

# Integrating Web Dynpro and SAP NetWeaver Portal

## Part II: Using Portal 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 portal navigation capabilities within a Web Dynpro application running in SAP NetWeaver Portal. Besides the standard navigation capabilities, we discuss how to suspend and resume a Web Dynpro application when running in SAP NetWeaver Portal and some advanced navigation possibilities.

**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.

Part II: Using Portal Navigation within a Web Dynpro Application .....	1
Applies to: .....	1
Summary.....	1
Author Bio .....	1
Introduction .....	2
The WDPortalNavigation Service .....	2
Absolute Portal Navigation .....	3
Relative Portal Navigation .....	6
Checking for Valid Navigation Targets .....	7
Accessing the Portal History.....	8
Suspending and Resuming a Web Dynpro Application.....	9
Using the wdDoExit() Plug.....	9
The Web Dynpro NavigationTester .....	10
Further Information .....	11
Copyright.....	12

## Introduction

One of the most important benefits when running your application inside the SAP NetWeaver Portal is the possibility to structure all your content in a role-based way. Using different roles you can define user-specific navigation structures, which offers each user a user-centric navigation to her needed applications.

Using the *Top-Level Navigation* (TLN) and the *Detailed Navigation* (DTN), the user can browse her specific navigation structure. Besides that, every application running as iView in SAP NetWeaver Portal can programmatically trigger navigation steps to other iViews or pages.

As SAP NetWeaver Portal could potentially run any type of content (regardless of its geographical location), the SAP NetWeaver Portal provides several client-side mechanisms to trigger such navigation steps.

Due to the fact that the client abstraction of the Web Dynpro programming model does not allow direct access to these specific client-side mechanisms, the Web Dynpro runtime environment provides the `WDPortalNavigation` service class to trigger navigation steps to other iViews or pages.

We describe below the usage of the `WDPortalNavigation` service class. In the end we will discuss some extended navigation capabilities like the suspending and resuming of a Web Dynpro application or the access to the portal navigation history.

## The WDPortalNavigation Service

To access any portal 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 environment.

Using portal navigation you have to differentiate two different ways to define the navigation target. Using absolute *portal navigation* you define the navigation target as an absolute URL. Obviously this kind of navigation target definition is not really flexible. As soon as the used navigation target is moved the navigation target URL is not longer valid.

Tip: Do not hardcode the used navigation targets inside your Web Dynpro application when using the absolute portal navigation. Make these navigation targets in general configurable to avoid code changes.

To get a more flexible definition of the used navigation targets the SAP NetWeaver portal offers *relative portal navigation* whereby all navigation targets are defined as *relative* URLs. The usage of the relative portal navigation makes especially sense when navigating between iViews or pages located in the folder or sub folder. Even if such a folder is moved completely to another location the relative URLs used for the relative portal navigation are still valid.

Besides the absolute and relative portal navigation the SAP NetWeaver Portal offers a third way to define navigation targets. Using *Object-Based Navigation (OBN)*, the navigation targets are defined by operations of a business object. We will discuss the usage of the OBN within a Web Dynpro application in the next part of this series.

## Absolute Portal Navigation

To trigger an absolute portal navigation using the `WDPortalNavigation` service you have to use one of the available `navigateAbsolute()` methods. Depending on the used signature there are the following parameters available:

### `navigationTarget`

The `navigationTarget` parameter specifies the absolute target URL (i.e. the URL to the iView or the page acting as navigation target). This absolute target URL points to the location of the iView or page in the Portal Content Directory (PCD) structure. You have to use the prefix `ROLES://`.

A valid absolute target URL is for example

```
ROLES://portal_content/administrator/super_admin/super_admin_role/com.sap.portal
.system_administration/com.sap.portal.support/com.sap.portal.web_dynpro_test_too
ls/com.sap.portal.portal_navigation.
```

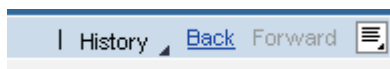
### `mode`

The navigation mode defines how the defined navigation target is displayed. Within NW04s the following modes are available:

```
WDPortalNavigationMode.SHOW_INPLACE
```

The navigation destination is displayed in-place, i.e., inside the same browser window. The Top-Level Navigation (TLN) and the Detailed Navigation (DTN) are updated accordingly.

You can use the **Back / Forward** functionality of the page header to navigate between pages that have already been visited in the session.



```
WDPortalNavigationMode.SHOW_EXTERNAL
```

The navigation destination is displayed in a new browser window without the portal frame. Only the specified iView or page is displayed.

```
WDPortalNavigationMode.SHOW_EXTERNAL_PORTAL
```

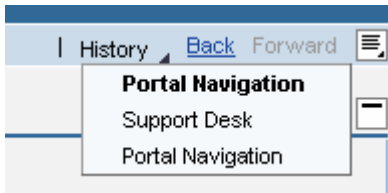
The navigation destination is displayed in a new browser window within the standard portal frame (containing TLN and DTN). The TLN a DTN are updated accordingly. As the navigation is started in a new browser window the navigation history is cleared and you cannot use the **Back / Forward** functionality of the page header to navigate between pages that have already been visited in the session.

```
WDPortalNavigationMode.SHOW_HEADERLESS_PORTAL
```

The navigation destination is displayed in a new browser window without the portal frame. Only the generic page header is displayed. As the navigation is started in a new browser window the navigation history is cleared and you cannot use the **Back / Forward** functionality of the page header to navigate back.

### historyMode

The history mode defines how the triggered navigation step is visible in the navigation history as part of the page header.



There are the following options available:

```
WDPortalNavigationHistoryMode.NO_HISTORY
```

The navigation step is *not* visible in the navigation history. You can use the **Back / Forward** functionality to navigate back.

```
WDPortalNavigationHistoryMode.NO_DUPLICATIONS
```

The navigation step is visible in the navigation history. If there are several navigation steps to the same navigation target there are only visible once in the navigation history.

```
WDPortalNavigationHistoryMode.ALLOW_DUPLICATIONS
```

The navigation step is visible in the navigation history. If there are several navigation steps to the same navigation destination but with different parameters they are visible as different entries in the navigation history.

This option makes sense if you for example navigation to an iView displaying details of a customer for example. The customer ID can be defined using a parameter and doing this several navigation steps to this details iView with different parameters (i.e., different customer IDs) results in different entries in the navigation history.

Using the `targetTitle` parameter as described below you can furthermore define the title of the entry in the navigation history (i.e., you can define titles like *Details for Customer 4711* and *Details for Customer 007* to make clear, that the same navigation destination is used but with different parameters).

### targetTitle

The `targetTitle` parameter defines the title of the entry in the navigation history.

### contextURL

The `contextURL` parameter defines the *navigation context* in which the navigation destination should be displayed. Using the `contextURL` parameter makes sense if your specified navigation destination is not visible in TLN or DTN. If you would trigger a navigation step to such a navigation destination the TLN and DTN would not be updated, because the navigation destination is not visible there. If you want to update the TDN and DTN although, you can use the `contextURL` parameter.

The `contextURL` has the same format as the absolute target URL.

## windowFeatures

If you start the navigation target in a new browser window, you can define the used window features such as window size, window position, or visible button bars.

The following table shows the most important options:

Name	Value	Description
<code>location</code>	<code>no</code> or <code>yes</code>	Defines whether or not the URL bar is displayed.
<code>menubar</code>	<code>no</code> or <code>yes</code>	Defines whether or not the menu bar is displayed.
<code>toolbar</code>	<code>no</code> or <code>yes</code>	Defines whether or not the toolbar is displayed.
<code>status</code>	<code>no</code> or <code>yes</code>	Defines whether or not the status bar is displayed.
<code>resizable</code>	<code>no</code> or <code>yes</code>	Defines whether or not the user could resize the opened browser window.
<code>scrollbars</code>	<code>no</code> or <code>yes</code>	Defines whether or not the browser windows shows (vertical and horizontal) scroll bars..
<code>width</code>	Pixel value	Defines the width of the opened browser window.
<code>height</code>	Pixel value	Defines the height of the opened browser window.
<code>posX</code>	Pixel value	Defines the horizontal position of the left corner of the opened browser window. The value 0 defines the left screen border.
<code>posY</code>	Pixel value	Defines the vertical position of the top corner of the opened browser window. The value 0 defines the top screen border.

## windowName

If you start the navigation target in a new browser window, you can define the name of this new browser window. Please keep in mind that the `windowName` parameter does *not* defines the title of the new browser window, but only the technical name of it.

### **launcherParameters**

You can define any parameters which should be passed to the specified navigation target. Navigating to an iView or page based on a SAP UI (like ITS, BSP, WebGui or Web Dynpro) you can define parameters, which should be passed only to the so-called *SAP Application Integrator* launching the specific SAP content (like a Web Dynpro application, a BSP application or an IAC based on the Internet Transaction Server (ITS)). The launcher parameters have to be defined like `NameSpace=local&ApplicationName=MyTestApp`.

Depending on the defined parameter values you have to encode the parameter values before putting together the `launcherParameters` string.

### **businessParameters**

If you want to define parameters, which are passed to the iView or page itself, you have to define them as business parameters like `CustomerID=4711&DisplayMode=Edit`.

Depending on the defined parameter values you have to encode the parameter values before putting together the `businessParameters` parameter.

### **useSAPLauncher**

If your navigation destination is content based on a SAP UI technology (i.e. BSP, ITS, BI or Web Dynpro Java or ABAP) you have to set this the `useSAPLauncher` parameter to `true`. Otherwise you have to switch it to `false`.

### **postParameters**

If you have a lot of parameters sometimes there is the need to post these parameters because of URL length restrictions. Using the `postParameters` parameter you can decide whether or not you want to post the parameters.

Tip: The correct definition of the needed parameters (both for the SAP Application Integrator and the called application itself) is very important to pass the defined parameters in the end correctly. For more details please have a look to the last chapter of this article, where we introduce the *Praxisbuch Web Dynpro for Java / Maximizing Web Dynpro for Java* book. This book contains a lot of more details and even running examples how to handle parameters correctly.

## **Relative Portal Navigation**

Using an absolute target URL could cause problems if you move the defined navigation target. Therefore in some scenarios it is much more useful to define the navigation target with a relative target URL. Such relative URLs are obviously much more stable.

For example, if you want to navigate from one page to another page stored in the same content folder, the relative navigation will work although if you move the whole folder to another place.

To trigger a relative portal navigation you have to use one of the `navigateRelative()` methods of the `WDPortalNavigation` service. Depending on the chosen signature there are different sets of parameters available.

To define the relative navigation URL you have to specify the following three parameters:

## baseUrl

The `baseUrl` parameter defines the starting point of the relative navigation target. The `baseUrl` parameter has the same format like the absolute target URL of an absolute navigation, i.e. something like

```
ROLES://portal_content/administrator/super_admin/super_admin_role/com.sap.portal
.system_administration/com.sap.portal.support/com.sap.portal.web_dynpro_test_too
ls/com.sap.portal.portal_navigation.
```

## levelsUp

Using the `levelsUp` parameter you define how many levels in the navigation hierarchy you have to go up for the relative navigation.

## path

The `path` parameter describes the path to the navigation target relative to the defined based target URL and the defined number of levels up.

All other (optional) parameters are the same as for the absolute portal navigation.

## Checking for Valid Navigation Targets

When using absolute or relative portal navigation you have to make sure that the defined navigation target is valid. Otherwise you get an empty screen. For some scenarios this is not acceptable and therefore it could make sense to check the used navigation target *before* triggering the navigation.

Unfortunately there is no simple API to check whether or not a navigation target is valid, but using a special portal service allows implementing the needed functionality.

Have a look at the Web Dynpro `PortalNavigationApp` test application as part of the attached `LocalDevelopment~extppnav~sap.com` Web Dynpro development component.

Using the `IPortalNavigation` portal service we can check a navigation target. The following code example (located in the `onActionCheckTarget()` action event handler of the `PortalNavigation` view controller) shows the details:

```

// Get the instance of the navigation service
INavigationService navService =
    wdContext.currentContextElement().getNavigationService();

// Define the needed environment
Hashtable environment = new Hashtable();
try {
    environment.put("User",
        WDClientUser.getCurrentUser().getSAPUser()
            .getUniqueName());
    environment.put("NavigationPrincipal",
        WDClientUser.getCurrentUser().getSAPUser());
} catch (WDUMException e) {
    wdComponentAPI.getMessageManager().reportException(
        "Failed to load user: " + e.getLocalizedMessage(),
        true);
}

// Try to get the navigation node
INavigationNode targetNode =
    navService.getNode(
        environment,
        wdContext.currentNavigationElement().getTarget());

// The navigation target is valid in case
// the navigation node is not NULL
wdContext.currentNavigationElement().setIsValidTarget(targetNode != null);

```

This test application is also a good example how to use a portal service inside your Web Dynpro application. We will discuss the usage of portal services in more detail in part six of this series.

## Accessing the Portal History

As described above you can decide whether or not a navigation step (absolutely or relatively defined) triggered by your Web Dynpro application is visible in the portal history. If you decided to make the navigation step visible in the portal history the user can easily navigate between the recent navigation steps using the generic portal history as part of the page header.

For some scenarios it could make sense to provide a **Back** button inside your Web Dynpro application to navigate back to the last showed iView or page.

You can do this by firing a specific portal event (we will describe the usage of portal eventing inside a Web Dynpro application in the part 5 of this series more in detail). The following code example shows the details:

```
WDPortalEventing.fire (
    "urn:com.sapportals:navigation",
    "historyNavigate",
    "-1" );
```

In the example above you navigate back one step by passing the `-1` value. Passing other values you can navigate back for more than one step. Using values greater than 0 you can navigate forward (as long as there is a navigation step available for that in the portal history).

## Suspending and Resuming a Web Dynpro Application

Using the *suspend* and *resume* plugs of a Web Dynpro application you can leave a Web Dynpro application without destroying it completely. Doing this it is for example possible to navigate to another (Web Dynpro) application, doing some work and navigate back in the end to the suspended Web Dynpro application.

To get a similar behavior when running your Web Dynpro application inside the SAP NetWeaver Portal the only thing to do is to add the application property `sap.suspendable.application` to the Web Dynpro application.

This application property marks the application as a *suspendable* one. Running this application inside the SAP NetWeaver Portal as a Web Dynpro iView (the NetWeaver 04s version based on the Web Dynpro page builder) does not destroy the application instance when navigating away using any portal capabilities like the TLN or DTN or any programmatic way to trigger a navigation step (as described in the document).

If you use the possibility to expose more than one Web Dynpro iView for the same Web Dynpro application it is even possible now to navigate between these iViews without losing the application instance (i.e. the application state). We recommend using this flexibility to navigate between Web Dynpro iViews sharing a big amount of data. Web Dynpro iViews exposed by one Web Dynpro application, which are located in the same PCD folder or same Web Dynpro page share the same application instance.

## Using the wdDoExit() Plug

Running outside the SAP NetWeaver Portal one typical way to leave a Web Dynpro application is the usage of the `wdDoExit()` plug.

Calling this plug destroys the current Web Dynpro application and presents an (optional) application-specific screen to inform the user that the application was successfully destroyed. Besides that you can define a URL, which should be showed after leaving the Web Dynpro application. By using a Web Dynpro application URL you can navigate to another Web Dynpro application directly. But as wrote above – this is only true for Web Dynpro application running *outside* the SAP NetWeaver portal.

When running a Web Dynpro application within the SAP NetWeaver portal the usage of the `wdDoExit()` plug is restricted. Calling the plug without an URL is in general possible and would destroy the Web Dynpro application (like in the standalone case).

Note that is only useful when running several Web Dynpro iViews on one Web Dynpro page. Imagine the following scenario:

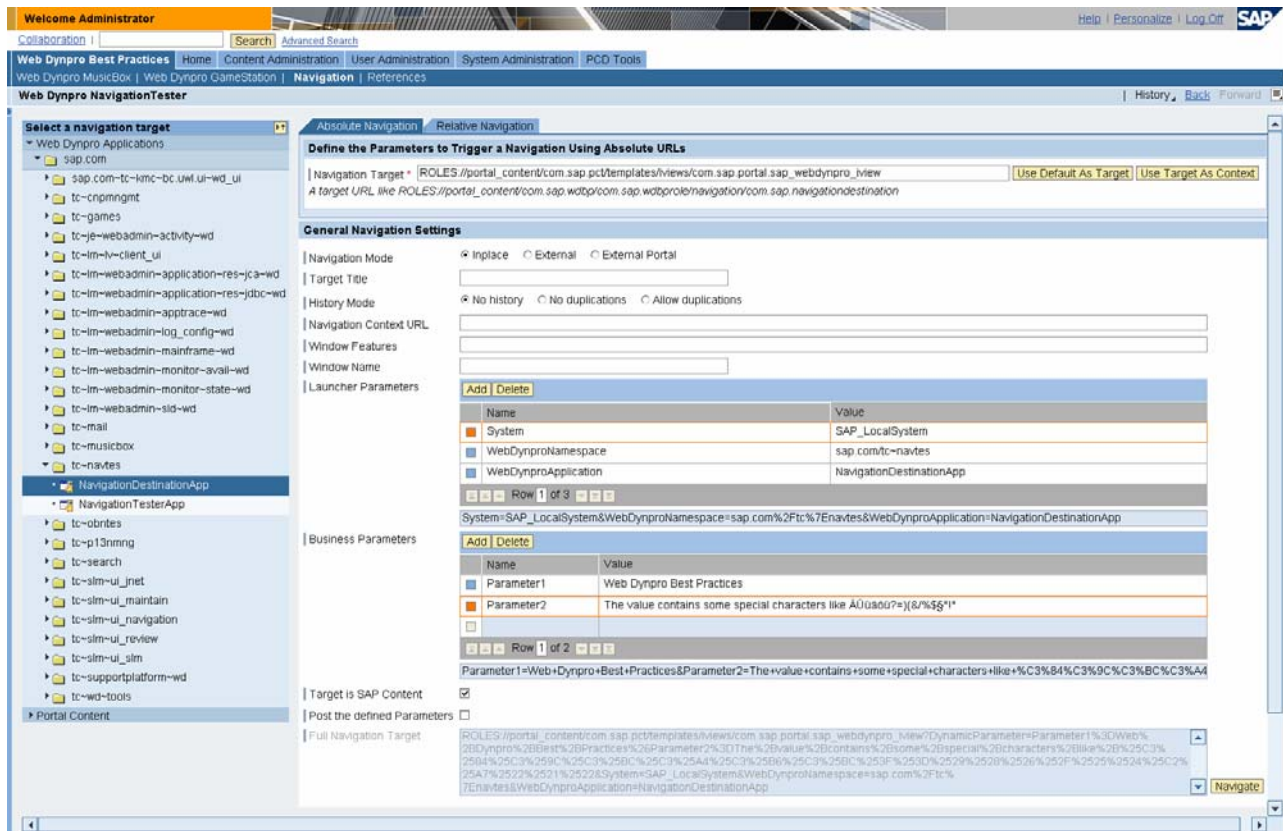
A Web Dynpro page contains several Web Dynpro iViews. All Web Dynpro iViews run as embedded iViews, i.e., triggering any action in one of the Web Dynpro iViews updates the whole page. In case one of the iViews is no longer needed you can call in the associated Web Dynpro application the `wdDoExit()` without a URL to destroy the application and to present a final screen. The Web Dynpro page builder recognizes this and shows the final screen of the iView again and again – as long as the user navigates away to another page or as long as the user restarts the iView using the **Refresh** iView tray entry.

Tip: The usage of the `wdDoExit()` plug with an URL is in general forbidden, when running a Web Dynpro application within the SAP NetWeaver portal. To navigate to another URL or even Web Dynpro application you should use one of the available portal navigation technologies (i.e. absolute or relative portal navigation or Object-based navigation).

## The Web Dynpro NavigationTester

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

The following figure shows the Web Dynpro NavigationTester running inside the SAP NetWeaver Portal.

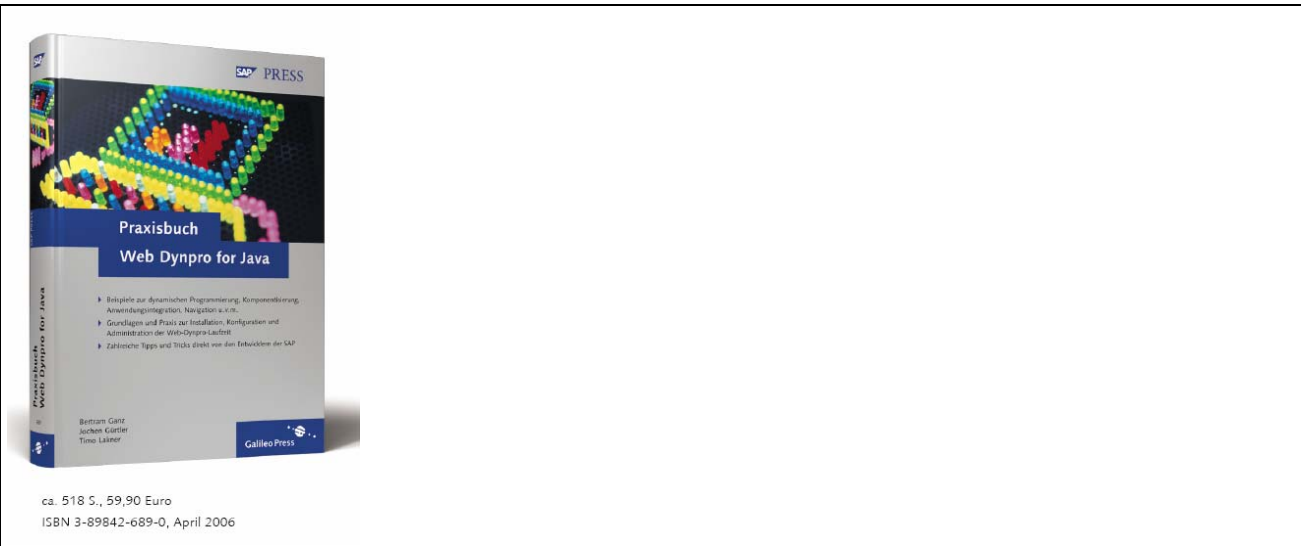


### The Web Dynpro NavigationTester

Using the NavigationTester you can browse both the deployed Web Dynpro content and the defined portal content. After selecting a Web Dynpro application or any specific iView or page you can define the needed navigation behavior. Furthermore you can define the parameters you want to pass to the selected navigation target. The NavigationTester shows automatically the correct encoding of the defined parameters and parameter values.

## 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.