

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 misled 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

Applies to:	1
Summary.....	1
Author Bio	1
Table of Contents	2
Overview	3
Details	3
What is OBN	3
Developer's point of view	4
The Catch.....	5
A basic OBN scenario.....	5
Solution Manager System	5
Solution Manager iViews.....	6
Workcenter.....	6
Work Launcher.....	7
Task	8
Workcenter Role	9
OBN Configuration	9
Business Object.....	9
BO Operations	10
Adding an operation to an iView	10
OBN in action.....	12
OBN Error Analysis.....	13
Tracing the navigation.....	13
OBN Investigation	13
Simulating an OBN error	15
OBN Troubleshooting Guidance.....	15
Related Content.....	16
Copyright.....	17

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 diagram consists of three blue callout boxes on the right side of the code block. The top callout box, labeled 'Alias Name in System Landscape', has a line pointing to the 'system' parameter in the 'exporting' section of the 'navigate_to_object' call. The middle callout box, labeled 'Business Object's Name', has a line pointing to the 'object_type' parameter. The bottom callout box, labeled 'Business Object's Operation', has a line pointing to the 'operation' parameter.

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:

Defined Aliases

Default	Alias Name
<input type="radio"/>	
<input checked="" type="checkbox"/>	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:

The screenshot displays the 'Property Editor - Workcenter' in SAP Solution Manager. The 'Property Category' is set to 'Content - Web Dynpro'. The configuration fields are as follows:

Property Name	Value
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	<input type="radio"/> Yes <input checked="" type="radio"/> No
Stylesheet	ur
Supply Portal Stylesheet	<input checked="" type="radio"/> Yes <input type="radio"/> 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	<input type="text"/>
▶ Character Encoding	<input type="text"/>
▶ Customer Exits for 'ParameterProvider'	<input type="text"/>
▶ Do Not Forward These Parameters to Transaction	<input type="text"/>
▶ Hand over Portal Stylesheet	<input checked="" type="radio"/> Yes <input type="radio"/> No
▶ Logon Language	English
▶ OK Code Field	/08
▶ Parameters Forwarded to Transaction	<input type="text"/>
▶ Process First Screen	<input type="radio"/> Yes <input type="radio"/> No
▶ SAP GUI Type	SAP GUI for HTML
▶ Show Debug Screen	<input type="radio"/> Yes <input checked="" type="radio"/> No
▶ Start Transaction Using SPO1	<input type="radio"/> Yes <input type="radio"/> No
▶ Stylesheet	ur
▶ System	SAP_SOLMAN
▶ Technique to Start SAP GUI for Windows	- Select -
▶ Transaction Code	AGS_WORK_LAUNCHER
▶ Transaction Supports Unicode Codepages	<input checked="" type="radio"/> Yes <input type="radio"/> No
▶ Use New Visual Design	<input type="radio"/> Yes <input checked="" type="radio"/> 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	<input checked="" type="radio"/> Yes <input type="radio"/> 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:

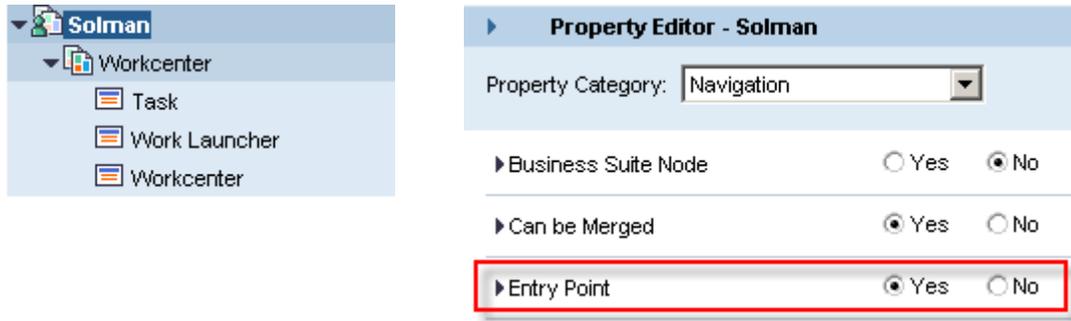
Property Editor - Task	
Property Category:	Content - BSP
▶ Application Namespace	sap
▶ Application Parameters	la
▶ BSP Definition Type	Application
▶ Business Server Page (BSP) Alias	
▶ Business Server Page (BSP) Application	DSWP_URL_LAUNCH
▶ Customer Exits for 'ParameterProvider'	
▶ Customer Namespace	sap
▶ Do Not Forward These Parameters to BSP	
▶ HTTP Request Method	HTTP: GET
▶ Logon Language	English
▶ Parameters Forwarded to BSP	
▶ PortalSessionID	
▶ Show Debug Screen	<input type="radio"/> Yes <input checked="" type="radio"/> No
▶ Start Page	start.htm
▶ Stylesheet	controls
▶ Supply Portal Stylesheet	<input checked="" type="radio"/> Yes <input type="radio"/> No
▶ System	SAP_SOLMAN
▶ Use URL-Based Session Tracking	<input checked="" type="radio"/> Yes <input type="radio"/> 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	<input checked="" type="radio"/> Yes <input type="radio"/> 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:



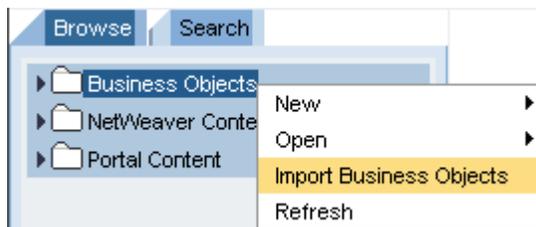
OBN Configuration

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.

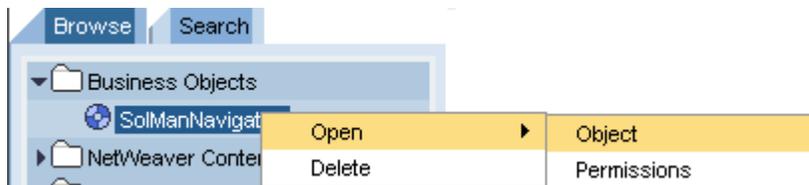
The creation screen is available at the Import BO Function of the PCD, so select it to open the BO Importer as follows:



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:

The image shows the 'Add Manually' dialog box for creating a Business Object. The 'Business Object Name' field contains 'SolManNavigation'. The 'System Alias' is set to 'Existing' with a dropdown menu showing 'SAP_SOLMAN'. The 'Business Object ID' field also contains 'SolManNavigation'. An 'Add' button is visible at the bottom left.

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
<input type="radio"/>	Navigate	Navigate	1
<input type="radio"/>	NavigateUrl	NavigateUrl	1

Page 1 / 1

Adding an operation to an iView

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

Overview Work Launcher

Save Close Preview Refresh Edit Mode Display: Object

Property Editor - Work Launcher

Property Category: Show All

► 'Report a Problem' iView Path

- Object
- Permissions
- Drag&Relate Targets
- Related Links
- Object-Based Navigation**
- Dynamic Navigation
- Delta Link Tracer

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

Browse Search Overview Work Launcher

Save Close Preview

Object-Based Navigation

Business Objects

- SolManNavigation
 - Navigate
 - Navigate
- NetWeaver Cor
- Portal Content

Open

Add Operation to iView

Delete

Add operations to "net.gerdau"

Operations

The operation should be displayed as follows:

Overview **Work Launcher**

Save Close Preview Refresh Edit Mode 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
<input checked="" type="radio"/>	Navigate	Navigate	SolManNavigation	1
<input type="radio"/>				
<input type="radio"/>				
<input type="radio"/>				

Page 1 / 1

Remove

Repeat the same procedure for iView "Task" and operation "NavigateUrl". The result should be something similar to this:

Overview **Task**

Save Close Preview Refresh Edit Mode 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
<input checked="" type="radio"/>	NavigateUrl	NavigateUrl	SolManNavigation	1
<input type="radio"/>				
<input type="radio"/>				
<input type="radio"/>				

Page 1 / 1

Remove

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 click on the "New" button from "Messages" context, then a new window appears with the BSP application for Task.

Workcenter → Message → New

The screenshot shows the SAP Solution Manager interface for Incident Management. The main area displays a table of Messages with columns for Priority, Text, ID, Description, User Status, Created On, Created By, and Changed On. The table is currently empty. A red arrow points to the 'New' button located in the top right corner of the Messages table area. The interface includes a left sidebar with navigation options like Overview, Messages, Queries, Reports, and Common Tasks. The top header shows 'SAP Solution Manager' and 'Incident Management'.

Task iView

The screenshot shows the Task iView form in SAP Solution Manager. The form is divided into several sections:

- System Information:** Fields for System and Client.
- Problem Description:** Fields for Priority (set to 4: low), Component, Short Text, and Long Text.
- Additional Information:** Fields for Reported By, Processor, Category (set to No Category Selected), and Subject (set to No Subject Selected).
- Attachments:** A section for adding attachments.

 The form includes 'Send' and 'Clear' buttons at the top left. The interface is clean and professional, typical of SAP software.

OBN Error Analysis

Tracing the navigation

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

Started	Time Chart	Time	Sent	Received	Method	Result	Type
00:00:00.000	Workcenter						
+ 0.000		1.196	1728	7332	POST		Button Click
+ 0.128		0.001	0	0	GET	(Cache)	application/x-jav..
+ 1.271		0.001	0	0	GET	(Cache)	application/x-jav..
+ 1.274		0.001	0	0	GET	(Cache)	application/x-jav..
+ 1.333		0.324	1896	1863	POST		OBN Call
+ 1.676		0.002	0	0	GET	(Cache)	text/css
+ 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 determines which iView should be called.

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://
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

Folder	Item	Action
Business Objects	SolManNavigation	Open
Business Objects	SolManNavigation	Delete
NetWeaver Center	Object	Permissions

Search for operation “NavigateUrl”.

Business Object Editor

Operations

	Operation Name	Operation ID	Priority
<input type="radio"/>	Navigate	Navigate	1
<input checked="" type="radio"/>	NavigateUrl	NavigateUrl	1

Page 1 / 1

Add Remove Display iViews/Pages

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
<input type="radio"/>	[Redacted] URL_LAUNCH	portal_content / [Redacted] Roles / Solman / Workcenter	Yes
<input type="radio"/>	[Redacted] URL_LAUNCH	portal_content / [Redacted] Worksets / Workcenter	No

Page 1 / 1

Edit Remove

Select the iView that has “In Navigation” set to ‘YES’ and click on the Edit Button.

On the editor, enter the “Delta Link Tracer”

Save Close Preview Refresh Edit Mode Display: Object

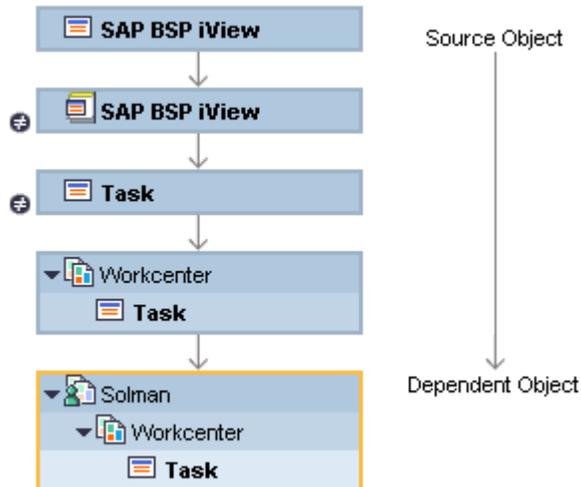
Property Editor - Task

Property Category: Show All

► 'Report a Problem' iView Path

- Object
- Permissions
- Drag&Relate Targets
- Related Links
- Object-Based Navigation
- Dynamic Navigation
- Delta Link Tracer

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



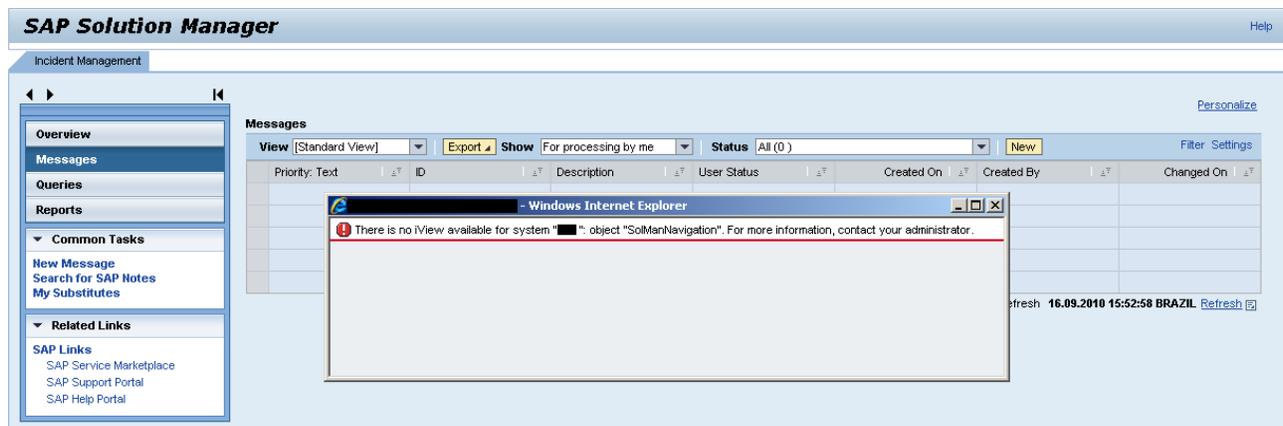
Simulating an OBN error

Let's take a business package like the one used on SRM. There 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" then 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:



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 its dependant objects on the PCD – including the BO itself.

Related Content

[Java OBN API](#)

[Navigating with OBN in Java](#)

[Webdynpro 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

© 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, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

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.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, 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 other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

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.