

Integrating Web Dynpro and SAP NetWeaver Portal

Part III: Using Object-Based Navigation within a Web Dynpro Application

Applies to:

Integration of Web Dynpro for Java applications into SAP NetWeaver Portal for the SAP NetWeaver 04s SP8 release.

Summary

This article describes the usage of the capabilities of Object-Based Navigation (OBN) within a Web Dynpro application running inside SAP NetWeaver Portal.

Author(s): Jochen Guertler

Company: SAP AG

Created on: 26 June 2006

Author Bio



Jochen Guertler works as a development architect within the Web Dynpro for Java team. His main responsibilities are the integration of Web Dynpro for Java with other components of SAP NetWeaver, especially the integration with the SAP NetWeaver Portal.

Jochen is co-author of the book "Maximizing Web Dynpro for Java" from SAP Press.

Table of Contents

Introduction	2
The WDPortalNavigation Service	3
Trigger an Object-based Navigation	3
Use the IUserObjectBasedNavigation Service	4
Defining a the Needed Reference	4
What is a Valid Operation?	5
Check for a Valid Default Operation	5
Check for a Valid Operation	5
Get the List of Valid Operations	6
Use a Web Dynpro iView as a Target	8
The Web Dynpro OBNTester	8
Further Information	9
Copyright	11

Introduction

Using SAP NetWeaver Portal you can structure all your content in a role-based manner. You can use the *Top Level Navigation (TLN)* or the *Detailed Navigation (DTN)* to browse the user-specific navigation structure. Using the portal navigation capabilities, every application running as portal content (i.e., every page or iView) can programmatically trigger navigation steps to other iViews or pages.

For some scenarios it is sufficient to navigate to a concrete iView or page using relative or absolute page navigation (as described in the last part of this series), but sometimes you need much more flexibility to define the needed navigation target.

Therefore there is *object-based navigation*, which allows defining navigation steps on a higher semantic level. Instead of defining a concrete target URL you define an *operation* of a *business object*. For example, you define that you want to trigger the *Display* operation of a *Customer* business object. The iView or page that is used as an implementation of the operation is configured within the SAP NetWeaver Portal and could be defined role- or even user-specific.

It is out of scope of this document to describe in detail how to configure the entire object-based navigation scenario. Look for specific documentation at help.sap.com or sdn.sap.com.

From the Web Dynpro perspective, the integration of object-based navigation is very similar as the usage of the absolute or relative portal navigation. Normally you have to add some Java Script coding to your iView, which triggers the object-based navigation. But as this is not possible for a Web Dynpro application (since Web Dynpro applications are implemented in a client–abstract manner), the Web Dynpro runtime offers a specific `WDPortalNavigation` service to define the required parameters.

The following document describes the triggering of object-based navigation within a Web Dynpro application.

The WDPortalNavigation Service

To access any page or object-based navigation functionality within a Web Dynpro application you have to use the `WDPortalNavigation` service. This service provides access to all needed parameters and functionalities. You do not have to define any specific reference to this service to access it, because it is a generic part of the Web Dynpro Runtime.

Trigger an Object-based Navigation

To trigger an object-based navigation for a specific operation of a certain business object using the `WDPortalNavigation` service you have to use one of the defined `navigateToObject()` or `navigateToObjectWithSpecificOperation()` methods. Depending on the used signature there are the following parameters available:

- **system**
You have to specify the system (alias), to which the business object is assigned. s a mandatory parameter.
- **businessobjType**
You have to define the used business object using this mandatory parameter.

Furthermore there are some optional parameters available:

- **objValue**
Normally there are a lot of different instances for one business object. A typical example would be the usage of a business object *Customer*. To specific which concrete customer should be used for the object navigation step you have to specify the object value, i.e. in our example the customer ID.
- **operation**
Using the `operation` parameter you can specify which operation should be used for the object navigation step. If you do not define the operation the so-called *default operation* of the defined business object is used automatically.
- **objValueName**
The specified object value is transported as one URL parameter to the iView or page implementing the defined operation of the business object. The default name of this parameter is `ObjectValue`. You can specify another parameter name using the `objValueName` parameter if needed.
- **businessParameters**
Besides the object value name you can define any other parameters, which should be passed through the object navigation step. An example for a possible parameter string is `Mode=Edit&ShowHeader=false`.

These parameters could be used by the object-based navigation destination. Please keep in mind that for the object-based navigation there are the same restrictions and guidelines regarding the encoding of the values of the defined business parameters. For more details please check the part 2 of this series describing absolute and relative portal navigation.
- **forwardOBNMetaData**
Sometimes it is useful for the object-based navigation destination to know more about the current navigation step. For example, if you implement a Web Dynpro application that acts as an implementation for different operations of a business object, the Web Dynpro application has to know

which operation is triggered by the navigation step. Therefore you can forward the following parameters:

- o `obn.system`

The system the business object is assigned.

- o `obn.bo_type`

The business object itself.

- o `obn.operation`

The name of the triggered operation. If the default operation is triggered the value is `_default_`.

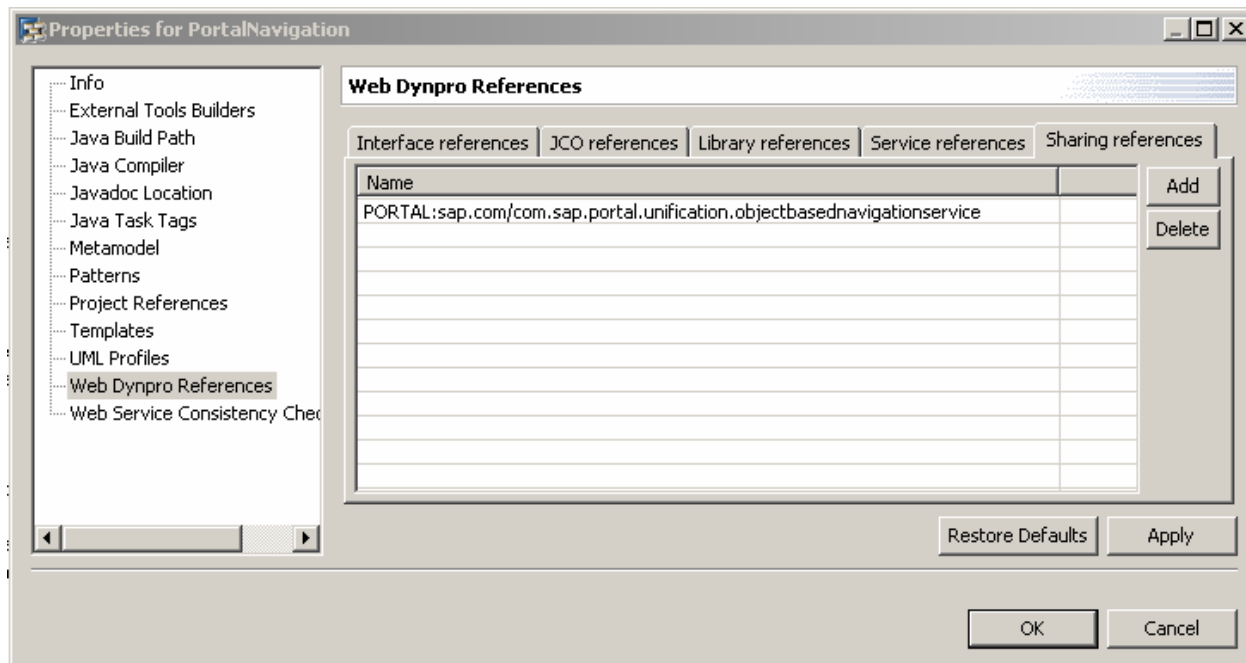
Use the IUserObjectBasedNavigation Service

Using the `WDPortalNavigation` service as described above you can trigger an object-based navigation for a certain operation of a business object. To realize a user-friendly user interface, a second service is available which provides a set of useful functionalities. This is the `IUserObjectBasedNavigation` service.

Defining a the Needed Reference

`IUserObjectBasedNavigation` is a service provided by the SAP NetWeaver Portal (i.e. a so-called *portal service*). You have to define a specific sharing reference for each Web Dynpro project using this service.

The following screenshot demonstrates the definition of this sharing reference:



To get more information how to access a portal service within a Web Dynpro Java application, please have a look to part 6 of this series.

What is a Valid Operation?

As mentioned above you define the navigation destination of an object-based navigation step on a higher semantic level as a concrete target URL. This concrete target URL depends on the configuration for the current role or user. Therefore sometimes it could happen, that for a specific role or user there is no destination configured for a certain operation of a certain business object. Therefore this operation is *invalid* for this role or user.

A *valid* operation is an operation with an defined destination (i.e. there is an iView or page in one of the user roles implementing the requested operation).

Check for a Valid Default Operation

Sometimes it is useful to know whether or not a business object has a valid target for any operation (i.e., if there is a valid target for the default operation).

The following code example demonstrates how to do this:

```
// Get a reference to the OBN service
IUserObjectBasedNavigation obnService
    =(IUserObjectBasedNavigation)
        WDPortalUtils.getServiceReference(IUserObjectBasedNavigation.KEY);

// Get the current user
IUser user = null;
try {
    user = WDClientUser.getCurrentUser().getSAPUser();
} catch (WDUMException e) {
    wdComponentAPI.getMessageManager().reportException(
        "Failed to get current user: " + e.getLocalizedMessage(), true);
}

// Define the system and the business object
String system = "MySystem";
String bo = "customer";

// Call the service
boolean hasValidDefaultOperation = obn.isTargetExist(system, bo, user);
```

You can also use this function to make sure to render the correct UI element depending on the configuration for the current user or to enable or disable a `LinkToAction` UI element, which is used to trigger the object-based navigation for the default operation.

Check for a Valid Operation

The following code example shows how to check if a certain operation has a valid target:

```

// Get a reference to the OBN service
IUserObjectBasedNavigation obnService
    =(IUserObjectBasedNavigation)
        WDPortalUtils.getServiceReference(IUserObjectBasedNavigation.KEY);

// Get the current user
IUser user = null;
try {
    user = WDClientUser.getCurrentUser().getSAPUser();
} catch (WDUMException e) {
    wdComponentAPI.getMessageManager().reportException(
        "Failed to get current user: " + e.getLocalizedMessage(), true);
}

// Define the system, the business object and the operation
String system = "MySystem";
String bo = "customer";
String operation "Display";

// Call the service
boolean operationHasValidTarget
    = isTargetExistsForOperation(system, bo, operation, user);

```

Get the List of Valid Operations

Besides the possibility to check for a valid default operation or for a valid target for any operation, you can use the `IUserObjectBasedNavigation` service also to get the list of valid operations for a certain business object. The following code example demonstrates how to do this:

```

// Get a reference to the OBN service
IUserObjectBasedNavigation obnService
    =(IUserObjectBasedNavigation)
        WDPortalUtils.getServiceReference(IUserObjectBasedNavigation.KEY);

// Get the current user
IUser user = null;
try {
    user = WDClientUser.getCurrentUser().getSAPUser();
} catch (WDUMException e) {
    wdComponentAPI.getMessageManager().reportException(
        "Failed to get current user: " + e.getLocalizedName(), true);
}

// Define the system and the business object
String system = "MySystem";
String bo = "customer";

// Get the list of valid operations
List operations = obn.getTargets(system, bo, user);

// Fill dynamically a context node with the operation information
IMyTestView.IOperationsElement newOperation = null;

for (Iterator iter = operations.iterator(); iter.hasNext();) {
    IOBNTarget target = (IOBNTarget) iter.next();

    newOperation = wdContext.nodeOperations().createOperationsElement();
    newOperation.setCaption(target.getOperationFriendlyName());
    newOperation.setName(target.getOperationName().lastIndexOf('/') + 1);
    wdContext.nodeOperations().addElement(newOperation);
}

```

You can display a list of operations in any way you want, for example in a `DropDownByIndex` UI element.

Use a Web Dynpro iView as a Target

To use a Web Dynpro based iView as a target for an object-based navigation you have to do the same steps for any other iView. To make sure that the forwarding of parameters works fine for the Web Dynpro iView you have to change the JS code, which is used by the object-based navigation to define the object value manipulation.

For each operation, your Web Dynpro iView should act as a target (or implementation). To do so, define the following JS code:

```
return \'DynamicParameter=\' + objValue;
```

The following screenshot shows the object-based navigation editor for a Web Dynpro iView. You can see also the defined JS code:

Object-Based Navigation

Add operations to "pcd:portal_content/com.sap.WebDynproTests/com.sap.wdtestsrole/com.sap.WebDynproNavigation" by choosing them from the Portal Catalog. This iView implements them at runtime.

Operation Name	Display Name	Business Object	Priority	Relation Resolving
<input checked="" type="radio"/> Details	Details	Customer	5	None
<input type="radio"/> Display	Display	Customer	10	None
<input type="radio"/> Send E-Mail	Send E-Mail	Customer	0	None

Remove Designate Relation Reset to Default

JavaScript for Operation

```
function OBNObjValueManipulation(objValue) {  
  return \'DynamicParameter=\' + objValue;  
}
```

The Web Dynpro OBNTTester

The *Web Dynpro OBNTTester* is a test and demo tool to get more familiar with all the described object-based navigation capabilities of the SAP NetWeaver Portal. The Web Dynpro OBNTTester is an example described in detail in the *Praxisbuch Web Dynpro for Java / Maximizing Web Dynpro for Java* (please have a look at the last chapter of this article).

The following screenshot shows the Web Dynpro OBNTTester.

Define the Needed Parameters to Trigger an Object-Based-Navigation

System: WebDynproBestPractices

Business Object: Artist

Operation: Search For Details

Object Value:

Object Value Name: ObjectValue

Business Parameters:

Name	Value
<input type="checkbox"/> query	Billy Joel
<input type="checkbox"/>	
<input type="checkbox"/>	

Row 1 of 1

Add OBN Meta Data

The Web Dynpro OBNTester

For example, using the Web Dynpro OBNTester you can easily check whether or not there is a valid operation for an operation of a business object or check the list of all operations.

Further Information

To get more information especially about extended features of the SAP Application Integrator we would like to bring to your attention to a new Web Dynpro for Java book. The German version *Praxisbuch Web Dynpro for Java* is available now – the English translation *Maximizing Web Dynpro for Java* will follow soon.

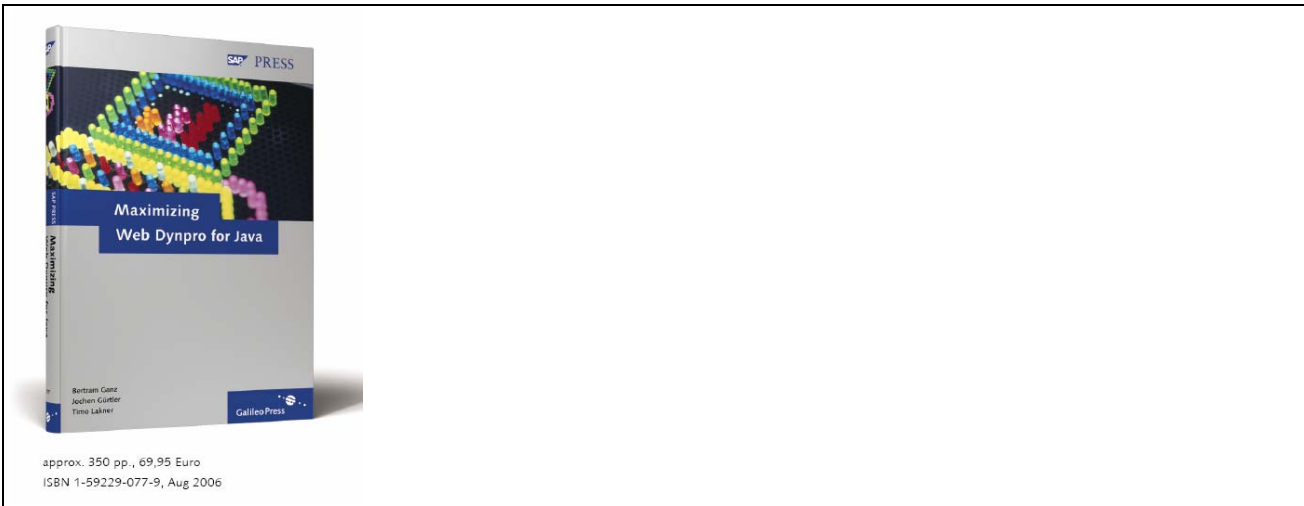


Praxisbuch Web Dynpro for Java

“Maximizing Web Dynpro for Java” is aimed at Web Dynpro developers who already have development experience but want to go to the next level. As members of the Web Dynpro development team, the authors of this book are expert at revealing tips and tricks for avoiding inefficiencies and pitfalls. They cover many areas including the SAP NetWeaver Development Infrastructure, component-oriented application design,

portal integration, object-based navigation, installation tips, configuration and administration, and more. There are also complete examples on how to write web service or portal-based applications.

Note: This book is currently available in German only. The English version will be released in August/September 2006. You can order the book through [SAP](#) or [Amazon.com](#).



Maximizing Web Dynpro for Java

Copyright

© Copyright 2006 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.

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.