



BEA Oracle[®] WebCenter Application Accelerator for Installation and Upgrade Guide .NET

Version 10.3
Revised: November, 2008

Contents

Introduction...5

About the Oracle WebCenter Application Accelerator for .NET.....	5
Hardware and Software Requirements for Oracle WebCenter Application Accelerator for .NET.....	8
Additional Documentation.....	9

Installation...11

Installing the Oracle WebCenter Application Accelerator for .NET.....	11
Installing the Oracle WebCenter Interaction Portlet Toolkit for .NET.....	12
Installing the .NET WSRP Producer.....	13

Upgrading Existing Applications...15

Upgrading WSRP Producer 1.0 Installations.....	15
Upgrading .NET Application Accelerator Portlets.....	17
Upgrading Oracle WebLogic Portal .NET Application Accelerator Portlets.....	18
Upgrading Existing WCC Applications (2.1.x and earlier).....	19

Introduction

About the Oracle WebCenter Application Accelerator for .NET

The Oracle WebCenter Application Accelerator for .NET is a collection of libraries and Microsoft Visual Studio 2005 integration features that simplify authoring of ASP.NET 2.0 portlets for Oracle WebCenter Interaction and Oracle WebLogic Portal.

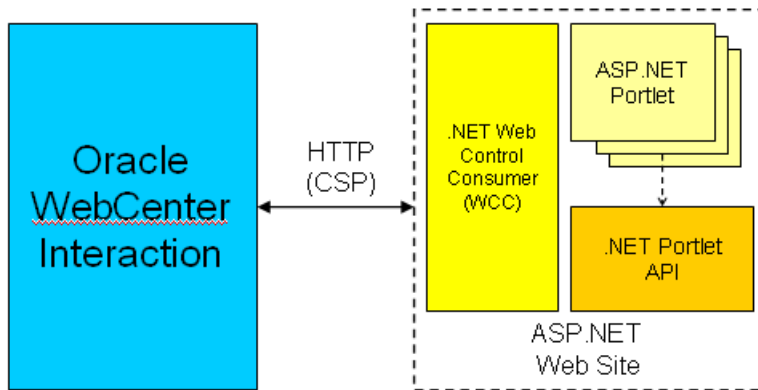
Each of the components in the Oracle WebCenter Application Accelerator for .NET serves a specific purpose:

- **.NET WSRP Producer:** Surfaces ASP.NET web applications to WSRP (Web Services for Remote Portlets) Consumers by transforming WSRP requests into HTTP requests, and rewriting HTTP response content for the WSRP Consumer. Requires use of the Oracle WebCenter .NET Portlet API.
- **.NET Portlet Toolkit:** Provides a set of project templates, file templates and a class library used to create portlets in Visual Studio. Includes separate templates and classes for WSRP (Oracle WebLogic Portal) and Oracle WebCenter Interaction portlet development.

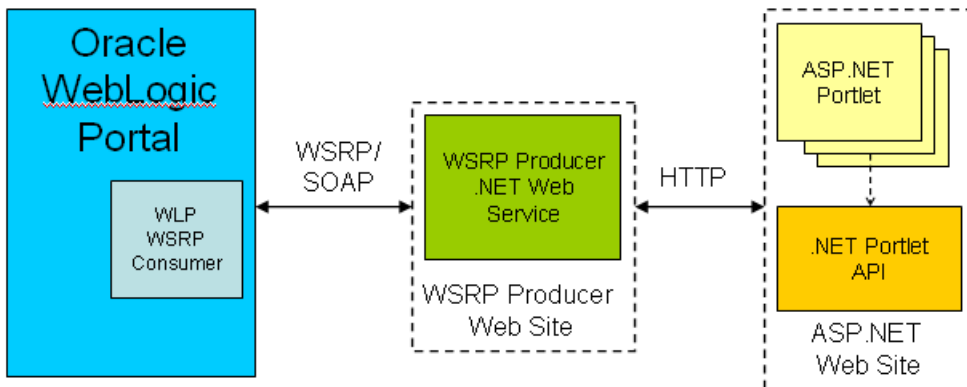
- **Web Control Consumer (WCC):** Transforms output rendered by Oracle WebCenter Interaction portlets into pages that can be consumed in a portal environment (Oracle WebCenter Interaction only). This component is not required for WSRP portlets.

A different set of components is required for each portal configuration, as shown below.

To build and deploy **Oracle WebCenter Interaction portlets**, the following components are required. In addition, the Image Service Files included in the .NET Portlet Toolkit must be installed on the portal Image Service.

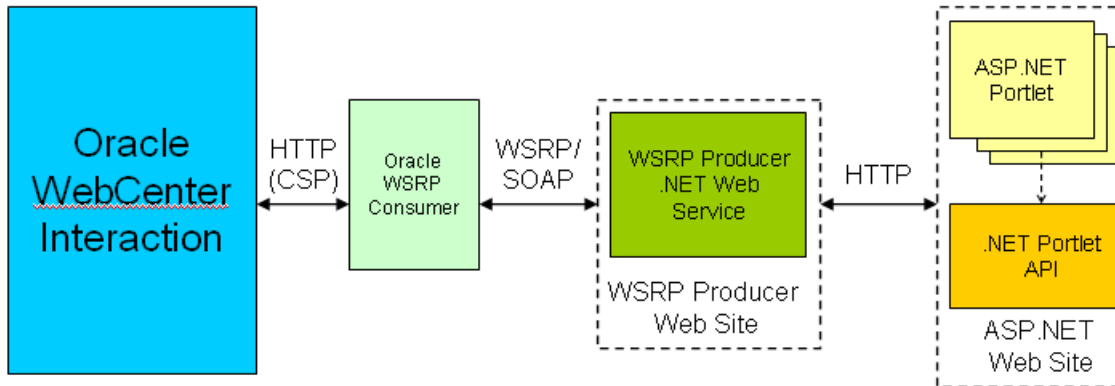


To build and deploy **WSRP portlets in Oracle WebLogic Portal**, the following components are required. In addition, the Microsoft Web Services Enhancements (WSE) 2.0 must be installed on the server that hosts the WSRP Producer and ASP.NET web sites.



To build and deploy **WSRP portlets in Oracle WebCenter Interaction**, the WSRP Producer and the Oracle WSRP Consumer (separate product and installation) are required. In addition, the Microsoft Web Services Enhancements (WSE) 2.0 must be installed on the server that hosts the WSRP Producer and ASP.NET web sites.

Note: This configuration is only efficient if you are building cross-platform WSRP portlets for both Oracle WebCenter Interaction and Oracle WebLogic Portal; if you are building portlets for Oracle WebCenter Interaction only, use the first Oracle WebCenter Interaction configuration shown above.



For a complete list of requirements, see *Hardware and Software Requirements for Oracle WebCenter Application Accelerator for .NET* on page 8.

Hardware and Software Requirements for Oracle WebCenter Application Accelerator for .NET

The following hardware and software is required to develop and deploy portlets using the Oracle WebCenter Application Accelerator for .NET.

Table 1: Hardware and Software Requirements

Component	Requirements
Host Computer	<ul style="list-style-type: none">• Windows Server 2003 SP2• Microsoft ASP.NET 2.0• Microsoft Web Service Enhancements (WSE) 2.0 SP3• IIS 6.0 (WSRP portlets only)• Microsoft Visual Studio 2005 (development machine only) <p>Note: Microsoft Web Services Enhancements (WSE) 2.0 is required on the server that hosts the WSRP Producer and/or .NET Portlet API. The WSE installation package is included in the Oracle WebCenter Application Accelerator for .NET installation package in the /thirdparty folder.</p>
Platform	<ul style="list-style-type: none">• Oracle WebCenter Interaction 10.3• BEA AquaLogic Interaction 6.0 and above• Oracle WebLogic Portal 10.3• BEA WebLogic Portal 10.2, 10 MP1 or 9.2



Additional Documentation

The following additional documentation is available.

Resource	Description
Installation Guide	This guide describes the prerequisites (such as required software) and procedures for installing and upgrading Oracle WebCenter Portlet Toolkit for .NET components.
Release Notes	The release notes provide information about new features, issues addressed, and known issues in the release.
Development Guide	The <i>Oracle WebCenter Portlet Toolkit for .NET Development Guide</i> provides detailed information on authoring portlets using the .NET Portlet Toolkit.

Installation

Installing the Oracle WebCenter Application Accelerator for .NET

To install the Oracle WebCenter Application Accelerator for .NET, complete the following steps.

Make sure your system meets the prerequisites for installation. For information on which components are required for your configuration, see [About the Oracle WebCenter Application Accelerator for .NET](#) on page 5.

The 1.1 MP1 release of the .NET Application Accelerator includes two installers:

- Install the .NET WSRP Producer. For detailed instructions, see [Installing the .NET WSRP Producer](#) on page 13.
- Install the .NET Portlet API, WCC and image files. For detailed instructions, see [Installing the Oracle WebCenter Interaction Portlet Toolkit for .NET](#) on page 12.

After installation, to upgrade existing applications, see the following topics:

- To upgrade your WSRP Producer installation, see [Upgrading WSRP Producer 1.0 Installations](#) on page 15.
- To upgrade Oracle WebCenter Interaction portlets built using the .NET Portlet Toolkit 1.0, see [Upgrading .NET Application Accelerator Portlets](#) on page 17.
- To upgrade WLP portlets built using the .NET Portlet Toolkit 1.0, see [Upgrading Oracle WebLogic Portal .NET Application Accelerator Portlets](#) on page 18.
- To upgrade portlets built using the Web Control Consumer 2.1 or earlier, see [Upgrading Existing WCC Applications \(2.1.x and earlier\)](#) on page 19.

Installing the Oracle WebCenter Interaction Portlet Toolkit for .NET

To install the Oracle WebCenter Portlet Toolkit for .NET component of the Oracle WebCenter Application Accelerator for .NET, including the .NET Portlet API, WCC and related files, run WebCenterPortletToolkitForDotNet_XX.exe.

The installation package installs the .NET Portlet API, WCC and related files. .

1. Copy the installation package to a location on the host server.
2. Launch the installer (**WebCenterPortletToolkitForDotNet_XX.exe**).
3. Complete the installation wizard pages as described in the table that follows.

Wizard Page	Description
Choose Components	<p>Choose the components to install:</p> <ul style="list-style-type: none"> • .NET Portlet Toolkit: Installs class libraries and VisualStudio templates for both Oracle WebCenter Interaction and WSRP portlets. • .NET Web Controls Consumer (WCC): Installs all runtime components required to run the WCC. If you are developing WSRP portlets, this component is not required. • Image Service Files for Image Server: Installs the URL-addressable resources that must be available for the WCC to operate correctly. This option can be used to install these resources on a separate static content server such as IIS or Apache. If you are developing WSRP portlets, this component is not required. <p>Note: If you are installing in a distributed system (Oracle WebCenter Interaction is on a different server than the server that hosts portlets), you must run the installer twice: 1) install the .NET Portlet Toolkit and WCC on</p>

Wizard Page	Description
	the remote server that hosts portlets, and 2) install the Image Service Files on the server that hosts the Image Service.
Choose Install Folder	Enter the path to the folder where the Portlet Toolkit files should be installed. (This page appears only if the .NET Portlet Toolkit and/or WCC options were selected.)
Path to Image Service Files	This page displays the location on the server from which static content is served. The location should be the same as the Image Service. (This page appears only if the Image Service Files for Image Server option was selected.)
Pre-Installation Summary	Review your choices before clicking Install to proceed.
Installation Complete	Choose whether or not to restart your system and click Done to close the installer.

Once installation is complete, a set of project and file templates will be available in Microsoft Visual Studio 2005, and the installation directory will contain necessary resources and libraries.

Installing the .NET WSRP Producer

To install the .NET WSRP Producer component of the Oracle WebCenter Application Accelerator for .NET, run WebCenterWSRPProducerForDotNet_XX.exe.

The WebCenterWSRPProducerForDotNet_XX.exe. installation package installs only the WSRP Producer. For instructions on installing the .NET Portlet API, WCC and related files, see [Installing the Oracle WebCenter Interaction Portlet Toolkit for .NET](#) on page 12.

1. Copy the installation package to a location on the host server.
2. Launch the installer (**WebCenterWSRPProducerForDotNet_XX.exe.**).
3. Complete the installation wizard pages as described in the table that follows.

Wizard Page	Description
Choose Install Folder	Enter the path to the folder where the WSRP Producer should be installed.

Wizard Page	Description
Select IIS Web Site	Select whether to deploy the WSRP Producer in the default IIS web site or another web site.
Specify IIS Web Site Information	If you chose to deploy to another web site (not the default IIS web site), enter the name for the WSRP Producer web site and its unsecured and secured ports. Note: This page does not appear if you selected the default IIS web site on the previous page.
Pre-Installation Summary	Review your choices before clicking Next to proceed.
Installation Complete	Click Done to close the installer. You will be prompted to restart your machine.

Once installation is complete, the WSRP Producer web site will be registered with IIS. The WSRP Producer web site will be available at the following URL:
<http://<iis-website-address>/wsrpproducer1.1>.

Note: If an incorrect web site port is specified during install, the Producer installation will be configured with incorrect service URLs and will not be accessible. The installer determines the hostname for Producer web service URLs from the fully-qualified domain name of the host machine. If this is not correct (for example, if the IIS instance uses a hostname alias) the service might not be accessible. The installer sets the Producer web service URLs using the http protocol; to use https, the service URLs must be updated. To update the Producer service URLs to resolve any of these issues, see *Configuring WSRP Producer Service URLs (WSRPService.wsdl)* in the *Oracle WebCenter .NET Portlet Toolkit Development Guide*.

Note: Microsoft Web Services Enhancements (WSE) 2.0 is required on the server that hosts the WSRP Producer and .NET Portlet API. The WSE installation package is included in the Oracle WebCenter Portlet Toolkit for .NET installation package in the /thirdparty folder.

Upgrading Existing Applications

Upgrading WSRP Producer 1.0 Installations

To upgrade an existing WSRP Producer 1.0 installation, you must update the `portlets.xml` file to use the latest `wsrp-producer.xml` file format.

In the 1.0 version of the WSRP Producer, portlet metadata was entered in the **`portlets.xml`** file located in the `\bin` subdirectory of the WSRP Producer web application. In the 1.1 MP1 and above versions of the WSRP Producer, this file has been renamed **`wsrp-producer.xml`** and contains additional configuration settings that control the functionality of the WSRP Producer.

The 1.0 `portlets.xml` format is not forward compatible with later versions of the WSRP Producer and must be upgraded to the `wsrp-producer.xml` format. The required changes are summarized below.

1. Change the `portlets.xml` file name to “`wsrp-producer.xml`”.
2. Wrap the 1.0 root element `<portlets>` in the new root element `<wsrp-producer>`.
3. Change the default namespace to
`http://www.bea.com/al/dotnet/wsrpproducer/1.1.`

4. Add a namespace declaration to the root element for the xml schema namespace using the “xs” prefix as follows: `xmlns:xs="http://www.w3.org/2001/XMLSchema"`
 5. Within each `<portlet>` element, make the following changes:
 - a) Remove all `<portlet-name>` elements. This element is no longer used.
 - b) If you have multiple `<display-name>` elements for a single portlet, remove all but one. Only a single `<display-name>` element is allowed for each portlet. If you included display name localization information in `portlets.xml`, it must be changed to the new format. For detailed instructions, see **Localizing Portlet Metadata** in the *Oracle WebCenter Portlet Toolkit for .NET Development Guide*.
 - c) Rename the `<view-url>` element to `<url>` and add the `id` attribute with the value “default”..
 - d) In the `<supports>` element, make the following changes in each `<portlet-mode>` element:
 1. Add the child element `<name>` and set the text content as follows:
 - If portlet-mode was “View” use “wsrp:view” as the value of the `<name>` element.
 - If portlet-mode was “Edit” use “wsrp:edit” as the value of the `<name>` element.
 - If portlet-mode was “Preview” use “wsrp:preview” as the value of the `<name>` element.
 2. Add the child element `<url>` with a single `idref` attribute with the value “default”.
- For example, this entry in `portlets.xml`:
- ```
<portlet-mode>View</portlet-mode>
```
- becomes this entry in `wsrp-producer.xml`:
- ```
<portlet-mode>
  <name>wsrp:view</name>
  <url idref="default"/>
</portlet-mode>
```
- e) In the `<portlet-info>` element, add a `lang` attribute with the appropriate locale (usually “en”) to the `<title>` and `<short-title>` elements.
 - f) Rename the `<portlet-preferences>` element to `<portlet-properties>`. Rename each `<preference>` child element to `<property>` and add a `type` attribute with the value “xs:string”.

For more information on `wsrp-producer.xml`, see *WSRP Producer Configuration Elements* (*wsrp-producer.xml*) in the *Oracle WebCenter Portlet Toolkit for .NET Development Guide*. For a detailed schema reference, see the `wsrp-producer.xsd` file in the WSRP Producer bin directory.

Upgrading .NET Application Accelerator Portlets

To upgrade existing Oracle WebCenter Interaction (or AquaLogic Interaction) portlets developed using the .NET Application Accelerator version 1.0, you must make modifications to your code.

If your portlet uses the profile provider, you must change the name and type of the profile provider defined in your project's `Web.config` file from `EDKProfileProvider` to `IDKProfileProvider` as shown in the code snippet below.

```
<profile enabled="true"
defaultProvider="IDKProfileProvider"
automaticSaveEnabled="true">
  <providers>
    <clear />
    <add name="IDKProfileProvider"
type="BEA.ALI.Web.Profile.IDKProfileProvider,
PortletAPI" description="IDK based profile provider"
/>
  </providers>
</properties>
  <clear />
</properties>
</profile>
```

The other namespace changes implemented in version 1.1 MP1 provide new classes and do not affect existing 1.0 portlets. (You may need to add namespaces to access new functionality.)

Upgrading Oracle WebLogic Portal .NET Application Accelerator Portlets

To upgrade existing Oracle WebLogic Portal portlets developed using the .NET Application Accelerator version 1.0, you must make modifications to your code.

If your portlet uses the profile provider, you must make the following changes:

1. Change the name and type of the profile provider defined in the project's Web.config file from EDKProfileProvider to IDKProfileProvider as shown in the code snippet below.

```
<profile enabled="true" defaultProvider="IDKProfileProvider"
automaticSaveEnabled="true">
  <providers>
    <clear />
    <add name="IDKProfileProvider"
type="BEA.ALI.Web.Profile.IDKProfileProvider, PortletAPI"
description="IDK based profile provider" />
  </providers>
  <properties>
    <clear />
  </properties>
</profile>
```

2. Upgrade property names to the 1.1 MPI syntax. In version 1.0, the property set name was appended to the property name. In 1.1 MPI, the property name is placed in a group element.
 - a) For each custom property set or user profile grouping, create a <group> element in the <properties> element within the <profile> element in the Web.config file for the portlet project. Add each user profile property to its respective group. Each property must be defined with a name and a type. If you provide a default value, it will be used if the property is not available.

Note: The name of the property **must** match the name sent by the source application. For more information about sending user profile properties from WebLogic Portal, see *Developing User Profiles in the Oracle WebCenter Portlet Toolkit for .NET Development Guide*.

```
<properties>
  <group name="homeInfo">
```

```

    <group name="online">
      <add name="email" type="string"/>
    </group>
  </group>
  <group name="CustomProperties ">
    <add name="title" type="string" defaultValue="No Title"/>
  </group>
  <group name="MyProfile">
    <add name="Name" type="String" defaultValue="Guest"/>
    <add name="Age" type="Int32"/>
  </group>
</properties>

```

- b) Reference the property in the portlet page by name using the Profile object. The correct syntax is shown below.

```

string title = Profile.CustomProperties.Title;
string name = Profile.MyProfile.Name;
int age = Profile.MyProfile.Age;

```

The other namespace changes implemented in version 1.1 MP1 provide new classes and do not affect existing 1.0 portlets. (You may need to add namespaces to access new functionality.)

Upgrading Existing WCC Applications (2.1.x and earlier)

To migrate existing applications built with the Web Control Consumer 2.1.x or earlier, you must make minor code modifications. Upgrading from versions 2.2.x and above requires no changes to the existing code.

Even though 2.1.x is very similar to later versions, there are some differences that might require some changes to existing code.

- **Modify JavaScript code to run only once:** In 2.1.x and earlier, JavaScript was executed only upon the first rendering of the page. Some applications may assume that inline code will be run only once; this code must be modified to run only once. (Many existing applications register for events upon the initial render; this code must be changed to prevent multiple registrations.)

This can be done easily on the server, by including the code `only if IsPostBack == false`. This can also be done on the client a few ways, such as:

```
<script>
  if(!already_run)
  {
    //run my code here...
    var already_run = true;
  }
</script>
```

- **Change `ptrender` setting:** In 2.1.x, buttons did not submit a form in-page by default. The default behavior now is to submit the form in-page, which can be disabled by added a `ptrender=false` attribute to the button. If there are buttons relying on the default behavior of not performing an in-page refresh, these must be changed to include the `ptrender=false` attribute (including a `ptrender=true` attribute now has no effect).