EP: Understanding & Troubleshooting OBN Scenarios

Applies to:

Netweaver Portal 6.x – release independent

Netweaver Portal 7.x – release independent

Summary

Object Based Navigation is often used by SAP in recent Business Packages like ERP's Plant Maintenance and SRM. But when errors occur in this component we are mislead into thinking it is an Application error instead of Portal's. What's a mystery to a few may be solved by a simple investigation. I will show you how to identify this kind of error and show you the actions to eliminate them.

Author: Ivan Mirisola

Company: SAP Consulting Brazil

Created on: 21 August 2010

Author Bio

Ivan has worked as a *Netweaver* Consultant and Developer since 2004 for SAP Brazil and is now acting as a *Netweaver* Technical Solution Architect. He majored on System Analysis and has extensive knowledge on Internet technologies. Specialized in *Netweaver* Portal and Java Custom Developments and now focused on SOA Methodology and *Netweaver* BPM implementation.

Table of Contents

Overview

Object Based Navigation (aka OBN) is vastly used in recent Business Packages like those for ERP's Plant Maintenance and SRM. OBN is by nature based in Java and is a powerful framework to develop integration between different components – being them Java/ABAP. Essentially OBN provides developers the means to navigate to a different iView residing in the Portal PCD without actually knowing where they are. Also with OBN it is also possible (and often used) to pass on data values between applications.

Details

What is OBN

Technically speaking OBN is a simple association of a Business Object Method with one or more iViews on the PCD. Let's take the Solution Manager Workcenter as an example:

Business Object	Operation (method)	Implementing iViews
SolManNavigation	Navigate	portal_content/ solman /roles /com.sap.solman.roles.workcenter /com.sap.solman.iview.ags_work_launcher
SolManNavigation	NavigateUrl	portal_content/ solman /roles /com.sap.solman.roles.workcenter /com.sap.solman.iview.dswp_url_launch

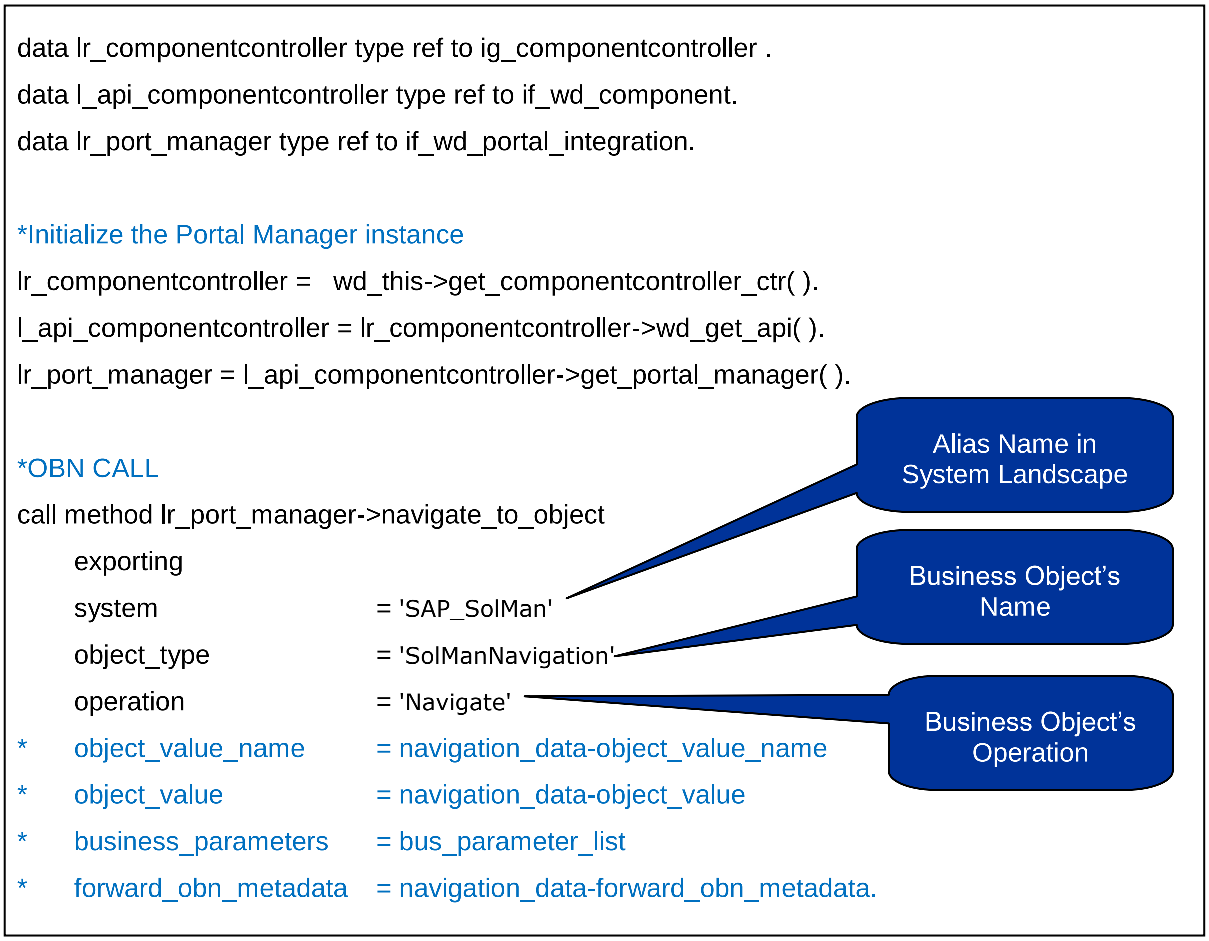
As you can see the table above means that there is an association between the operation "Navigate" with the iView that calls AGS_WORK_LAUNCHER (a Webdynpro ABAP application). The same happens for operation "NavigateUrl".

Now the developer has an abstraction layer between the source code and the PCD so the code remains clean and the Administrator has the means to easily change PCD objects locations.

If you've implemented this scenario before you may have noticed that every time you add an iView to the operation, not just the actual iView will be associated, but all instances of the same on every role. But why is this so. The reason behind this is that the portal runtime can determine the navigation based on the operation of the OBN, but it cannot deliver a navigation to the end-user for whom there is no associated role.

Developer’s point of view

The developer is able to call the BO’s methods in order to retrieve the end user’s navigation and perform the proper navigation between Portal Applications. Here is how the API works in ABAP:



```
data lr_componentcontroller type ref to ig_componentcontroller .
data l_api_componentcontroller type ref to if_wd_component.
data lr_port_manager type ref to if_wd_portal_integration.

*Initialize the Portal Manager instance
lr_componentcontroller =   wd_this->get_componentcontroller_ctr( ).
l_api_componentcontroller = lr_componentcontroller->wd_get_api( ).
lr_port_manager = l_api_componentcontroller->get_portal_manager( ).

*OBN CALL
call method lr_port_manager->navigate_to_object
      exporting
      system                  = 'SAP_SolMan'
      object_type             = 'SolManNavigation'
      operation               = 'Navigate'
*     object_value_name       = navigation_data-object_value_name
*     object_value            = navigation_data-object_value
*     business_parameters     = bus_parameter_list
*     forward_obn_metadata    = navigation_data-forward_obn_metadata.
```

The above code will create an EPCM JavaScript Code that can be raised on any event the developer has programmed (such as screen button’s click). There will be a call to “*EPCM.doObjBasedNavigate = function(systemAlias, businessObjName,objValue, operation)*” which is automatically generated by the framework.

Notice that there are several commented parameters that can be used by the developer to send information from the current screen to the Portal and subsequently to the next application to pick it up.
In Java the same can be applied with the following code:

```
WDPortalNavigation.navigateToObject(
                  "MyTargetSystem",
                  "MyTargetObject",
                  "AObjectValue",
                  "parameter1=value&parameter2=value2");
```

The Catch

Portal Administrators tend to train their users not mix up errors from applications with errors from Portal. But OBN errors are at first hard to be identified because it seems as though it is an application error where it is not.

A basic OBN scenario

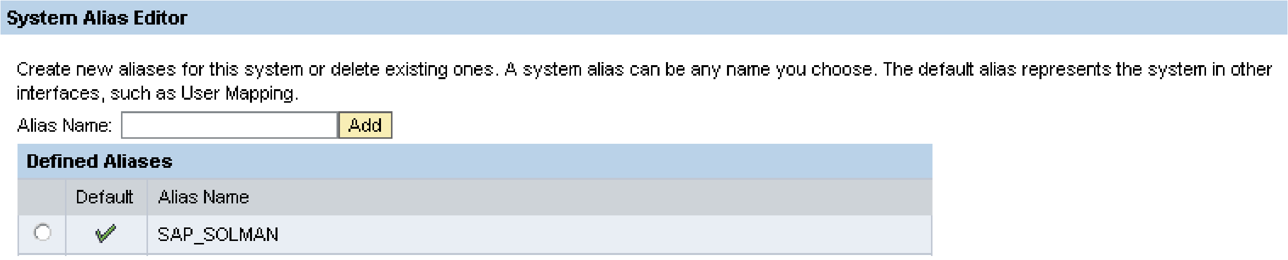
First you will learn how to create the necessary PCD objects for Solution Manager Workcenter and then it's related configuration for OBN just to depict what could go wrong.

I chose the Solution Manager Scenario because it is very simple to implement and simulate any errors.

The solution to this scenario is described on Note 1399645 - *SAP Solution Manager in SAP Enterprise Portal*

Solution Manager System

The first thing to do is to make sure that an ERP system exists in the Portal System Landscape Editor with an attached System Alias. It is not very common, but sometimes it is necessary to attach a predefined System Alias name because it is hardcoded in the application.



System Alias Editor

Create new aliases for this system or delete existing ones. A system alias can be any name you choose. The default alias represents the system in other interfaces, such as User Mapping.

Alias Name: Add

Defined Aliases

	Default	Alias Name
○	✓	SAP_SOLMAN

You must also supply information for the following items on the system:

- Connector
- ITS
- User Management
- Web AS

If you need assistance, follow the instructions on this weblog:

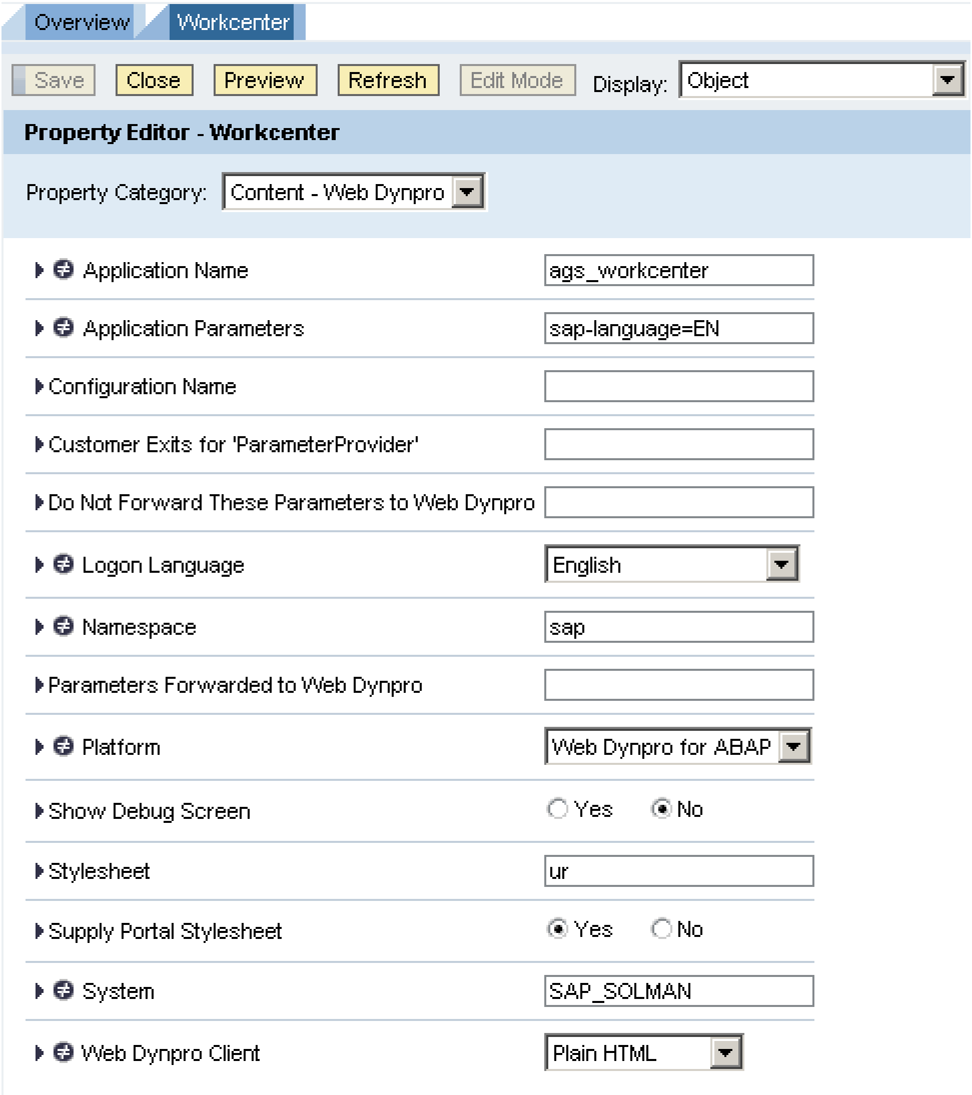
http://wiki.sdn.sap.com/wiki/display/EP/How+to+Create+System+Object+in+the+Portal+for+Connecting+to+SAP+backend+System

Make sure you have created a system with enabled SSO to avoid double authentication.

Solution Manager iViews

Workcenter

Create an iView of type Webdynpro for ABAP for the application “ags_workcenter” as follows:



Display: Object

Property Editor - Workcenter

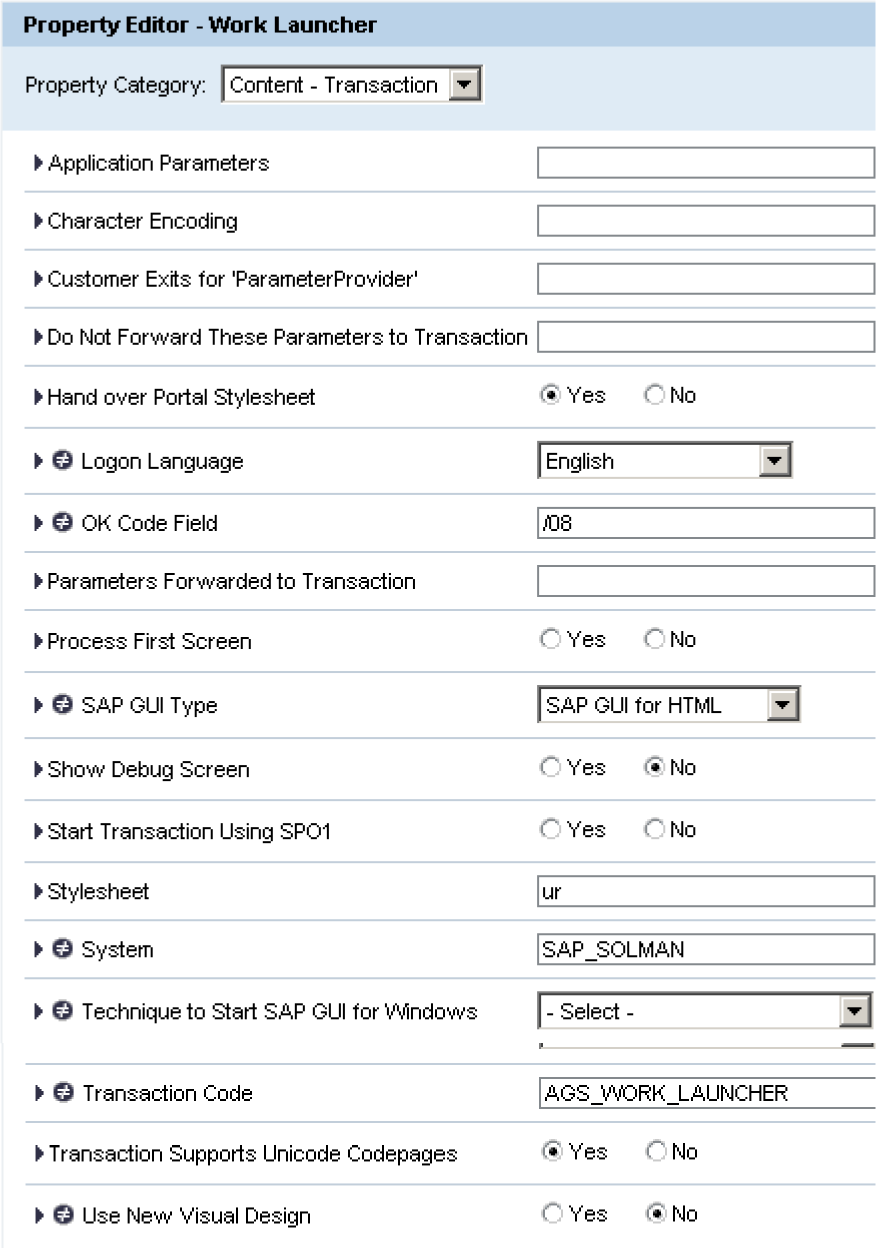
Property Category: Content - Web Dynpro

Application Name	ags_workcenter
Application Parameters	sap-language=EN
Configuration Name	
Customer Exits for 'ParameterProvider'	
Do Not Forward These Parameters to Web Dynpro	
Logon Language	English
Namespace	sap
Parameters Forwarded to Web Dynpro	
Platform	Web Dynpro for ABAP
Show Debug Screen	○ Yes ● No
Stylesheet	ur
Supply Portal Stylesheet	● Yes ○ No
System	SAP_SOLMAN
Web Dynpro Client	Plain HTML

Notice the use of the recently created System Alias Name. Here I used the language definitions for English, as we needed to enforce it. But you can also use AI parameters to define the user’s logon language such as `<Request.Language>`.

Work Launcher

Create an iView of type Transaction iView for the transaction code “AGS_WORK_LAUNCHER” as follows:

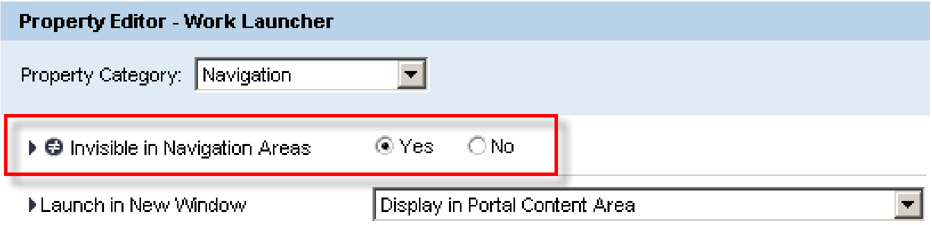


Property Editor - Work Launcher

Property Category: Content - Transaction

Application Parameters	
Character Encoding	
Customer Exits for 'ParameterProvider'	
Do Not Forward These Parameters to Transaction	
Hand over Portal Stylesheet	● Yes ○ No
Logon Language	English
OK Code Field	/08
Parameters Forwarded to Transaction	
Process First Screen	○ Yes ○ No
SAP GUI Type	SAP GUI for HTML
Show Debug Screen	○ Yes ● No
Start Transaction Using SPO1	○ Yes ○ No
Stylesheet	ur
System	SAP_SOLMAN
Technique to Start SAP GUI for Windows	- Select -
Transaction Code	AGS_WORK_LAUNCHER
Transaction Supports Unicode Codepages	● Yes ○ No
Use New Visual Design	○ Yes ● No

Usually the target iView for OBN has the property “Invisible in Navigation Areas” set to true, because we do not need to view the link for this iView during navigation. So go to the Navigation Category and set it to true as follows:



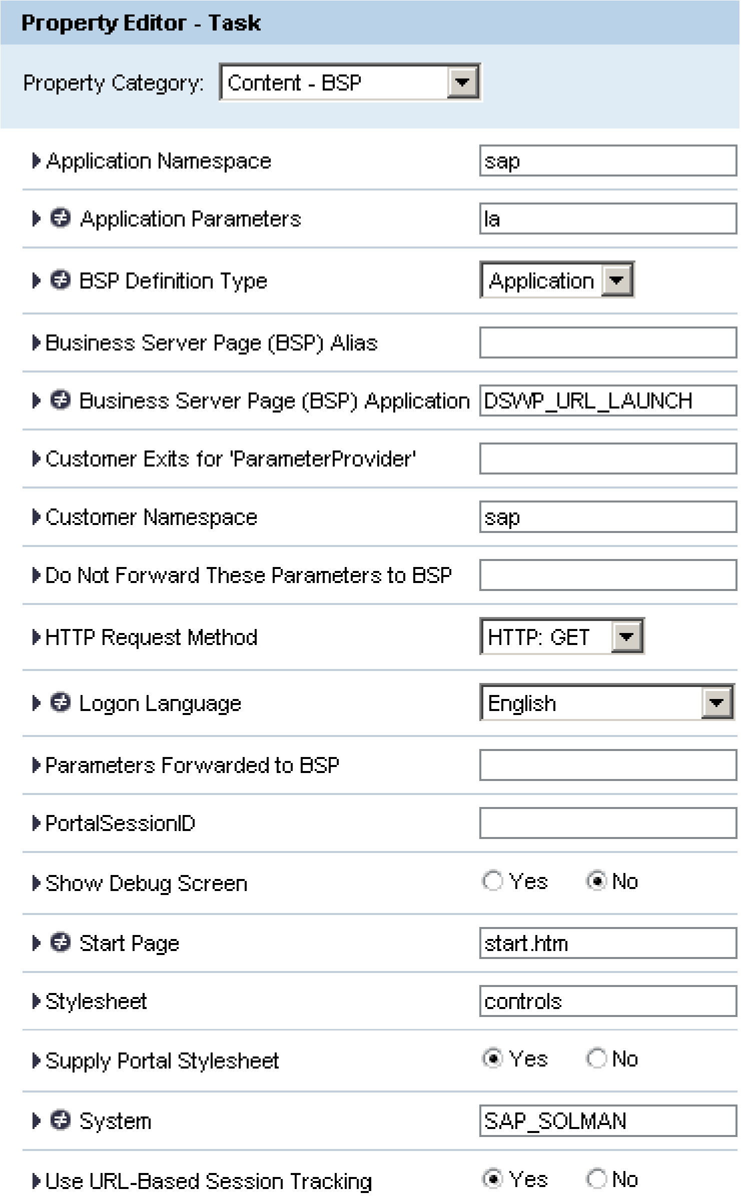
Property Editor - Work Launcher

Property Category: Navigation

Invisible in Navigation Areas	● Yes ○ No
Launch in New Window	Display in Portal Content Area

Task

Create an iView of type BSP iView for the application “DSWP_URL_LAUNCH” as follows:

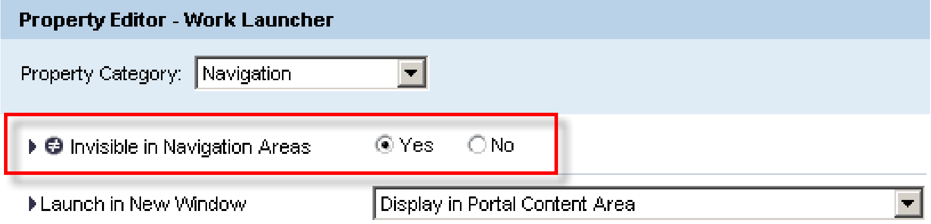


Property Editor - Task

Property Category: Content - BSP

Application Namespace	sap
Application Parameters	la
BSP Definition Type	Application
Business Server Page (BSP) Alias	
Business Server Page (BSP) Application	DSWP_URL_LAUNCH
Customer Exits for 'ParameterProvider'	
Customer Namespace	sap
Do Not Forward These Parameters to BSP	
HTTP Request Method	HTTP: GET
Logon Language	English
Parameters Forwarded to BSP	
PortalSessionID	
Show Debug Screen	○ Yes ● No
Start Page	start.htm
Stylesheet	controls
Supply Portal Stylesheet	● Yes ○ No
System	SAP_SOLMAN
Use URL-Based Session Tracking	● Yes ○ No

Usually the target iView for OBN has the property “Invisible in Navigation Areas” set to true, because we do not need to view the link for this iView during navigation. So go to the Navigation Category and set it to true as follows:



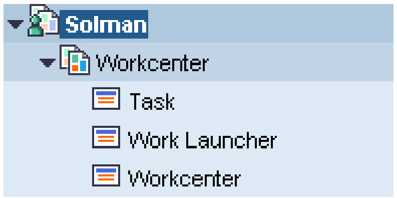
Property Editor - Work Launcher

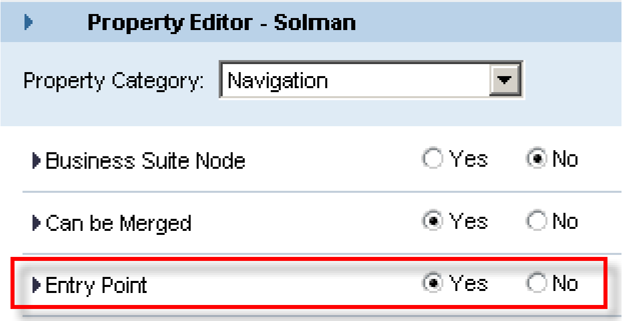
Property Category: Navigation

Invisible in Navigation Areas	● Yes ○ No
Launch in New Window	Display in Portal Content Area

Workcenter Role

Add the recently created iViews to a role of your choice. Make sure that the role you have chosen has an entry point set such as the following:





Property Editor - Solman

Property Category: Navigation

Business Suite Node ○ Yes ● No

Can be Merged ● Yes ○ No

Entry Point ● Yes ○ No

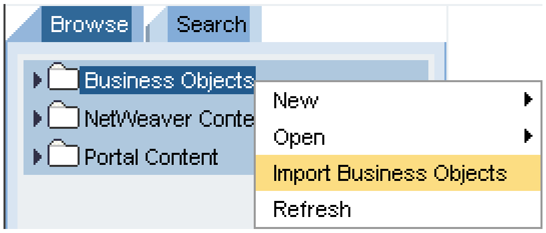
OBN Configuration

Now that we have the role created, we need to create the Business Object defined in the ABAP code of the Workcenter Application in the Portal Content Directory.

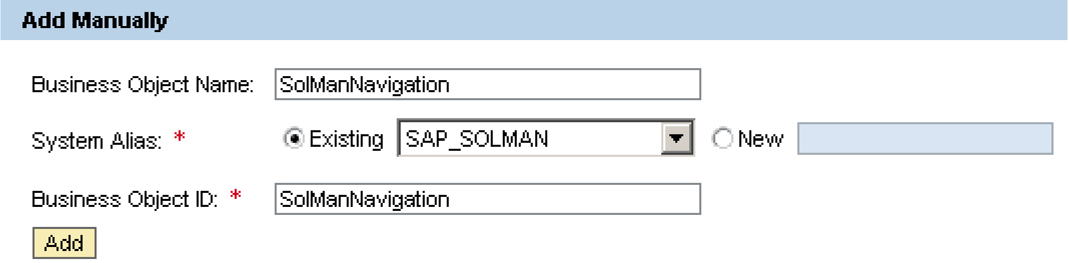
Business Object

The BO for Workcenter was not defined in Solution Manager, so we need to manually create it.

The creation screen is available at the Import BO Funtion of the PCD, so select it to open the BO Importer as follows:



Now at the bottom of the screen you will be able to create the BO required by Workcenter - named “SolManNavigation” – by clicking at the ADD button:



Add Manually

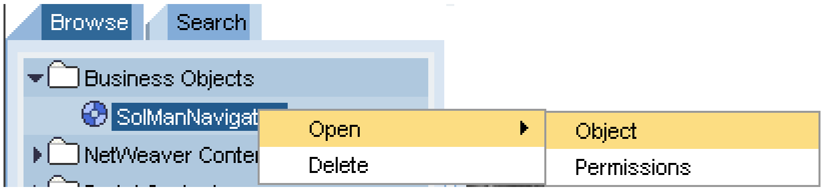
Business Object Name: SolManNavigation

System Alias: * ● Existing SAP_SOLMAN ○ New

Business Object ID: * SolManNavigation

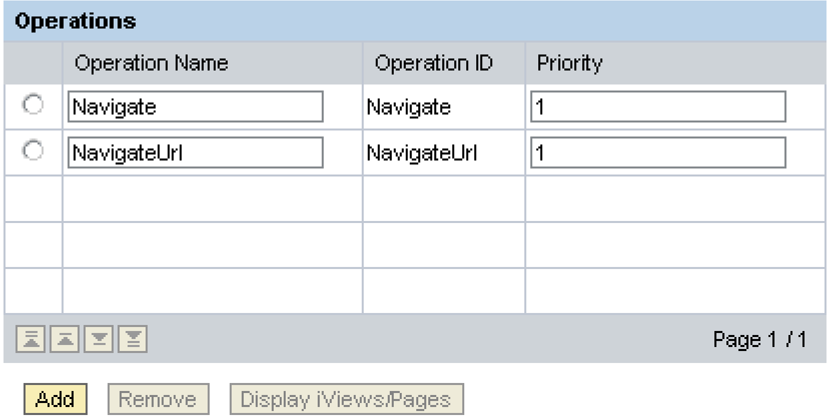
Add

Now the object is available at the PCD Folder Named “Business Objects” and is ready to be edited. Select the function to open the Object as follows:



BO Operations

Now add the following operations (which again are expected by the BO in ABAP code for navigation) by simply writing its name of the operation in field “Operation Name” and specify the priority as “1”. To add another operation, simply click on the add button below.

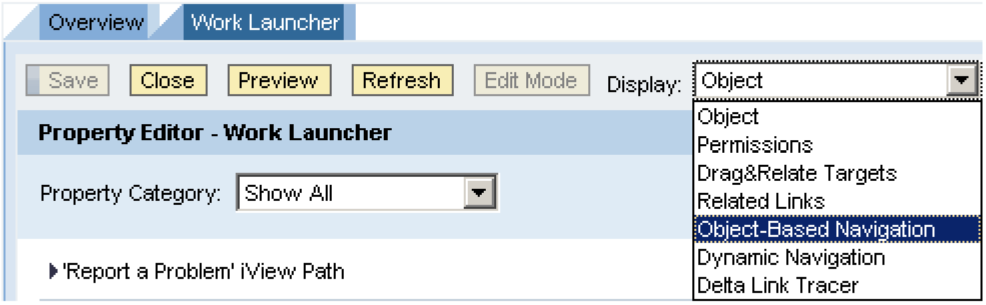


Operations

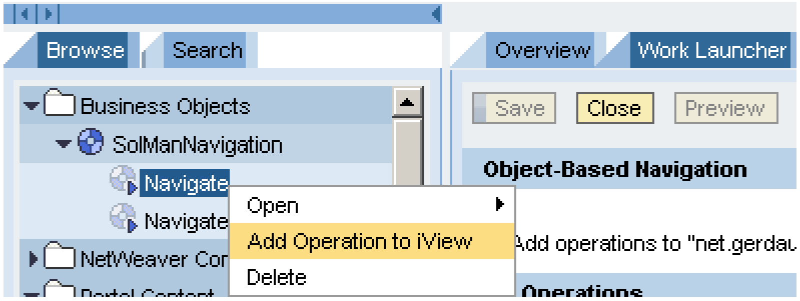
	Operation Name	Operation ID	Priority
○	Navigate	Navigate	1
○	NavigateUrl	NavigateUrl	1

Adding an operation to an iView

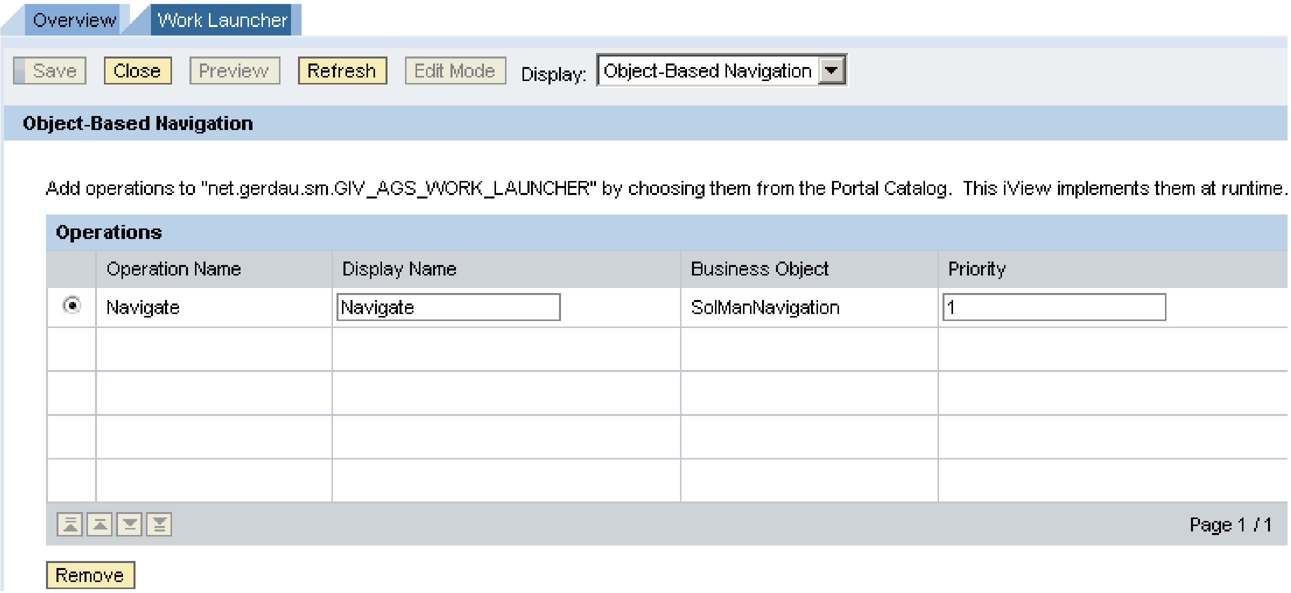
Open the editor for “Work Launcher” iView and select the display mode for “Object Based Navigation”:



Now select the operation available at the Business Object SolManNavigation and the operation “Navigate” to the “Work Launcher” iView:



The operation should be displayed as follows:



Display: Object-Based Navigation

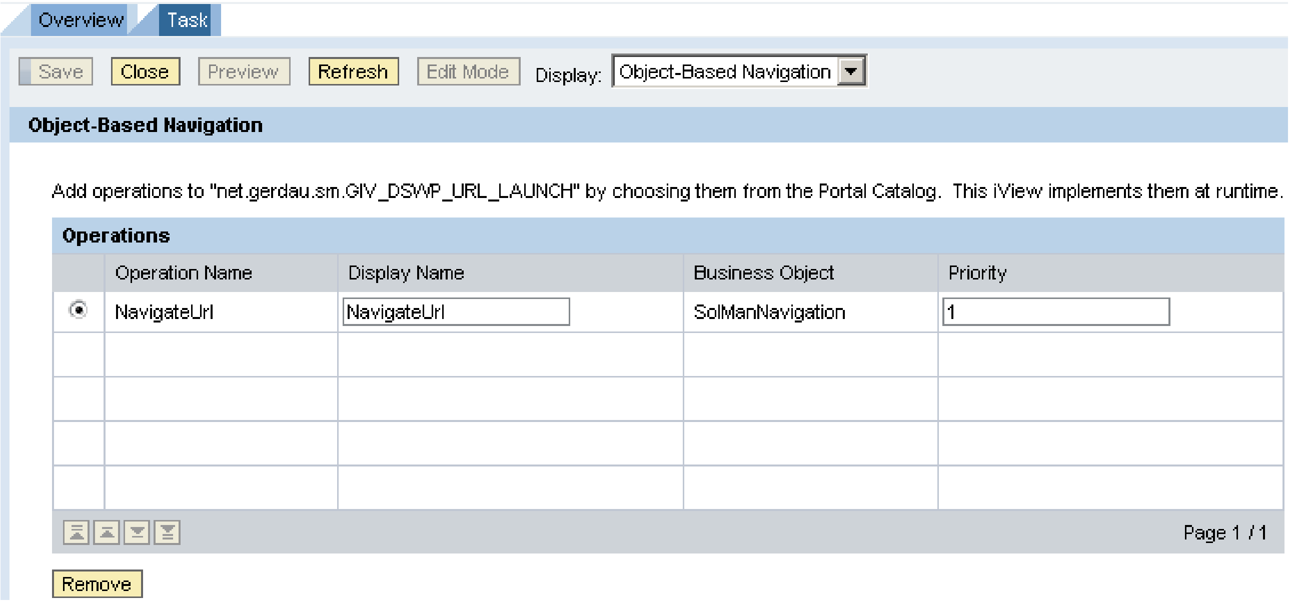
Object-Based Navigation

Add operations to "net.gerdau.sm.GIV_AGS_WORK_LAUNCHER" by choosing them from the Portal Catalog. This iView implements them at runtime.

Operations

	Operation Name	Display Name	Business Object	Priority
●	Navigate	Navigate	SolManNavigation	1

Repeate the same procedure for iView “Task” and operation “NavigateUrl”. The result should be something similar to this:



Display: Object-Based Navigation

Object-Based Navigation

Add operations to "net.gerdau.sm.GIV_DSWP_URL_LAUNCH" by choosing them from the Portal Catalog. This iView implements them at runtime.

Operations

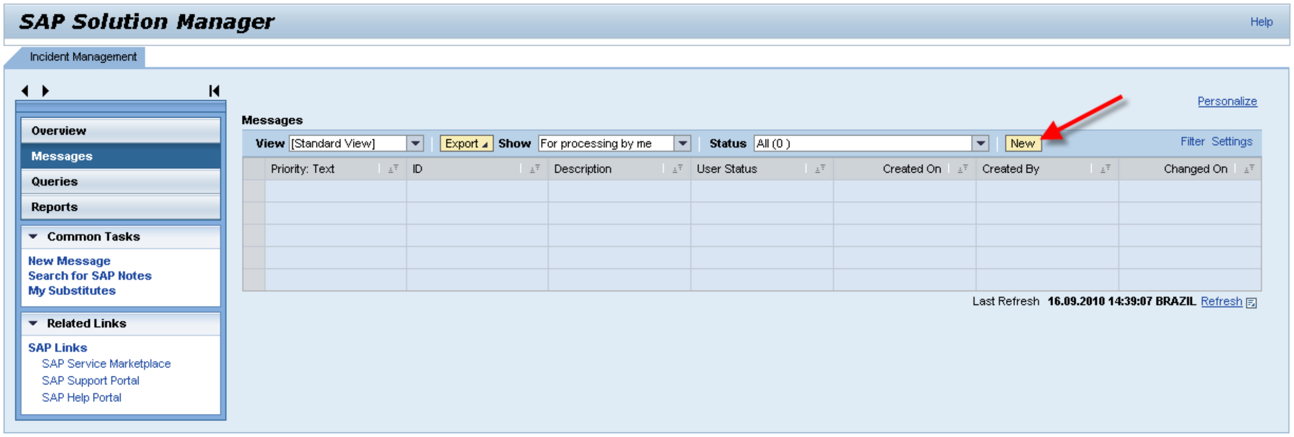
	Operation Name	Display Name	Business Object	Priority
●	NavigateUrl	NavigateUrl	SolManNavigation	1

OBN in action

When click on a particular function of an application that uses OBN, a new iView is called and parameters are passed to it by this application.

For instance, let’s take a look at Workcenter. When you chick on the “New” button from “Messages” context, then a new window appears with the BSP application for Task.

Workcenter → Message → New

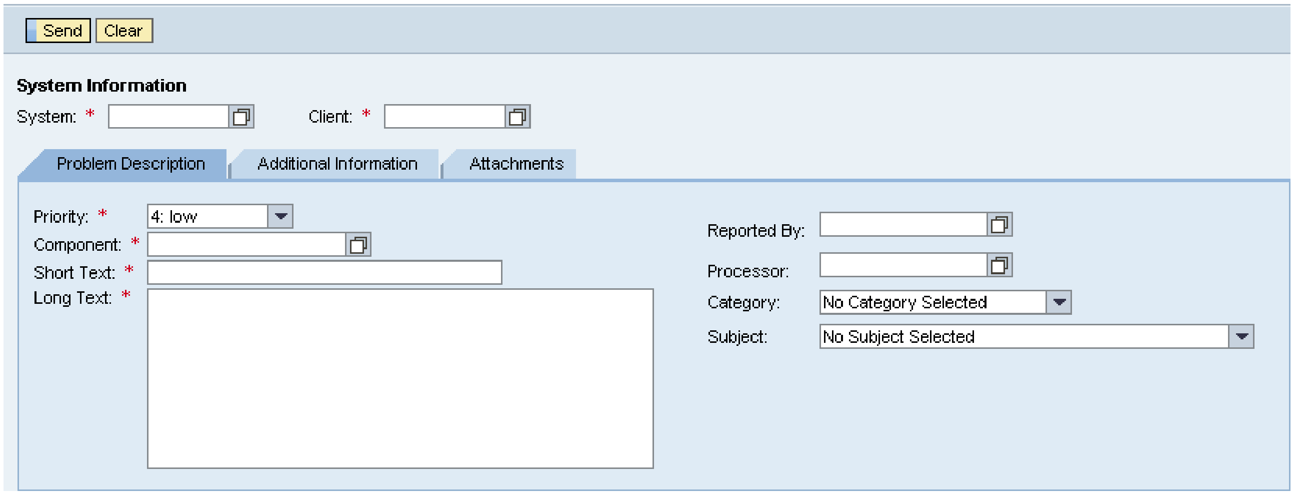


View [Standard View] Show For processing by me Status All (0)

Priority: Text	ID	Description	User Status	Created On	Created By	Changed On

Last Refresh 16.09.2010 14:39:07 BRAZIL

Task iView



System: *

Client: *

Priority: * 4: low

Component: *

Short Text: *

Long Text: *

Reported By:

Processor:

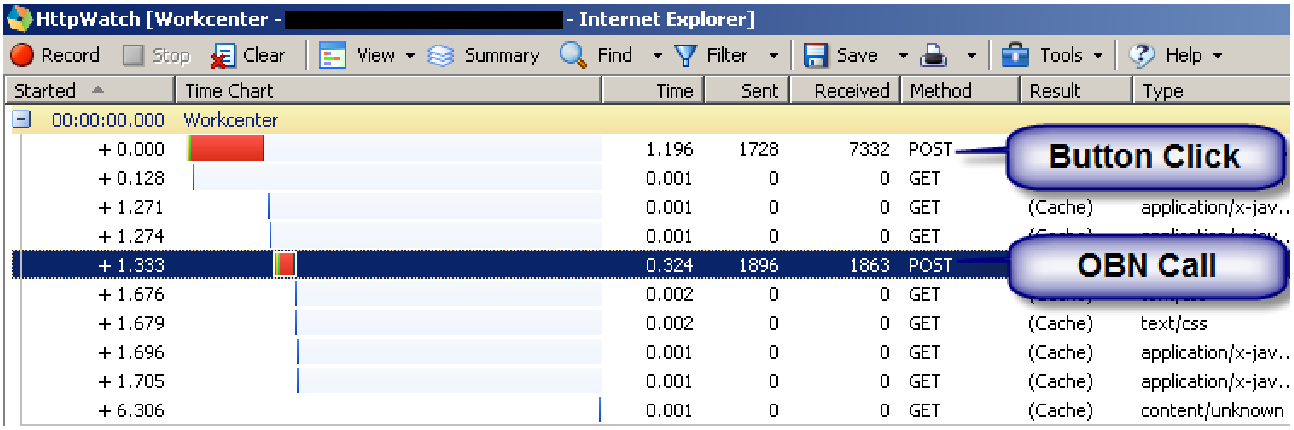
Category: No Category Selected

Subject: No Subject Selected

OBN Error Analysis

Tracing the navigation

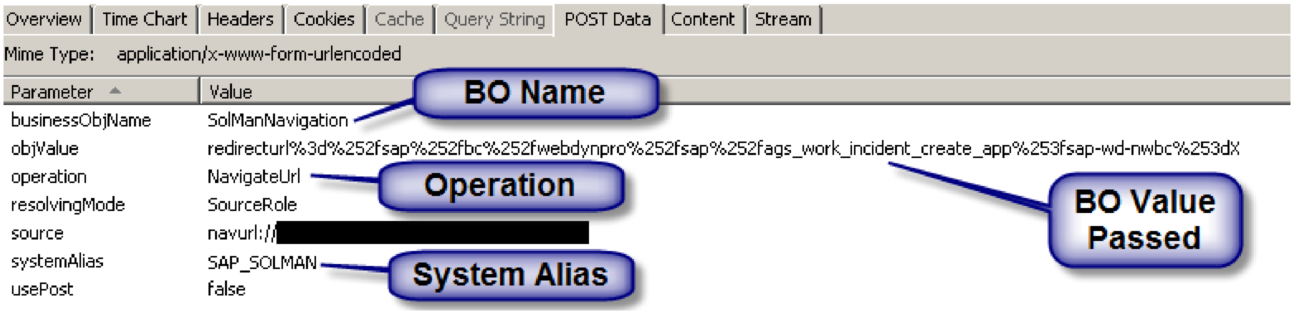
Using an HTTP Tracing Tool, we are able to “see” what happens under the hood:



HttpWatch [Workcenter - [redacted] - Internet Explorer]

Started	Time Chart	Time	Sent	Received	Method	Result	Type
00:00:00.000	Workcenter						
+ 0.000		1.196	1728	7332	POST		
+ 0.128		0.001	0	0	GET		
+ 1.271		0.001	0	0	GET	(Cache)	application/x-jav..
+ 1.274		0.001	0	0	GET		
+ 1.333		0.324	1896	1863	POST		
+ 1.676		0.002	0	0	GET		
+ 1.679		0.002	0	0	GET	(Cache)	text/css
+ 1.696		0.001	0	0	GET	(Cache)	application/x-jav..
+ 1.705		0.001	0	0	GET	(Cache)	application/x-jav..
+ 6.306		0.001	0	0	GET	(Cache)	content/unknown

Looking at the second POST HTTP call you are able to see that data is sent to “http://<host>:5<sys_number>00/irj/servlet/prt/portal/prtroot/com.sap.portal.navigation.objbased.ObjBasedNavigation”. This application is a standard application that parses the posted data, retrieves the Business Object’s operation being called and determinates which iView should be called.



Mime Type: application/x-www-form-urlencoded

Parameter	Value
businessObjName	SolManNavigation
objValue	redirecturl%3d%252fsap%252fbc%252fwebdynpro%252fsap%252fags_work_incident_create_app%253fsap-wd-nwbc%253dX
operation	NavigateUrl
resolvingMode	SourceRole
source	navurl://[redacted]
systemAlias	SAP_SOLMAN
usePost	false

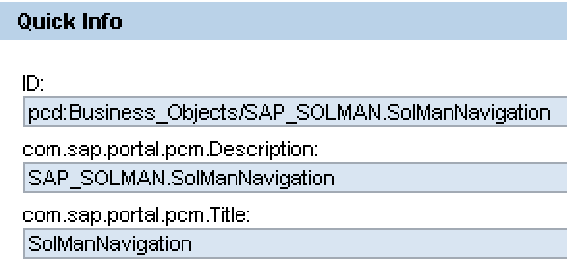
After that, the “ObjBasedNavigation” application redirects that flow to the correct iView based on operation specified by “Operation” and passes on the contents of variable “objValue”.

OBN calls the “Task” iView which will perform an action based on the value received by “objValue”. In this case the iView performs a redirect to an ABAP application called “AGS_WORK_INCIDENT_CREATE_APP”. But it is up to the iView what action will be taken based on the value and operation received.

OBN Investigation

With this type of information it is easy to trace back what is being called in the Portal.

Open the Business Object folder and find the BO with name “SolManNavigation. Notice that the HTTP trace will display the technical name of the Object whilst the PCD will display the “Friendly Name”. Find the technical name of BO from the “Quick Info” pane if necessary:

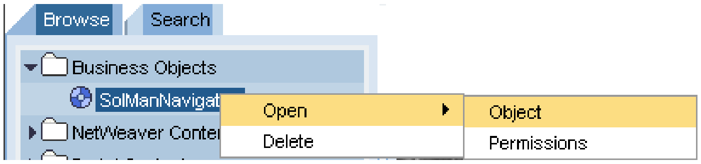


Quick Info

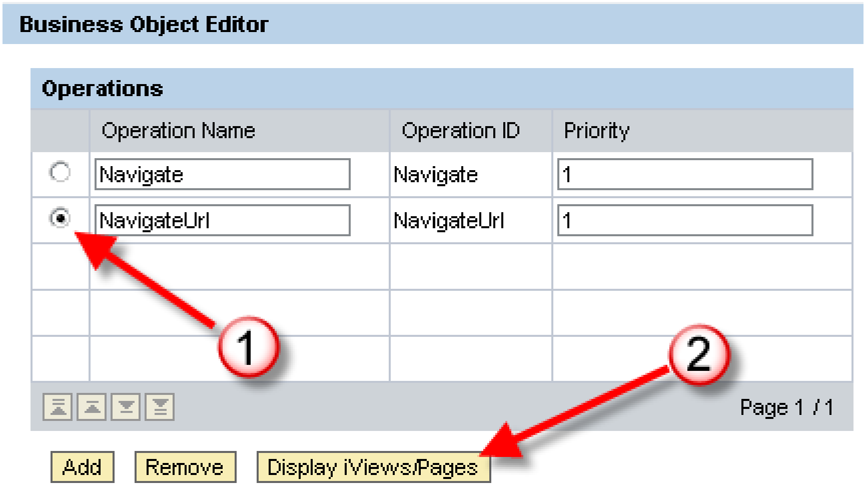
ID:
pcd:Business_Objects/SAP_SOLMAN.SolManNavigation

com.sap.portal.pcm.Description:
SAP_SOLMAN.SolManNavigation

com.sap.portal.pcm.Title:
SolManNavigation



Search for operation "NavigateUrl".



Business Object Editor

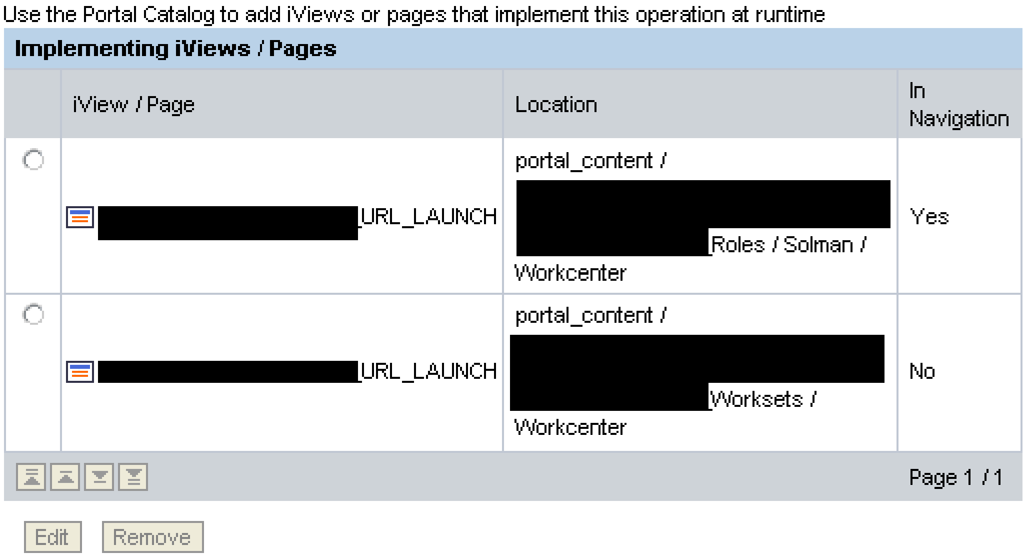
Operations

	Operation Name	Operation ID	Priority
○	Navigate	Navigate	1
●	NavigateUrl	NavigateUrl	1

Page 1 / 1

A list of iViews that implement the selected operation will be displayed.

The Column "In Navigation" indicates an iView that is to be found inside a role (YES) or to be found on the PCD (No).



Use the Portal Catalog to add iViews or pages that implement this operation at runtime

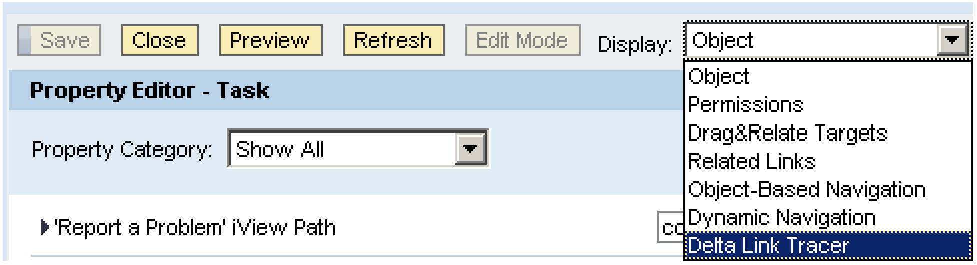
Implementing iViews / Pages

	iView / Page	Location	In Navigation
○	[redacted] URL_LAUNCH	portal_content / [redacted] Roles / Solman / Workcenter	Yes
○	[redacted] URL_LAUNCH	portal_content / [redacted] Worksets / Workcenter	No

Page 1 / 1

Select the iView that has "In Navigation" set to 'YES' and click on the Edit Button.

On the editor, enter the "Delta Link Tracer"



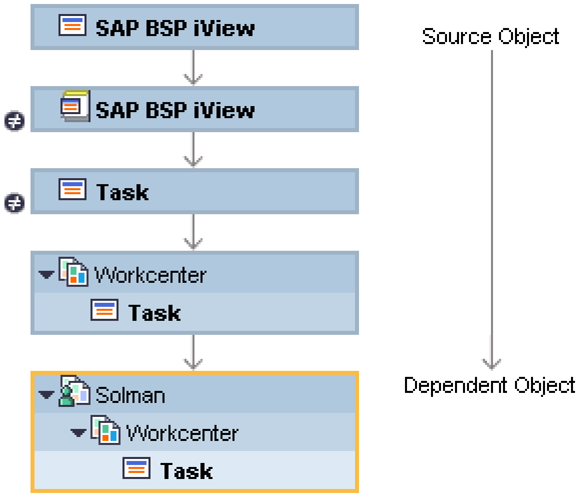
Display: Object

Property Editor - Task

Property Category: Show All

▸ 'Report a Problem' iView Path

This shows you which role the end-user should have so OBN would work correctly.



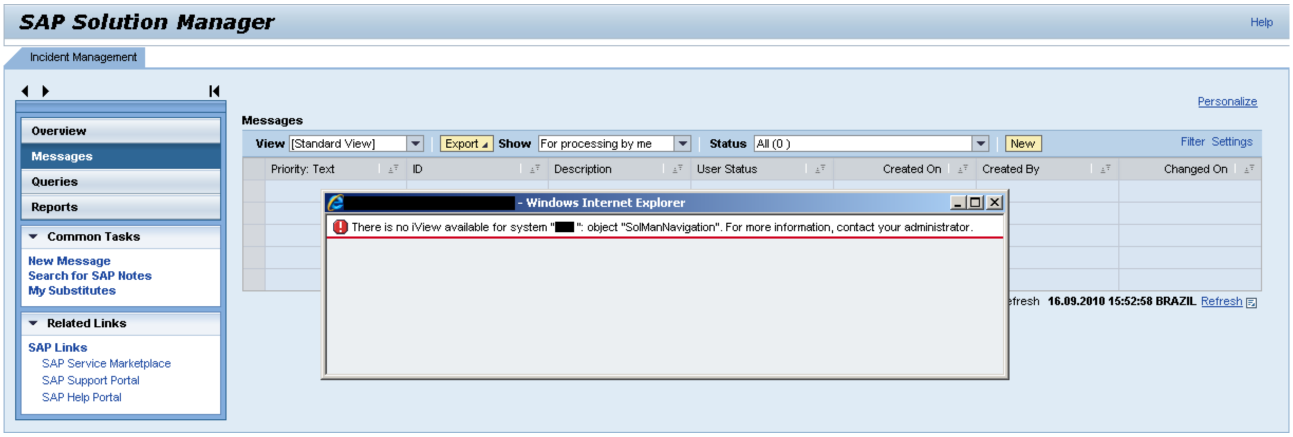
Simulating an OBN error

Let's take a business package like the one used on SRM. The are several roles in this BP and most of them use a "Service Role" that groups together all the PCD objects that are involved with OBN.

The "Service Role" never has an entry point or iViews that can be shown in the navigation areas. So the Portal Administrator is not instantly aware that there is a missing role for the end-user.

If the end user doesn't have the "Service Role" than Portal Navigation will fail.

Let's simulate this error by removing the Task iView from the Role "SolMan" and click again on the "New" Button:



View [Standard View] Show For processing by me Status All (0)

[redacted] - Windows Internet Explorer

There is no iView available for system "[redacted]": object "SolManNavigation". For more information, contact your administrator.

efresh 16.09.2010 15:52:58 BRAZIL

The error message indicates the object for which an iView was not found. It is usually useless information, unless you have performed a trace analysis and found which operation has triggered this error.

OBN Troubleshooting Guidance

To correct this type of issue, you may follow the procedure below:

1. Perform a trace analysis and find out which BO, System and Operation was called.
2. Based on the above information, check which role has the iView that implements this operation
3. Check if the end-user is assigned to this role
4. Check if the end-user has read permissions for the role and it's dependant objects on the PCD – including the BO itself.

Related Content

Java OBN API

Navigating with OBN in Java

Webdyrnpro OBN API

Webdynpro ABAP OBN API

Webdynpro Code Wizard

Note 1399645 - SAP Solution Manager in SAP Enterprise Portal

Wiki on How to Create System Object in the Portal for Connecting to SAP backend System

Copyright

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.