

How to Create an Interactive Flash Island Slider in a Web Dynpro for Java Application



Applies to:

Web Dynpro for Java 7.11 For more information, visit the [Web Dynpro Java homepage](#).

Summary

The flash slider application explains the interaction between a Flash component and the embedding Web Dynpro Java application.

Author: Web Dynpro Java Team

Company: SAP AG

Created on: 29 June 2010

Table of Contents

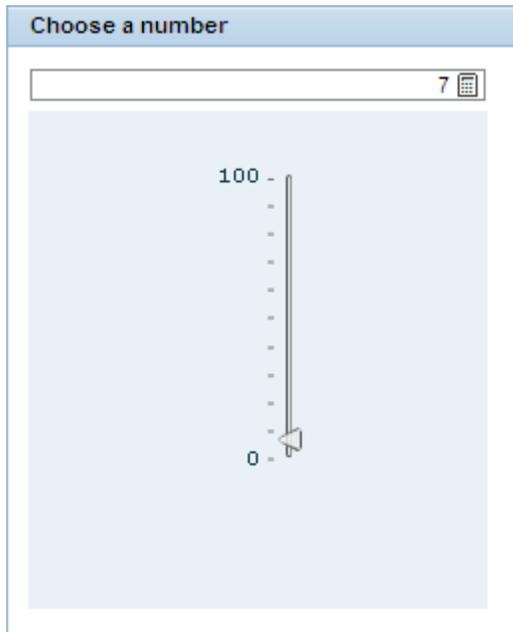
| | |
|---|----|
| Introduction | 3 |
| Prerequisites | 4 |
| Objectives | 4 |
| Prepare the Flex Project..... | 4 |
| Create the Slider | 5 |
| Wrapping the Flex component into a Flash Island..... | 6 |
| Handle the Flash Event | 7 |
| Create the FlashIsland UI element in Web Dynpro Java | 8 |
| Define the data flow | 9 |
| Enable data flow between FlashIsland and Web Dynpro component | 9 |
| Running the application..... | 10 |
| Appendix..... | 11 |
| Tutorial Result..... | 12 |
| Further Information..... | 12 |
| Legend of Text Symbols..... | 13 |
| Copyright | 14 |

Introduction

This small sample is designed to explain the interaction between a Flash component and the embedding Web Dynpro Java application.

As a Flash component we will use a slider that allows choosing a value between 0 and 100. On Web Dynpro side an *InputField* displays this value. If you now change the value of the *InputField* and press Enter, the slider's thumb moves to this value, and the other way round: If you change the slider's value, this value is also displayed in the *InputField*.

With this sample you can see how the data has to be bound to enable this connection and how the event on Flash side has to be handled to let Web Dynpro know that the data has changed.



There are different steps to establish this connection between Web Dynpro and the embedded Flash Island:

- The Flex component itself has to be wrapped into a Flash Island.
- This Flash Island has to be embedded into Web Dynpro Java.

Prerequisites

You need to install the NetWeaver Developer Studio (Version 7.11 or later) in order to compile and deploy the tutorial application. The SAP Java AS to which this application is deployed should have the same or newer version as the NWDS. You must make the corresponding settings for the server instance home name and the instance number in *Window* → *Preferences* → *SAP AS Java*, and also select the corresponding flag if two or more servers have been entered in the server list.

The tutorial application is available as a Development Component (DC). You need to import the Software Component HM-WDUIDMKT CNT, which contains the DC `tc/wd/tut/flash/sld`. The exact steps are described in a separate document.

Objectives

After working through this tutorial, you should be able to:

- Create a Flex / Flash Slider
- Integrate the user interface element *FlashIsland* in a view
- Enable the data flow between Flex / Flash and Web Dynpro Java

Prepare the Flex Project

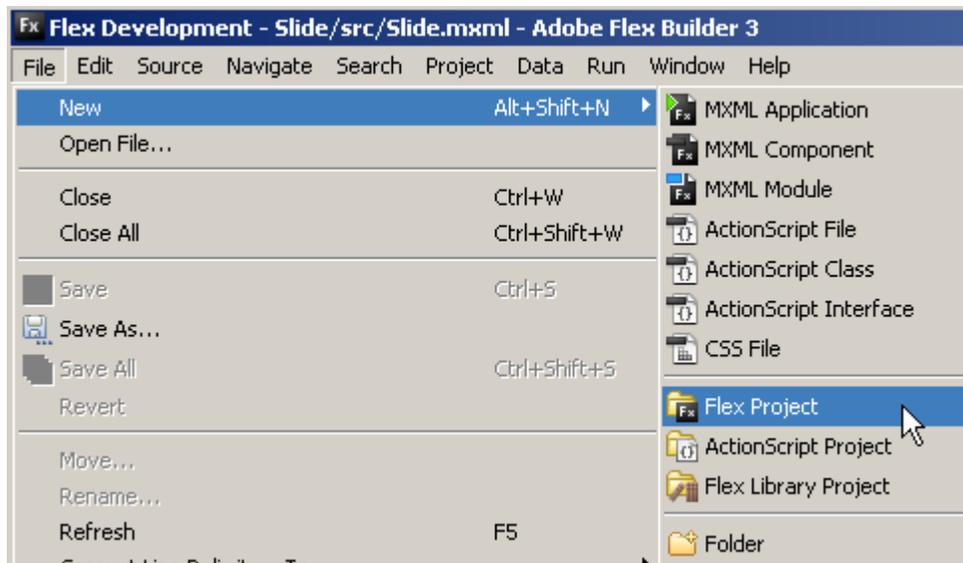
To implement a Flash Island for Web Dynpro Java applications integrating Flash Islands, you require the following:

- Adobe Flex Builder 2 or 3
- Adobe Flash Player 9 Update 3
- Flash Island Flex library `WDIslandLibrary.swc`.

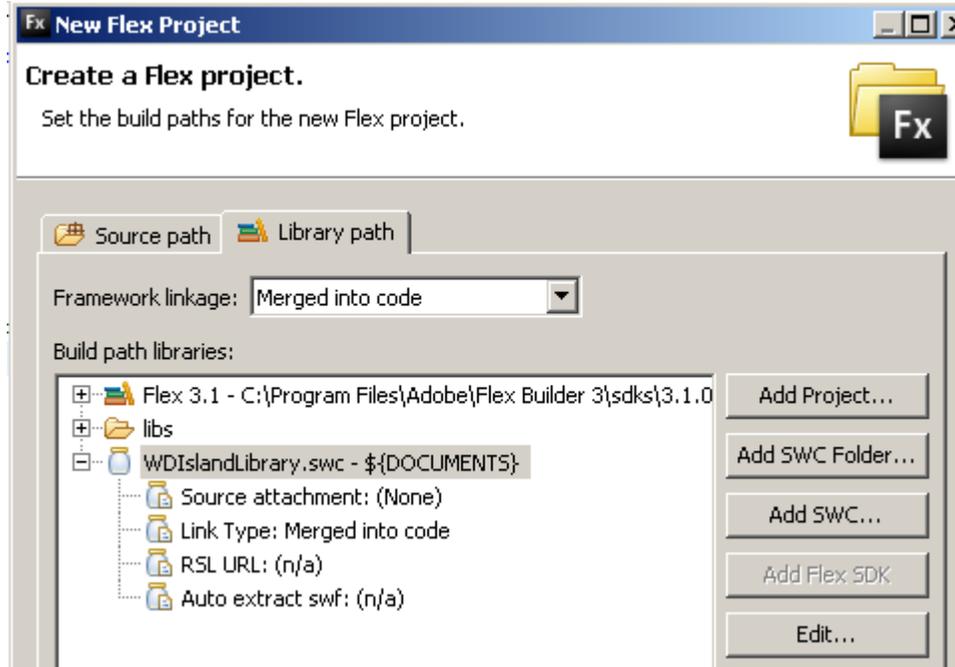
To enable the use of a Flash/Flex component in Web Dynpro, the component must exist in the form of a *FlashIsland*. This wrapping enables the communication between the Flex component and the Web Dynpro Framework. This library is available on the SAP NetWeaver Application Server CE under the following path:

...:\usr\sap\ce1\j00\j2ee\cluster\apps\sap.com\tc~wd~dispwda\servlet_jsp\webdynpro\resources\sap.com\tc~wd~dispwda\root\global\activeComp

1. In your Adobe Flex Builder, create a new Flex Project, e. g. called `Slider`.



- To add `WDIslandLibrary.swc` to the Library path, choose *Add SWC...*, browse to the directory where you have stored the library and add it.



- To specify the Flex Compiler, select the project, open the context menu and choose Flex Compiler. Use *Flex 2.0.1. Hotfix 3 as SDK*.
- Confirm with Finish. The Flex project is created.

Create the Slider

In this sample we will use a vertical slider which is part of Adobe Flex control library.



You can copy the complete code from the appendix of this document.

- You can now switch to the *Source* tab for the *Slider.mxml* editor.
- To insert the vertical slider, add the following lines of code into the `mx:Application` tag:

```
<mx:Script>
    <![CDATA[
        import mx.events.SliderEvent;
        import mx.controls.sliderClasses.Slider;
        [Bindable]
        public var xvalue:int;
    ]]>
</mx:Script>
<mx:VSlider
    id="slideIn"
    value="{xvalue}"
    maximum="100"
    minimum="0"
    snapInterval="1"
    tickInterval="10"
    dataTipPrecision="0"
    labels="['0', '100']"
    width="50" >
```

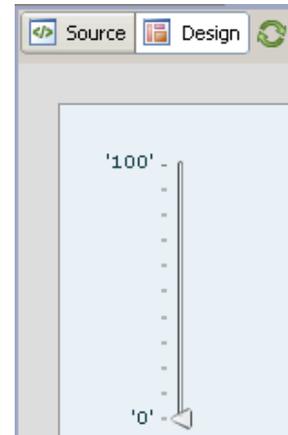
```
</mx:VSlider>
```

You declared a variable `xvalue` that is used as value for the slider's thumb. This variable will be bound in a later step to the Web Dynpro context.



The attributes `snapInterval`, `tickInterval`, `dataTipPrecision` and `labels` alter the design of the Slider. `snapInterval` defines the steps in which values can be chosen ("1" referring to integers only). `tickInterval` adds ticks to the Slider track in the interval defined (here every 10 steps of the track). `dataTipPrecision` allows you to specify the number of decimal places displayed by the data tip ("0" shows integers only). With `labels` you can add an arbitrary number of strings to label your Slider track. These alterations can be helpful if you want to display specific values only.

Switching to the *Design* tab, your slider should now look like this:



Wrapping the Flex component into a Flash Island

To establish and manage the connection to Web Dynpro, you have to implement some steps:

1. To register your Flex component as a Flash Island, create a method called `initApp()`. In this method call the *Flash Island's* method `register()` with the parameter `this`. This method is called during initialization of the Flex application:

```
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="vertical"
  width="100%"
  height="100%"
  initialize="initApp()"
  backgroundColor="#EAF1F6">
<mx:Script>
<![CDATA[
  import sap.FlashIsland;
  import mx.events.SliderEvent;
  import mx.controls.sliderClasses.Slider;
  [Bindable]
  public var xvalue:int;
  public function initApp():void
  {
    FlashIsland.register(this);
  }
  .....
```

The background is set to the SAP's standard background color to get a seamless display on the screen, in case you are not using a Group or Tray UI element.

Handle the Flash Event

1. To be able to react on the event that is triggered if the user changes the thumb's position, create the following function:

```
<mx:Script>
  <![CDATA[
    .....
    private function sliderChange(event:SliderEvent):void
    {
      var currentSlider:VSlider=event.currentTarget as VSlider;
      xvalue = currentSlider.value;
      FlashIsland.storeProperty(this,"xvalue",xvalue);
      FlashIsland.fireEvent(this, 'change', null );
    }
  ]]>
</mx:Script>
```

First the current value is stored in the `xvalue` variable and then the method `storeProperty` is called that ensures that the value is saved before the event is fired. At least the event itself is fired.

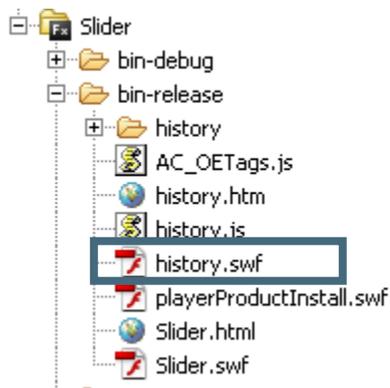
2. To call this function with the slider's change event, add the following properties to the slider:

```
<mx:VSlider
  id="slideIn"
  value="{xvalue}"
  maximum="100"
  minimum="0"
  width="50"
  snapInterval="1"
  tickInterval="10"
  dataTipPrecision="0"
  labels="['0', '100']"
  liveDragging="false"
  change="sliderChange(event)">
</mx:VSlider>
```

`liveDragging` has to be set to false to ensure that only one event is triggered when the user releases the slider's thumb.

The Flash component is now finished and you can build the `swf` file that is needed in Web Dynpro. To do this:

1. Select *Project -> Export Release Build...* and follow the wizard. You now can find the `Slider.swf` file in the `bin-release` folder.

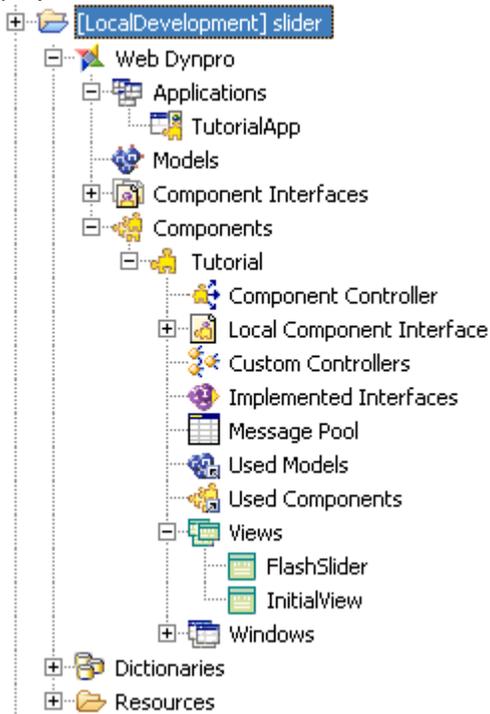


Create the FlashIsland UI element in Web Dynpro Java

You now have finished all steps needed on Flex site for creating a Flash Island that can be used in Web Dynpro Java.

The Flash Island UI element spans an entire view, that's why you have to create a separate view for it.

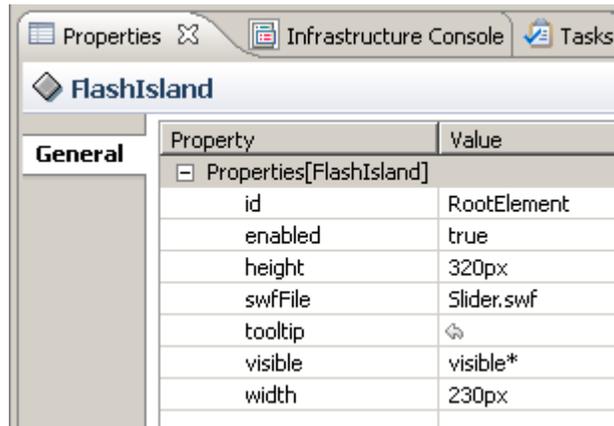
1. Create a Web Dynpro Development component (DC) and add a new view to the window. Your DC project should look like this:



2. Add a *Group UI element* to the `InitialView`, adjust the heading and insert a *ViewContainer UI element* into the group. Add the `FlashSlider` view into it.
3. Open the `FlashSlider` view and replace the `RootElement`'s default container with a *FlashIsland*. To do so, select the `RootElement`, open the context menu and choose *Replace With... -> FlashIsland*.
4. Specify appropriate *width* and *height* values to avoid scroll bars to appear.
5. Insert the following view elements into this *RootElement*:
 - *Property1* (`GACProperty`)
 - *Event* (`GACEvent`)

The *Property* will help you bind the slider's value and the *Event* has to be connected to the event you've created in the Flash component.

6. Copy the *Slider.swf* file into the mime repository (The relevant path is: `<Web Dynpro DC name>` → *Resources* → *src* → *mimes* → *Components* → `<component name>`) and specify *Slider.swf* as value for the *FlashIsland*'s *swfFile* property.



Define the data flow

We now have to create a context attribute to be able to store and transfer the slider's value.

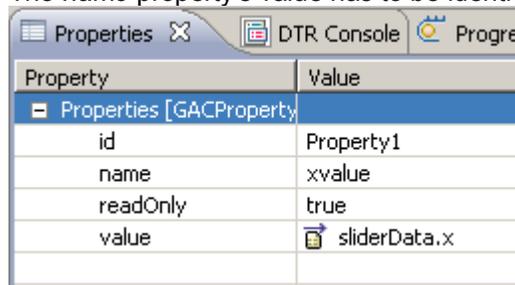
1. In your component controller, create a context node called `sliderData` and an attribute called `x` of type *integer* inside this node.
2. Set the node's cardinality property to *1:n*
3. Map this node to both view's contexts.
4. Create an *InputField* in the `InitialView` and bind the value to the `x`-attribute.

Enable data flow between FlashIsland and Web Dynpro component

1. Switch back to the `FlashSlider` view and select the `Property1` element and set the following values:

- Enter `xvalue` as *name*.

The *name* property's value has to be identical to the variable's name of the Flash component!



- Bind the *value* property to the `x` attribute.
2. For the *Event* view element set `change` as *name*. Ensure that this value is identical to the slider's event name!
 3. Create an action as event handler to be able to trigger a roundtrip when the user triggers the event.
 4. Assign this action also to the *InputField's* `onEnter` event.
 5. Initialize the value of the slider by adding the following code to the `wdDoInit()` of your component controller:

```
public void wdDoInit()
{
    //@@begin wdDoInit()
    ISliderDataElement elem;
    elem = wdContext.createAndAddSliderDataElement();
}
```

```
    elem.setX(7);  
    //@@end  
}
```

Running the application

1. Save your DC, choose *Deploy New Archive and Run* from the context menu.
2. Play around and have fun!

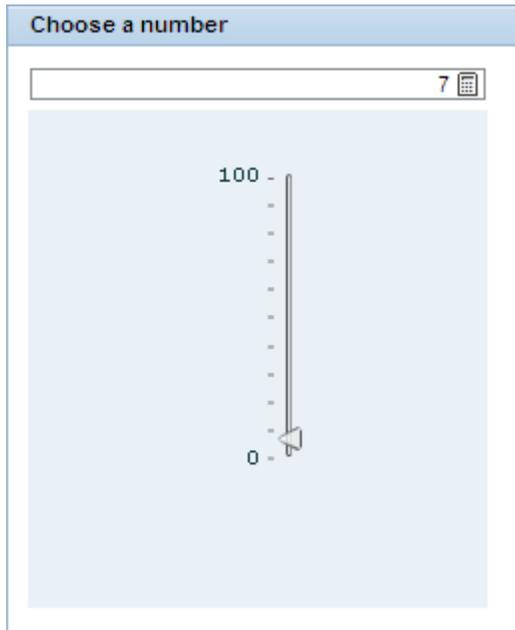
Appendix

Complete Flash code:

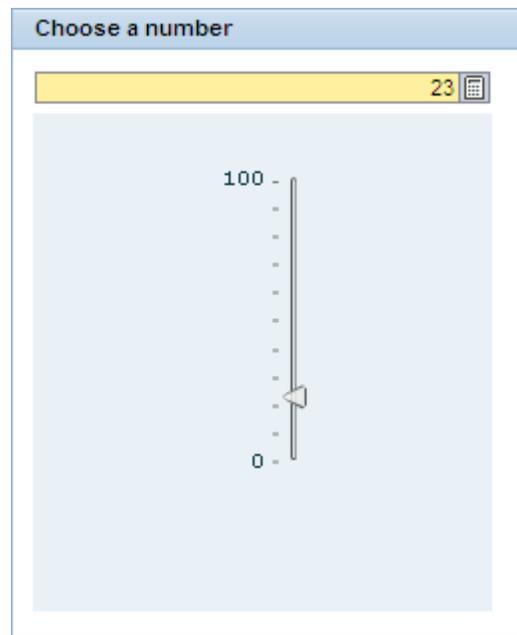
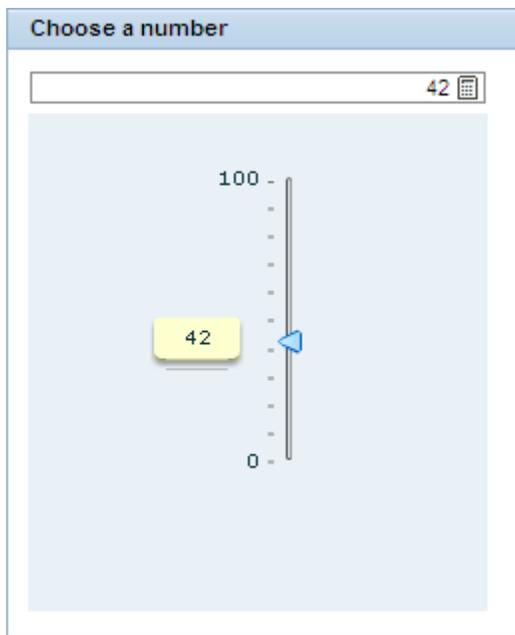
```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
  xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="vertical"
  width="100%"
  height="100%"
  initialize="initApp()"
  backgroundColor="#EAF1F6">
<mx:Script>
<![CDATA[
import sap.FlashIsland;
import mx.events.SliderEvent;
import mx.controls.sliderClasses.Slider;
[Bindable]
public var xvalue:int;
public function initApp():void
{
    FlashIsland.register(this);
}
private function sliderChange(event:SliderEvent):void
{
    var currentSlider:VSlider=event.currentTarget as VSlider;
    xvalue = currentSlider.value;
    FlashIsland.storeProperty(this, "xvalue", xvalue);
    FlashIsland.fireEvent(this, 'change', null );
}
]]>
</mx:Script>
<mx:VSlider
  id="slideIn"
  value="{xvalue}"
  maximum="100"
  minimum="0"
  snapInterval="1"
  tickInterval="10"
  dataTipPrecision="0"
  labels="['0', '100']"
  width="50"
  liveDragging="false"
  change="sliderChange(event)">
</mx:VSlider>
</mx:Application>
```

Tutorial Result

The complete application looks as follows: There is a Group UI element with a header that displays the title “Choose a number”. Within the Group UI element there are an InputField UI element and a ViewContainer UI element that displays the FlashIsland UI element. The initial value of the Slider and InputField is 7.



You can either change the value of the Slider and InputField by dragging the thumb to a desired position (while holding down the mouse button, the dataTip will appear) or you can enter a value into the InputField and press Enter.



Further Information

FlashIsland is one of a large set of user interface elements for Web Dynpro for Java. For more information about *FlashIsland* properties or any other user interface elements that Web Dynpro provides, see the UI Element Guide at

http://help.sap.com/saphelp_nwce711/helpdata/en/9f/656442a1c4de54e10000000a155106/content.htm.

Alternatively, you can navigate directly to the *FlashIsland* topic in the SAP Help Portal at

http://help.sap.com/saphelp_nwce711/helpdata/en/48/8303caacb21b5ae10000000a42189c/content.htm.

Legend of Text Symbols

| Symbol | Usage |
|---|----------------|
|  | Note |
|  | Recommendation |
|  | Warning |
|  | See also |

Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, 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.