResource jspResource = request.getResource("jsp",
        "jsp/InstantNavigationLauncher.jsp");
    response.include(request, jspResource);
}
```

Note

If you have used any JavaScript file(s) eg. instantnavigationlauncher.js, you can attach them to Portal Component using the code below. Here we assume that the JavaScript file is created in the scripts folder.

```
IResource jsResource = request.getResource(IResource.SCRIPT,
    "scripts/instantnavigationlauncher.js");
response.include(request, jsResource);
```

4.5 Creating UI and calling JavaScripts in JSP

Add the following code fragments to the InstantNavigationLauncher.jsp file:

1. Create an input text box element named auto_text_box within a <table> element. Set the OnKeyPress event to the quickLaunchNavigate function call that performs navigation to the node specified in the text box.

```
<%@ page
import="com.sapportals.portal.prt.component.IPortalComponentRequest" %>
<table>
<tr>
<td nowrap="nowrap" id="auto_complete">
<input type='text' style='font-family:Arial;width:150px;font-size:11px' id='auto_text_box' value=''
```

```
        onKeyPress='quickLaunchNavigate(this,event)'/>
    </td>
</tr>
</table>
```

2. Declare and initialize the global variables needed to store the navigation tree nodes, and add the quickLaunchInit function. This function wraps the EPCM.getSAPTop().LSAPI.AFPPlugin.model.getNavigationSubTree method call that retrieves the initial nodes of the user's navigation tree and passes the results to the parseNavigationNodes callback function.

```
<script language="javascript">
    var arrayIndex = 0;
    var navigationNodesArray = [];
    var navigationNodesHash = [];
    var quickLaunchInit = function() {
        // get all the initial nodes
        EPCM.getSAPTop().LSAPI.AFPPlugin.model.getNavigationSubTree
            (null,parseNavigationNodes);
    }
}
```

3. Add the parseNavigationNodes callback function that receives the initial nodes and then parses all nodes to retrieve the entire navigation tree and store the data locally in two arrays. One of the arrays stores the titles of the nodes for the autoComplete function, and the other stores the URLs for actual navigation to the selected nodes.

```
var parseNavigationNodes = function(nodes, args) {
    if (!nodes) return;
    for (var i = 0; i < nodes.length; i++) {
        var node, title, nodeUri, hasChildren;
        node = nodes[i];
        nodeUri = node.getNodeURI();
        hasChildren = node.hasChildren();
        title = node.getTitle();
        if (navigationNodesHash[title] == null) {
            navigationNodesArray[arrayIndex++] = title;
            navigationNodesHash[title] = nodeUri;
        }
        // add the node to the array
        // if the node has children, perform recursive call
    }
}
```

```
    if (hasChildren){
        node.getChildren(parseNavigationNodes);
    }
}
var obj = autoComplete(document.getElementById
('auto_text_box'),navigationNodesArray);
}
```

4. Add the autoComplete function that completes input of the text box element with the valid values from the array of the node titles.

```
var autoComplete = function(textBoxElm, validValues){
    textBoxElm.onkeyup = function(e){
        var evt = e || window.event;
        var match = false;
        validValues.selectedIndex = -1;
        for(var i = 0; i < validValues.length; i++){
            if(validValues[i].toUpperCase().indexOf(textBoxElm.value.toUpperCase())
            == 0){
                match=true;
                validValues.selectedIndex = i;
                break;
            }
        }
        if (textBoxElm.createTextRange()){
            if(!match){
                textBoxElm.value=textBoxElm.value.substring(0,textBoxElm.value.length-
                1);
                return;
            }
            var ignoredKeys ="8;46;37;38;39;40;33;34;35;36;45;";
            if (ignoredKeys.indexOf(evt.keyCode+";") == -1){
                var curMatchedText = textBoxElm.createTextRange();
                var oldValue = curMatchedText.text;
                var newValue = match ? validValues[i] : oldValue;
                if(newValue != textBoxElm.value){
                    textBoxElm.value = newValue;
                }
            }
        }
    }
}
```

```
        var addedChars = textBoxElm.createTextRange();
        addedChars.moveStart('character', oldValue.length);
        addedChars.select();
    }
}
}
```

5. Add the quickLaunchNavigate function that performs navigation to the specified node title using the navigate method of the LSAPI.

```
var quickLaunchNavigate = function() {
    var keycode;
    if (window.event) keycode = window.event.keyCode;
    if (keycode == 13) {
        var title = document.getElementById('auto_text_box').value;
        var nodeURI = navigationNodesHash[title];
        // performing the navigation
        if (nodeURI != null)
            EPCM.getSAPTop().LSAPI.AFPPlugin.service.navigate(nodeURI);
    }
}
```

6. Add the clearLaunchedURL function to clear the input text box after navigation.

```
var clearLaunchedURL = function() {
    document.getElementById('auto_text_box').value = '';
}
```

7. Use the registerOnNavigate method of LSAPI to call the clearLaunchedURL function after each navigation event.

```
EPCM.getSAPTop().LSAPI.AFPPlugin.controller.registerOnNavigate(clearLaunchedURL);
```

8. Subscribe to the browser load event that triggers the quickLaunchInit function.

```
EPCM.subscribeEvent("urn:com:sapportals.portal:browser", "load", quickLaunchInit);
```

9. Call the `notifyPanelIfHasContent` method of LSAPI to ensure that the `iView` created from your portal component is displayed in the Left Navigation Panel Container.

```
EPCM.getSAPTop().AFPLayoutAPI.notifyPanelIfHasContent(true);  
</script>
```

4.6 Deploy the application

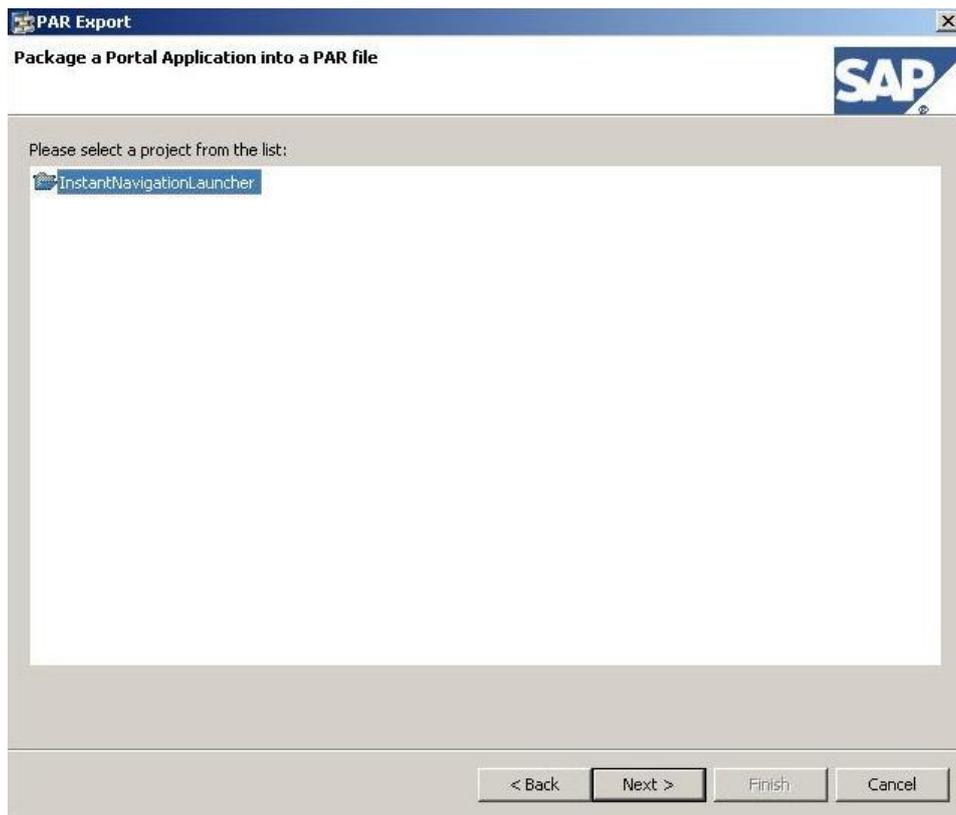
Once you have created a Portal Application, you need to deploy it on the Portal J2EE Server.

A single PAR can be uploaded from the NetWeaver Developer Studio directly into the Portal via the PAR Export feature of the portal plug-in for Eclipse.

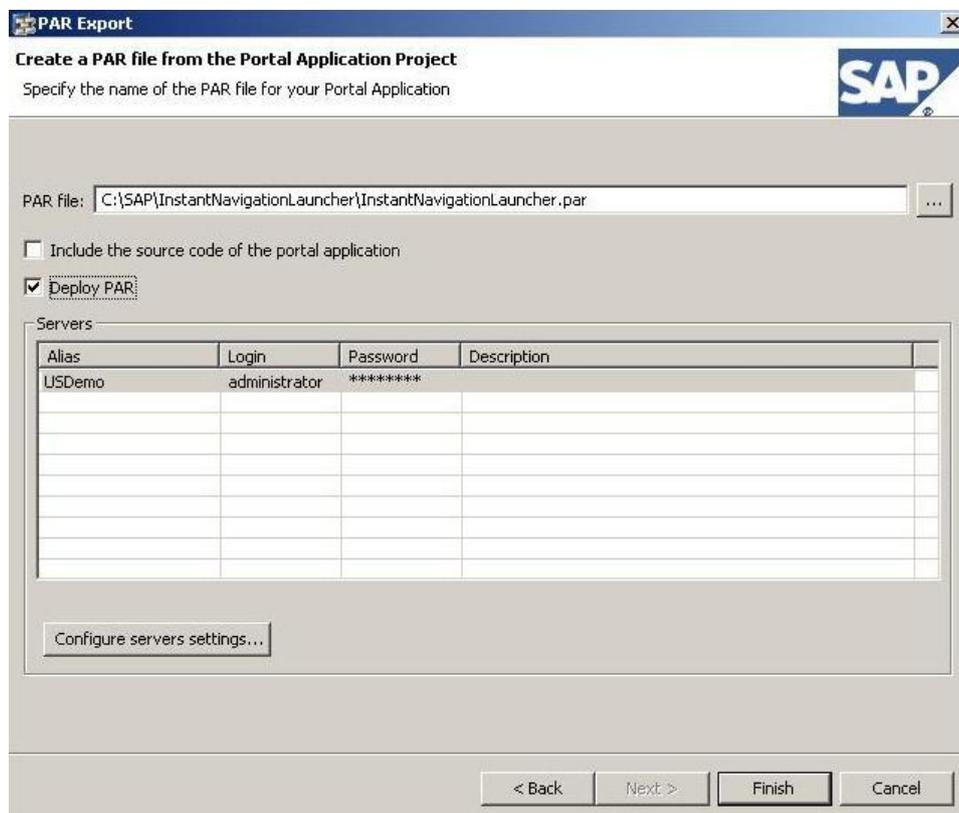
1. Choose File → Export → PAR File. Choose Next.



2. Select the project where the PAR file is located. In this case, `InstantNavigationLauncher`. Choose Next.



3. Enter data as required. If you want to deploy the PAR, choose Deploy PAR.

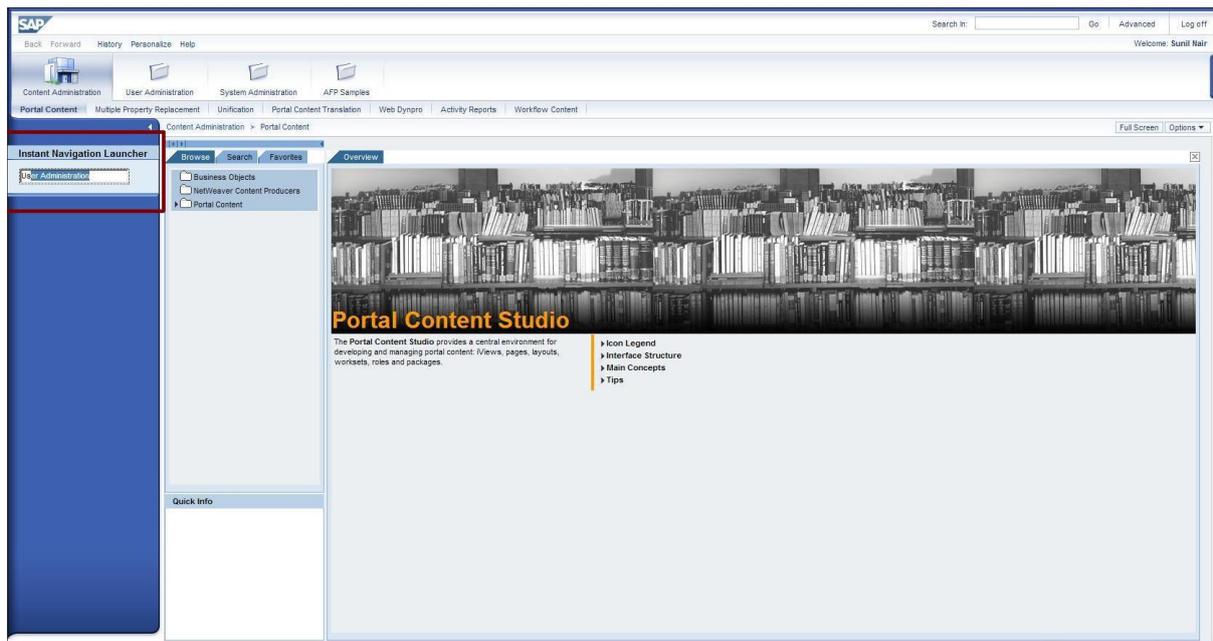


4. Choose Finish.

4.7 Integrate and test the application

We can now create an iView for the InstantNavigationLauncher Portal application and test the application.

1. Create an iView from the InstantNavigationLauncher Portal application.
 - In the Portal Content Catalog, open the Ajax Framework Page for editing.
 - Add the InstantNavigationLauncher iView to the page as a delta link.
 - Choose Page Layout and drag the InstantNavigationLauncher iView to the Left Navigation Panel Container.
2. Test your application.
 - You can see the iView in the left navigation panel.
 - Key in the initial characters of the navigation node where you want to navigate. For example, as shown in the image below, to navigate to User Administration, enter Us and the text box automatically fetches the navigation node for you and fills the text box.
 - Press the enter button and you are taken to the desired navigation node.



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