



# Infor Enterprise Server Connector for Web Services Administration and User Guide

---

Copyright © 2013 Infor

## Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

## Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

## Publication Information

---

<b>Document code</b>	U9209I US
<b>Release</b>	10.3
<b>Publication date</b>	August 7, 2013

---

---

# Table of Contents

## About this document

<b>Chapter 1 Introduction.....</b>	<b>7</b>
Description of the Connector for Web Services.....	7
Concept and Use of the Connector for Web Services.....	7
Development time architecture.....	9
<b>Chapter 2 Requirements.....</b>	<b>11</b>
Overview.....	11
LN back end support.....	11
Baan IV or Baan 5.0 back end support.....	12
SLM License.....	12
Proxy Development Tools.....	12
<b>Chapter 3 Installation.....</b>	<b>13</b>
Installation overview.....	13
Installing the connector using the setup program.....	14
Installation program.....	14
Manual installation.....	14
To deploy the connector to an existing Tomcat server.....	15
Updating or uninstalling the connector.....	17
<b>Chapter 4 Configuration and deployment.....</b>	<b>19</b>
Configuring the Connector for Web Services.....	19
Web Services Status page.....	20
ERP Servers Configuration.....	22
Configuring Infor LN or Baan application servers.....	22
ERP Servers page.....	23
Add/Modify an ERP Server page.....	24
Web Services Deployment.....	28
Deployed Web Services page.....	28

---

---

Deploy BDE Web Services page.....	29
Deploy BOI Web Services page.....	30
Log configuration.....	30
Logging.....	30
Log Configuration properties.....	31
<b>Chapter 5 Employment.....</b>	<b>33</b>
Setting up web services for BDEs.....	33
Overview.....	33
BDE access through web services.....	34
To deploy a web service.....	34
Setting up web services for BOIs.....	36
Overview.....	36
BOI access through web services.....	37
To deploy a web service.....	37
Overruling ERP servers configuration.....	39
Developing a Web Client.....	40
Starting and stopping the connector.....	40
<b>Chapter 6 Troubleshooting.....</b>	<b>43</b>
Tracing down errors.....	43
Miscellaneous.....	43
<b>Appendix A Web Service Concepts.....</b>	<b>45</b>
<b>Appendix B Definitions, abbreviations and acronyms.....</b>	<b>47</b>
<b>Index</b>	

---

---

# About this document

## Objectives

This document describes how you use the Connector for Web Services to set up and run web services to access BDE business objects on an Infor LN back end or BOI's on a Baan IV or Baan 5.0 back end. The guide additionally discusses the installation and configuration of the connector.

*BDE stands for Business Data Entity, which is an LN-specific representation of a business object.*

*BOI stands for Business Object Interface, which is a Baan specific representation of a business object.*

## Pre-requisites

### **Connector for Web Services applies to BDE interfaces version 8.4 or later**

The connector provides web services if the back end conforms to the following pre-requisites:

- BDE business interfaces implementations generated with Infor Business Studio version 8.4 (or later)
- Enterprise Server version 8.4 (or later)

### **Connector for Web Services applies to all versions of BOI interfaces**

The connector provides web services if the back end conforms to the following pre-requisites:

- Baan IV
- Baan 5.0

## Document summary

This document includes the following chapters:

- **Chapter Introduction (p. 7)**  
Provides general product information on the Connector for Web Services.
- **Chapter Requirements (p. 11)**  
Describes the pre-requisites of the Connector for Web Services.
- **Chapter Installation (p. 13)**  
Describes the installation of the Connector for Web Services. Both automated and manual installation are discussed. The chapter also mentions updating and uninstalling the connector.
- **Chapter Configuration (p. 19)**  
Describes how to configure the Connector for Web Services to be able to run web services for particular BDE or BOI business objects.
- **Chapter Employment (p. 33)**  
Describes how to run a web service for a BDE business object on an LN back end and a webservice for a BOI on a Baan IV and Baan 5.0 back-end.
- **Chapter Troubleshooting (p. 43)**  
Deals with finding the source of an error and solving known problems.

Appendix *Web Service Concepts* (p. 45) provides some background information on web service standards, such as WSDL and SOAP.

### Related documents

- *Infor Enterprise Server Installation Guide (U9484 US)*
- The Infor LN Business Studio documentation

For more information about BDE business objects and the messages they apply to exchange business data, refer to:

- Infor Integration 6.2 - *Integration Guide for BDE Clients* (U9051 US)
- Infor Integration 6.2 - *Integration Guide for BOI Clients* (U9145 US)

### Note

The terms BDE and BOI are used throughout this document, but are used for different software versions: BDE is used for LN, and BOI is used for Baan IV and Baan 5.0. The term Infor LN or Baan contains the software versions Infor LN, Baan 5.0, and Baan IV. If one specific version is mentioned, it means that the other versions do not support this case.

### Caution!

The web services for BDE business objects discussed in the *Integration Guide for BDE Clients* does not apply to the Connector for Web Services, because Infor Integration uses its own web application to provide BDE web services, called the *Web Services Toolkit (wstk)*. To run this connector, you require the transport and messaging infrastructure of Infor Integration .

### Comments?

We continually review and improve our documentation. Any remarks/requests for information concerning this document or topic are appreciated. Please e-mail your comments to [documentation@infor.com](mailto:documentation@infor.com) .

In your e-mail, refer to the document number and title. More specific information will enable us to process feedback efficiently.

### Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at [www.infor.com/inforxtreme](http://www.infor.com/inforxtreme) .

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

This chapter introduces the Infor ES Connector for Web Services and provides a brief conceptual overview.

## Description of the Connector for Web Services

This section gives a brief description of the Infor ES Connector for Web Services.

The Connector for Web Services is a Java servlet, which implements a web application to access business objects through a web service. This web service is SOAP-based and exposes data from BDE or BOI business interfaces implemented on Infor LN or Baan back ends. The web services consist of server requests from a web client that call methods on BDE or BOI business objects implemented on the Infor LN or Baan application back end. The connector receives and unwraps the SOAP messages from a web client, redirects the contained method calls to the Infor LN or Baan application, wraps the reply to the call in a SOAP message again, and sends the reply back to the web client.

## Concept and Use of the Connector for Web Services

This section describes the concept of the Connector for Web Services and sketches in general terms how the connector is used to build a web-based integration with an Infor LN or Baan back end.

Infor ES Connector for Web Services is a Java servlet that enables you to access BDE or BOI business objects on an Infor LN or Baan back end. This Java servlet is called *c4ws* and runs in the managed environment of a Tomcat server.

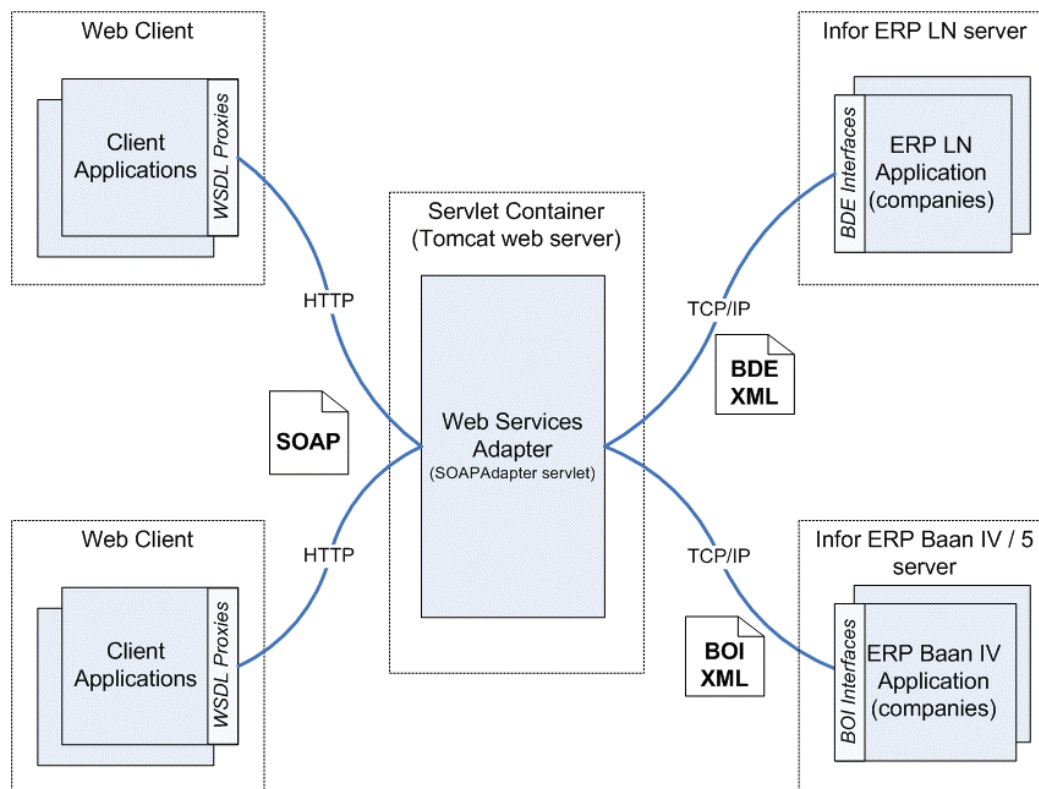
To implement a web service for a specific business object, you use the configuration pages of the Connector for Web Services. On these pages, you define which business objects need to be deployed in the web server. For each deployed web service a WSDL file will be available. The WSDL file completely defines the web service and has been generated using the Business Studio. After deployment, the service is running and ready to be used.

Next, any web client that has access to the web service can use the service. This means that a web client can send a SOAP request message to the web service and that, if all goes well and no errors occur, the web service sends back a SOAP response message that contains the return value of the invoked method on the BDE or BOI business object. If something went wrong, the web service returns a SOAP result message that contains information about the errors. Web client applications can use the deployed WSDL file to generate WSDL proxies, which form a Client API of the web services for the involved BDE or BOI business object.

To view the current status of the deployed web services and to monitor incoming requests for a particular service, the connector also provides a status page. This *Web Services Status page* (p. 20) shows which web services are currently active and specifies, for each service, the Infor LN or Baan specific back end, the BDE or BOI business object, the URL of the service and number of requests handled by the service. If you click the WSDL link of a web service, you can view the WSDL file of this service in your web browser, and save it to a local .wsdl file to use it for the generation of proxies.

## Runtime architecture

The following figure illustrates the various runtime components that build up a web-based integration for the business objects of an Infor LN or Baan application server.



The Connector for Web Services receives and unwraps a SOAP request message from a web client and redirects the contained method call to an Infor LN or Baan application server (an application component or *company* on this server). The server executes the call and returns a reply to the web



service, which consists of the method output and any associated error messages. In the web service, the Connector for Web Services wraps the reply in a SOAP message and sends it back to the web client.

BDE and BOI messages contained in the SOAP message represent the request from the client or the reply of the Infor LN or Baan server. These BDE and BOI messages have an XML vocabulary that is specific to the BDE and BOI business interfaces. For more information about these messages, refer to the:

- *Infor Integration - Integration Guide for BDEs (U9051 US).*
- *Infor Integration - Integration Guide for BOIs (U9145 US).*

The exchange of SOAP messages between the web client and the web server is based on the HTTP transport protocol. The exchange of BDE and BOI messages between the web server and the Infor LN or Baan back end occurs through a TCP/IP connection. The communication over TCP/IP is realized through a built in JCA implementation. This implementation replaces the communication through Infor Integration.

## Development time architecture

When you build a web-based integration, you must first set up the connections between the Connector for Web Services and the Infor LN or Baan back ends from which you want to access BDE and BOI business objects. To configure a back end connection, you use the *Add/Modify an ERP Server (p. 24)* configuration page of the connector. On this page, you enter the host name of the Infor LN or Baan server, indicate which activation protocol you want to use and specify various other Baan configuration properties, such as the company number, the authentication data to log on to the application, the BSE location, and the Bshell name. Make sure that the system that runs the connector is connected with the desired Infor LN or Baan back ends.

Next, you create a web service for a particular BDE or BOI business object by deploying it on the Deploy DBEs or Deploy BOIs configuration page. This deployment will make a WSDL file available, which describes the BDE and BOI business object, its properties, the functions to manipulate these properties and the address of the server application that implements the business object.

### **Important! for BDE web services:**

You can only deploy a web service for a BDE business interface from Infor LN or Baan if the implementation of this interface was generated with Business Studio version 8.4 or later. You cannot import WSDL files for BOD business interfaces.

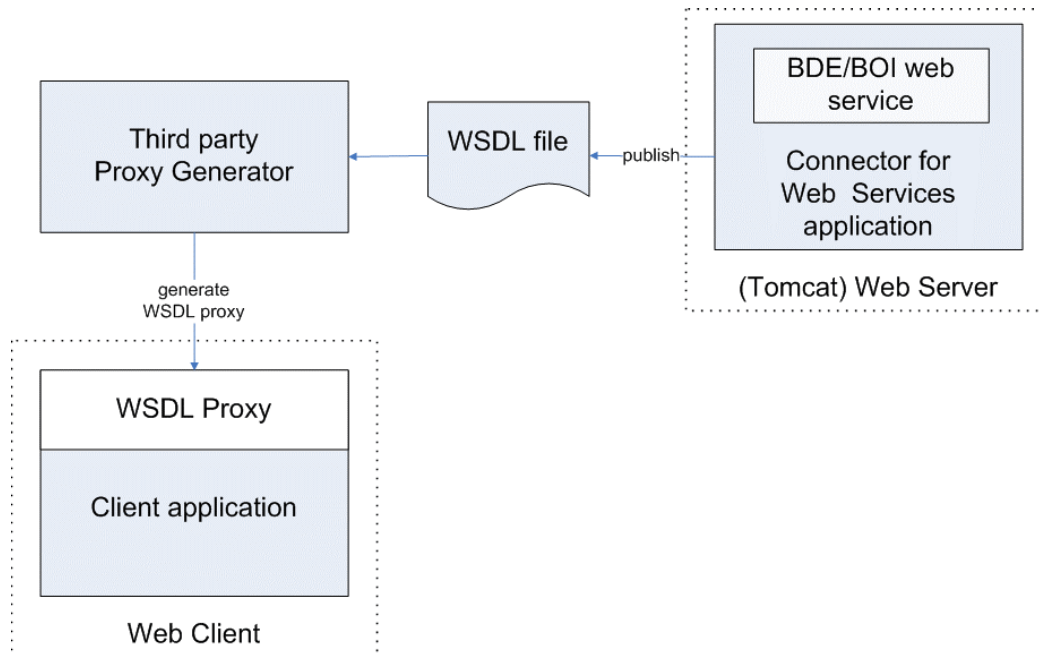
You can use the WSDL file to generate WSDL proxies for the web client. To generate these proxies, you must use a proxy generator provided by a third party web services toolkit vendors, for example, the Proxy Generator of AXIS 1.4, a Java SOAP implementation provided by Jakarta. You first retrieve the WSDL file from the connector, copy the WSDL file to your local system and then import the file in the Proxy Generator.

To test the web services, you can also use third party web services toolkit applications. Some toolkits have test facilities to generate default requests for all web service operations from the WSDL file that defines the service.

**Note**

The *Web Services Status page* (p. 20) of the connector displays the URL address of the web service, which is the SOAP address for the endpoint of the message. The URL of the WSDL file that defines the web service corresponds with the URL address your web browser displays after you clicked the WSDL link on the Web Services Status page of the connector.

The following figure illustrates the basic procedure of importing WSDL files for business objects to generate WSDL proxies for a web client.



This chapter describes the requirements of the Connector for Web Services. For the platforms supported by the installer, refer to *Infor Enterprise Server Installation Guide (U9484 US)*.

## Overview

To install and run the Connector for Web Services, your system must fulfill the following requirements:

- Java Runtime Environment 1.6.0 or later (Java version 6).
- A web server that supports the Servlet 2.4 specification, such as Apache Tomcat 7.0.30 or Sun Java System Web Server 7.0.
- An LN server supported by the Enterprise Server Installer. For details, see the *Infor Enterprise Server Installation Guide (U9484 US)*.
- On this back end, a license to access the LN business objects.
- Development tools for generation of client proxies should support SOAP 1.1 and WSDL 1.1.

If you use the Enterprise Server Installer, you do not require a supported Java VM. or Web Server, because the setup program installs these components as well. For the platforms supported by this installer, refer to *Infor Enterprise Server Installation Guide (U9484 US)*.

## LN back end support

The Connector for Web Services supports LN back ends that meet the following preconditions:

- Enterprise Server 8.4 (or later)
- BDE Business Interface Implementations generated with Business Studio 8.4 or later.

To access the BDE business objects on an LN application server, you require an SLM license on this server (see section SLM License).

## Baan IV or Baan 5.0 back end support

The Connector for Web Services supports Baan IV and Baan 5.0 back ends that meet the following preconditions:

- BaanIV portingset 6.1c.08 or higher
- Baan 5 portingset Enterprise Server 8.4 (or later)
- BOI Business Interface Implementations generated with Business Studio 2.5 or later.

To access the BOI business objects on a Baan IV or Baan 5.0 application server, you require an SLM license on this server (see section SLM License).

## SLM License

The Connector for Web Services is licensed through Infor Solution License Manager (SLM). To access the business objects on an Infor LN or Baan server application, you require an SLM Server license with Product ID 7056. This license must be implemented on the Infor LN or Baan server.

## Proxy Development Tools

Development tools for proxy generation need to support the following specifications:

- SOAP 1.1 (specification found at <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>)
- WSDL 1.1 (specification found at <http://www.w3.org/TR/wsdl>).

In addition the SOAP messages that are supported by the Connector for Web Services must use the RPC binding style. This is also defined in the SOAP binding section of the published WSDL files.

This chapter describes the installation of the Connector for Web Services.

## Installation overview

The software of the Connector for Web Services consists of a WAR file representing Java servlets. You must run these servlets in the managed environment of a Java Servlet Container. You can install the WAR file and a Tomcat Servlet Container using the Enterprise Server Installer. A Java virtual machine is included, which runs automatically during the setup process. The installer not only installs the software, but also deploys the servlets (web applications) to the Servlet Container (web server).

The installation of the connector consists of the following actions:

1. *Fulfill the requirements*

Make sure your system is supported and fulfills all software pre-requisites.

For more details, refer to section Requirements or *Infor Enterprise Server Installation Guide (U9484 US)*.

2. *Prepare the installation and run the installer*

First, make the installation disk available. Then, run the installer and follow the instructions of the installation wizard.

For some additional information, refer to section *Installing the connector using the setup program (p. 14)* . For detailed information, refer to the documentation of the installer.

You can also install the connector manually by deploying the WAR files of the connector to a SOAP-based, servlet-compliant web server. You can, for example, prefer a manual installation in the following situations:

- you do not want to run an installer
- you already use a web server
- you have a complex setup with multiple web servers

**Note**

Updating and uninstalling with the Enterprise Server Installer requires some files that were installed during the fresh install. If you do not use the installer, you cannot update or uninstall using the installer.

For more details, refer to section *Manual installation* (p. 14) .

You can also use the Enterprise Server Installer to update or uninstall the software. For details, refer to *Updating or uninstalling the connector* (p. 17) .

## Installing the connector using the setup program

This section briefly describes how you install the Connector for Web Services using the Enterprise Server Installer. For details, refer to the documentation of the installer.

### Installation program

To install the Connector for Web Services, run the setup program that applies to your platform and follow the instructions of the installation wizard. During the installation procedure, you can select which software component you want to install. The Connector for Web Services is one of these components.

The installer can perform the following tasks:

- Install a pre-configured instance of a Tomcat web server version 7.0.30. On Windows platforms, the web server can run as a Windows service.
- Copy the connector web application to your system and deploy it to the Tomcat web server.
- Update or uninstall previous installations.

You can choose which tasks the installer must carry out. Note that update and uninstall tasks are only available if a previous installation exists.

To start the installer on systems that support HTML, open the `install.htm` file and click **Startup Installer (<size>) for <platform> (includes Java VM)**. If your system does not support HTML, you must select the setup program that is specific to your platform.

## Manual installation

This section describes the manual installation of the Connector for Web Services.

The connector runs as a Java web application (also known as servlet) in the managed environment of a web server. The web application is packed in the following Web ARchive (WAR) file on the installation disk.

WAR file name	Web Application name	Location
c4ws.war	c4ws	Staging_Area/ InstallableUnits/c4ws

Use this WAR file to deploy the connector to an existing web server. To do this, follow the instructions in the documentation of the web server that you are using. For Tomcat web servers this process is described in the following section because some additional deployment steps are required.

## To deploy the connector to an existing Tomcat server

The following instructions explain how to deploy the web application of the Connector for Web Services to an existing Tomcat server.

To keep the Tomcat application 'clean' and avoid overwriting of your configuration, install the web applications in a directory structure outside the Tomcat file hierarchy.

### Step 1: Create directory structure

On the server that runs Tomcat, create the following directory structure:

UNIX folder	Windows folder	Use
/infor/ese/warfiles	C:\infor\ese\warfiles	Temporary storage war files.
/infor/ese/c4ws/web	C:\infor\ese\c4ws\web	Connector for Web Services web application
/infor/ese/c4ws/conf	C:\infor\ese\c4ws\conf	Connector for Web Services configuration data

The previous directory structure is used here to demonstrate manual deployment of the WAR file to the Tomcat web server. You can, of course, use any other suitable directory structure.

The /infor/ese/ and C:\infor\ese\ folders are further referred to as the *Infor installation folder*.

### Step 2: Copy WAR files to temp folder

Copy the c4ws.war file from the Enterprise Server Installer installation disk to the warfiles subdirectory of the Infor installation folder. You find this WAR file in the c4ws subdirectory of the Staging\_Area/InstallableUnits folder on the installation disk.

The `c4ws.war` file represents the Connector for Web Services.

### Step 3: Extract WAR files in application folders

Manually extract the WAR file into its application folder, `c4ws/web`.

For UNIX, use the following commands:

```
cd /infor/ese/c4ws/web
$JAVA_HOME/bin/jar xf /infor/ese/warfiles/c4ws.war
```

For Windows, use the following commands:

```
cd /D C:\infor\ese\c4ws\web
%JAVA_HOME%\bin\jar.exe xf C:\infor\ese\warfiles\c4ws.war
```

### Step 4: Create a context file

Create a `c4ws.xml` context file in the folder that contains the WAR files. Copy this XML file to the `Conf/Catalina/localhost` folder of your Tomcat installation, for example `C:\infor\ese\apache-tomcat-7.0.30\Conf\Catalina\localhost` for Windows.

Tomcat can deploy a web application using a context configuration XML file. If you use such a file, you can distribute the Tomcat software, the web applications and the configuration data over different folders. The context file is used to specify the location of the web applications and the location of the configuration data these applications use. You are advised to use a context file to store the connector web applications and their configuration data in separate folders outside the Tomcat file hierarchy, to avoid that you overwrite your configuration data when updating Infor or Tomcat software or lose web applications when re-installing Tomcat.

The following `c4ws.xml` file is an example of a context file for the Connector for Web Services running on a Windows platform. The file indicates that the web application is located in `C:\infor\ese\c4ws\web` and that the configuration data is stored in `C:\infor\ese\c4ws\conf`.

```
<Context docBase="C:\infor\ese\c4ws\web">
  <Parameter name="configLocation"
    value="C:\infor\ese\c4ws\conf"
    override="false"/>
</Context>
```

### Step 5: Restart the Tomcat Manager

Stop and restart Tomcat to make sure the web application is deployed.

#### Note

The configuration file of the Connector for Web Services is located in the `< Infor installation folder>/c4ws/web/WEB-INF` folders.



## Updating or uninstalling the connector

If you previously used an Enterprise Server Installer to install the software of the connector, you can re-run use the same setup program to update or remove the software. For detailed information on the subject, refer to the documentation of the Enterprise Server Installer.

If you manually installed the connector software, you remove the connector by deleting the directories containing the connector web application and the connector configuration data. To update the connector software, you remove the content of the web application directory and extract in this directory the WAR file of the new version of the connector.



# Chapter 4

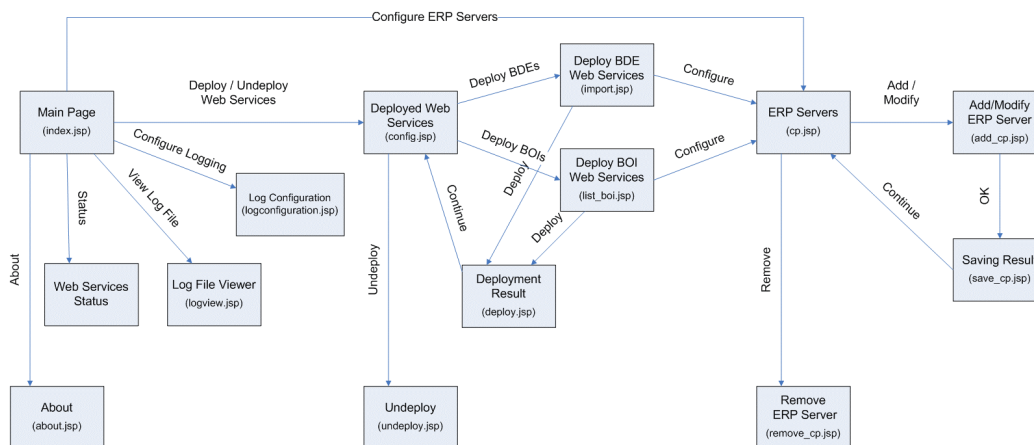
## Configuration and deployment

# 4

This chapter describes how to configure the Connector for Web Services to run BDE web services, which includes the deployment of web services.

## Configuring the Connector for Web Services

This section describes the configuration of the Connector for Web Services. The connector is used to run web services that expose BDE or BOI business objects implemented on Infor LN or Baan application servers.



Pages and navigation relationships

The configuration of the Connector for Web Services includes:

- **ERP Servers Configuration**

The connector must be configured to define the connection between the connector and the Infor LN or Baan application servers. These back ends, which are referred to as the ERP Servers, host the BDE or BOI business interface implementations for which the connector can provide web services.

For more information, refer to section *Configuring Infor LN or Baan application servers* (p. 22) , the *ERP Servers page* (p. 23) and the *Add/Modify an ERP Server page* (p. 24) .

- **Web Services Deployment**

The connector must also be configured to deploy a BDE or BOI web service for a specific BDE or BOI business interface.

To carry out this configuration, complete the following steps:

- a. Select a BDE or BOI interface from a connected ERP Server.
- b. Import the WSDL file that defines the web service for this BDE or BOI.
- c. Deploy the WSDL file to the connector.

For more information, refer to the *To deploy a web service* (p. 34) section, the *Deployed Web Services page* (p. 28) and the *Deploy BDE Web Services page* (p. 29) .

- **Logging Configuration**

The runtime processes of the connector are logged in a log file. To view and set the log configuration, you use the *c4ws log configuration page*.

For details on logging configuration, refer to section *Log configuration* (p. 30) .

The connector additionally displays the current status of all deployed BDE web services on the following page:

- **Web Services Status**

The Web Services status page displays the deployed web services, the status of the services and the number of requests handled by the services. Use this page to view and monitor the web services.

For more information, refer to *Web Services Status page* (p. 20)

The home page of the Connector for Web Services includes links through which you can reach previous configuration pages. The connector's home page is the first page that appears when you start the web application of the connector and has the following URL address:

`http://<host web server>:<port number>/c4ws`

## Web Services Status page

This section describes the **Web Services Status** page of the Connector for Web Services.

To reach the status page, on the connector's index page (starting page), click **status page**. On Windows systems, you can also open the home page by starting the connector's web interface from the Windows Start menu (Start > Programs > Infor > ese > c4ws).

Use the status page of the connector to maintain and monitor the BDE or BOI web services provided by the connector. The page displays the current status of all deployed BDE or BOI web services. Each web service has a link to the WSDL file that defines the service. The page also shows the number of requests that are handled by a service.

The page shows the following web service properties:

Field	Description
Service name	The name of the web service. This name corresponds with the BDE or BOI business interface that is exposed through the web service.
Type	The type of business object, which is either BDE or BOI
ERP Server	The name of the Infor LN or Baan application server that implements the BDE or BOI business interface associated with the web service. This name corresponds with the <b>Identification ERP Server</b> field you specified on the <i>Add/Modify an ERP Server page (p. 24)</i> .
Status	<p>The current status of the web service, which can be:</p> <ul style="list-style-type: none"><li>■ <b>ACTIVE</b> The web service is up and running.</li><li>■ <b>STILL ACTIVE</b> This state can occur after undeploying a web service. You must restart the Web Server to deactivate the web service.</li><li>■ <b>NO WSDL</b> The WSDL file of the web service is not available.  To fix this problem, go to the <i>Deployed Web Services page (p. 28)</i> of the connector and click <b>Refresh</b> to reload the WSDL file from the ERP Server.</li></ul>
Requests	The number of SOAP request messages that have been processed.
WSDL	The hyperlink to the WSDL file.
Address	The URL of the web service.

# ERP Servers Configuration

## Configuring Infor LN or Baan application servers

This section describes how you can configure the connection between the Connector for Web Services and an Infor LN or Baan application server. The configuration of such a connection is referred to as an ERP Server configuration.

### ERP Servers configuration

Before you can use the Connector for Web Services to run BDE or BOI web services, you must configure the connections between the connector and the Infor LN or Baan servers that host the BDE or BOI implementation for which web services are provided. The configuration of such a connection is called an *ERP Server configuration*. To set up a new ERP Server configuration or to modify an existing one, you must fill in several fields on the *Add/Modify an ERP Server page (p. 24)* of the Connector for Web Services (the c4ws servlet).

To add and set a new ERP Server configuration, follow these steps:

#### Step 1: start c4ws Servlet

Make sure the Connector for Web Services is running. For instructions about how you can start this web application, refer to section *Starting and stopping the connector (p. 40)*.

When the connector is started, you can view its index page by entering the following URL address in your web browser:

```
http://<web server host>:<port nr>/c4ws/
```

The access of the web pages of the connector is restricted with Tomcat authentication, which means you must log on as a user with Tomcat Manager privileges.

#### Step 2: Navigate to ERP Servers page

To navigate from the connector's index page to the *ERP Servers page (p. 23)*, click the **ERP Servers** link.

The *ERP Servers page (p. 23)* lists all Infor LN or Baan Server connections for which a configuration has been set. To view or adjust a configuration, click the **Modify ERP Server** icon (a pencil). This opens the **Modify an ERP Server** page, which shows all configuration fields. To add a new configuration, click **Add ERP Server**.

In these instructions, you will learn how to add a new ERP Server configuration. You modify an existing configuration in a similar way.

### Step 3: Add new ERP Server configuration

On the **ERP Servers** page, click **Add ERP Server**. This brings you to the **Add an ERP Server** page of the connector. The page displays an ERP Server form, in which all configuration fields are still empty (unset).

### Step 4: Set ERP Server configuration properties

On the **Add an ERP Server** page, fill in the required configuration fields. For a detailed description of the Infor LN or Baan Server configuration properties, refer to the *Add/Modify an ERP Server page (p. 24)*.

### Step 5: Save the ERP Server configuration


To save the ERP Server configuration, on the **Add an ERP Server** page, click **Add**. If all required configuration fields are properly set, the connector will return to the *ERP Servers page (p. 23)*, which will display the ERP Server connection you just configured.

## ERP Servers page

This section describes the **ERP Servers** page of the Connector for Web Services.

You reach the **ERP Servers** page by clicking **Configure (ERP Server)** on the **Import Web Services** page.

The **ERP Servers** page lists all Infor LN or Baan application servers (ERP Servers) that are properly configured and can be used by the Connector for Web Services. On this page, you can additionally carry out the following commands:

Command		Description
<b>Add ERP Server</b>		Navigates to the <b>Add an ERP Server</b> page. Use this page to add a new ERP Server configuration. For more information, refer to the <i>Add/Modify an ERP Server page (p. 24)</i> .
modify icon		Navigates to the <b>Modify an ERP Server</b> page, which contains the configuration properties for the selected ERP Server. Use this page to modify an existing ERP Server configuration (the one you selected here). For more informa-

tion, refer to the *Add/Modify an ERP Server page (p. 24)* .

remove icon



Removes the selected ERP Server configuration.

You cannot remove a server configuration, if this server is used by a deployed web service. You must first undeploy the web service, before you can remove the associated server.

Back

Returns to the previous page.

For an explanation of the displayed ERP Server properties, refer to the *Add/Modify an ERP Server page (p. 24)* .

## Add/Modify an ERP Server page

This section describes the **Add/Modify an ERP Server** page of the Connector for Web Services.

The Connector for Web Services provides web services that expose BDE business interfaces implemented on LN application servers. On the *Add/Modify an ERP Server page (p. 24)* , you can configure the connections between the connector and one of these LN application servers (ERP Servers). You use the **Add an ERP Server** page to add a new ERP Server configuration. You use the **Modify an ERP Server** page to edit an existing one.

To configure a connection with an ERP Server, fill in these fields.

Field	Description
ERP Server *	<p>The unique name of the LN application server that holds the BDEs or BOIs for which web services are provided.</p> <p>In the WSDL files that define these services, this name corresponds with the last part of the URL specified as the location attribute of the &lt;soap:address&gt; element (see <i>To deploy a web service (p. 34)</i> ).</p>
Host name *	The TCP/IP name of the computer that hosts the LN application server.



---

**Activation type**

The activation type used to start a bshell on the LN application server. You can choose one of the following types:

- **rexec**

Activation type `rexec` starts a bshell using the rexec protocol, which sends the username and password over the network without encryption.

The `rexec` type is the default activation type.

- **baanlogin**

Activation type `baanlogin` starts a bshell using the BaanLogin protocol, which is more secure than rexec protocol because the username and password are first encrypted before sent over the network.

- **none**

If you select value `none`, remote activation is disabled, which means that you must manually start a bshell and an `ottstpcadaemon` on the LN application server.

You will only use this option for back end debugging.

---

**Port**

The number of the TCP/IP port to the ERP application server.

- If **Activation type** is `rexec`, the port number is read-only and fixed to 512.
- If **Activation type** is `baanlogin`, the default port number is 7150. You can, however, select any other valid port number.
- If **Activation type** is `none`, this port number must correspond with the port number used by the `ottstpcadaemon`.

---

**User \***

The user name used for authentication on the ERP application server

---

**Password \***

The password used for authentication on the ERP application server.

---

BSE*	The path of the Baan Environment (BSE) of the bshell.
Bshell*	The logical name of the bshell (as specified in configuration file \${BSE}/lib/ipc_info).
Environment	<p>Optional field containing the environment settings for the bshell process.</p> <p>For example:</p> <ul style="list-style-type: none"><li>■ To set the environment variable TRACEJCA, entering the following value: <code>-set TRACEJCA=1</code>.</li><li>■ To activate extra tracing options for the bshell, enter the following value <code>-dbgmulact -keeplog -logfile log.out -dbgjvmi -tracelevel 1</code>.</li></ul> <p>To set here the environment variable TRACEJCA to 1 has the same effect as checking the <b>Trace</b> option.</p>
Company	The Baan company number used by the LN application. If this field remains empty or has value -1, the default user company applies.
Max connections	<p>The maximum number of physical connections for this connection point.</p> <p>If the value is 0, the number of physical connections is not limited and connection pooling is disabled. If the <b>Activation type</b> is <code>none</code>, this field is read-only and fixed to value 0.</p>
Max references	<p>The maximum number of references to one physical connection.</p> <p>This restriction on the number of references only applies when the maximum number of physical connections (as specified by <b>Max connections</b>) is not yet reached. When the maximum number of physical connections is reached, the connector selects the connection with the lowest number of references.</p> <p>If the <b>Activation type</b> is <code>none</code>, this field is read-only and fixed to value 1.</p>

Keep alive time (seconds)	<p>The number of seconds the connection with the ERP application server remains in the idle pool after it is closed.</p> <p>If you enter here a value larger than 0, you enable connection pooling and set the timer for the idle pool. The default value is 0, which means that no connection pooling is used.</p>
Trace	<p>If this check box is selected, a trace file is generated on the server in which the activities of the ottstpjcadaemon and ottstpbdeamon processes are traced. This file is called <code>trace.jca&lt;bshell.pid&gt;</code> and is located in the <code>\${BSE}/log</code> directory.</p>
Allow anonymous requests	<p>If this check box is selected, "anonymous" SOAP requests are processed. Anonymous requests are requests in which no username/password has been specified in the SOAP header. In this case the username/password as specified above is used for authentication.</p> <p>If this check box is cleared, all SOAP requests must contain the username and password in the SOAP header. Anonymous requests are blocked.</p> <p>If the Activation type is "none", this field is read-only and its value is not used.</p>

All fields marked with an asterisk (\*) are mandatory, which means that you must specify a valid value. Although **Activation type**, **Connection port**, **Keep alive time**, **Max connections** and **Max references** are also required, you do not need to fill in these fields, because they use default values.

You use the **Max connections**, **Max references** and **Keep alive time** fields to enable connection pooling, which means that multiple SOAP requests can be processed through one shared physical connection. For more information, refer to the connection pooling page of the connector's online Help.

To save a new configuration, click **Add**. To save a modified existing configuration, click **Modify**. Any configuration changes will not take effect until one of these buttons is pressed. After successfully saving a configuration, the connector returns to the **ERP Servers** page.

Click **Cancel** to return to the *ERP Servers page (p. 23)* without saving the added or modified configuration.

# Web Services Deployment

This section describes the Connector for Web Services pages you use to deploy BDE and BOI web services to the web server, which are the Deployed Web Services page, Deploy BDE Web Services page and the Deploy BOI Web Services page. The former page gives an overview of all current BDE and BOI web services, the latter provides functionality to import a WSDL file that defines a web service for a BDE or BOI object and to deploy this web service to the web server.

For detailed instructions, refer to section *To deploy a web service (p. 34)*.



## Deployed Web Services page

This section describes the **Deployed Web Services** page of the Connector for Web Services.

To reach the **Deployed Web Services** page, on the connector's index page (starting page), click **configuration page**. In your web browser, you can directly navigate to the index page by entering the following URL: `http://<web server host>:<port nr>/c4ws`. On Windows systems, you can also open the home page by starting the connector's web interface from the Windows Start menu (Start > Programs > Infor > ese > c4ws).

The **Deployed Web Services** page is the main configuration page of the connector and lists all BDE web services that are currently deployed to the web server. For each service, the page displays the name of the web service and the ERP Server on which the related BDE interface has been implemented.

The page also provides functionality to deploy, refresh and undeploy web services. To carry out these tasks, use the following commands:

Command		Description
Undeploy		Undeploys all selected web services. The web services will be removed from the web server and from the <b>Web Services Status</b> page. To carry out the command, click <b>Undeploy</b> or the remove icon.
refresh		Reloads the WSDL files for the selected web services. To carry out the command, click the refresh icon.  Reloading or refreshing web services means that the process of importing and deploying the web service is carried out again. Use

this option if the web service status is 'NO WSDL' (see *Web Services Status page (p. 20)* ).

Deploy BDEs	Navigates to the <b>Deploy BDE Web Services</b> page. For more information on how you import and deploy web services, refer to sections <i>Deploy BDE Web Services page (p. 29)</i> and <i>To deploy a web service (p. 34)</i> .
Deploy BOIs	Navigates to the <b>Deploy BOI Web Services</b> page. For more information on how you import and deploy web services, refer to sections <i>Deploy BOI Web Services page (p. 30)</i> and <i>To deploy a web service (p. 34)</i> .
Back	Navigates to the connector's index page.


## Deploy BDE Web Services page

This section describes the Deploy BDE Web Services page of the Connector for Web Services.

You reach the **Deploy BDE Web Services** page by clicking **Deploy BDEs** on the **Deployed Web Services** page.

On the **Deploy BDE Web Services** page, you can import WSDL files from ERP Servers and deploy the corresponding web services to the Connector for Web Services. These WSDL files define web services for BDE business interfaces that are implemented on the selected ERP Servers.

The general procedure is as follows:

- 1. Select ERP Server**  
 Select an LN application server to list all web services of this server. To select a server, use the drop-down list.  
 If the LN server you want to use is not available, you can click **Configure** to configure a connection to this server. For more information, refer to the *ERP Servers page (p. 23)* .
- 2. Deploy Web Services**  
 Select one or more web services from the list and click **Deploy** or .

The details of the import-and-deploy procedure are explained in section *To deploy a web service (p. 34)* .


## Deploy BOI Web Services page

This section describes the Import Web Services page of the Connector for Web Services.

You reach the **Deploy BOI Web Services** page by clicking **Deploy BOIs** on the **Deployed Web Services** page.

On the **Deploy BOI Web Services** page, you can select the available BOI business objects that you want to deploy as webservice. The page will show only those BOIs that are visible in the CLASSPATH of the web server. To make BOIs available you must copy the jar file (that contains the BOI domain classes) to the c4ws/web/WEB-INF/lib folder. This folder can be found under the installation folder created by the Enterprise Server Installer. WSDL files will be generated based on the metadata of the BOI business object during deployment. These WSDL files define web services for BOI business interfaces that are implemented on the selected ERP Servers.

The general procedure is:

- 1. Select ERP Server**  
Select an Infor LN or Baan application server to list all web services of this server. To select a server, use the drop-down list.  
If the Infor LN or Baan server you want to use is not available, you can click **Configure** to configure a connection to this server. For more information, refer to the *ERP Servers page (p. 23)* .
- 2. Deploy Web Services**  
Select one or more web services from the list and click **Deploy** or .

The details of the import-and-deploy procedure are explained in section *To deploy a web service (p. 34)* .

## Log configuration

This section discusses how you can change the log configuration of a connector.

### Logging

To debug programming code and to log errors, the runtime components use the standard logging interface for Java applications (log4j). The interface is designed so that logging does not heavily affect the performance of the application and that logging behavior can be controlled by changing the log configuration, without altering the application binary.

On the Connector Log Configuration page, you can specify the log settings of a connector. You can change the default name and path of the log file and you can indicate on which level the connector must

be logged. To reach the Log Configuration page, click **log configuration page** or **Configure logging** on the connector's starting page.

### Note

The starting page of the connector is the opening page of the connector web interface. For the Connector for Web Services, the starting page is the *c4ws* index page (`http://<web server host>:<port nr>/c4ws`).

## Log Configuration properties

To view and change the log configuration of a connector, the Connector Log Configuration page shows the following fields:

Field	Description
Application log level	<p>The used log level for the connectors. You can select one of the following levels.</p> <ul style="list-style-type: none"> <li> <b>Error</b> <p>The <i>Error</i> level designates error events that can still allow the connector to continue running. The error message indicates that parts of the application are not working or working incorrectly. The <i>Error</i> level also includes the logging of fatal errors.</p> </li> <li> <b>Info</b> <p>The <i>Info</i> level designates informational messages that highlight the progress of the application at coarse-grained level. Such informational messages include, for example, the initialization of a configuration or the end of a long import process. The <i>Info</i> level includes <i>Error</i> logging.</p> </li> <li> <b>Debug</b> <p>The <i>Debug</i> level designates fine-grained informational events that are most useful to debug the connector. Use this level only for developing purposes to follow the program execution. The <i>Debug</i> level includes the <i>Info</i> logging.</p> </li> </ul>
Log file location (Web Server)	<p>The name and location of the log file of a particular connector. The log file must be located on the computer hosting the web server that runs the connector.</p>

Maximum log file size (Mb)	Enter the maximum size of the log file in Mb. Leave the field empty or use the value '0' for an unlimited log file size. If the log file reaches the specified size a backup will be made and the logfile will be cleared automatically.
Maximum number of backup files	This field is used when the maximum log file size is specified. Once the maximum size of the logfile is reached, a backup of the log file is created. The filename of these backup files is the log file name extended with a number. The log files are stored in the same folder as specified by the log file location.

---

Connector for Web Services uses as default value for the **Log file location** the following file path and file name:

`<installation directory web application>/WEB-INF/c4ws.log.`

By default, the log level of the connector is set to '*Error*'. To trace down errors, change the logging to a lower level.



This chapter describes in general terms how you use the Connector for Web Services to run web services for BDE and BOI business interfaces that have been implemented on LN back ends.

The following subjects are discussed:

- Setting up web services for BDEs.
- Setting up web services for BOIs.
- Overruling ERP Servers Configuration
- Developing a Web Client
- Starting and stopping the Connector web application.

## Setting up web services for BDEs

This section describes how to set up web services for the BDE's of LN using the Connector for Web Services. The section first summarizes the concept of BDE access through the connector and then explains how you can develop a web service for a BDE.

### Overview

The basic procedure to set up a web service for a BDE business interface is as follows:

1. Make sure the appropriate BDE business interface implementations are available on the LN applications server you want to use for the web-based integration. Check whether the LN server and BDE implementations meet the requirements of the connector.
2. Configure the connections between the Connector for Web Services and the LN back ends.
3. Create the web service for a BDE interface by importing the corresponding WSDL file and deploying this file to the Tomcat web server that runs the Connector for Web Services (a web application called *c4ws*).

After you carried out previous steps, you have a running web service that is ready to be consumed by a web client application. The development of such a web client application simply consists of standard web services development procedures and will therefore not be explained in detail.

For the configuration of the connection with the LN back ends, refer to section *Configuring Infor LN or Baan application servers* (p. 22) .

If no suitable BDE interfaces are available, you can implement a BDE interface on an LN server using the Implementation Generator of Infor LN Business Studio. For more information about this subject, refer to the Infor LN Business Studio documentation.

## BDE access through web services

A Business Data Entity (BDE) business interface of Infor LN consists of collection of related business data and the methods to access this data. On the LN server, the exact implementation of the BDE is hidden and may consist of any combination of DLLs and database tables. The public interface, however, shows a standard behavior and appearance.

The public layer of a BDE interface exposes the business data to external applications and is message-based. This implies that an external application can access the BDE by sending a request message to this BDE, and that the BDE on the back end can reply to this request by returning a response and a result message to the client. These messages are XML documents that conform to a BDE-specific vocabulary. The data and functions of this public interface are described in a WSDL file. LN provides this WSDL file, in a repository, to external applications.

The Connector for Web Services implements a web application to run web services with which you can access BDE business interfaces. A BDE web service performs the message-based communication with the BDE in the following way. The web service receives a SOAP message from a web client, converts this SOAP message into a BDE-specific XML message, and sends this BDE message to the back end. The LN application on the back end forwards this message to a BDE implementation and thus invokes one of the methods of the BDE interface. The BDE interface replies by returning output for the invoked method in another BDE-specific XML message. This BDE response or result message is sent back to the web service, which converts it into a SOAP message and then forwards the SOAP message to the web client. The format of the SOAP messages that are sent to and from are completely defined by the WSDL file of the BDE web service.

### Note

For more conceptual information about BDE interfaces and their XML-based method arguments, refer to the *Integration Guide for BDEs*. Notice that the solution for creating web services for BDEs discussed in this guide refers to the Connector for Web Services of Infor Integration 6 and does not apply to the Connector for Web Services.

## To deploy a web service

To implement a BDE web service, you must import the WSDL file that defines this service and then deploy the WSDL file to the web server. You carry out both tasks in the web application of the Connector for Web Services (on various web pages of the c4ws servlet).

To deploy a BDE web service, follow these steps:

## Step 1: Start c4ws web application

Make sure the Connector for Web Services is running. For detailed information on how to start and stop the connector web application, refer to section *Starting and stopping the connector* (p. 40) .

When the connector is started, you can view its index page by entering the following URL address in your web browser:

`http://<web server host>:<port nr>/c4ws/`

The access of the web pages of the connector is restricted with Tomcat authentication, which means you must log on as a user with Tomcat Manager privileges.

## Step 2: Check LN connections

On the *ERP Servers page* (p. 23) of the connector, check whether the connections to the required LN Servers are appropriately configured. To navigate from the connector's index page to the *ERP Servers page* (p. 23) , click the **configuration page** link, the **Import and Deploy** button, and the **Configure** button.

For more information about the configuration of LN Server connections, refer to section *Configuring Infor LN or Baan application servers* (p. 22) and the *Add/Modify an ERP Server page* (p. 24) .

### Note

You can also directly navigate to the *ERP Servers page* (p. 23) by entering the page's URL address `http://<web server host>:<port nr>/c4ws/cp.jsp`.

## Step 3: Deploy a BDE Web Service

On the **Deploy BDE Web Services** page of the connector, select a BDE web service from an LN server, import the corresponding WSDL file in the connector application and deploy the WSDL file to the Tomcat web server.

Follow this procedure:

1. Navigate from the connector's index page to the **Deploy BDE Web Services page** by clicking **Deployed Web Services page** and **Deploy BDEs**.
2. On the **Deploy BDE Web Services page**, indicate which LN back end you want to access by selecting a server from the drop-down list of the ERP Server field. This list only contains the ERP Servers for which the connection with the connector has been configured. After you selected an LN server, the connector lists all the BDE web services that are available on this server. The web services are named after the associated BDE business objects, for example, Customer or Purchase Order.
3. Select one of the available BDE web services.
4. Click **Deploy**. The **Deployment Result** page shows whether the deployment of the web service was successful.

After successful deployment, a web client can use the web service to call methods on the involved BDE business object by sending SOAP messages that conform to the imported WSDL file. To retrieve this

WSDL file, click WSDL on the connector's *Web Services Status page (p. 20)* (`http://<web server host>:<port nr>/c4ws/services`).

The `<soap:address>` in the `<ws: port>` element of the WSDL file denotes the end point of the SOAP message, which is the URL address where the message is delivered. This address contains the location of the web server, the name of the web application and the location of the back end. In our case, the location of the back end is indicated by the **Identifier** of the LN server hosting the BDE business object that is exposed by the web service.

The following example shows the `<soap:address>` in a WSDL file of a web service to expose Areas BDEs:

```
<wsdl:service name="AreasService">
  <wsdl:port binding="bo:AreasSoapBinding" name="AreasSoapPort">
    <soap:address location="http://localhost:8080/c4ws/services/Areas/MyBackEnd123" />
  </wsdl:port>
</wsdl:service>
```

## Setting up web services for BOIs

This section describes how to set up web services for the BOIs of Baan IV or Baan 5.0 using the Connector for Web Services. The section first summarizes the concept of BOI access through the connector and then explains how you can develop a web service for a BOI.

### Overview

Where Baan IV is mentioned you can also read Baan 5.0:

1. **The basic procedure to set up a web service for a BOI business interface is as follows:** Make sure the appropriate BOI business interface implementations are available on the Baan IV applications server you want to use for the web-based integration. Check whether the Baan IV server and BOI implementations meet the requirements of the connector.
2. Configure the connections between the Connector for Web Services and the Baan IV back ends.
3. Create the web service for a BOI interface by importing the corresponding WSDL file and deploying this file to the Tomcat web server that runs the Connector for Web Services (a web application called *c4ws*).

After you carried out previous steps, you have a running web service that is ready to be consumed by a web client application. The development of such a web client application simply consists of standard web services development procedures and will therefore not be explained in detail.

For the configuration of the connection with the Baan IV back ends, refer to section *Configuring Infor LN or Baan application servers (p. 22)*.

If no suitable BOI interfaces are available, you can implement a BOI interface on an Baan IV server using the Implementation Generator of Infor LN Business Studio. For more information about this subject, refer to the Infor LN Business Studio documentation.

## BOI access through web services

A Business Object Interface (BOI) business interface of Baan IV consists of collection of related business data and the methods to access this data. On the Baan IV server, the exact implementation of the BOI is hidden and may consist of any combination of DLLs and database tables. The public interface, however, shows a standard behavior and appearance.

This public interface for BOIs is implemented in BOI proxy classes. Usually these classes are packaged together in domain packs. To be able to access these BOI proxies from the Connector for Web Services, the domain packs (these are jar files) must be available in the classpath of the web application.

The data and functions of this public interface are described in a WSDL file.

The Connector for Web Services implements a web application to run web services with which you can access BOI business interfaces. A BOI web service performs the message-based communication with the BOI in the following way. The web service receives a SOAP message from a web client, converts this SOAP message into a BOI-specific XML message, and sends this BOI message to the back end. The Baan IV application on the back end forwards this message to a BOI implementation and thus invokes one of the methods of the BOI interface. The BOI interface replies by returning output for the invoked method in another BOI-specific XML message. This BOI response or result message is sent back to the web service, which converts it into a SOAP message and then forwards the SOAP message to the web client. The format of the SOAP messages that are sent to and from are completely defined by the WSDL file of the BOI web service.

### Note

For more conceptual information about BOI interfaces, refer to the *Integration Guide for BOIs*. Notice that the solution for creating web services for BOIs discussed in this guide refers to the Connector for Web Services of Infor Integration 6 and does not apply to the Connector for Web Services.

## To deploy a web service

To implement a BOI web service, you must generate the WSDL file that defines this service and then deploy the WSDL file to the web server. You carry out both tasks in the web application of the Connector for Web Services (on various web pages of the c4ws servlet).

To deploy a BOI web service, follow these steps:

### Step 1: Start c4ws web application

Make sure the Connector for Web Services is running. For detailed information on how to start and stop the connector web application, refer to section *Starting and stopping the connector* (p. 40) .

When the connector is started, you can view its index page by entering the following URL address in your web browser:

`http://<web server host>:<port nr>/c4ws/`

The access of the web pages of the connector is restricted with Tomcat authentication, which means you must log on as a user with Tomcat Manager privileges.

## Step 2: Check Baan IV connections

On the *ERP Servers page (p. 23)* of the connector, check whether the connections to the required Baan IV Servers are appropriately configured. To navigate from the connector's index page to the *ERP Servers page (p. 23)*, click the **configuration page** link, the **Import and Deploy** button, and the **Configure** button.

For more information about the configuration of Baan IV Server connections, refer to section *Configuring Infor LN or Baan application servers (p. 22)* and the *Add/Modify an ERP Server page (p. 24)*.

### Note

You can also directly navigate to the *ERP Servers page (p. 23)* by entering the page's URL address `http://<web server host>:<port nr>/c4ws/cp.jsp`.

## Step 3: Deploy a BOI Web Service

On the **Deploy BOI Web Services** page of the connector, select a BOI web service from an Baan IV server, generate the corresponding WSDL file in the connector application and deploy the WSDL file to the Tomcat web server.

Follow this procedure:

1. Navigate from the connector's index page to the **Deploy BOI Web Services** page by clicking **Deployed Web Services page** and **Deploy BOIs**.
2. On the **Deploy BOI Web Services** page, indicate which Baan IV back end you want to access by selecting a server from the drop-down list of the ERP Server field. This list only contains the ERP Servers for which the connection with the connector has been configured. The connector lists all the BOI proxies that are available on this web server. The web services are named after the associated BOI business objects, for example, Customer or Purchase Order.
3. Select one of the available BOI web services.
4. Click **Deploy**. The **Deployment Result** page shows whether the deployment of the web service was successful.

After successful deployment, a web client can use the web service to call methods on the involved BOI business object by sending SOAP messages that conform to the imported WSDL file. To retrieve this WSDL file, click WSDL on the connector's *Web Services Status page (p. 20)* (`http://<web server host>:<port nr>/c4ws/services`).

The `<soap:address>` in the `<ws: port>` element of the WSDL file denotes the end point of the SOAP message, which is the URL address where the message is delivered. This address contains the location of the web server, the name of the web application and the location of the back end. In our case, the

location of the back end is indicated by the **Identifier** of the Baan IV server hosting the BOI business object that is exposed by the web service.

The following example shows the <soap:address> in a WSDL file of a web service to expose Areas BOIs:

```
<wsdl:service name="AreasService">
  <wsdl:port binding="bo:AreasSoapBinding" name="AreasSoapPort">
    <soap:address location="http://localhost:8080/c4ws/services/Areas/MyBackEnd123" />
  </wsdl:port>
</wsdl:service>
```

## Overruling ERP servers configuration

Activation related ERP Servers configuration properties can be overruled at runtime. Provide the settings for the Activation properties in the SOAP Message (Header).

These Activation properties can be overruled:

- User name
- Password
- Company

If the ERP Server has been configured to allow anonymous requests, using the overrule mechanism and changing the properties is optional. The User name and Password activation properties must be used together.

### Note

If the ERP Server does not allow anonymous requests, overruling the user name and password is mandatory.

When the overrule mechanism is active, the changed properties overrule the configured ERP Server configuration settings. This enables the creation of more flexible connection towards the ERP Server.

The SOAP Header definition of these properties is specified in the WSDL. Therefore clients using generated proxy code based on the WSDL will have some API extensions to set the properties.

### Note 1:

Changing user or company will have impact on the ERP Server for either installed software (package VRC) or data (companies).

### Note 2:

Specifying one of the properties creates a new 'virtual' ERP Server configuration. Passing the same property values again will point to the same 'virtual' ERP Server configuration and enable connection pooling

# Developing a Web Client

To use the web service, you must develop web client applications that consume the web service. The development of the web client applications includes two steps:

- **Generate WSDL Proxies**  
Generate WSDL Proxies from the WSDL file. On the client, these proxies form local API that mirror the data and functionality of the web service. The client uses this API to compose a SOAP message containing a request to invoke a BDE or BOI method.  
There are several tools available to generate WSDL Proxies. For Java applications, for example, you can use the proxy generator of Apache AXIS (see <http://ws.apache.org/>).
- **Write client applications**  
To write a web client application that calls these WSDL Proxies.

## Note

To test a web service without the necessity of implementing WSDL proxies and writing a dedicated web client application, you can use various Web Services Testing and Diagnostics toolkits. To test web services, you can also use more generic development environments, such as Microsoft Visual Studio or Eclipse.

The development of Web clients is a standard web services practice. Describing this practice is not in the scope of this document.

# Starting and stopping the connector

This section describes how you start and stop the Connector for Web Services.

The Connector for Web Services runs as a web application in the managed environment of a Tomcat web server. To start or stop the connector, you can start or stop the Tomcat server or, if Tomcat is running, you can start or stop the connector's web application.

To start or stop Tomcat (Windows)

## Step 1: Open the Services window

On Windows systems, you can open the **Services** window in various ways. For Windows 2003, for example, you can apply the following method:

1. Click **Start**
2. Point to **Settings**
3. Click **Control Panel**
4. Double-click **Administrative Tools**



## 5. Double-click **Services**

### Step 2: Start or Stop the Tomcat service

If you used the Enterprise Server Installer, the Tomcat service that runs the connector servlets is called **Infor Enterprise Server**. In the **Services** window, use the shortcut menu to start or stop this service. To start an inactive Tomcat service, right-click the service and then click **Start**. To stop an active Tomcat service, right-click the involved service and then click **Stop**.

You can also start and stop the Tomcat service through commands in the **Start -> All Programs -> Infor -> ese -> apache-tomcat-7.0.30** menu, or through batch files in the <installation-directory>\apache-tomcat-7.0.30\bin directory.

To start or stop Tomcat (UNIX)

Before you use the following commands to start and stop the connector, add the webapp folder that contains the commands to the PATH environment variable.

### Step 1: Start Tomcat server

On UNIX systems, you start the Tomcat server with the following command:

```
$ sh startup.sh
```

### Step 2: Stop Tomcat server

On UNIX systems, you stop the Tomcat server with the following command:

```
$ sh shutdown.sh
```

You can also stop the server, though less gracefully, by killing the corresponding job.

To start or stop the connector web application

### Step 1: Open the Tomcat home page

Make sure the Tomcat server is running.

Open the Tomcat home page in your browser. The URL address that points to this home page is `http://<host>:<port>`, where <host> refers to the name of the computer that runs the Tomcat server and <port> to the port number used by the server.

A typical example of the Tomcat home page URL is:

```
http://localhost:8080
```

Note: The Enterprise Server Installer installs the Connector for Web Services on port 8312.

## Step 2: Open the Tomcat Web Application Manager

On the Tomcat home page, in the left column, click Tomcat Manager and log on to the manager as a user with `manager-gui` privileges.

## Step 3: Start or Stop the connector web application

In the Tomcat Manager console, the Connector for Web Services is represented as a web application (servlet) called `c4ws`.

To start a stopped web application, click the servlet's **Start** command.

To stop a running web application, click the servlet's **Start** command.

This chapter deals with tracing down errors and solving known problems.

## Tracing down errors

To find the source of an error, you must read the connector's log file.

You can simply view the current log file using the web interface of the connector: in the Log configuration section on the connector's starting page, click **log file** or **Show log file**.

To obtain more information, you can set the logging to a more detailed level. In most cases, log level *Info* will provide sufficient information. Infor advises not to use log level *Debug*, because this level can provide too much information and will seriously affect the performance of the connector.

To change the log level, navigate to the Log configuration page of the connector and select another value for the **Application log level** field. For more information on the log level, refer section *Log configuration* (p. 30) .

### Note

The connector log file is not the only log file you can consult to trace down the source of an error. On the ERP back end, and if applicable in the ESB message bus, errors and warnings will be logged as well.

## Miscellaneous

If the installation fails, or if the Connector for Web Services does not start, check whether the prerequisites are installed. For details, refer to Requirements and to the *Infor Enterprise Server Installation Guide (U9484 US)*.



# Appendix A

## Web Service Concepts

# A

This appendix gives an short description of web services and the related web standards WSDL and SOAP, and how the Connector for Web Services adheres to these standards.

- **Web Service**

A web service is a server application that runs somewhere on the Internet or an intranet and that directly interacts with a client application (a web client) connected to this network. The service is based on various Web standards, such as XML, SOAP, WSDL and HTTP.

The Connector for Web Services provides web services to access BDE and BOI business interfaces that are implemented on Infor LN or Baan application servers. Through these services, a web client can invoke the public methods of a specific BDE business interface. Each web service corresponds with a single BDE implementation. The BDE business interface and the associated web service have the same name. You reach the web services through the following URL ("SOAP address"):

```
http://<web server host>/c4ws/services/<BDE name>/< Infor LN or Baan  
Server name>,
```

The Infor LN or Baan Server name used in the previous URL represents the identifier of the Infor LN or Baan server you specified in the configuration of the connection between the connector and this server (the **ERP Server** field on the connector's Add/Modify an ERP Server page).

- **XML**

Extensible Markup Language (XML) is a markup language to describe data. All documents web services use for exchanging and verifying data are XML based.

- **SOAP**

SOAP is an XML-based protocol used to exchange information across the internet.

You use SOAP for remote procedure calls (RPC) on the web service that is implemented by the Connector for Web Services.

- **WSDL**

Web Service Definition Language is an XML vocabulary to define web services. A WSDL file describes the complex data types of the web service and the methods to manipulate their values. You use WSDL to implement a web service for adding, retrieving, modifying and removing specific business data through method calls.

In the context of the Connector for Web Services, the WSDL file describes the methods and the method arguments to invoke BDE or BOI methods. This information is used to build request SOAP messages. The required information includes:

- The business object (BDE or BOI)
- The method with the required input arguments
- The server itself

■ **HTTP**

The HyperText Transfer Protocol (HTTP) is the protocol used by the World Wide Web that defines how messages are formatted and transmitted, and what actions must be taken in response to various commands. If, for example, you submit an URL address in your web browser, HTTP is used to retrieve the corresponding website.

Knowledge about WSDL and SOAP standards is required to be able to understand and use the Connector for Web Services.

---

## Appendix B

### Definitions, abbreviations and acronyms

# B

#### BDE

See: *Business Data Entity* (p. 47)

#### BOI

See: *Business Object Interface* (p. 47)

#### Business Data Entity

A Business Data Entity (BDE) is a standard Business Interface (BI) of Infor LN that conforms to the Infor Uniform Connectivity Architecture.

Acronym: BDE

See: BI, Uniform Connectivity

#### Business Object Interface

A Business Object Interface (BOI) is the Business Interface (BI) of Baan IV or Baan 5.0 that conforms to the Infor Uniform Connectivity Architecture.

Acronym: BOI

#### Simple Object Access Protocol

Simple Object Access Protocol is a lightweight XML-based messaging protocol used to encode the information in Web service request and response messages before sending them over a network.

Acronym: SOAP

#### SOAP

See: *Simple Object Access Protocol* (p. 47)

## Web Service Description Language

An XML document to define and locate Web services. Web Service Description Language can include all type declarations of XML Schema.

Acronym: WSDL

## Web Service Description Language

An XML document to define and locate Web services. Web Service Description Language can include all type declarations of XML Schema.

Acronym: WSDL

## WSDL

See: *Web Service Description Language (p. 48)*

## WSDL

See: *Web Service Description Language (p. 48)*



---

# Index

- Add an ERP Server**, 24
  - Add ERP Server**, 23
  - Baan Settings and Environment**, 24
  - Back End**, 23
  - BDE**, 33, 36, 47
  - BDE Web Services**
    - running web services, 33
    - troubleshooting, 43
  - BOI**, 47
  - Business Data Entity**, 47
  - Business Data Entity (BDE)**, 33, 36
  - Business Object Interface**, 47
  - Business Studio**, 33, 36
  - Configuration of ERP Servers**, 23
  - Configuration of Infor LN or Baan Server**, 22
  - Configuration of Logging**, 30
  - Connection with Back End**, 23
  - Connector for Web Services**, 7
    - concepts, 7
    - definition, 7, 7
    - install, 13
    - installation using installer, 14
  - Deployed Web Services**, 28
  - Deploy Web Services**, 28, 29, 30
  - Development**
    - of web services, 33, 36
  - ERP Server Configuration**, 23, 24
  - ERP Server Identification**, 24
  - Error Logging**, 43
  - Handled Requests**, 20
  - Import & Deploy Web Services**, 28
  - Import Web Services**, 29, 30
  - Infor LN or Baan Servers**
    - configuration, 22
  - Installation**, 13
    - installer, 14
    - manual installation, 14
  - Installer**, 13
  - Log Configuration**, 30
  - Log File**, 30
  - Log Levels**, 30
  - Modify an ERP Server**, 24
  - Modify ERP Server**, 23
  - override**, 39
  - Remove ERP Server**, 23
  - Request Handled by Service**, 20
  - Run BDE Web Services**, 33
  - Setup of Web service**, 33, 36
  - Simple Object Access Protocol**, 47
  - SOAP**, 47
  - Start Connector**, 40
  - Status page**, 20
  - Tracing Errors**, 43
  - Troubleshooting**, 43
    - web services, 43
  - Undeploy Web Services**, 28
  - Uninstall Connector for Web Services**, 17
  - Update Connector for Web Services**, 17
  - webclient**, 40
  - Web Service Description Language**, 48, 48
  - Web service**
    - development, 33, 36
    - setup, 33, 36
  - Web Services**
    - deploy, 28
    - reload, 28
    - undeploy, 28
  - Web Services for BDEs**
    - running web services, 33
    - troubleshooting, 43
  - WSDL**, 33, 36, 48, 48
-

