Infor Enterprise Server Connector for Web Services Administration and User Guide

Release 10.6.x

Contents

About this Guide

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 server or BOI's on a Baan IV or Baan 5 server. 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

Note: Connector for Web Services applies to BDE interfaces version 8.4 or later

The connector provides web services if the server conforms to these pre-requisites:

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

Note: Connector for Web Services applies to all versions of BOI interfaces

The connector provides web services if the server conforms to these pre-requisites:

- Baan IV
- Baan 5.0

Organization

This table shows the chapters and appendices of the guide:

Section	Description
Introduction on page 7	Provides general product information on the Connector for Web Services.
Requirements on page 11	Describes the pre-requisites of the Connector for Web Services.
Installation on page 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.

Section	Description
Configuration and deployment on page 17	Describes how to configure the Connector for Web Services to run web services for particular BDE or BOI business objects.
Employment on page 30	Describes how to run a web service for a BDE business object on an LN server and a webservice for a BOI on a Baan IV and Baan 5 server.
Troubleshooting on page 39	Deals with finding the source of an error and solving known problems.
Web service concepts on page 40	Provides some background information on web service standards, such as *WSDL* and *SOAP*.

Related documents

You can find the documents in the product documentation section of the Infor Xtreme Support portal:

- *Infor Enterprise Server Installation Guide*
- The Infor LN Studio documentation

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

- *Infor Integration 6.2 - Integration Guide for BDE Clients*
- *Infor Integration 6.2 - Integration Guide for BOI Clients*

Points of attention

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. 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 *Infor Integration 6.2 - 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 .

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal.

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

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1: Introduction

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 servers. 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 server. 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 server.

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 server. 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 must 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 LN Studio. After deployment, the service is running and ready to be used.

Web clients that have access to the web service can use the service. Therefore, a web client can send a SOAP request message to the web service and, 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. These proxies 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 shows which web services are currently active and specifies, for each service, the Infor LN or Baan specific server, the BDE or BOI business object, the URL of the service and number of requests handled by the service.

See Web Services Status page on page 18.

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

This diagram shows the various runtime components that build up a web-based integration for the business objects of an Infor LN or Baan application server:
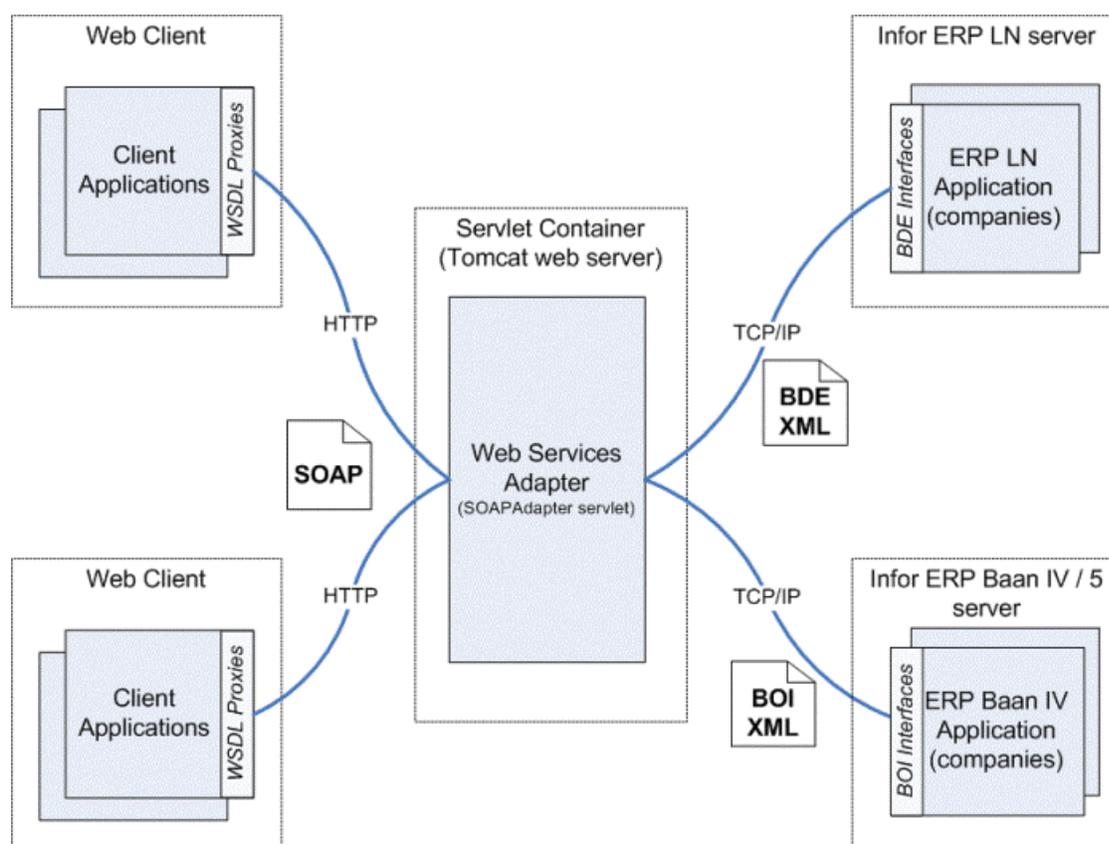
The Connector for Web Services receives and unwraps a SOAP request message from a web client and redirects the contained method call to an Infor LN or Baan application server (an application component or company on this server). The server executes the call and returns a reply to the web

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, see these documents:

- *Infor Integration 6.2 - Integration Guide for BDE Clients*.
- *Infor Integration 6.2 - Integration Guide for BOI Clients* .

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 server 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 servers from which you want to access BDE and BOI business objects. To configure a server connection, you use the **Add/Modify an ERP Server** page of the connector. On this page, you specify the host name of the Infor LN or Baan server, indicate which activation protocol you want to use and specify various other Infor LN or Baan configuration properties, such as the company number, the authentication data to log on to the application, the BSE location, and the Bshell name. Ensure that the system that runs the connector is connected with the desired Infor LN or Baan servers.

Next, you create a web service for a particular BDE or BOI business object by deploying it on the **Deploy BDE Web Services** or **Deploy BOI Web Services** page. This deployment makes 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.

Note: 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 LN 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 of the connector shows the URL address of the web service, which is the SOAP address for the endpoint of the message.
Note: See Web Services Status page on page 18.
Note: The URL of the WSDL file that defines the web service corresponds with the URL address that is displayed in your web browser after you clicked the WSDL link on the **Web Services Status** page of the connector.

This diagram shows the basic procedure of importing WSDL files for business objects to generate WSDL proxies for a web client:
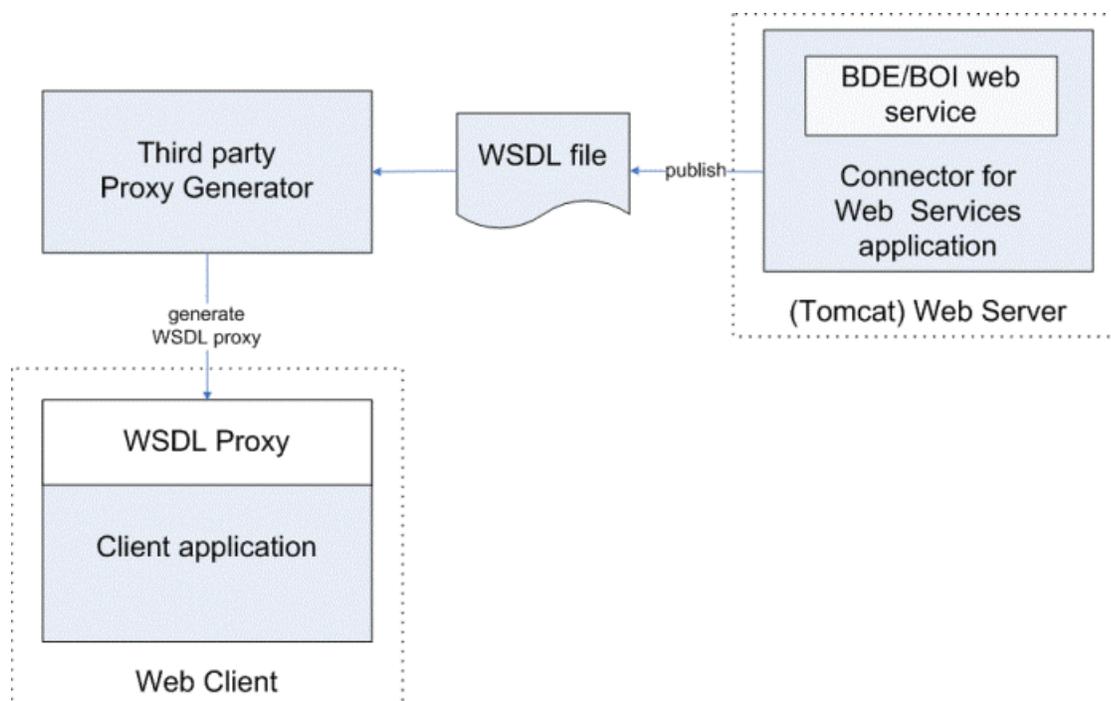
Chapter 2: Requirements

This chapter describes the requirements of the Connector for Web Services. For the platforms supported by the installer, see the *Infor Enterprise Server Installation Guide*.

Overview

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

- Java Runtime Environment 1.8.0 or later (Java version 8).
- A web server that supports the Servlet 2.4 specification, such as Apache Tomcat 8.0.21.
- An LN server supported by the Enterprise Server Installer. See the *Infor Enterprise Server Installation Guide*.
- On this server, 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, see the *Infor Enterprise Server Installation Guide.*

LN server support

The Connector for Web Services supports LN servers that meet these preconditions:

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

Baan IV or Baan 5.0 server support

The Connector for Web Services supports Baan IV and Baan 5.0 servers that meet these preconditions:

- Baan IV portingset 6.1c.08 or higher
- Baan 5 portingset Enterprise Server 8.4 (or later)
- BOI Business Interface Implementations generated with LN 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 SLM license on page 12.

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 must support these 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.

Chapter 3: Installation

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).

To install the connector:

1 Fulfill the requirements.

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

See Requirements on page 11, or see the *Infor Enterprise Server Installation Guide*.

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, see Installing the connector using the setup program on page 14. For detailed information, see 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 these 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, see Manual installation on page 14.

You can also use the Enterprise Server Installer to update or uninstall the software. See Updating or uninstalling the connector on page 16.

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, see 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 installer. 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 these tasks:

- Install a pre-configured instance of a Tomcat web server version 8.0.21. 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 perform. Note that update and uninstall tasks are only available if a previous installation with the installer exists.

To start the installer, open `install.bat` (Windows) or `install.sh` (UNIX).

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 a Web ARchive (WAR) file on the installation disk.

This table shows this WAR file:

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.

Deploying 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.

Complete these steps:

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 Enterprise Server Installer installation disk to the warfiles subdirectory of the Connector for Web Services.

2 Extract WAR files in application folders.

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

For UNIX, use these commands:

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

For Windows, use these commands:

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

3 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:\in for\ese\apache-tomcat-8.0.21\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. We recommend that you 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:\in for\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>
```

4 Restart Tomcat.

Stop and restart Tomcat to ensure the web application is deployed.

Updating or uninstalling the connector

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

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 in on-premises mode

The connector is used to run web services that expose BDE or BOI business objects implemented on Infor LN or Baan application servers.

To configure the Connector for Web Services in on-premises mode, you must use several configuration pages.

The configuration of the Connector for Web Services includes:

- ERP servers configuration

  You must configure the connector to define the connection between the connector and the Infor LN or Baan application servers. These servers, 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, see these sections: .
  - Configuring Infor LN or Baan application servers on page 19
  - ERP Servers page on page 20
  - Add/Modify an ERP Server page on page 21

- Web services deployment

  You must configure the connector to deploy a BDE or BOI web service for a specific BDE or BOI business interface.

  To perform this configuration:
  1 Select a BDE or BOI interface from a connected ERP Server.
  2 Import the WSDL file that defines the web service for this BDE or BOI.
  3 Deploy the WSDL file to the connector.

  For more information, see these sections:
  - Deploying a web service on page 31
  - Deployed Web Services page on page 24
  - Deploy BDE Web Services page on page 25

- Advanced settings

From version 10.5.1 some additional settings are available related to OAuth 1.0a authentication, as used in (multi-tenant) cloud installations or in on-premises installations with ION-API.

For more information, see Advanced Settings on page 26.

- 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, see Log configuration on page 27.

Viewing and monitoring the web services

The connector additionally shows the current status of all deployed BDE web services on the **Web Services Status** page. This page shows 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, see Web Services Status page on page 18.

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 is displayed when you start the web application of the connector and has this 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 shows 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 is automatically refreshed every 15 seconds. To disable this feature, click Disable Auto-Refresh at the top of the page.

These web service properties are displayed on the page:

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 **Identification ERP Server** field you specified on the **Add/Modify an ERP Server** page.

Status

The current status of the web service, which can be:

- **ACTIVE**

  The web service is up and running.

- **STILL ACTIVE**

  This state can occur after undeploying a web service. You must restart the Web Server to deactivate the web service.

- **NO WSDL**

  The WSDL file of the web service is not available.

  To fix this problem, go to the **Deployed Web Services** page of the connector and click **Refresh** to reload the WSDL file from the ERP Server.

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 of the Connector for Web Services (the c4ws servlet).

To add and set a new ERP Server configuration:

1 Start c4ws Servlet.

Ensure the Connector for Web Services is running. For instructions about how you can start this web application, see Starting and stopping the connector on page 37.

When the connector is started, you can view its index page by specifying this 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. Therefore, you must log on as a user with Tomcat Manager privileges.

2   Navigate to ERP Servers page.

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

The **ERP Servers** page 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). The **Modify an ERP Server** page, which shows all configuration fields, is displayed. 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.

3   Add new ERP Server configuration.

On the **ERP Servers** page, click **Add ERP Server**. The **Add an ERP Server** page is displayed. The page shows an ERP Server form, in which all configuration fields are still empty (unset).

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, see Add/Modify an ERP Server page on page 21.

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 returns to the **ERP Servers** page, 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 the ERP Servers page link on the index page. You can also reach this page by clicking **Configure** on the **Deploy BDE Web Services** or **Deploy BOI 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.

This table shows the commands you can perform on this page:

Command	Description
Add ERP Server	Navigates to the **Add an ERP Server** page. Use this page to add a new ERP Server configuration. For more information, see Add/Modify an ERP Server page on page 21.

Command	Description
Modify	Navigates to the **Modify an ERP Server** 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 information, see Add/Modify an ERP Server page on page 21.
Remove	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, see Add/Modify an ERP Server page on page 21.

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, 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, specify this information:

ERP Server *
The unique name of the LN application server that holds the BDEs or BOIs for which web services are provided.

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 `<soap:address >` element.

See Deploying a web service on page 31.

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 these 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.

- **baanlogin_ssl**

  Activation type **baanlogin_ssl** starts a bshell using the BaanLogin protocol using a "secure sockets" connection. This activation type can only be used in environments where ION-API performs the user authentication. Typically, this is the case in a multi-tenant cloud environment, but ION-API can also be installed in an on-premises environment.

  ION-API must be configured in such a way that user information, tenant information, and OAuth 1.0a tokens are passed to the Connector for Web Services through HTTP headers.

- **none**

  If you select value **none**, remote activation is disabled. Therefore, you must manually start a bshell and an ottstpjcadaemon on the LN application server.

  You will only use this option for server 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** or **baanlogin_ssl**, 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 ottstpjcadaemon.

User *

The user name used for authentication on the ERP application server. This account is used when deploying BDEs. It is also used as the account for anonymous requests.

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 the `${BSE}/lib/ipc_info` configuration file.

Environment

Optional field containing the environment settings for the bshell process.

For example:

- To set the environment variable TRACEJCA, specify this value: `-set TRACEJCA=1`.
- To activate extra tracing options for the bshell, specify this value:

  `-dbgmulact -keeplog -logfile log.out -dbgjvmi -tracelevel 1`

To set here the environment variable TRACEJCA to 1 has the same effect as checking the **Trace** option.

Company

The company number used by the LN application. If this field remains empty or has value -1, the default user company applies.

Max connections
The maximum number of physical connections for this connection point.

If the value is 0, the number of physical connections is not limited and connection pooling is disabled. If the **Activation type** is `none`, this field is read-only and fixed to value 0.

Max references
The maximum number of references to one physical connection.

This restriction on the number of references only applies when the maximum number of physical connections (as specified by **Max connections**) is not yet reached. When the maximum number of physical connections is reached, the connector selects the connection with the lowest number of references.

If the **Activation type** is `none`, this field is read-only and fixed to value 1.

Max threads
The maximum number of requests that can be placed in a thread queue while waiting for a physical connection to become available. As soon as a connection becomes available, the oldest request is removed from the queue and assigned to the connection. A value of -1 means that there is no limit to the amount of requests that can be placed in the thread queue. The default value is 0, which means that no queueing is used.

This field is typically used in combination with **Max references** set to 0. This prevents that multiple requests are handled concurrently in the same bshell.

If the **Activation type** is `none`, this field is read-only and fixed to value 0.

Keep alive time (seconds)

The number of seconds the connection with the ERP application server remains in the idle pool after it is closed.

If you specify 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.

Trace
If this check box is selected, a trace file is generated on the server in which the activities of the ottstpjcadaemon and ottstpbdeman processes are traced. This file is called `trace.jca<bshell.pid>` and is located in the `${BSE}/log` directory.

Allow anonymous requests
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 or in the "Basic Authentication" HTTP header. In this case the username/password as specified above is used for authentication.

If this check box is cleared, all SOAP requests must contain the username and password in the SOAP header. From version 10.5.1.0042 you can also pass the username and password in a "Basic Authentication" HTTP header. Anonymous requests are blocked.

If the **Activation type** is `none` or `baanlogin_ssl`, this field is read-only and its value is not used. In case of `baanlogin_ssl` activation, the user account as passed by ION-API is always used.

Use UPN instead of User ID
This check box is only available if the **Activation type** is `baanlogin_ssl`.

If this check box is selected, the "User Principal Name" of the user is used instead of the "User ID". This option is intended for multi-domain installations.

All fields marked with an asterisk (*) are mandatory, which means that you must specify a valid value. The following fields are also required, but you do not have to fill in these fields, because they use default values.

- **Activation type**
- **Connection port**
- **Keep alive time**
- **Max connections**
- **Max references**
- **Max threads**

You use the following fields to enable connection pooling, which means that multiple SOAP requests can be processed through one shared physical connection:

- **Max connections**
- **Max references**
- **Max threads**
- **Keep alive time**

For more information, see 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 do not take effect until you click one of these buttons. After successfully saving a configuration, the connector returns to the **ERP Servers** page.

Click **Cancel** to return to the **ERP Servers** page 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, see

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 Deployed Web Services. In your web browser, you can directly navigate to the index page by specifying this 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 shows the name of the web service and the ERP Server on which the related BDE interface has been implemented. In a multi-tenant cloud installation, no web services are explicitly defined, so no web services are displayed here.

The page also provides functionality to deploy, refresh, and undeploy web services. This table shows the commands to perform these tasks:

Command	Description
Undeploy 🗑	Undeploys all selected web services. The web services are removed from the web server and from the **Web Services Status** page. To run the command, click **Undeploy** or the remove icon.
Refresh 🔃	Reloads the WSDL files for the selected web services. The process of importing and deploying the web services is performed again. To run the command, click the refresh icon. Use this option if the web service status is `NO WSDL`; see Web Services Status page on page 18.
Deploy BDEs	Navigates to the **Deploy BDE Web Services** page. For more information on how you import and deploy web services, see Deploy BDE Web Services page on page 25 and Deploying a web service on page 31.
Deploy BOIs	Navigates to the **Deploy BOI Web Services** page. For more information on how you import and deploy web services, see Deploy BOI Web Services page on page 26 and Deploying a web service on page 31.
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.

To deploy BDE web services:

1 Select an 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, click **Configure** to configure a connection to this server. See <u>ERP Servers page</u> on page 20.

2 Deploy web services.

Select one or more web services from the list and click **Deploy** or

 .

For details about the import-and-deploy procedure, see <u>Deploying a web service</u> on page 31.

Deploy BOI Web Services page

This section describes the **Deploy BOI 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 web service. 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. 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.

To deploy BOI web services:

1 Select an 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, click **Configure** to configure a connection to this server. See <u>ERP Servers page</u> on page 20.

2 Deploy web services.

Select one or more web services from the list and click **Deploy** or

 .

For details about the import-and-deploy procedure, see <u>Deploying a web service</u> on page 34.

Advanced Settings

This section describes the additional settings that are available from version 10.5.1.

To reach this page, click <u>Advanced Settings</u> on the connector's index page, that is, the starting page.

You can specify these settings on this page:

MT Cloud Enabled
Select this option if the Connector for Web Services is running in a multi-tenant cloud environment.

Typically, you do not select this option. You can use it for "cloud-like" test environments. In the "real" cloud, the configuration pages are not available, so this option is set directly in the configuration file.

LN UI configuration file
If LN UI is installed on the same machine, you can inherit several settings from the LN UI configuration file. Specify the path to the LN UI configuration file, either as an absolute path, or as a path relative to the configuration folder of the Connector for Web Services.

Keystore location
The location of the Keystore/Truststore that is required for establishing SSL connections to the ERP Server (configured with "baanlogin_ssl" activation). You can leave this blank to "inherit" the corresponding setting from the "LN UI Configuration file".

Keystore password

The password for accessing the Keystore/Truststore.

Consumer key
The key for OAuth 1.0a authentication. This key and the secret that belongs to it are used for validating the OAuth 1.0a data that is sent in an "Authorization" HTTP header.

Consumer secret

The secret for OAuth 1.0a authentication.

These actions can be performed on this page:

Modify
Save the changes made to this page. To ensure all changes are active, you must restart the web application.

Cancel
Ignore the changes made on this page.

Note: Most of these settings are evaluated only once, during startup of the web application. To ensure all changes are applied, you must restart the Connector for Web Services web application.

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

These fields are displayed on the **Connector Log Configuration** page:

Application log level
The used log level for the connectors. You can select one of these levels.

- **Error**

  The **Error** 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 **Error** level also includes the logging of fatal errors.

- **Info**

  The **Info** 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 **Info** level includes **Error** logging.

- **Debug**

  The **Debug** 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 **Debug** level includes the **Info** logging.

Log file location (Web Server)
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.

Maximum log file size (Mb)
Specify 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 is made and the log file is cleared automatically. The file name of these backup files is the log file name extended with a number. The log files are stored in the folder that is specified in the **Log file location (Web Server)** field.

Maximum number of backup files
This field is used when the maximum log file size is specified. After the maximum size of the log file is reached, a backup of the log file is created. This process is repeated until the maximum number of backup files is reached. When reaching this maximum number, the oldest backup is deleted.

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

`<Tomcat installation directory>/logs/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.

Configuration in multi-tenant cloud mode

From version 10.5.1 you can run the Connector for Web Services in multi-tenant cloud mode. In this mode, you do not have to deploy BDEs. All BDEs on the LN system are implicitly available for use.

Typically, the configuration pages are not accessible in the cloud. Therefore, all configuration is performed by editing the configuration file directly.

For a description of the configuration file, see Configuration file on page 44.

Logging configuration

Logging is configured in the `log4j.xml` file in the configuration folder of Connector for Web Services. See the Log4J website, https://logging.apache.org/log4j/1.2/manual.html.

Changes to the `log4j.xml` file are not effective immediately. A restart of the web application is required.

Chapter 5: Employment

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 servers.

These 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 BDEs 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

To set up a web service for a BDE business interface:

1 Ensure 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 servers

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).

   Note: If the Connector for Web Services is running in a multi-tenant cloud environment, this step is not required. In a multi-tenant cloud environment all BDEs on the LN server are available if the user is authorized to use them.

After you performed the 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 is therefore not explained in detail.

For the configuration of the connection with the LN servers, see Configuring Infor LN or Baan application servers on page 19.

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

BDE access through web services

A Business Data Entity (BDE) business interface of 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 server 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 server. The LN application on the server 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 fro 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, see the *Infor Integration 6.2 - Integration Guide for BDE Clients*. Notice that the solution to create web services for BDEs discussed in this guide refers to the Web Services Toolkit of Infor Integration 6 and does not apply to the Infor Enterprise Server Connector for Web Services.

Deploying 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 perform 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:

1 Start the c4ws web application.

Ensure the Connector for Web Services is running.

For detailed information on how to start and stop the connector web application, see Starting and stopping the connector on page 37.

When the connector is started, you can view its index page by specifying this 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. Therefore, you must log on as a user with Tomcat Manager privileges.

2 Check LN connections.

On the **ERP Servers** page 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, click ERP Servers.

For more information about the configuration of LN Server connections, see Configuring Infor LN or Baan application servers on page 19 and the Add/Modify an ERP Server page on page 21.

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

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.

Complete these steps:

a Navigate from the connector's index page to the **Deploy BDE Web Services** page by clicking **Deployed Web Services page** and **Deploy BDEs**.

b On the **Deploy BDE Web Services** page, indicate which LN server 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.

c Select one of the available BDE web services.

d 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 (`http://<web server host>:<port nr>/c4ws/services`).

In a multi-tenant cloud environment, no web services are displayed on the **Web Services Status** page. In this case you can download the WSDL file by completing these steps:

1 Start the **Business Objects (ttadv7500m000)** session in LN UI.

2 Select the appropriate Business Object for the BDE.

3 Select **Actions > Download WSDL**.

Afterwards, you can edit the `soap:address` in the downloaded file to suit your needs.

The `<soap:address>` in the `<wsdl: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 server. In our case, the location of the server is indicated by the **Identifier** of the LN server hosting the BDE business object that is exposed by the web service.

This 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/My▶
BackEnd123" />
  </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.

To set up a web service for a BOI business interface:

1 Ensure 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 servers.
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 is therefore not explained in detail.

For the configuration of the connection with the Baan IV servers, see Configuring Infor LN or Baan application servers on page 19.

If no suitable BOI interfaces are available, you can implement a BOI interface on a Baan IV server using the Implementation Generator of Infor LN Studio. For more information about this subject, see the LN 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 access these BOI proxies from the Connector for Web Services, the domain packs, which 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 server. The Baan IV application on the server 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 fro are completely defined by the WSDL file of the BOI web service.

Note: For more conceptional information about BOI interfaces, see the *Infor Integration 6.2 - Integration Guide for BOI Clients*. Notice that the solution to create web services for BOIs discussed in this guide refers to the Web Services Toolkit of Infor Integration 6 and does not apply to the Infor Enterprise Server Connector for Web Services.

Deploying 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 perform 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:

1 Start the c4ws web application.

Ensure the Connector for Web Services is running. For detailed information on how to start and stop the connector web application, see Starting and stopping the connector on page 37.

When the connector is started, you can view its index page by specifying this 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. Therefore, you must log on as a user with Tomcat Manager privileges.

2 Check Baan IV connections.

On the **ERP Servers** page 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, click ERP Servers.

For more information about the configuration of Baan IV server connections, see Configuring Infor LN or Baan application servers on page 19 and Add/Modify an ERP Server page on page 21.

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

3 Deploy a BOI web service.

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

Complete these steps:

a Navigate from the connector's index page to the **Deploy BOI Web Services** page by clicking **Deployed Web Services page** and **Deploy BOIs**.

b On the **Deploy BOI Web Services** page, indicate which Baan IV server 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.

c Select one of the available BOI web services.

d 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 (`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 server. In our case, the location of the server is indicated by the **Identifier** of the Baan IV server hosting the BOI business object that is exposed by the web service.

This 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/My▶
BackEnd123" />
  </wsdl:port>
</wsdl:service>
```

Overruling ERP servers configuration

You can overrule activation-related ERP servers configuration properties at runtime. Specify the settings for the activation properties in the *SOAP* message (header).

You can overrule these activation properties:

- User name
- Password
- Company

If the ERP server is configured to allow anonymous requests, you can optionally use the overrule mechanism and change the properties. You must use the User name and Password activation properties together.

Note:
- If the ERP server does not allow anonymous requests, you must overrule the user name and password.
- If the ERP server has been configured with "baanlogin_ssl" activation, you cannot overrule the user name and password. Any such values in the SOAP header are ignored.

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* have some API extensions to set the properties.

From version 10.5.1.0042 of the Connector for Web Services, you can also override the user name and password in an HTTP header using Basic Authentication. For a specification of Basic Authentication, see https://www.ietf.org/rfc/rfc2617.txt.

If the user name and password are overridden in both the SOAP header and the HTTP header, the value from the SOAP header is used.

Note:
- Changing user or company has effect on the ERP Server for either installed software (package VRC) or data (companies).
- 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.

To develop web client applications:

1 Generate WSDL Proxies.

Generate WSDL Proxies from the WSDL file. On the client, these proxies form a local API that mirrors 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.

You can use several tools to generate WSDL Proxies. For Java applications, for example, you can use the proxy generator of Apache AXIS; see `http://ws.apache.org/`.

2 Write client applications.

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.

Starting or stopping Tomcat (Windows)

To start or stop Tomcat on a Windows system:

1 Open the **Services** window.

On Windows systems, you can open the **Services** window in various ways. For Windows 2003, for example, you can complete these steps:

a   Click **Start**.
b   Point to **Settings**.
c   Click **Control Panel**.
d   Double-click **Administrative Tools**.
e   Double-click **Services**.

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-8.0.21** menu, or through batch files in the `<installation-directory>\apache-tomcat-8.0.21\bin` directory.

Starting or stopping Tomcat (UNIX)

To start or stop Tomcat on a UNIX system:

1 Start the Tomcat server.

Go to the `<Tomcat>/bin` folder. Run this command:

```
$ sh startup.sh
```

2 Stop the Tomcat server.

Go to the `<Tomcat>/bin` folder. Run this command:

```
$ sh shutdown.sh
```

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

Starting or stopping the connector web application

To start or stop the connector web application:

1 Open the Tomcat home page.

Ensure the Tomcat server is running.

To open the Tomcat home page in your browser, open this URL:

```
http://<host>:<port>
```

`<host>` refers to the name of the computer that runs the Tomcat server. `<port>` refers 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 by default.

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.

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 **Stop** command.

Chapter 6: Troubleshooting

This chapter describes how you can trace down errors and solve known problems.

Tracing down errors

To find the source of an error, 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, set the logging to a more detailed level. Usually, the `Info` log level provides sufficient information.

The `Debug` log level can provide too much information, and seriously affects the performance of the connector. Therefore, we recommend that you do not use this log level.

To change the log level, open the **Log configuration** page of the connector and select another value for the **Application log level** field.

For more information on the log level, see Log configuration on page 27.

Note: The connector log file is not the only log file you can consult to trace down the source of an error. Errors and warnings are also logged on the ERP server and, if applicable, in the ESB message bus.

Miscellaneous

If the installation fails, or if the Connector for Web Services does not start, check whether the prerequisites are installed.

See Requirements on page 11 and the *Infor Enterprise Server Installation Guide.*

Appendix A: Web service concepts

This section gives a short description of web services and the related WSDL and SOAP web standards, 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. To access the web services, open this URL ("SOAP address"):

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

The Infor LN or Baan server name 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; see 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 that web services use for exchanging and verifying data are XML-based.

SOAP

SOAP is an XML-based protocol 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 WDSL 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 a URL address in your web browser, HTTP is used to retrieve the corresponding website.

To understand and use the Connector for Web Services, knowledge about WSDL and SOAP standards is required.

Appendix B: Handling non-standard SOAP requests

Since version 10.3.0.0060 you can handle SOAP requests that do not conform to the WSDL of the service that is invoked. To handle these requests, you must perform an XSL transformation on the XML contents of the SOAP body. This transformation must result in a format that does conform to the WSDL. The transformed request is then processed as usual. The resulting response is transformed again, resulting in a format that can be handled by the client software.

Note: Only BDE services are supported; you cannot use the old BOI services with non-standard requests.

This is useful in the following situation. A third party has defined a web service (WSDL), and client software is available. LN has a BDE that implements the required logic. The WSDL belonging to the BDE does not match the third party WSDL that is used by the client software. To use the existing client software to communicate with the service offered by the BDE, you must write a transformation stylesheet and deploy it to the Connector for Web Services application.

The non-standard SOAP requests must adhere to these requirements:

* The XML Namespace URI of the element within the SOAP body must not start with
  `"http://www.infor.com/businessinterface/"` or `"http://www.infor.com/boi_in
  terface/"`. These namespace URIs are reserved for the standard SOAP requests that conform
  to the WSDL of the BDE.
* If any of the ERP server's activation properties must be overruled, you must provide these settings
  as usual in the SOAP header. See <u>Overruling ERP servers configuration</u> on page 36. You cannot
  change these settings during the XSL transformation because the SOAP header is not affected by
  the transformation; only the SOAP body is being processed.

You must specify the XSL transformation in a stylesheet file with the same name as the service that is being invoked. You can find the service name on the **Deployed Web Services** configuration page. The service name is equal to the name of the BDE; if the same BDE has been deployed for multiple ERP Servers, the name is possibly extended with "_2" or "_3", etc. The stylesheet files have a "`.xsl`" extension and are located in the "`WEB-INF/wsdl`" folder within the c4ws application. Example: if you want to process non-standard requests for a service called "AreasService_2", you must specify transformation instructions in "`<Infor Installation folder>/c4ws/web/WEB-INF/wsdl/Ar
easService_2.xsl`".

You can define a generic stylesheet called "`_default_.xsl`". If a non-standard request arrives, and the addressed service does not have its own stylesheet, the generic stylesheet is used to transform the request and response. If no applicable stylesheet is present, the non-standard request cannot be processed and a SOAP fault is returned.

For the transformation stylesheets only version 1.0 of XSLT is supported. Therefore, the stylesheet must begin with: `<xsl:stylesheet version="1.0" xmlns:xsl=http://www.w3.org/1999/XSL/Transform` …

After adding or changing stylesheets in the `WEB-INF/wsdl` folder, you must restart the web application to pick up the changes.

XML Namespaces

When processing a request, the XML namespace URI of the request data is used to determine whether a standard request, which conforms to the WSDL of the BDE, or a non-standard request is handled.

The rules in the XSL stylesheet must transform a non-standard request into a standard request. This transformation can ignore any namespaces of the XML elements within the request data, because these are not used when further processing the data.

When a response is returned, it has these namespaces:

- The top-level element has the same namespace as the top-level element in the request. The top-level element in the response is called "`xxResponse`", where `xx` is the invoked method of the BDE. For "standard" requests, the namespace URI starts with "`http://www.infor.com/businessin terface/`"; for non-standard requests, the namespace URI starts with something else.
- All child elements have an empty namespace.

If the client software requires specific namespaces in the response, the transformation must add the necessary namespaces in the response.

Appendix C: Configuration file

From Connector for Web Services 10.5.1 a new-style configuration file is used. The format of this file is identical to the format that is used for the configuration of LN UI. Connector for Web Services can inherit several settings from LN UI: you can use the "environment" settings of LN UI for the "ERP Server" settings of Connector for Web Services.

The configuration file is modified when applying changes to the configuration pages of Connector for Web Services. In a multi-tenant cloud installation, you typically use a text editor to change the configuration file.

The configuration file is called `c4wsProperties.xml` and is located in the folder specified by the `configLocation` parameter of the web application. For a default installation, the file is located at `c:\Infor\ese\c4ws\config\c4wsProperties.xml`.

Example contents:

```
<?xml version="1.0" encoding="utf-8"?>
<c4ws>
  <general>
    <cloudEnabled>true</cloudEnabled>
    <lnuiConfigFile>../../lnui/config/webuiProperties.xml</lnuiConfigFile>

  </general>

  <!-- overrides/additions to LN UI settings -->
  <environments>
    <environment name="infor">
      <!-- (settings like "host", "port", "bshell", "bse" are inherited
           from the LN UI settings)
      -->
      <keepalive>60</keepalive>
    </environment>
  </environments>

  <!-- C4WS settings, not present in LN UI -->
  <cloud>
    <environmentID>infor</environmentID>
    <consumerkey>c4wsclient</consumerkey>
    <consumersecret cleartext="true">T0p$3cr3t</consumersecret>
  </cloud>

</c4ws>
```

Available settings

This table shows settings that are inherited from `webuiProperties.xml`:

Path (relative to "c4ws" root node)	Description
general/cloudEnabled	If "true", this is a, multi-tenant, cloud installation.
environments	Contains one or more LN UI environments.
environments/environment	Contains the settings for an LN UI environment.
environments/environment/@name	Name (ID) of the environment.
environments/environment/host	Hostname of LN server.
environments/environment/port	Port number to connect to, such as 512 for "rexec".
environments/environment/activation	Activation type: "rexec", "baanlogin", "baanlogin_ssl".
environments/environment/bse	BSE directory.
environments/environment/bshell	Bshell name.
environments/environment/command	Command line parameters for bshell.
environments/environment/username	User name to be used for Connector for Web Services admin tasks, that is, deploying BDEs. This user name is also used for Keystore management in LN UI.
environments/environment/password	Encrypted password for user.

This table shows settings for Connector for Web Services:

Path (relative to "c4ws" root node)	Description
general/lnuiConfigFile	Path to `webuiProperties.xml` to be included.
cloud/environmentID	ID of the environment to be used in the multi-tenant cloud installation. This ID must match some "environments/environment/@name".
cloud/consumerkey	The consumer key to be used for OAuth1.0a authentication.
cloud/consumersecret	The consumer secret, that is, password, for OAuth1.0a authentication. Usually stored in encrypted format.
cloud/consumersecret/@cleartext	If "true", the consumer secret is stored in cleartext format. Only used when manually editing the file.
cloud/keystore	Path to keystore/truststore for baanlogin_ssl connections.
cloud/keystorepassword	Password for keystore/truststore for baanlogin_ssl connections. Usually stored in encrypted format.

Path (relative to "c4ws" root node)	Description
cloud/keystorepassword/@cleartext	If "true" the password is stored in cleartext format. Only used when manually editing the file.
environments/environment/company	Company number.
environments/environment/keepalive	Keep-alive timeout for connection pooling.
environments/environment/maxconnections	Maximum number of connections.
environments/environment/maxreferences	Maximum number of references for shared connections.
environments/environment/maxthreads	Maximum number of waiting threads.
environments/environment/anonymous	If "true", anonymous requests, that are using the admin account, are allowed
environments/environment/trace	If "true", server tracing is active. `$BSE/log/trace.jca.xxxx`
backends	Contains one or more LN server specifications for on-premises/single-tenant mode.
backends/backend/@name	Server name, that is, "connectionpoint", as used in request URL.
backends/backend/@ref	Reference to LN UI environment. This reference must match some "environments/environment/@name".

This table shows settings that are related to Infor Reporting:

Path (relative to "c4ws" root node)	Description
reporting/certificatesfolder	Path to the folder containing certificates for accessing Infor Reporting
reporting/userstore	Name of "UserStore" file within certificates folder
reporting/userstorepassphrase	Passphrase for UserStore
reporting/truststore	Name of "TrustStore" file within certificates folder
reporting/truststorepassphrase	Passphrase for TrustStore

Glossary

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.

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.

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.

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.