



BEA AquaLogic® Data Services Platform

Installation Guide

Note: Product documentation may be revised after the release.
The most recent product information is always available from
the BEA [e-docs](#) site.

Contents

1. Preparing to Install ALDSP

Installation Overview	1-2
Supported Configurations	1-3
Installation Notes	1-3
ALDSP Installation Components.....	1-4
Preconfigured Samples Domain	1-5
Internationalization Support.....	1-5

2. Installing ALDSP Using GUI Mode

Before You Install	2-2
Installing Using GUI Mode	2-2

3. Installing ALDSP Using Console or Silent Mode

Before You Install	3-2
Installing Using Console Mode	3-3
Using Silent Console Mode to Install ALDSP.....	3-6
Using Silent Mode on Windows and UNIX Systems.....	3-6
Exploring the Silent Mode Installer Properties File.....	3-9

4. Post-Installation Tasks

Verifying the Installation	4-2
Starting the Development of a Data Integration Solution	4-2

5. Uninstalling ALDSP

Automated Uninstallation.	5-2
-----------------------------------	-----

6. Upgrading from ALDSP 3.0 or 2.5 to ALDSP 3.2

Upgrading from ALDSP 3.0 to 3.2.	6-2
Upgrading from ALDSP 2.5 Prerequisites.	6-2
Upgrading ALDSP 2.5 Artifacts to ALDSP 3.2.	6-2
Upgrading an ALDSP 2.5 Application	6-4
Upgrading an ALDSP Control from ALDSP 2.5 to 3.2	6-9
Importing ALDSP 2.5 Configuration to ALDSP 3.2	6-14
Post-Upgrade Artifact Mapping in ALDSP 3.2	6-16
Post-Upgrade Tasks for Java Web Service	6-21
ALDSP 2.5 Upgrade: Known Issues and Workarounds.	6-22

Preparing to Install ALDSP

Before you install ALDSP, you need to verify that your system meets minimum requirements. This chapter provides information about supported platforms, system hardware and software requirements, and installation prerequisites. It also provides an overview of the ALDSP components that get installed.

The following sections are included:

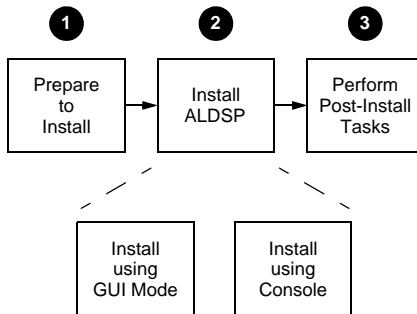
- [Installation Overview](#)
- [Supported Configurations](#)
- [Installation Notes](#)
- [ALDSP Installation Components](#)

Note: ALDSP was originally named Liquid Data. Some instances of the original name remain in the product, installation path, and components.

Installation Overview

This section provides a high-level overview of the tasks required to install ALDSP, illustrated in [Figure 1-1](#).

Figure 1-1 Overview of Installation Process



Task 1: Prepare to install

1. Preparing to install ALDSP involves checking the supported platforms and databases, and verifying that you have the installation prerequisites in place. In addition, you can familiarize yourself with a description of the components that get installed on your system.
2. Uninstall any previous versions of ALDSP. For details see [Chapter 5, “Uninstalling ALDSP.”](#)

Note: The uninstall option does not remove any ALDSP or other BEA WorkSpace Studio projects that you have created.

Task 2: Install ALDSP 3.2

You can install ALDSP in the following modes:

- GUI mode — Features an easy-to-use graphical installer application
- Console mode — Is suitable on UNIX systems without a graphics (windowing) workstation
- Silent mode — Reads an installer properties file to determine the install options on Windows or UNIX systems

For more information about using GUI mode, see [Chapter 2, “Installing ALDSP Using GUI Mode.”](#) For more information about using Console mode and Silent mode, see [Chapter 3, “Installing ALDSP Using Console or Silent Mode.”](#)

Task 3: Perform post-installation tasks

After installation, you can verify that the installation was successful and explore Windows shortcuts and UNIX paths to key components. For more information, see [Chapter 4, “Post-Installation Tasks.”](#)

Supported Configurations

For information on supported operating systems, DBMS vendors, and other configurations, see [Supported Configurations](#) on BEA’s e-docs web site.

Installation Notes

The ALDSP 3.2 installer application installs the following additional software, as required:

- WebLogic Server 10.0 MPI, if it is not already present in the <bea_home> directory you select
- The latest JRockit version ([JRockit version 5.0 R27.4 JDK](#)) to ensure higher ALDSP performance

The ALDSP 3.2 IDE runs as a Workspace Studio plug-in. (Note that in ALDSP 3.0, the ALDSP IDE was named Data Services Studio.)

Note: If you are using Windows 2000, the maximum classpath size can be exceeded. For this reason, it is recommended that when you install ALDSP you install it into the <bea_home> directory with a directory name of four characters or less, such as <bea>.

ALDSP Installation Components

Table 1-1 provides a list of components that are installed with ALDSP.

Table 1-1 ALDSP Installation Components

Component Name	Sub-components
ALDSP Runtime	<ul style="list-style-type: none">ALDSP runtime including the XQuery engineConfiguration frameworkALDSP Administration ConsoleALDSP base domain template
Note: Installing the runtime is mandatory for ALDSP and is a prerequisite for installing other components.	Note: WebLogic Server 10.0 MP1 is a prerequisite for ALDSP runtime.
ALDSP IDE and Client Tools	<ul style="list-style-type: none">Common WorkSpace Studio 1.1 pluginsALDSP IDE pluginsALDSP Microsoft Excel add-inWorkSpace Studio common componentsAquaLogic tooling common components including AquaLogic Sync and AquaLogic Enterprise Registry browser
	Note: Eclipse 3.2.2/WTP 1.5.4 or WorkSpace Studio 1.1 is required to run the ALDSP perspective.
ALDSP Samples	<ul style="list-style-type: none">Samples domainSample project predeployed in the samples domainRetail Dataspace Sample Application

When you install ALDSP, a valid evaluation license is automatically included with the installation.

Preconfigured Samples Domain

The full ALDSP installation provides a preconfigured samples domain, as shown in [Table 1-2](#).

Table 1-2 ALDSP Samples Preconfigured Domain and Start Commands for Samples Server

Platform	Windows and UNIX Paths to Start in Each Domain	Description
Windows	Start > All Programs > <i>BEA Products</i> > Examples > AquaLogic Data Services Platform 3.2 > Start Examples Server	Starts the ALDSP samples server on Windows
	or Run <code>startWebLogic.cmd</code> from any one of the following locations: <aldsp_home>\samples\domains\aldsp\ or <aldsp_home>\samples\domains\aldsp\bin	Starts the ALDSP samples server on Windows
UNIX	Run the <code>startWebLogic.sh</code> command from any of the following locations: <aldsp_home>/samples/domains/aldsp/ or <aldsp_home>/samples/domains/aldsp/bin	Starts the ALDSP samples server on UNIX

Note: <aldsp_home> is the home directory for installing ALDSP.

For a detailed explanation of domains, see “[Creating WebLogic Configurations Using the Domain Control Wizard](#)” in the WebLogic Server documentation on the BEA e-docs site.

Internationalization Support

ALDSP is internationalized and supports multi-byte data from the underlying data sources. Specifically, ALDSP works with Japanese character sets, where the underlying databases are running in Japanese locales.

Preparing to Install ALDSP

Installing ALDSP Using GUI Mode

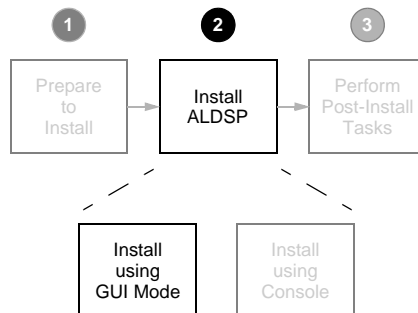
This chapter describes how to install ALDSP using GUI mode.

It includes the following sections:

- [Before You Install](#)
- [Installing Using GUI Mode](#)

[Figure 2-1](#) illustrates the current task in the overall installation process.

Figure 2-1 GUI Mode Installation Task



Before You Install

Before you begin installing ALDSP, confirm that the following prerequisites are met:

- WebLogic Server 10.0 MPI is installed in the `<aldsp_home>` directory in which you are going to install ALDSP. The ALDSP 3.2 installer application installs WebLogic Server 10.0 MPI if it is not present.
- Any previous installations of ALDSP have been uninstalled. For details see [Chapter 5, “Uninstalling ALDSP.”](#)
- For Windows (and UNIX and Linux), an entry for the JDK 150_11 bin directory (for example: `<bea_home>/JDK 150_11/bin`) is included in your PATH environment variable setting before any other JDK bin directories.

Note: By default, WorkSpace Studio (Eclipse 3.2.2) is available for installation with ALDSP 3.2 in the `<bea_home>\workSpaceStudio_1.1\workSpaceStudio` directory. If you are planning to use an Eclipse version that is not bundled with ALDSP, then you need to manually clean up the Eclipse configuration directory to avoid some known issues in Eclipse 3.2.2. All files except `config.ini` and files under `.settings` subdirectory should be deleted.

For additional information about installation prerequisites, see [Installation Notes](#) in [Chapter 1, “Preparing to Install ALDSP.”](#)

Installing Using GUI Mode

This section describes how to install ALDSP using the GUI mode on Microsoft Windows and supported UNIX and Linux-based platforms.

To install ALDSP in GUI mode:

1. Launch the GUI installer.

On Microsoft Windows-based systems:

- a. Navigate to the folder to which you downloaded the ALDSP installer using Windows Explorer.
- b. Double-click the installer executable file. For example:
`aldsp_320_win32.exe`

On UNIX and Linux-based systems:

- a. Verify that the console attached to the machine on which you are installing the software supports a Java-based GUI.
- b. Open a command window, and change (cd) to the directory to which you downloaded the ALDSP installer file.
- c. Start the installer in a new shell by entering the following:

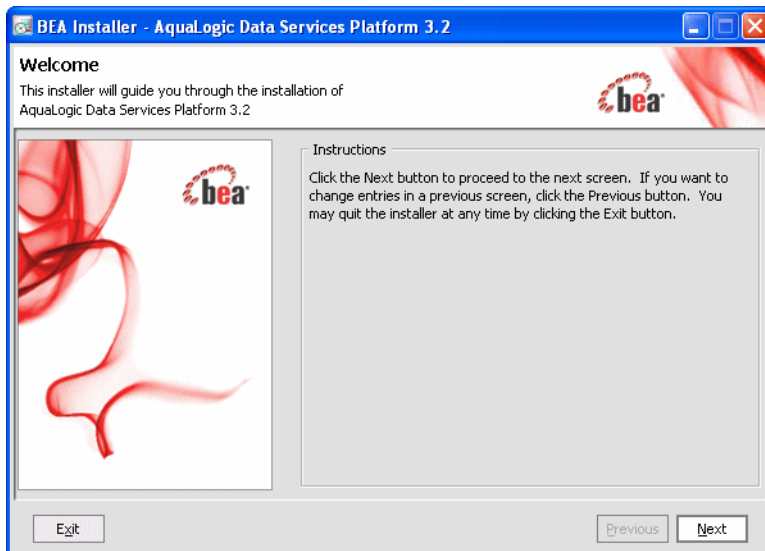
```
sh filename.bin
```

where *filename.bin* is the name of the ALDSP installation program specific to your platform. For example, enter the following to start the Solaris version of the ALDSP installation program:

```
sh aldsp_320_solaris32.bin
```

The installer application welcomes you to install ALDSP, as illustrated in [Figure 2-2](#).

Figure 2-2 ALDSP Installation Welcome Screen

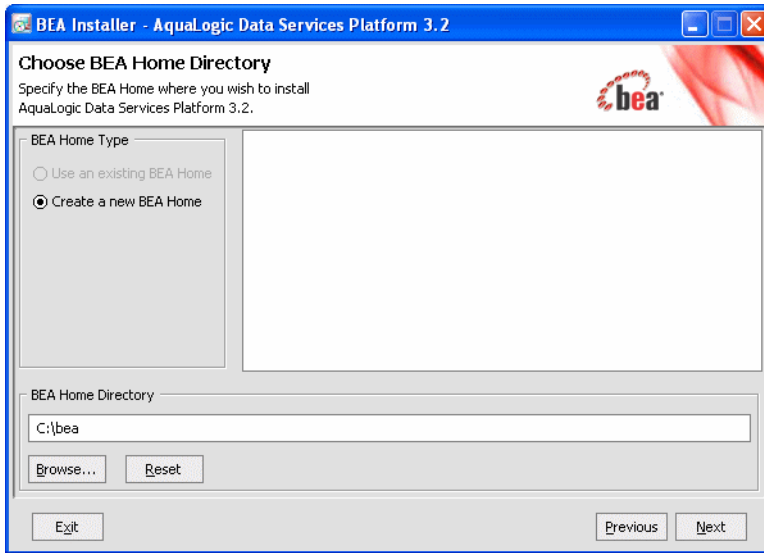


2. Click **Next** to begin the installation. The License Agreement appears.

3. Review the License Agreement. Choose **Yes** to accept, and click **Next**.

A screen appears enabling you to specify the BEA home directory, as illustrated in [Figure 2-3](#).

Figure 2-3 Choose BEA Home Directory



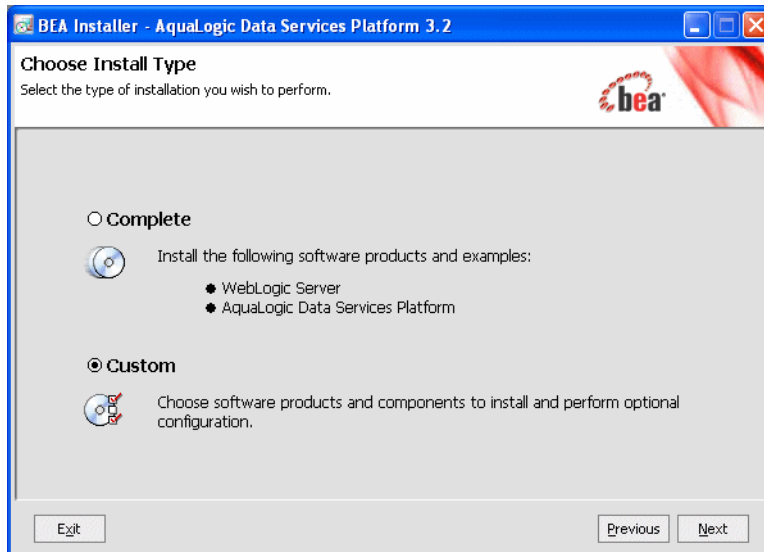
The BEA Home directory (<bea_home>) serves as the central support directory for all the BEA products installed on your system.

4. Specify the BEA Home directory, and click **Next**.

You can also choose to create a new BEA Home directory if this is an initial installation.

Or you can choose a directory from the list or type the directory name in the BEA Home Directory field. You must install ALDSP in the same home directory where you installed WebLogic Platform 10.0 MP1.

A screen appears enabling you to choose the type of installation, as illustrated in [Figure 2-4](#).

Figure 2-4 Choose Install Set

5. Choose either Complete or Custom as the install type, and click **Next**.

The following describes the options:

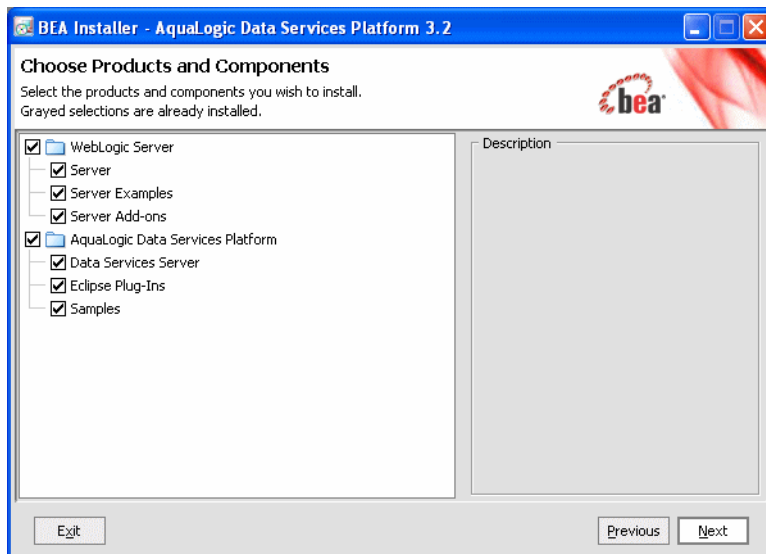
- Complete

This option installs all ALDSP components mentioned in [ALDSP Installation Components](#).

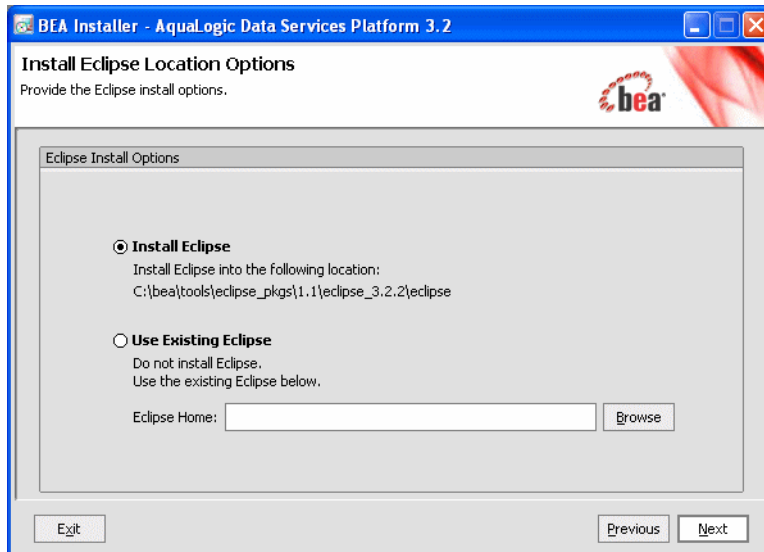
- Custom

Enables you to choose the ALDSP components you need to install. If you select this option and click **Next**, the screen shown in [Figure 2-5](#) is displayed.

Figure 2-5 Choosing Products and Components to Install



6. Choose the components and click **Next**. The next screen allows you select or specify the location of your Eclipse installation as shown in [Figure 2-6](#).

Figure 2-6 Choosing Eclipse Installation Location

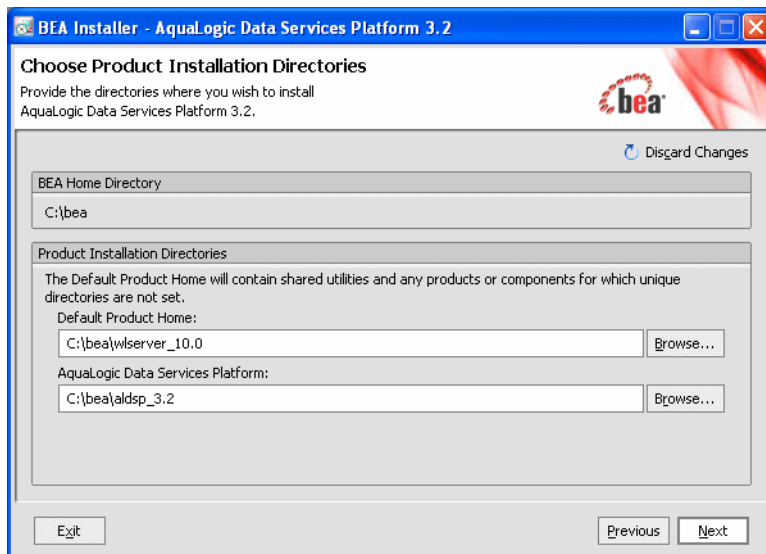
If you want to install a new Eclipse instance, then select the **Install Eclipse** option. This option installs Eclipse at the following location:

```
<bea_home>\tools\eclipse_pkgs\1.1\eclipse_3.2.2\eclipse
```

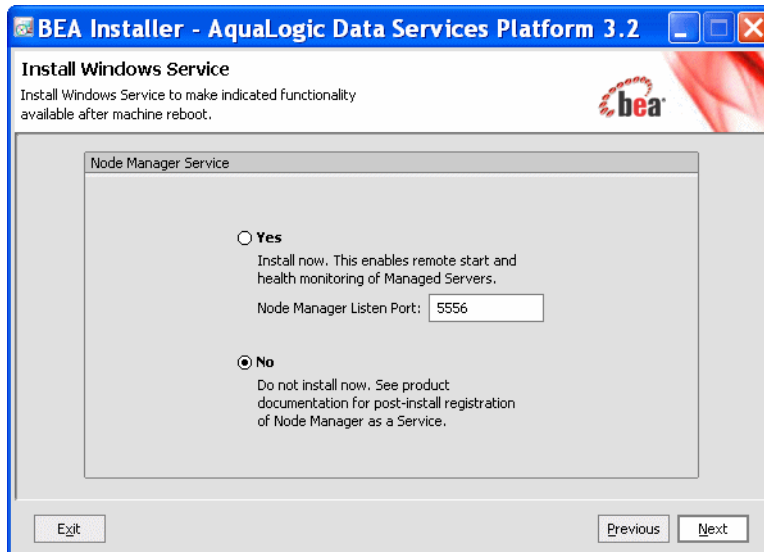
If you already have Eclipse installed on your machine and want to use the same instance of Eclipse to work with ALDSP, then select the **Use Existing Eclipse** option and enter the location where Eclipse is installed.

7. Click **Next**. This screen allows you to specify the directory where you want to install ALDSP. By default, ALDSP is installed in `<bea_home>\aldsp_3.2`. See [Figure 2-7](#).

Figure 2-7 Choosing Installation Directories



8. The Install Window Service screen displays as shown in [Figure 2-8](#). By default, choose **No**.

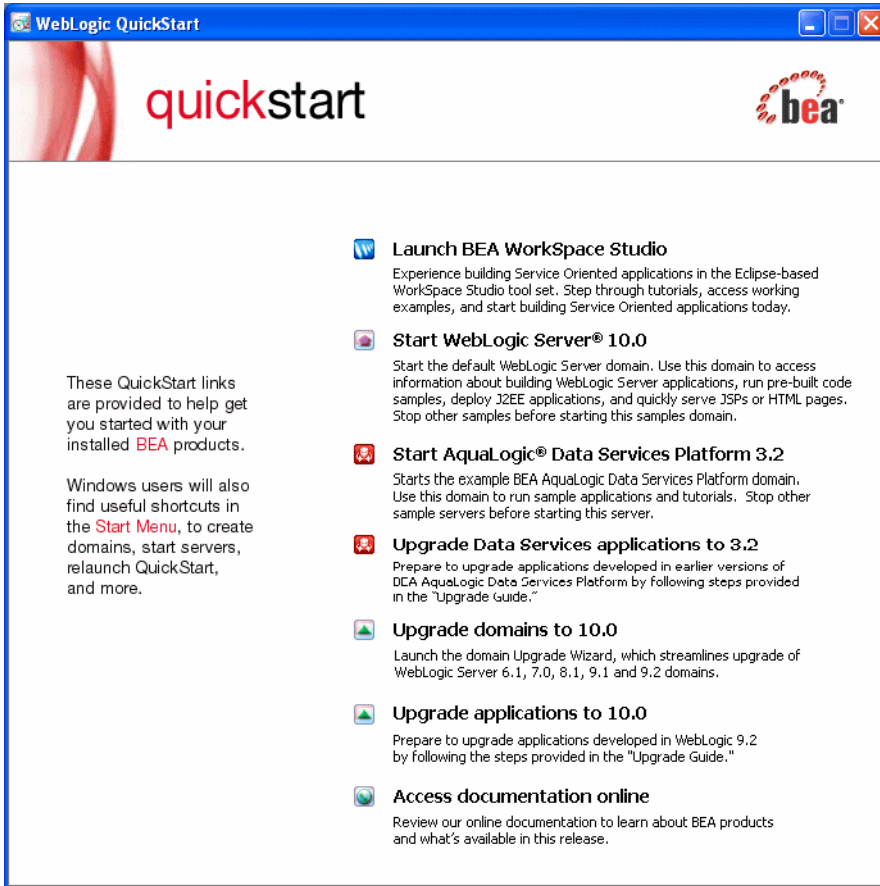
Figure 2-8 Installing Windows Service

9. The installer displays a progress screen, and begins installing the files on your system. When the installation is complete, the Installation Complete screen appears. Click **Done**.

You have successfully installed an instance of ALDSP.

10. After you have installed ALDSP, you may choose to launch quickstart. If you choose this option, the quickstart window displays, showing links to start applications, upgrade applications, or access product documentation. See [Figure 2-9](#).

Figure 2-9 Launching quickstart



Installing ALDSP Using Console or Silent Mode

This section describes how to install ALDSP using either Console mode or Silent mode. Console mode is an interactive installation that you can use on UNIX systems without a graphics (windowing) workstation.

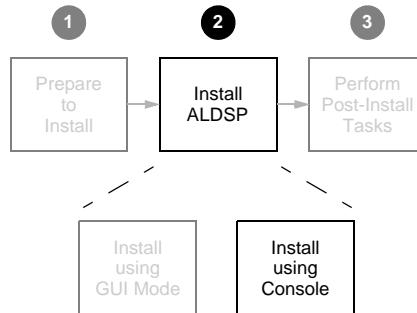
Silent mode is a non-interactive installation on a Windows or UNIX system. In Silent mode, the installer application uses a properties file to obtain installation parameters.

The chapter includes the following sections:

- [Before You Install](#)
- [Installing Using Console Mode](#)
- [Using Silent Console Mode to Install ALDSP](#)

Figure 3-1 illustrates the current step in the overall installation process.

Figure 3-1 Console Mode Installation Task



Before You Install

Before you begin installing ALDSP, confirm that the following conditions are met:

- WebLogic Server 10.0 MP1 is installed in the <bea_home> directory in which you are installing ALDSP. The ALDSP 3.2 installer application installs WebLogic Server 10.0 MP1 if it is not present.

- ALDSP is not running on the system.

You must remove ALDSP in cases where it is located in the same <aldsp_home> directory to which you plan to install. For details see [Chapter 5, “Uninstalling ALDSP.”](#)

- An entry for the JDK 150_11 bin directory (for example <bea_home>/JDK 150_11/bin) must be included in your PATH environment variable setting before any other JDK bin directories.
- By default, WorkSpace Studio (Eclipse 3.2.2) is available for installation with ALDSP 3.2 in the <bea_home>\workSpaceStudio_1.1\workSpaceStudio directory. If you are planning to use an Eclipse version that is not bundled with ALDSP, then you need to manually clean up the Eclipse configuration directory to avoid some known issues in Eclipse 3.2.2. All files except config.ini and files under .settings subdirectory should be deleted.

For additional information about installation prerequisites, see [“Installation Notes” on page 1-3 in Chapter 1, “Preparing to Install ALDSP.”](#)

Installing Using Console Mode

This section describes the console-mode installation procedure.

Note: You cannot install ALDSP in console mode on Solaris systems running in a Japanese locale. In cases when you are installing on a Japanese Solaris system, change your locale in the shell in which you are installing to English, and then install the product. After ALDSP is installed, you can run it in a Japanese locale.

To install ALDSP on a UNIX system in console mode:

1. Change to the directory (cd) that contains the ALDSP installer.
2. Start the installation program.

The installer file name has the following form:

filename.bin

where *filename* is the name of the ALDSP installation program specific to your platform. For example, enter the following to start the Solaris version of the ALDSP installation program:

```
sh aldsp_320_solaris32.bin -mode=console
```

Similarly, enter the following to start the Linux version of the ALDSP installation program:

```
./aldsp_320_linux32.bin -mode=console
```

A message appears indicating that the software is being extracted, followed by a welcome message, as shown in the following text:

```
<---- BEA Installer - BEA AquaLogic Data Services Platform 3.2 ---->
Welcome:
-----
This installer will guide you through the installation of BEA AquaLogic
Data Services Platform 3.2. Type "Next" or enter to proceed to the next
prompt. If you want to change data entered previously, type "Previous".
You may quit the installer at any time by typing "Exit".

Enter [Exit][Next]>
```

3. Press Enter to continue.

The license agreement is displayed. Review the entire agreement. When you reach the end of the agreement, you are prompted to accept or reject the terms.

Select Option:

- 1 - Yes, I agree with the terms of the license
- 2 - No, I do not agree with the terms of the license

Enter option number to select OR [Down][Exit][Previous]>

4. Enter 1 if you accept the license agreement.

You are prompted to choose the BEA home directory, as shown in the following text:

```
<---- BEA Installer - AquaLogic Data Services Platform 3.2 ---->
Choose BEA Home Directory:
-----
```

"BEA Home" = [Enter new value or use default /usr/local/bea]

Enter new BEA Home OR [Exit][Previous][Next]>

The <bea_home> directory serves as the central support directory for all the BEA products installed on your system. For a detailed description of how this directory is used, see “BEA Home Directory” in “Preparing to Install WebLogic Server” in [Installing BEA WebLogic Server](#).

A warning message is displayed in cases when a previous installation of ALDSP is found.

5. Press Enter to accept the default BEA home directory, or enter an alternative directory.

When prompted, confirm the directory you selected. You are prompted to choose whether you want to perform a complete or custom installation, as shown in the following text:

Select the type of installation you want to perform.

```
->1|Complete
   | Install the following software products and examples:
   | - WebLogic Server
   | - AquaLogic Data Services Platform
   |
   |
2|Custom
   | Choose software products and components to install and perform
   | optional configuration.
```

Enter index number to select OR [Exit][Previous][Next]>

6. Enter 1 or 2 to choose a complete or custom installation respectively.

When you specify a custom installation, the installer enables you to choose the components to install, as shown in the following text:

```
<----- BEA Installer - AquaLogic Data Services Platform 3.2 ----->
```

```
Choose Components to install:
```

```
-----
```

```
Release 3.2.0.0
+----WebLogic Server [1] v
|   +----Server [1.1] v
|   +----Server Examples [1.2] v
|   +----Server Add-ons [1.3] v
+----AquaLogic Data Services Platform [2] v
      +----Data Services Server [2.1] v
      +----Eclipse Plug-Ins [2.2] v
      +----Samples [2.3] v
```

```
Enter number exactly as it appears in brackets to toggle selection OR
[Exit][Previous][Next]>
```

Enter the value for the component, for example, 1.1 for the ALDSP Server. The installer prompts you for additional component values. Continue selecting components, and press Enter without a value when you are done.

7. Enter the Eclipse installation options. Option 1 allows you to install Eclipse bundled with ALDSP. Option 2 allows you to specify a different Eclipse home directory and not install the Eclipse bundle available with ALDSP.
8. Review the ALDSP product installation directories and press Enter.
9. The installation begins and the installation status is displayed as it progresses. When the installation is complete, "Congratulations!" is displayed. Press Enter to exit the installer.

This completes the Console mode installation process.

Using Silent Console Mode to Install ALDSP

The Windows, UNIX, and Linux versions of the installer provide a noninteractive, or *silent-mode* installation, that you can use in cases when you want to install ALDSP without needing to supply information from the keyboard during the installation process. Instead, the installer gets the required information from a properties file that you provide.

Before launching a silent-mode installation, make sure that all installation prerequisites are met and that all the information in the properties file is correct. After the silent installer is started, it proceeds in the background and does not report exceptions. Some exceptions are ignored.

However, if a previously installed copy of ALDSP is detected, a dialog box appears asking if you want to override the old version. Other exceptions cause the installer to fail. For example, if the specified BEA home directory `<bea_home>` or the specified ALDSP install directory (`USER_INSTALL_DIR`) do not exist or are incorrect, the installer fails.

Note: On UNIX systems, the installer displays the message `Installation Complete` when it finishes. This message does not necessarily indicate that the installer was successful; it means only that the process has finished running.

If a fatal exception occurs during installation, the installer displays a message and no changes are made to the system.

Using Silent Mode on Windows and UNIX Systems

This section describes how to install ALDSP on Windows and UNIX systems using Silent Mode.

To install using silent mode:

1. Create the required installer properties file.

The content of the file is described in [“Exploring the Silent Mode Installer Properties File” on page 3-9](#). You can use any legal file name for the installer properties file. Verify that the `<BEAHOME>`, `<USER_INSTALL_DIR>`, and other values specified in the properties file are correct and that all requirements have been met.

2. Open a command window.

3. Navigate to the directory containing the ALDSP installer.

- On Windows, the installer application is:

`aldsp_320_win32.exe`

- On UNIX and Linux, the installer application is:

`filename.bin`

where *filename.bin* is the name of the ALDSP installation program specific to your platform; for example, `aldsp_320_solaris32.bin` for the Solaris version of ALDSP.

You will find similarly-named installer files for all UNIX and Linux versions of ALDSP.

4. Run the installer application, specifying the properties file and the log file name as options.

- On Windows, run the following command:

```
aldsp_320_win32.exe -mode=silent
-silent_xml=<drive:\properties_file_path>
-log=<drive:\log_file_path>
```

where *drive* is the letter that identifies the hard disk drive, *properties_file_path* is the complete pathname of the ALDSP silent installation properties file, and *log_file_path* is the complete pathname of the log file.

For example, you could enter the following:

```
aldsp_320_win32.exe -mode=silent
-silent_xml=c:\temp\silent.xml
-log=c:\temp\logfile.txt
```

You are returned to the command prompt and the ALDSP installation preparation dialog box is briefly displayed. The installation proceeds in the background with the information specified in the installer properties file.

To verify that the installer is running, open the Windows Task Manager. The installer is listed as the `javaw.exe` process.

Installing ALDSP Using Console or Silent Mode

- On UNIX, run the following command:

```
sh aldsp_320_solaris32.bin -mode=silent  
-silent_xml=complete_properties_file_path  
-log=complete_log_file_path
```

where *complete_properties_file_path* is the complete pathname of the properties file and *complete_log_file_path* is the complete pathname of the log file. A complete path is required, even when the file resides in the same directory as the `aldsp_320_solaris32.bin` file.

The message Preparing to Install is displayed. After the installer decompresses the required files, the installation proceeds with the information specified in the installer properties file. When the process is successful, Installation Complete is displayed.

- On Linux, run the following command:

```
./aldsp_320_linux32.bin -mode=silent  
-silent_xml=complete_properties_file_path  
-log=complete_log_file_path
```

where *complete_properties_file_path* is the complete pathname of the properties file and *complete_log_file_path* is the complete pathname of the log file. A complete path is required, even when the file resides in the same directory as the `aldsp_320_linux32.bin` file.

The message Preparing to Install is displayed. After the installer decompresses the required files, the installation proceeds with the information specified in the installer properties file. When the process is successful, Installation Complete is displayed.

Exploring the Silent Mode Installer Properties File

[Table 3-1](#) describes the required installer properties used with a Silent Mode installation.

Table 3-1 Installer Properties

Sample Property Setting	Description
BEAHOME	<p>The BEA Home directory. This is the complete pathname to the BEA Home directory that will contain this installation.</p> <p>For Windows, specify the absolute path, including the drive.</p> <p>For UNIX, specify the absolute path.</p>
USER_INSTALL_DIR	<p>The WebLogic Server installation directory.</p> <p>For Windows, specify the absolute path, including the drive.</p> <p>For UNIX, specify the absolute path.</p>
ALDSP_INSTALL_DIR	<p>The ALDSP 3.2 installation directory.</p> <p>For Windows, specify the absolute path, including the drive.</p> <p>For UNIX, specify the absolute path.</p>
INSTALL_MERCURY_PROFILING_TOOLS	<p>Optionally install the Mercury profiling tools. Possible values are <code>true</code>, <code>yes</code>, <code>false</code>, and <code>no</code>. The default is <code>true</code>.</p>
INSTALL_NODE_MANAGER_SERVICE	<p>Install Node Manager as a Windows service. The default value is set to <code>no</code>; you need Administrative privileges to set this value to <code>yes</code>.</p>
NODEMGR_PORT	<p>Set the Node Manager listen port number. If you do not specify a port, then the installer uses default port 5556. To set this value the <code>INSTALL_NODE_MANAGER_SERVICE</code> option needs to be set to <code>yes</code>.</p>

Table 3-1 Installer Properties (Continued)

Sample Property Setting	Description
COMPONENT_PATHS	Specify the components and subcomponents to install. To install multiple components, separate the components with a bar (). To install subcomponents, specify a component/subcomponent combination. For example, to install Web Server Plug-Ins, use “WebLogic Server/Web Server Plug-Ins”.
INSTALL_SHORTCUTS_IN_ALL_USERS_FOLDER	<p>The Windows Start menu folder in which you want the Start menu shortcuts created. The user performing the installation must have Administrator privileges to install the shortcuts in the All Users folder.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> • true/yes—The shortcuts are created in the All Users folder (default) • false/no—The shortcuts are created in the local user's folder
USE_EXTERNAL_ECLIPSE	<p>Specify whether to use an existing Eclipse installation, from among the following values:</p> <ul style="list-style-type: none"> • true—Only the Eclipse plug-ins are installed in the Eclipse instance located at EXTERNAL_ECLIPSE_DIR. • false (default)—The complete Eclipse instance is installed in the default location.
EXTERNAL_ECLIPSE_DIR	The directory where the existing external Eclipse instance is installed.
ECLIPSE32_HOME	The home directory where Eclipse should be installed. This setting is for backward compatibility; you should typically use EXTERNAL_ECLIPSE_DIR instead.

The following shows a sample Silent Mode installer properties file (silent.xml):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bea-installer>
  <input-fields>
    <data-value name="BEAHOME" value="C:/bea100"/>
    <data-value name="ALDSP_INSTALL_DIR" value="C:/bea100/aldsp_3.2"/>
    <data-value name="USER_INSTALL_DIR" value="C:/bea100/wlserver_10.0"/>
    <data-value name="COMPONENT_PATHS" value="WebLogic Server|
AquaLogic Data Services Platform/Data Services Server|AquaLogic Data
Services Platform/Eclipse Plug-Ins|AquaLogic Data Services
Platform/Samples"/>
    <data-value name="INSTALL_SHORTCUT_IN_ALL_USERS_FOLDER" value="yes"/>
  </input-fields>
</bea-installer>
```

Installing ALDSP Using Console or Silent Mode

Post-Installation Tasks

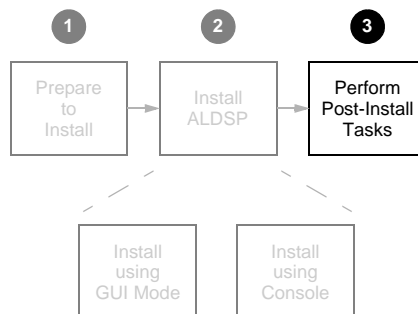
This chapter describes tasks you can do after installation of an instance of ALDSP, including verifying the installation, starting development of applications, and exploring Windows shortcuts and UNIX paths.

The chapter includes the following sections:

- [Verifying the Installation](#)
- [Starting the Development of a Data Integration Solution](#)

Figure 4-1 illustrates the current step in the overall installation process.

Figure 4-1 Post-Installation Task



Verifying the Installation

To verify that the ALDSP 3.2 installation is follow the steps described in “[How To Configure the Retail Dataspace Sample Application for ALDSP 3.2](#)” in the *Data Services Developer’s Guide*.

Starting the Development of a Data Integration Solution

Developing your own data integration solution consists of a design phase and an implementation phase.

- When you are ready to start setting up and configuring your own data sources, refer to the AquaLogic Data Services Platform [Administration Guide](#).
- When you are ready to start mapping source and target XML schemas and constructing queries using WorkSpace Studio, refer to [Data Services Developer’s Guide](#).
- When your application developer is ready to access data services, see the [Client Application Developer’s Guide](#).
- For information on the XQuery engine, functions and the XQSE language see the [XQuery and XQSE Developer’s Guide](#).

Uninstalling ALDSP

For most platforms you can automatically remove system-installed, unmodified ALDSP artifacts from your system. After ALDSP has been uninstalled, you can manually remove any extra files, such as modified samples or other user-created files, which were not removed automatically by the uninstaller.

You should undeploy any ALDSP applications before uninstalling ALDSP from your system. For information about undeploying, see the topic [“Undeploying Deployed Applications”](#) in the WebLogic Server documentation.

Automated Uninstallation

To uninstall ALDSP:

1. Back up any data you want to retain for use with a new installation.
2. Shut down your server.
3. Run the ALDSP uninstaller.
 - On Windows systems, choose the following:

Start > All Programs > BEA Products > Uninstall BEA Products

- On UNIX and Linux systems, run the following command in cases when you installed using GUI mode (as described in [Chapter 2, “Installing ALDSP Using GUI Mode”](#)):

```
<aldsp_home>/uninstall/uninstall.sh
```

Alternatively, run the following command with the `-i` `CONSOLE` option in cases when you installed using Console mode (as described in [Chapter 3, “Installing ALDSP Using Console or Silent Mode”](#))

```
<aldsp_home>/uninstall/uninstall.sh -i CONSOLE
```

4. After the uninstallation, delete the `aldsp_3.2` folder in the `<bea_home>` directory. In addition, you may need to remove some other files manually.

When the uninstall is completed, you may be notified that some files could not be removed. This could be a result of new files that were either generated or user-created after ALDSP was installed.

For example, if you added new target schemas, stored queries, Web services, and so on to the ALDSP repository make sure that you save them elsewhere before proceeding to the next step.

Note: The Uninstall option does not remove any ALDSP dataspace or other WorkSpace Studio applications that you have created if they are created outside the `<aldsp_home>` directory. It is advised that you created your domains and other projects outside `<aldsp_home>` directory.

Upgrading from ALDSP 3.0 or 2.5 to ALDSP 3.2

ALDSP 3.0 applications are created in the Data Services Studio environment and deployed on WebLogic Server 9.2.

ALDSP 2.5 applications are created in the WebLogic Workshop 8.x environment and deployed on WebLogic Server 8.1.

ALDSP 3.2 applications are created within the WorkSpace Studio framework and are deployed on WebLogic Server 10.0 MP1.

While upgrading an ALDSP 2.5 application to ALDSP 3.2, the data service projects within the data service application are upgraded to single a dataspace project. The dataspace project contains all the data services along with the relevant artifacts which need to be deployed on WebLogic 10.0 MP1.

This chapter provides the steps to upgrade ALDSP 2.5 applications to ALDSP 3.2 dataspaces. It includes the following sections:

- [Upgrading from ALDSP 3.0 to 3.2](#)
- [Upgrading from ALDSP 2.5 Prerequisites](#)
- [Upgrading ALDSP 2.5 Artifacts to ALDSP 3.2](#)
- [Upgrading an ALDSP Control from ALDSP 2.5 to 3.2](#)
- [Importing ALDSP 2.5 Configuration to ALDSP 3.2](#)
- [Post-Upgrade Artifact Mapping in ALDSP 3.2](#)

- [ALDSP 2.5 Upgrade: Known Issues and Workarounds](#)

Upgrading from ALDSP 3.0 to 3.2

The upgrade path from ALDSP 3.0 to 3.2 is very simple:

- **Development environment.** To upgrade to the ALDSP 3.2 development environment, uninstall ALDSP 3.0 and install ALDSP 3.2, as described in [Chapter 1, “Preparing to Install ALDSP.”](#) You can then use an existing ALDSP workspace, or create a new one and import the contents of a 3.0-level workspace.
- **Deployment environment.** From the ALDSP 3.2 Workspace Studio, you can build and deploy project created under ALDSP 3.0. Alternatively, you can deploy a JAR file created under ALDSP 3.0 to an ALDSP-enabled WebLogic 10.0 MP1 server.

Upgrading from ALDSP 2.5 Prerequisites

Before you start upgrading an ALDSP 2.5 data service application to ALDSP 3.2, make sure that you perform Build > Clean Application in Workshop 8.1 environment. This ensures that application upgrade is successful and does not require manual clean up of files later.

Upgrading ALDSP 2.5 Artifacts to ALDSP 3.2

ALDSP source upgrade allows you to upgrade ALDSP 2.5 data service projects and bring them into ALDSP 3.2 WorkSpace Studio as a dataspace project. It also upgrades data service related artifacts such as schema, model diagram, .java files, and WSDL files.

Note: To upgrade any ALDSP 2.5 data service applications developed in versions prior to 2.5, you need to first upgrade the application to ALDSP 2.5. For more information about upgrading to ALDSP 2.5, refer to [Upgrading AquaLogic Data Services Platform Application To Version 2.5](#) in the *ALDSP 2.5 Installation Guide*.

This section describes the upgrade of the following artifacts from ALDSP 2.5 to ALDSP 3.2:

- Applications
- Projects
- Schemas
- Java Files within the ALDSP Project
- JAR Data Service Projects

- Shredder Based Artifacts
- Other Artifacts

Notes:

- You cannot use Data Services Studio with Eclipse WTP to upgrade non-data service projects such as Portal and Java web services. To upgrade non-data service projects, refer to the [Upgrading WebLogic Workshop 8.1 Applications](#) chapter in the *Workshop for WebLogic Platform Programmer's Guide*.
- Java projects and Schema projects on which the data service project depends, can be upgraded along with data service projects.

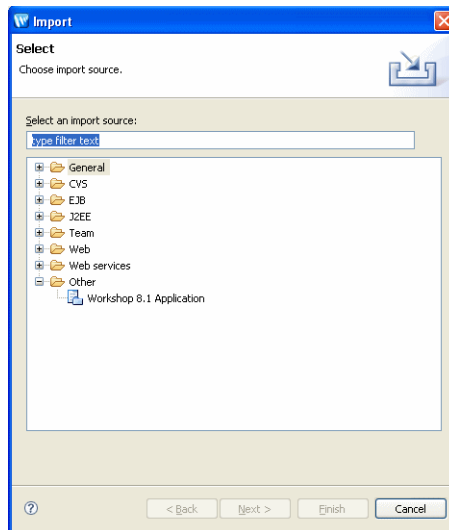
For steps to upgrade an ALDSP Control to version 3.2, refer to [Upgrading an ALDSP Control from ALDSP 2.5 to 3.2](#).

Upgrading an ALDSP 2.5 Application

To upgrade using WorkSpace Studio, do the following:

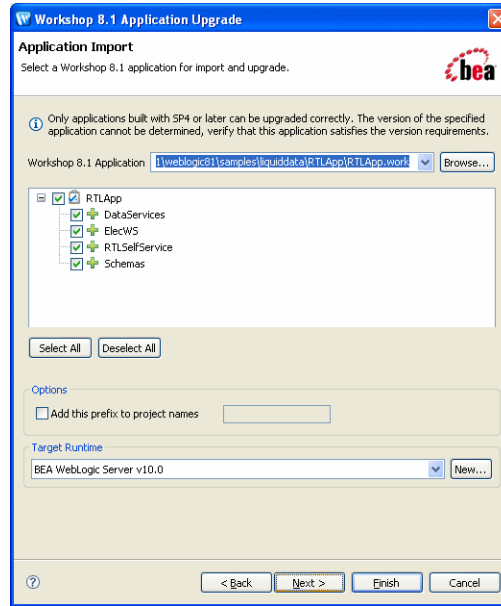
1. Launch WorkSpace Studio.
2. Right-click in the Project Explorer, and choose Import > Import. This displays the Import dialog box, as shown in [Figure 6-1](#).

Figure 6-1 Import Dialog



3. Expand the Other option, select Workshop 8.1 Application, and click Next. The Workshop 8.1 Application Upgrade: Application Import dialog is displayed.
4. Browse and select the ALDSP 2.5 application to be upgraded, as shown in [Figure 6-2](#).

Note: All projects within an ALDSP 2.5 application, along with dependent schemas and Java projects, are upgraded to the same dataspace in ALDSP 3.2. Select all the Data Service project dependents such as Java projects and Schema projects, if any. If you do not want to upgrade all the projects within the application to the same dataspace, then select only the projects that you want to upgrade.

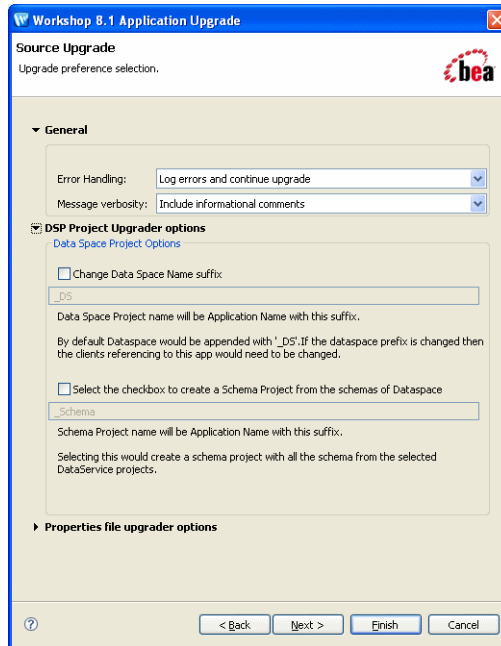
Figure 6-2 Workshop 8.1 Application Upgrade: Application Import Dialog

5. Select the Add this prefix to project names option if you want to specify a prefix for the project being upgraded.

Note: All project files upgraded from ALDSP 2.5 will add the specified prefix, including Java and schema projects created from Data Service projects. The Deployment name property also includes the prefix. However, the ALDSP 2.5 Data Service project folder within the dataspace name will not carry the prefix and therefore, the LD namespace URI does not need to be changed.

6. Ensure that runtime environment is set to BEA WebLogic Server v10.0 for ALDSP 3.2.
7. Click Next. The Workshop 8.1 Application Upgrade: Source Upgrade dialog is displayed as shown in [Figure 6-3](#).

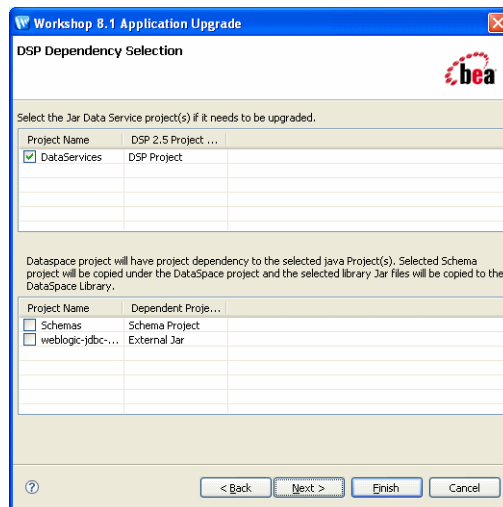
Figure 6-3 Workshop 8.1 Application Upgrade: Source Upgrade Dialog



8. In the Source Upgrade dialog box:
 - a. Select the Error Handling option, which includes:
 - Log errors and continue upgrade
 - Log errors and abort upgrade
 - Display dialog on error
 - b. Select the Message verbosity option from the list. The options include:
 - Include informational comments
 - Include warning comments
 - Include error comments

- c. Select the Change Dataspace Name suffix option to change the name of the dataspace suffix. By default, the dataspace project name will be <application_name>_DS. You can also select a checkbox to create a schema project with the name <Application_Name>_Schema from the selected Data Service projects. For example, you can use this option when the schema or the XMLBeans created are to be used outside of the project.
9. Click Next. This displays the Workshop 8.1 Application Upgrade: DSP Dependency Selection dialog.

Figure 6-4 Workshop 8.1 Application Upgrade: DSP Dependency Selection Dialog

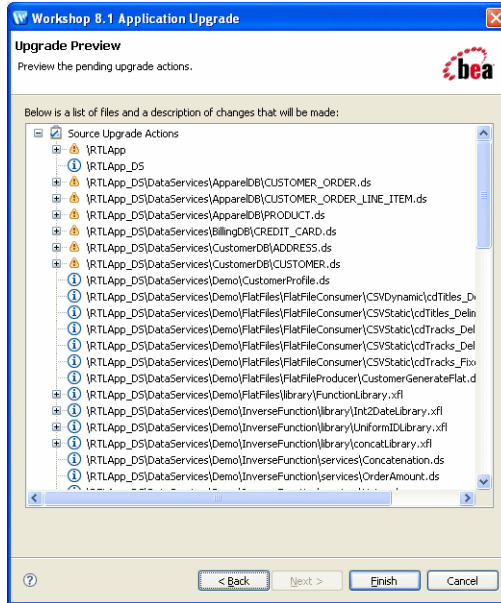


The dialog lists the data service project selected for upgrade along with project JARs including the catalog services project (if available). By default, the data service JAR projects are selected for upgrade. You need to review and deselect them if you do not want them to be upgraded. Because the data service projects are already selected, you cannot edit the selection here.

The dialog also lists the dataspace dependencies such as schemas, Java projects, and library JARs. You need to select the Java projects, schema projects or any library JARs if they are used by one or more data service projects that are brought up. However, internal JAR files such as `ldclient.jar` and `ld-server-app.jar` are not listed.

10. Click Next. This completes the upgrade process and a summary of the upgraded files is displayed in the Upgrade Preview, as shown in [Figure 6-5](#).

Figure 6-5 Application Upgrade: Upgrade Preview Dialog



The Upgrade Preview provides the following details:

- List of the actions performed during the upgrade
- List of files added and deleted
- List of files that do not require an upgrade

11. Click Finish. The imported dataspace shows up in the Project Explorer.

Note: After completing upgrade, an empty EAR project folder is also created. This EAR project is not relevant to the upgraded dataspace and can be deleted if you are not planning to use it at a later stage.

Upgrading an ALDSP Control from ALDSP 2.5 to 3.2

You need to upgrade an ALDSP 2.5 control created in Workshop for WebLogic 8.1 to use it in WorkSpace Studio 1.1. This section describes the steps to upgrade projects using an ALDSP 2.5 LD control (.jcx).

After the upgrade, all the old annotations are also upgraded to the new annotations. The upgraded ALDSP control is then available in WorkSpace Studio 1.1.

Before you begin the upgrade process, you need to install the control plug-in and the control upgrade plug-in:

1. Copy the following control plug-in file:

`<aldsp_home>\eclipse-plugins\workshop9\com.bea.aldsp32.eclipse.plugins.workshop9.link`

or

`<aldsp_home>\eclipse-plugins\workshop102\com.bea.aldsp32.eclipse.plugins.workshop102.link`

where `<aldsp_home>` is the home directory for your ALDSP installation. For example, `C:\bea\aldsp_3.2`.

2. Paste this file to the following location:

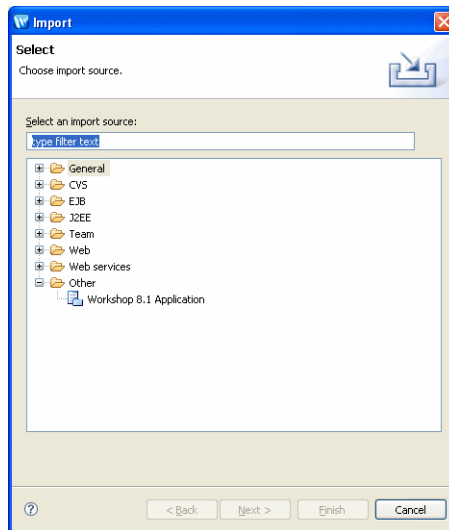
`<bea_home>\tools\eclipse_pkgs\1.1\eclipse_3.2.2\eclipse\links`

Upgrading from ALDSP 3.0 or 2.5 to ALDSP 3.2

To upgrade the control:

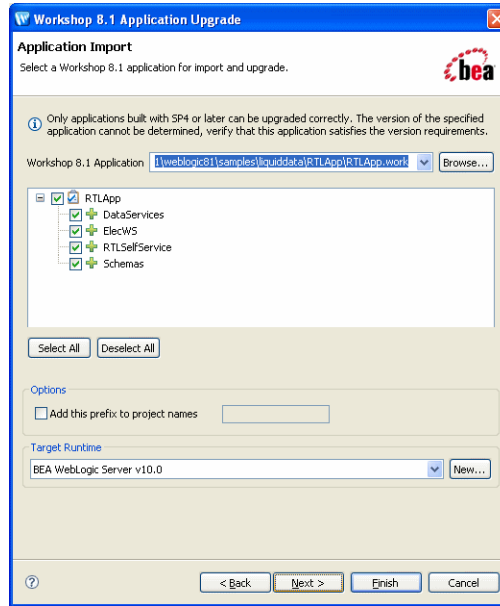
1. Launch WorkSpace Studio.
2. Right-click in the Project Explorer, and choose Import > Import. This displays the Import dialog box, as shown in [Figure 6-6](#).

Figure 6-6 Import Dialog



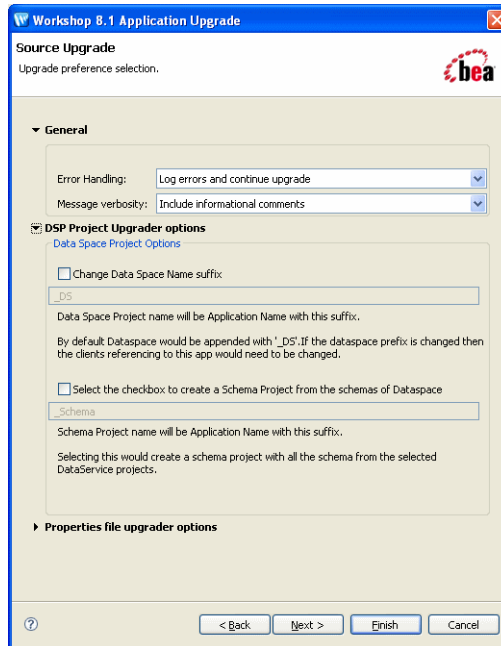
3. Expand the Other option, select Workshop 8.1 Application, and click Next. The Workshop 8.1 Application Upgrade: Application Import dialog is displayed.

Figure 6-7 Workshop 8.1 Application Upgrade: Application Import Dialog



4. In the Application Import dialog:
 - a. Browse and select the application that needs to be upgraded.
 - b. If you want to enter a prefix for projects, select the Add this prefix to project names option and enter the prefix.
 - c. Click Next. The Workshop 8.1 Application Upgrade: Source Upgrade dialog box is displayed as shown in [Figure 6-8](#).

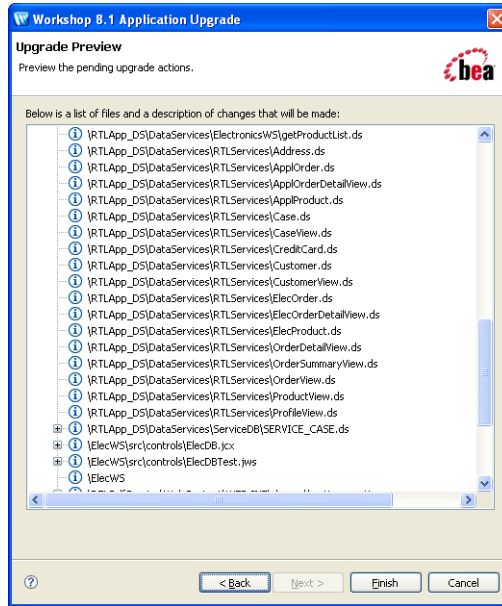
Figure 6-8 Workshop 8.1 Application Upgrade: Source Upgrade Dialog Box



5. In the Source Upgrade dialog box:
 - a. Select the Error Handling option. The options are same as mentioned in [step a](#) of [Upgrading an ALDSP 2.5 Application](#).
 - b. Select the Message verbosity option. The options are same as mentioned in [step b](#) of [Upgrading an ALDSP 2.5 Application](#).
 - c. Click Next. This starts the process of importing configuration for the project folders and report generation of the files being upgraded.

After the report is generated, the Upgrade Preview dialog box is displayed as shown in [Figure 6-9](#).

Figure 6-9 Application Upgrade: Upgrade Preview Dialog



The control, ElecDB.jcx, is listed and the actions to be performed by the upgrade utility on the associated files are mentioned.

6. Click Finish. This completes the upgrade process. The new control in WorkSpace Studio is displayed in the Project Explorer. After the upgrade, the control is converted to a . java file.

Note: If the return type of the control is complex, then make sure that the upgraded control has the associated schema JAR or XMLBeans JAR in the `WebContent\WEB-INF\lib` folder. For details on XMLBeans JAR for the control, refer to [Key Differences for WebLogic Workshop 8.1 Users](#) topic in the WebLogic Workshop documentation.

Importing ALDSP 2.5 Configuration to ALDSP 3.2

After you upgrade an ALDSP 2.5 application to 3.2, you may want to import your existing administration and project configuration to the new ALDSP environment. You can import the ALDSP 2.5 configuration file using the ALDSP Administration Console in ALDSP 3.2. The ALDSP 2.5 configuration file is located at:

```
<domain_dir>\liquiddata\<applicationname>config.xml
```

where <domain_dir> is usually located at <bea_home>\weblogic81\samples\domains\

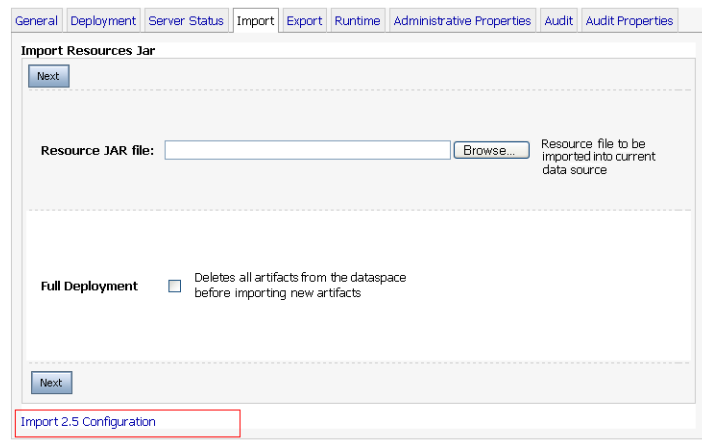
Note: The Allow Default Anonymous Access option from ALDSP 2.5 is not imported into ALDSP 3.2. For more information about setting default anonymous access in ALDSP 3.2, refer to “Allowing Anonymous Access” section in [Securing ALDSP Resources](#).

ALDSP configuration import requires the ALDSP 2.5 data service projects to be upgraded to a ALDSP 3.2 dataspace project and deployed in a ALDSP 3.2 runtime environment.

To import the ALDSP 2.5 configurations:

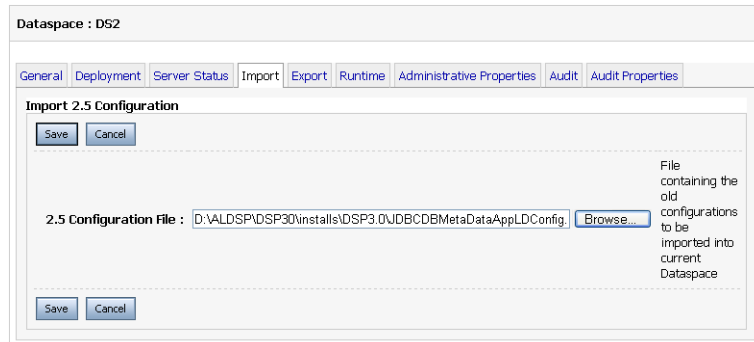
1. Launch ALDSP Administration Console using the following URL and then log in:
`http://localhost:7001/dspconsole`
2. Select the upgraded dataspace project.
3. Select the System Administration category and then click the Import tab.
4. Click Lock & Edit to acquire the lock.
5. Click the Import 2.5 Configuration link as shown in [Figure 6-10](#).

Figure 6-10 Import Tab



6. In the Import 2.5 Configuration section, browse and specify the path where the 2.5 configuration file is located, as shown in [Figure 6-11](#).

Figure 6-11 Importing 2.5 Configuration



7. Click Save > Activate Changes to complete upgrading the configuration file.

Post-Upgrade Artifact Mapping in ALDSP 3.2

This section highlights the results after upgrading from ALDSP 2.5 to 3.2 and provides information about mapping artifacts between ALDSP 2.5 and ALDSP 3.2.

Note: After upgrade, you can continue to use ALDSP 2.5 static clients to communicate with the ALDSP-enabled server. If you want to change the upgraded dataspace to use ALDSP 2.5 clients, then you need to apply the changes to the original ALDSP 2.5 application, generate the 2.5 static clients, and use the clients in ALDSP 3.2.

[Table 6-1](#) provides details about the upgrade and the mappings of artifacts after upgrading to ALDSP 3.2.

Table 6-1 Upgrade Features and Artifact Mapping

ALDSP 2.5 Artifact	ALDSP 3.2 Mapping	Features
Application	Dataspace	The dataspace name is the 2.5 application name with the _DS suffix. For example, RTLApp_DS. You can change the suffix during the upgrade process. For more information, refer to Upgrading an ALDSP 2.5 Application .
Projects	Dataspace projects under the upgraded dataspace.	<ul style="list-style-type: none">Multiple data service projects are upgraded to a single dataspace. The data service project becomes a folder under the dataspace.The project retains the URI for data service and schemas.To allow ALDSP 2.5 mediator APIs access to the upgraded application, the dataspace project uses the application name (EAR project name) for deployment.

Table 6-1 Upgrade Features and Artifact Mapping (Continued)

ALDSP 2.5 Artifact	ALDSP 3.2 Mapping	Features
Project Dependencies	JAR files, schema projects, and Java projects	<ul style="list-style-type: none"> Schema projects that you mark as dependent projects are copied under the dataspace project. For information about marking or selecting project dependencies, refer to step 9 in Upgrading an ALDSP 2.5 Application. Library files marked as dependent are copied to the dataspace project's DSP-INF\lib folder. Dataspace project depends on the created Java projects and dependant Java projects. All the created Java projects and dependant Java projects depends on the schema project, if created.
Other Project Files	index.xml	<ul style="list-style-type: none"> Individual index.xml files are removed from projects within the upgraded dataspace. Instead, a single index.xml file is created for the entire dataspace when you build the project. For different projects within a dataspace, the sdo.xsdconfig.xml file is removed at the root level. The xquery-types.xsd file is removed for individual projects and one file is created for the dataspace. The SQL index file from each of the projects in a dataspace are merged and placed under DSP-INF/sql/sql-index.xml for the dataspace.

Table 6-1 Upgrade Features and Artifact Mapping (Continued)

ALDSP 2.5 Artifact	ALDSP 3.2 Mapping	Features
Java Files	<code><dataservice_ project_name>_ java</code>	<ul style="list-style-type: none"> In ALDSP 3.2, dataspace projects cannot store Java files. Therefore, during upgrade of ALDSP 2.5 data service projects, they are moved under a new Java project for each data service project. This project has the name: <code><dataservice_project_name>_java</code>. For the Java projects that are created from the data service project, the dependency is set at the dataspace level. While building the dataspace project, the dependent JAR or binary files, are copied to the <code>DSP-INF/lib</code> folder.
Schema Files	Same as 2.5	<p>The folder structure and name of the schema remains the same after upgrade.</p> <p>The schema files from ALDSP 2.5 applications must be recompiled with XMLBeans v2 to run them inside the ALDSP 3.2 server environment. You can do this using the XMLBeans Builder facet.</p> <p>If you choose to create a separate schema project, then all the schema files from different selected data service projects are placed in a separate schema project with name <code><dataspace_name>_Schema</code>. For information about selecting separate schema project, refer to step c in Upgrading an ALDSP 2.5 Application.</p>
Catalog Data Service Projects	Exploded as <code>_catalogservices</code> folder under dataspace project	<p>Catalog services JAR under <code>APP-INF/lib</code> in ALDSP 2.5 are upgraded in the exploded form under the dataspace.</p> <ul style="list-style-type: none"> All catalog services are upgraded like other data services. All class files are filtered out and deleted after upgraded and a new <code>catalogservicelib.jar</code> is added under <code>DSP-INF/lib</code> folder.

There are other artifacts such as data services, xfl and Java files, schemas, and WSDL files, which are outlined in [Table 6-2](#).

Table 6-2 Mapping and Features of Other Artifacts Post-Upgrade

Other Files	Features
.ds	<ul style="list-style-type: none"> The .ds files are upgraded as entity data services with the .ds extension. Data service namespaces are not changed. Pragma information that contains the version number changes to include the 3.2 stamp. All read functions are upgraded with <code>visibility=public</code>, <code>kind=retrieve</code> All navigation functions are retained with <code>visibility=public</code>, <code>kind=navigate</code> All procedures (side effecting functions) have <code>visibility=public</code>, <code>kind=library</code>. In addition: <ul style="list-style-type: none"> Function declarations use the following declaration for external functions: <pre>procedure declare procedure for external functions</pre> Function declarations use the following declaration if the function is not external and has a body: <pre>function (declare function)</pre> <p>Note: After the upgrade, mark the <code>kind</code> for side-effecting functions as <code>CUD</code> if they are from the <code>libraryProcedure</code>.</p> <ul style="list-style-type: none"> All functions are marked as <code>primary = false</code> with the new validation rules. A key schema file is created and added, if key element is present in the data service pragma. <p>This key schema is <code>TargetTypeSchemaName_KEY</code>. It shares the same namespace as the target type schema such as XML or other return type schemas.</p>

Table 6-2 Mapping and Features of Other Artifacts Post-Upgrade (Continued)

Other Files	Features
.xfl	<ul style="list-style-type: none"> • These files are changed to library data service files and have no return type pragma. • The namespace prefix are changed from lib to ld and all its references are also modified. • The name of the xfl is retained if there are no DS files with the name in the same folder. If another data service with the same name exists, then it is resolved with numbered suffixes. • All functions are set to <code>visibility=protected</code> during upgrade and in case of private functions, are retained as private. • All functions are set to <code>kind=library</code> and function declaration is <code>function(declare function)</code>
.java files	These files are moved to separate Java projects.
WSDL files	WSDL files are upgraded as is. However, in some cases, you may need to upgrade the WSDL manually.
XML, CSV	No change occurs to these files after upgrade.
Model Diagram	Model diagrams are upgraded as is.

Note: If you have configured the max thread per application property in ALDSP 2.5 using the ALDSP Administration Console, then this configuration is not upgraded to ALDSP 3.2.

Post-Upgrade Tasks for Java Web Service

After completing the upgrade of Java Web Service (JWS), you need to perform the following steps to retrieve values from a dynamic JWS client:

1. Save the WSDL file for the JWS locally and remove the `parameterOrder` attribute from operations tag.
2. Modify the client code to use the saved WSDL to create the service. Examples of the ALDSP 2.5 configuration and the modified version are shown in [Listing 6-1](#) and [Listing 6-2](#).

Listing 6-1 ALDSP 2.5 Client

```
QName serviceName = new QName(targetNamespace, 'testSimplePhyCtrlTest');
URL wsdlLocation = new
URL('http://localhost:7001/testWSProj/controls/testSimplePhyCtrlTest.jws?W
SDL');
Service service = factory.createService(wsdlLocation, serviceName);
```

Listing 6-2 Modified ALDSP 2.5 Client

```
QName serviceName = new QName(targetNamespace, 'testSimplePhyCtrlTest');
URL wsdlLocation = new
URL('file:///D:/dynwsclient/testSimplePhyCtrlTest.wsdl');
Service service = factory.createService(wsdlLocation, serviceName);
```

Note: While modifying the client, ensure that the port name matches the port name of the migrated JWS.

3. Recompile and run the client with the ALDSP 2.5 classpath. This allows you to run the JWS successfully.

ALDSP 2.5 Upgrade: Known Issues and Workarounds

[Table 6-3](#) are some of the known issues after you upgrade various components from ALDSP 2.5 environment to ALDSP 3.2:

Table 6-3 Known Issues and Workarounds

Known Issues	Workaround
An XQueryTypeException occurs when a read function, which is shredded from a custom Java function, is invoked.	<p>The schema (.xsd) file needs to be recompiled using XMLBeans v2, using the following command:</p> <pre>java -classpath \${WL_HOME}/common/lib/javax.xml.stream_ 1.0.0.0.jar;\${WL_HOME}/server/lib/xbean .jar com.bea.xbean.tool.SchemaCompiler *.xsd -d testbin -src testsrc -out <jarfilename>.jar</pre> <p>where <WL_HOME> is root directory, which contains the WebLogic Server 10.0 MP1 installation and <jarfilename> is the name of the schema JAR file.</p>
ALDSP 2.5 update properties are not displayed in the Properties view in WorkSpace Studio. For example, DS pragma elements are not displayed.	No workaround available.
An ALDSP 2.5 Control will work if the SDO is exposed over JWS, however upgraded 2.5 applications will not support exposing SDO over JWS.	No workaround available.

Table 6-3 Known Issues and Workarounds (Continued)

Known Issues	Workaround
The max thread per application property in ALDSP 2.5 is not imported into the ALDSP 3.2 configuration.	<p>You can obtain similar results by configuring the dataspace Work Manager. However, the value is not the same because the pool of thread is shared across async/timeout spawned threads and the top-level EJB threads.</p> <p>In addition, because the Work Manager is self-tuning, it is advised that you do not specify a maximum- or minimum-threads constraint unless you have a specific need to do so.</p> <p>For additional information on the WLS 10.0 Work Manager see:</p> <p>http://edocs.bea.com/wls/docs100/config_wls/self_tuned.html</p>
Schema projects that only exist in JAR format in the Libraries folder are not upgraded to create schemas in the dataspace project.	Obtain the original expanded schema project and include it in the application before starting upgrade.

Upgrading from ALDSP 3.0 or 2.5 to ALDSP 3.2