IONA®

Artix™ ESB

Installation Guide

Version 5.0, July 2007

Making Software Work Together™

While the information in this publication is believed to be accurate, IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Updated: January 9, 2009

Contents

Preface

What is Covered in This Book

This book describes the prerequisites for installing Artix and the procedures for installing Artix on supported systems.

Who Should Read This Book

This guide is intended for all users of Artix.

How to Use This Book

This guide is divided into the following chapters:

- Chapter 1, Installation Prerequisites, which details the supported operating systems, compilers, and required patches.
- Chapter 2, Installing Artix, which provides the steps to install Artix and describes the installation options.
- Chapter 3, Uninstalling Artix, describes how to uninstall Artix.

The Artix Documentation Library

For information on the organization of the Artix library, the document conventions used, and finding additional resources, see Using the Artix Library.

CHAPTER 1

Installation Prerequisites

Before you install Artix, check the system requirements and familiarize yourself with the steps involved in installing the product.

In this chapter

This chapter discusses the following topics:

Before You Begin

Read the release notes

Before installing Artix:

- Visit the IONA Product Documentation web page at:

  http://www.iona.com/support/docs/artix/5.0/index.xml

- Read the *Artix Release Notes* for late-breaking information on new features, known problems, and other release-specific information.

There may also be updates to this *Installation Guide* available at the Web address above.

Save your license file

You will receive your Artix license file by e-mail. When the e-mail arrives, save the attached license file to a safe location on a local or network disk. During installation, the Artix installer prompts for the location of the license file.

Supported Systems and Compilers

Platforms and compilers

Artix 5.0 is supported on Windows, Linux, and UNIX. Table 1 shows the supported platforms and compilers.

For the latest information on supported platforms and compilers, see http://www.iona.com/products/artix/platforms.htm.

Table 1: *Supported Platforms and Compilers*

Operating System	Hardware	C++ Compilers	JDK/JRE [1]
Windows Server 2003	x86_32 [2]	Visual C++ 6.0 SP3 Visual C++ .NET 2003 (7.1)	1.5.0_11 or higher (For C++/JAX-RPC, 1.4.2_13 or higher, or 1.5.0 or higher)
Windows XP	x86_32		
Red Hat Enterprise Linux AS 4.0 (32-bit)	x86_32, x86_64	GCC 3.4.x	1.5.0_10 (32-bit) or higher (For C++/JAX-RPC, 1.4.2_13 or higher, or 1.5.0 or higher)
Red Hat Enterprise Linux AS 4.0 (64-bit)	x86_64		1.5.0_10 (64-bit) or higher
Red Hat Enterprise Linux AS 3.0 (32-bit)	x86_32, x86_64	GCC 3.2.3	1.5.0_10 (32-bit) or higher (For C++/JAX-RPC, 1.4.2_13 or higher, or 1.5.0 or higher)
Red Hat Enterprise Linux AS 3.0 (64-bit)	x86_64		1.5.0_10 (64-bit) or higher
SuSE Linux Enterprise Server 9 (32-bit)	x86_64	GCC 3.3.3	1.5.0_10 (32-bit) or higher (For C++/JAX-RPC, 1.4.2_13 or higher, or 1.5.0 or higher)
SuSE Linux Enterprise Server 9 (64-bit)	x86_64		1.5.0_10 (64-bit) or higher

Table 1: *Supported Platforms and Compilers*

Operating System	Hardware	C++ Compilers	JDK/JRE [1]
Solaris 10 (32-bit)	SPARC	Sun Studio 8, C++ 5.5, Sun Studio 11, C++ 5.8	1.5.0_09 (32-bit) or higher (For C++/JAX-RPC, 1.4.2_13 or higher, or 32-bit 1.5.0 or higher)
Solaris 9 (32-bit)		Sun Studio 8, C++ 5.5	
Solaris 8 (32-bit)		Sun Studio 8, C++ 5.5, Sun Studio 11, C++ 5.8	
Solaris 10 (64-bit)	SPARC	Sun Studio 8, C++ 5.5, Sun Studio 11, C++ 5.8	1.5.0_09 (64-bit) or higher (For C++/JAX-RPC, 1.4.2_13 or higher, or 32-bit 1.5.0 or higher)
Solaris 9 (64-bit)		Sun Studio 8, C++ 5.5	
Solaris 8 (64-bit)		Sun Studio 8, C++ 5.5, Sun Studio 11, C++ 5.8	
AIX 5.3	PowerPC	XL C/C++ 7.0	IBM JDK 1.5.0 or higher (For C++/JAX-RPC, IBM JDK 1.4.2 or higher, or IBM JDK 1.5.0 or higher)
AIX 5.2		Visual Age 6.0.2 (32-bit)	
HP-UX 11i [3]	PA-RISC	aCC 3.56, aCC 3.73	HP JDK 5.0.05 or higher (For C++/JAX-RPC, HP JDK 1.4.2.12 or higher, or HP JDK 5.0.05 or higher)

[1] Artix Designer 5.0 requires a Java 1.5.x JRE to run. Artix 5.0's Java JAX-WS runtime requires a 1.5.x JDK for development and a 1.5.x JRE for running applications. The Artix C++/JAX-RPC runtime for hosting containers and services can run with a 1.4.2 or 32-bit 1.5.x JRE. See "Java JRE and JDK Requirements" on page 14 for more information.

[2] In the Hardware column, X86_32 refers to the 32-bit Pentium architecture, while X86_64 includes the AMD64 (Opteron, Athlon 64), Intel Pentium Extreme Edition 64-bit, and Intel EMT64 (Xeon, Core 2) architectures.

[3] The Artix 5.0 C++/JAX-RPC runtime libraries support applications built with both the HP-UX Standard and Classic C++ runtimes.

Operating System Patch Requirements

Patches

Table 2 shows the operating system patches and runtime components for C++ and Java required to run Artix for the supported platforms.

Note: See also the Operating System/Compiler Patch Policy Web page.

Table 2: *Required OS Patches*

Operating System[1]	Hardware[2]	OS Patches; C++/Java Runtime Environment
Windows Server 2003	x86_32	No patches required.
Windows XP	x86_32	SP2
Red Hat Enterprise Linux Advanced Server 4.0	x86_32	No patches required.
Red Hat Enterprise Linux Advanced Server 4.0	x86_64	
Red Hat Enterprise Linux Advanced Server 3.0	x86_32	GCC 3.2 runtime (`libstdc++.so.5` and `libgcc_s.so[.1]`)
Red Hat Enterprise Linux Advanced Server 3.0	x86_64	No patches required.
SUSE Linux Enterprise Server 9	x86_64	SP1
Solaris 10 (32-bit)	SPARC	
Solaris 9 (32-bit)	SPARC	111685-01 patch
Solaris 8 (32-bit)	SPARC	108827-12; 108434-09 (32-bit C++ runtime); 108827-12 (libthread patch); 111685-01 patch
Solaris 10 (64-bit)	SPARC	
Solaris 9 (64-bit)	SPARC	
Solaris 8 (64-bit)	SPARC	No patches required.

Table 2: *Required OS Patches (Continued)*

Operating System[1]	Hardware[2]	OS Patches; C++/Java Runtime Environment
AIX 5.3	PowerPC	April 2007 IBM C++ Runtime Environment Components for AIX PTF (ref. no. 4015708)
AIX 5.2	PowerPC	Fix for IY57576 January 2006 IBM C++ Runtime Environment Components for AIX PTF (ref. no. 4011532)
HP-UX 11i	PA-RISC	PHSS_24638 (aCC runtime); PHCO_24402 (1.0 libc cumulative header file patch 60); PHCO_25452 (1.0 libc cumulative patch 23632); PHSS_24304 (1.0 ld(1) and linker tools cumulative patch 21234)

[1] In the Operating System column, “32-bit” refers to an installation of the 32-bit version of Linux or Solaris onto 64-bit capable hardware. “64-bit” refers to an installation of the 64-bit version of Linux or Solaris onto 64-bit hardware.

[2] In the Hardware column, X86_32 refers to the 32-bit Pentium architecture, while X86_64 includes both AMD64 (Opteron and Athlon 64) and Intel EMT64 (Xeon and Core 2) architectures.

Java, Compiler, and Artix Designer Requirements

In this section

This section discusses the Artix requirements for Java and C++ compilers, and the requirements to run Artix Designer. This section contains the following topics:

Java JRE and JDK Requirements

Artix Designer Java requirements

Artix Designer 5.0 requires a Java 1.5.x JRE to run.

The Eclipse environment that hosts Artix Designer uses an internal Java compiler, and can generate and compile Java code without the presence of a Java Development Kit (JDK). However, a JDK is still required for other Artix purposes, as described in "Java JDK requirements".

JRE bundled with Artix installer (and alternatives)

A 32-bit 1.5.0.x Java Runtime Environment (JRE) is bundled with Artix for optional installation. The installer also allows you to instead specify the location of a previously installed system JRE or JDK.

Note: Developing programs for the Artix 5.0 Java JAX-WS runtime requires a 1.5.x JDK, not just a JRE. If you intend to use the Java JAX-WS runtime, do not allow the installer to install a JRE. Instead, point the installer toward an existing JDK.

If you decline to install the bundled JRE, then you must specify the location of an existing JRE or JDK during Artix installation. Make sure the one you specify is at the required release level for your operating system, as specified in Table 1 on page 9.

If, during Artix installation, you specify the location of an existing JRE or JDK, you may also need to specify its location in the JAVA_HOME environment variable. See the discussion in "Setting the JAVA_HOME Environment Variable" on page 43.

Java JDK requirements

You must install a JDK to use with Artix in these cases:

- To compile and run the Artix Java sample code from the shell command line or when imported into Artix Designer.
- To compile your own Java code at the shell command line.
- To run certain Artix command-line tools.

A JDK at release level 1.5.x is:

- required when developing for the Java JAX-WS runtime
- strongly recommended when developing Java applications for the C++/JAX-RPC runtime

The C++/JAX-RPC runtime for hosting containers and services can run with a 1.4.2 or 32-bit 1.5.x JRE. If you are developing Java code for use with existing 1.4.2-based Artix services, you can optionally develop with a 1.4.2 JDK.

Specific Java suppliers supported

Artix supports the specific JRE and JDK versions listed in Table 1 on page 9. In particular, Table 1 specifies:

Java supplier	Operating system
Sun Microsystems	Windows, Linux, Solaris
IBM	AIX
HP	HP-UX

Licensing restrictions from Sun Microsystems prevent IONA from including a JDK with Artix installations. You must download and install a JDK from Sun Microsystems, or from the operating system's vendor. You must install a separate JDK even if you allow the Artix installer to install a dedicated JRE for use with Artix Designer.

For more information on Java from Sun, see Sun Microsystems' Java site at http://java.sun.com.

Override default JRE for Red Hat systems

Red Hat Enterprise Linux ships with a GCC-based Java compiler, `gjc`, which is set up by default to provide the system default `java` and `javac` commands. Artix command-line tools and Artix Designer do not support the `gjc` compiler, so you must install a Sun JDK and must take steps to ensure that its `java` and `javac` commands are used by Artix.

The simplest override method is to install a supported Sun JDK in its installer's default location, and then specify the location of the Sun JDK during Artix installation.

You can also manage your Sun JDK installation with one or more of the following methods:

- Specify the Sun JDK's location in a global JAVA_HOME environment variable, as described in "Setting the JAVA_HOME Environment Variable" on page 43.
- Replace the default Java-related symbolic links in `/etc/alternatives`.
- Integrate the Sun JDK into Red Hat's alternatives system, as described in the alternatives(1) man page.

C++ Compiler Requirements

C++ development requirements

If you plan to develop Artix applications in C++, or if you want to compile and run any of the Artix C++ samples, you must have a C++ compiler installed on the target machine. Table 1 on page 9 shows the C++ compilers supported by Artix.

When using Visual C++ with Artix on Windows, the Visual C++ environment must be set before starting Artix Designer, as described in "Setting up for Windows C++ Development" on page 46.

First run of artix_env script

Certain Artix-specific makefile settings are generated and set up the first time you run the `artix_env[.bat]` script, as described in "First Run of artix_env Script for C++ Development" on page 45.

JDK not needed for C++ only

If you will develop only in C++ (or in a language supported by Artix Mainframe), you can install a Java JRE, not JDK, such as the one supplied by the Artix installer. In this case, installing a Java JDK is not required.

Exception: to run the Artix Orchestration BPEL server, you must specify a JDK, not just a JRE.

Artix Designer Requirements

Artix Designer as installed with Artix

The Artix Designer development tool ships as a set of plug-ins for the Eclipse open source development environment. Artix Designer is shipped with the Windows, Linux, and Solaris versions of Artix.

Unlike previous releases, Artix Designer 5.0 requires a Java 1.5.x JRE or JDK to run, as described in "Java JRE and JDK Requirements" on page 14.

The Artix installer installs an Eclipse 3.2.2 environment, the Artix Designer plug-ins, and all necessary supporting plug-ins into the following directory:

```
    ArtixInstallDir\tools\eclipse
```

Solaris and Linux requirements for Artix Designer

To run Artix Designer on Solaris, you must have GTK 2.0 or later installed, as well as the prerequisites of GTK, which are ATK, glib, libgcc (or GCC), libiconv, libintl, and Pango. Install GTK and its prerequisites using the method defined by Sun Microsystems for your version of Solaris.

Running Artix Designer on Linux has the same requirement for GTK 2.0 or later and its prerequisites. For the supported versions of Linux, these subsystems are already installed in the default configuration.

Using Artix Designer in your existing Eclipse

If you have other Eclipse-based tools and you want to add Artix Designer to that environment, you can add the Artix Designer plug-ins to your existing Eclipse installation as described in "Installing Artix Designer into an Existing Eclipse Platform" on page 53.

Disk Space Requirements

Overview

This section lists the amount of permanent and temporary disk space required for different installations of Artix 5.0.

Artix installation disk space

The disk space requirements for Artix depend on the installation options selected. Table 3 shows the approximate disk space in megabytes for full and runtime only installations. These numbers include the bundled JRE optionally installed with Artix.

Table 3: *Disk space used by Artix installations in megabytes*

Installation Type	Windows	Linux	Linux 64	Solaris	AIX	HP-UX
Artix full installation	706	790	798	884	1005	904
Artix C++/JAX-RPC runtime-only installation[1]	273	417	441	527	864	784
Artix Java JAX-WS runtime-only installation[2]	158	176	155	177	155	209

[1] This means selecting everything under the "C++ / JAX-RPC" installation option except "Development libraries and samples," with all other option check boxes blank.

[2] This means selecting everything under the "Java JAX-WS" installation option except "Development libraries and samples," with all other option check boxes blank.

Artix installer disk space

The disk space used by Artix installer package is shown in Table 4. This table also shows the disk space used by the Java runtime environment installed with Artix. The JRE numbers are included in the totals in Table 3.

Table 4: *Disk space used by the Artix installer and JRE in megabytes*

Installation Type	Windows	Linux	Linux 64	Solaris	AIX	HP-UX
Artix installer	487	534	525	551	416	418
Dedicated JRE installed with Artix	71	86	68	90	68	121

Temporary disk space

In addition to the requirements in Table 3 and Table 4, you will need 30 to 50 megabytes of temporary work space for the installer. By default, this work space is the Windows `TEMP` directory or the UNIX `/tmp` directory.

On UNIX, if the required temporary space is not available on `/tmp`, you can specify a different partition for the Artix installer by setting the `IATEMPDIR` environment variable. For example:

```
IATEMPDIR=/local2/tmp
export IATEMPDIR
```

RAM Requirements

RAM requirements for development tools

Artix is a development environment that is used in conjunction with other development tools, such as compilers. As such, the Artix tools do not consume more RAM than the associated toolset.

Check with the vendor of the compiler and JDK for your operating system for their minimum RAM requirements. A typical minimum RAM requirement for compilers and JDKs is 512 MB.

RAM requirements for Artix Designer

Artix Designer is set by default to use a minimum of 128 MB, up to at least 256 MB of RAM. Thus, a practical minimum requirement for running Artix Designer is 512 MB.

RAM used by Artix container and servers

The Artix container for the C++/JAX-RPC runtime, `it_container[.exe]`, is lightweight, and consumes about 21 KB (Windows) or 43 KB (Linux) on first start. Each hosted server adds another few KB (Windows) or 20+ KB (Linux). A complex Artix bus with many containers and services might consume several hundred KB of RAM. Memory consumption for Solaris, AIX, and HP-UX is comparable to the Linux numbers.

Using Artix with Other Products

This section outlines the Artix support for third-party products and protocols. This information helps you plan for running some of the Artix sample code.

This section includes important information on installing Artix on a machine that hosts other IONA products.

Messaging

Artix supports the following messaging product versions:

- IBM WebSphere MQ 5.3
- BEA Tuxedo
    - 6.5 on Windows and HP-UX
    - 8.1 on all supported platforms except AIX
- TIBCO Rendezvous 7.2
- SonicMQ 5.x, 6.x

Transports

Artix supports these transports:

- SOAP 1.1 and 1.2 (MTOM is not supported for SOAP 1.2)
- IIOP 1.1 and 1.2
- HTTP
- RMI

Application servers

The Artix J2EE Connector supports the following application servers:

- JBoss 4.0.1
- BEA WebLogic 8.1 SP3
- IBM WebSphere 5.1

Security

Artix supports the following security products and protocols:

- SiteMinder 4.6.1, 5.5
- Kerberos 5
- LDAP 3.0

Web services

Artix supports these Web services products and protocols:

- Apache Axis 1.3
- jUDDI 0.9rc3

Artix and Microsoft .NET

Artix ships with an assembly that developers can use to build interactions between Artix and Microsoft .NET. The assembly provides a set of helper libraries that facilitate interaction between the Artix session manager and locator services, and an IS2 Kerberos adapter, using Microsoft Active Directory.

The Microsoft environments supported for .NET integration are:

- Development environment: Visual Studio .NET 2003
- Runtime environment: .NET Framework 1.1
- Operating systems: Windows XP or Windows Server 2003

Installing Artix with other IONA products

If you have another IONA product installed on the machine where you are installing Artix 5.0, remember the following:

- Do not install Artix 5.0 under the same directory tree as an existing Artix installation. Either uninstall the existing version, or install Artix 5.0 under a separate directory structure.
- Do not install Artix 5.0 under the same directory tree as any other IONA product.

Note: Unlike previous releases, Artix 5.0 does not support a combined installation with Orbix.

Installing Artix

This chapter describes how to install Artix.

In this chapter

This chapter discusses the following topics:

Running the Artix Installer

Downloading the installation package

The Artix 5.0 installation package is available for download from the IONA Product Download Center at http://www.iona.com/downloads/.

The following installation packages are available:

Table 5: *Artix Installation Packages*

Platform	Use with
Windows	Windows XP and Server 2003
Linux	Supported distributions of 32-bit Linux installed on 32-bit or 64-bit hardware
Linux 64-bit	Supported distributions of 64-bit Linux installed on 64-bit hardware
Solaris	Solaris 10, 9, 8
AIX	AIX 5.3, 5.2
HP-UX	HP-UX 11i

Installation issues

The following are known issues with the installation of Artix 5.0:

- Artix 5.0 cannot be installed in the same directory tree as Artix 1.x or 2.x. IONA recommends that you remove any 1.x or 2.x installations from your system before installing Artix 5.0.
- Artix 5.0 should not be installed in the same directory tree as Artix 3.x or 4.x. Those versions of Artix can remain on your system; install Artix 5.0 in a separate top-level directory.
- Artix 5.0 and Orbix should not be installed into the same directory. In addition. Artix 5.0 libraries and Orbix libraries should not be loaded into the same process-space.
- When installing Artix 5.0 on Windows Server 2003, you must run the installer in Windows XP compatibility mode.

- When installing Artix 5.0 on Windows platforms, do not install into a top-level folder whose pathname contains a space. For example, do not install into `C:\Program Files\IONA`. If you do, the settings of `PATH` and `CLASSPATH` in the `artix_env.bat` file, and the sample build scripts will be incorrect.
- When using the console installation for UNIX systems, only Full and Runtime-only installation options are available.

Installation modes

You can run the Artix installer in three modes, as described in the following topics:

Installing in GUI Mode

Overview

You can run the Artix installer in graphical user interface mode on all supported platforms.

Running the installer

To install Artix in GUI mode:

1. Navigate to the directory into which you extracted the installation package and run the installer:

   Windows

   ```
   artix_platform.exe
   ```

   UNIX

   ```
   ./artix_platform.bin
   ```

2. Follow the onscreen instructions and respond to each prompt.

Use the information in Table 6 as a guide when selecting installation options as the installation proceeds.

Table 6: *Artix Installation Options*

Platform	Installation Option	Default	Notes

All	Top-level directory for your Artix installation	Windows:
		`C:\IONA\artix_50`
		UNIX:
		`/opt/iona/artix_50`
			On Windows, do *not* specify a directory whose pathname contains spaces, For example, do not specify a directory under `C:\Program Files`.
			On UNIX, specify the absolute path to a directory in which your current login name has full read and write permissions. Do not use the ~ abbreviation for home directory.

Windows only	Location for product icons	The **Start | (All) Programs | IONA** menu for all users	You can select only one location. Some of the location options also allow you to select the **Set for all system users** checkbox. The default is to set up the shortcuts for the current user only.

Table 6: *Artix Installation Options (Continued)*

Platform	Installation Option	Default	Notes
All	Full, Custom, or Runtime options	Full	Specify the runtime-only option when deploying an Artix service for testing or production on a system other than your development system. See Table 8 on page 34 for help in deciding which options to install.
All	Install or select a JVM	Install a bundled JRE for use by Artix Designer.	Java JRE and JDK issues are discussed in "Java JRE and JDK Requirements" on page 14. The installer may not identify all JVMs on your system. If you know the exact location of your JRE or JDK, it is faster to navigate to that location than to let the installer search the entire disk.
All	Save installation options?	No	Allows you to save a properties file containing entries for the installation you just completed. This properties file can be used with future automated or silent installations of Artix as described in "Installing in Silent Mode" on page 32.

3. When the installer finishes installing the Artix files, it prompts for the location of your Artix license file. Click **Browse** to locate the license file you saved, as described in "Save your license file" on page 8. The installer copies your license information into the file `ArtixInstallDir\etc\licenses.txt`.

   If you prefer to install the license later, click **Cancel**. For more information see "Installing Artix License Keys" on page 40.

4. Click **Done** to finish the installer.

Installing in Console Mode

Overview

UNIX users can run the Artix installer in console mode if no windowing environment is available.

Running the installer

To run the Artix installer in console mode:

1. Go to the directory into which you extracted the installation package and run the installer as follows:

   ```
   ./artix_platform.bin -i console
   ```

2. Follow the onscreen instructions and respond to option prompts. Use the information in Table 6 on page 28 as a guide when selecting installation options as the installation proceeds.
3. The installer prompts you to specify the options to install using a table like the one in Figure 1. By default, all options are specified, which is the same as a Full installation. To customize your installation, enter a

list of numbers representing the group or feature you want to *exclude* from the installation. Use Table 8 on page 34 as a guide when deciding which group or feature options to install.

Example 1: *Installation options menu in console installation mode*

```
 1- [X] C++ / JAX-RPC
 2-  |-[X] Runtime
 3-  |-[X] Development libraries and samples
 4-  |-[X] Artix Locator
 5-  |-[X] Security Framework
 6-  |-[X] Artix Router
 7-  |-[X] Transaction Manager
 8-  |-[X] High Availability
 9-  |-[X] Session Manager
10- [X] Java JAX-WS
11-  |-[X] Runtime
12-  |-[X] Development libraries and samples
13-  |-[X] Artix Locator
14-  |-[X] Security Framework
15-  |-[X] Artix Router
16- [X] Mainframe
17- [X] Development Tools
18-  |-[X] Artix Designer
19-  |-[X] Command-line Tools
20- [X] Third-party Integrations
21-  |-[X] EMS Integration
22-  |-[X] Amberpoint
23-  |-[X] CA-WSDM
```

WARNING: Console installation only works on UNIX systems. Using `-i console` when installing on Windows simply runs a silent installation with default options.

Installing in Silent Mode

Overview

Silent installations are installations that run without user intervention. Their advantage is that they allow you to automate the process of installing Artix on more than one machine.

In an interactive installation, the installer receives necessary user input in response to questions posed in a GUI or console. In a silent installation, you must provide the same information in a properties file.

Creating the properties file

First, create a properties file to contain the response values for the silent installation. You can use any name for your properties file and invoke it with the -f option when running the installer. Or you can use the reserved file name installer.properties, which is automatically used by the installer.

The easiest way to create a properties file is to go through the steps of an Artix installation, then save the properties of that installation to a file when so prompted at the end of the installation. You can then edit the saved properties file to adjust the way you want your silent installation to proceed.

You can also create a properties file with any text editor.

Contents of properties file

The properties file must contain entries for the variables listed in Table 7:

Table 7: *Properties File Variables*

Variable	Description
JDK_HOME	The path to the root of a JDK or JRE installation. If this variable is set, the installation uses the JDK or JRE specified. If unset, the installation installs a dedicated JRE.
SILENT_ACCEPT_LICENSE_AGREEMENT	Set to true to accept the Artix license agreement
USER_INSTALL_DIR	Absolute path to the directory where Artix will be installed on the user's machine
INSTALLER_UI	Set to silent for a silent installation
USER_INPUT_SAVE_PROPERTIES_YES_NO	Set to No for a silent installation

Table 7: *Properties File Variables (Continued)*

Variable	Description
`CHOSEN_INSTALL_FEATURE_LIST`	This entry must be one long string containing a comma-separated list of feature codes, with no spaces between entries. The valid feature codes for this variable, shown in Table 8, specify the Artix components you want to install. To specify a Full installation, you must list all group components in Table 8.
`ECLIPSE_SUPPORT`	Set to **`true`** to install Eclipse and the Artix Designer plug-ins[a]
`MAINFRAME_SUPPORT`	Set to `true` to install the Artix Mainframe component[b]

a. You must also add the `Designer` attribute in the `CHOSEN_INSTALL_FEATURE_LIST` variable to install Artix Designer.

b. You must also add the `Mainfra` attribute in the `CHOSEN_INSTALL_FEATURE_LIST` variable to install Artix Mainframe.

The valid values for the `CHOSEN_INSTALL_FEATURE_LIST` variable are shown in Table 8.

Note: The codes in Table 8 are either group codes or individual feature codes. If you specify a group code, you specify all features in that group.

Table 8: *CHOSEN_INSTALL_FEATURE_LIST feature codes*

Feature Code	Group	Feature	Description
CXXCore	C++ / JAX-RPC		Specify options in the C++ / JAX-RPC category to develop Java services compatible with previous releases of Artix, or to develop all C++ services, and all client applications for Artix Mainframe services.
CXXRT		Runtime	The infrastructure based on the C++ / JAX-RPC runtime that allows services and service consumers to interact in a distributed environment. This includes a bus, which handles the delivery of messages between different middleware systems; support for Artix containers; and support for the transports and payload formats supported by this version of Artix.
CXXDev		Development libraries and samples	The standard foundation classes, XML Schema-based type system, WSDL API, and sample code that allow you to build Web service applications or to service-enable existing applications, based on the C++ / JAX-RPC runtime.
CXXLocator		Artix Locator	An Artix service that allows clients to locate registered services independent of their deployed location.
CXXSecurity		Security Framework	The IONA Security Framework, which includes support for the WS-Security SOAP header format; support for single sign-on and mutual authentication; the IONA Security Service, which provides role-based access control and authentication; and plug-ins to support File Adapter, Netegrity, and LDAP.

Table 8: *CHOSEN_INSTALL_FEATURE_LIST feature codes (Continued)*

Feature Code	Group	Feature	Description
`CXXRouter`		Artix Router	An Artix intermediary service that redirects messages based on rules defined in the router's contract. An Artix router can be used as a bridge between different communication protocols.
`CXXTX`		Transaction Manager	An Artix plug-in that supports interoperability with a CORBA OTS transaction system.
`CXXHA`		High Availability	Support for service replication, which allows services to remain operational despite hardware or communication failures.
`CXXSM`		Session Manager	A group of Artix plug-ins that work together to control the number of clients that can access a group of services concurrently. The session manager can be used to ensure a given instance is used by only one client at a time, which is useful for service-enabling single-threaded applications.

Table 8: *CHOSEN_INSTALL_FEATURE_LIST feature codes (Continued)*

Feature Code	Group	Feature	Description
`JavaCore`	Java JAX-WS		Choose options in the Java JAX-WS category to support newly developed Java applications based on the JAX-WS compliant runtime introduced in Artix 5.0.
`JavaRT`		Runtime	The infrastructure based on the Java JAX-WS runtime that allows services and service consumers to interact in a distributed environment. This includes a bus, which handles the delivery of messages between different middleware systems; support for containers; and support for the transports and payload formats supported by this version of Artix.
`JavaDev`		Development libraries and samples	The classes, XML Schema-based type system, WSDL API, and sample code that allow you to build Web service applications or to service-enable existing applications, based on the Java JAX-WS runtime.
`JavaLocator`		Artix Locator	An Artix service that allows clients to locate registered services independent of their deployed location.
`JavaSecurity`		Security Framework	The IONA Security Framework, which includes support for the WS-Security SOAP header format; support for single sign-on and mutual authentication; the IONA Security Service, which provides role-based access control and authentication; and plug-ins to support File Adapter, Netegrity, and LDAP.
`JavaRouter`		Artix Router	An Artix intermediary service that redirects messages based on rules defined in the router's contract. An Artix router can be used as a bridge between different communication protocols.
`Mainfra`[a]	Mainframe		Tool and Artix Designer support for creating clients that communicate with Artix Mainframe services.

Table 8: *CHOSEN_INSTALL_FEATURE_LIST feature codes (Continued)*

Feature Code	Group	Feature	Description
`Develop`	Development Tools		Tools and utilities for developers of Artix applications, including command-line tools and the Eclipse-based tool suite, Artix Designer (for supported operating systems).
`Designer`[b]		Artix Designer	A suite of GUI tools for creating and editing WSDL contracts, and for generating code to implement the consumer and server sides of the WSDL contract. Artix Designer is integrated into the Eclipse development environment.
`Command`		Command-line Tools	Tools used at the shell prompt to generate code from WSDL contracts, and to generate WSDL from code.
`Third-p`	Third-party Integrations		Support for integrating Artix with Enterprise Management Systems from several vendors. (Support for SNMP traps is already included in the base installation.)
`EMS`		EMS Integration	Support for the following separately licensed components: Operational logging; integration with IBM Tivoli and BMC Patrol.
`Amber`		AmberPoint Integration	Support for integrating Artix with the AmberPoint SOA Management System.
`CAWSDM`		CA-WSDM Integration	Support for integrating Artix with the Computer Associates Web Services Distributed Management system (CA WSDM).

a. You must also set the `MAINFRAME_SUPPORT` variable to `true` to install the Artix Mainframe component.

b. You must also set the `ECLIPSE_SUPPORT` variable to `true` to install Artix Designer.

Example properties file

An example of a properties file is shown below:

```
JDK_HOME=
SET_PATH=
SILENT_ACCEPT_LICENSE_AGREEMENT=true
USER_INSTALL_DIR=C\:\\IONA\\artix_5.0
CHOSEN_INSTALL_FEATURE_LIST=CXXCore,JavaCore,Designer,Command,CX
   XLocator,CXXSecurity,CXXHA,CXXRouter,CXXTX,CXXSM,EMS,Amber,CA
   WSDM,Mainfra,Develop,Third-p,CXXRT,CXXDev,JavaRT,JavaDev,Java
   Locator,JavaSecurity,JavaRouter
USER_INPUT_SAVE_PROPERTIES_YES_NO=No
INSTALLER_UI=silent
```

Note: When including directory paths in the properties file, you can represent path separators in the format `$/$`. This is read by the Artix installer as the correct path separator independent of operating system convention. For example: `C:$/$IONA`

If you instead use backslashes in a properties file targeted for Windows systems, you must escape the backslashes by doubling them, and escape the colon in drive letters with a backslash. For example: `C\:\\IONA`

Running the installer

To run the Artix installer in silent mode:

1. Save the properties file to the directory into which you extracted the installation package.
2. From the same directory, run the Artix installer with its `-f` option:

   Windows

   ```
   artix_platform.exe -f your_properties_file
   ```

   UNIX

   ```
   ./artix_platform.bin -f your_properties_file
   ```

   As an alternative, if you used the reserved file name `installer.properties`, you do not need to use the `-f` option:

   Windows

   ```
   artix_platform.exe
   ```

UNIX

```
./artix_platform.bin
```

When the Artix installation is complete, you need to install the Artix license file. For more information see “Installing Artix License Keys” on page 40.

Uninstalling a Silent Installation

After performing a silent installation, the next uninstallation also runs silently.

Note: When running a silent uninstallation in Windows, the Add/Remove Control Panel’s dialog box may appear to be hung. In fact, the silent uninstallation is proceeding silently. Control is returned to the dialog box when the uninstallation completes.

Installing Artix License Keys

Overview

Before you can begin using Artix, you must install a valid product license. The license is a text file containing keys for the individual components that you have purchased. A 30-day evaluation installation also requires a license file.

Typically, you receive your Artix license from IONA by e-mail. Save it to a disk location accessible from the machine on which you are installing Artix. Then install your licenses in one of the following ways:

- Automatically, from within the Artix installer. See "Installing in GUI Mode" on page 28.
- By running the License Installer script (See below)
- By manually copying the license file to the default location. See "Installing the license file manually" on page 41.

Running the License Installer

If you did not install your license keys during Artix installation, you can use the license installer script:

To install a license using the license installer:

1. Run the license installer as follows:

   Windows

   From the Windows **Start** menu, select **(All) Programs|IONA|Artix 5.0|License Installer**.

   UNIX

   Run the following script:

   ```
   ArtixInstallDir/artix_version/cxx_java/bin/
      license_installer
   ```

2. In the **Install Artix Licenses** dialog box, click the **Browse** button.
3. Browse to the directory where you saved your license file.
4. Select the license file, then click **Select**.
5. The license file is added to the default license location. Click **OK** to close the license installer.

Installing the license file manually

You can install your license manually by copying the license file to the default location:

```
ArtixInstallDir/etc
```

If you want to save the license file to an alternative location on a local disk, you must set the `IT_LICENSE_FILE` environment variable to point to the alternate location.

Windows

```
set IT_LICENSE_FILE=path\license_filename.txt
```

UNIX

```
export IT_LICENSE_FILE=path/license_filename.txt
```

Post-Installation Settings and Tasks

In this section

This section discusses environment and configuration settings that are required for command-line development and for certain other cases. This section contains the following topics:

Setting the JAVA_HOME Environment Variable

Windows and JAVA_HOME

Set the JAVA_HOME environment variable before running Artix Designer, or before running the `artix_env.bat` or `artix_java_env.bat` scripts to set up your command-line development environment.

If you opted to have the bundled JRE installed by the Artix installer, then the JAVA_HOME variable in the artix_env.bat file points to the bundled JRE, not to a JDK. You must set the global environment variable JAVA_HOME to point to a your JDK installation.

To set the JAVA_HOME environment variable globally for your Windows system, use the **System** Control Panel, **Advanced** tab, **Environment Variables** button. It is not enough to set the variable at the Windows command prompt. Use the 8.3 version of space-containing directory names.

For example:

```
JAVA_HOME=C:\Progra~1\Java\jdk1.5.0_11
```

You can use the `dir /x` command at the Windows command prompt to determine the 8.3 version of file and directory names.

Note: The JRE and JDK installers from Sun Microsystems do *not* set the JAVA_HOME environment variable.

UNIX/Linux and JAVA_HOME

The Artix installer sets a value for the JAVA_HOME environment variable near the top of the `artix_env` and `artix_java_env` environment-setting scripts. The path value set is either the path to the bundled JRE, or the path to the alternate JRE or JDK you specified to the installer. Any setting of JAVA_HOME in the shell’s global environment takes precedence over the setting in the `artix_env` script.

Red Hat Linux systems ship with a non-Sun Java JRE and JDK based on GCC `gcj`. To avoid using the Red Hat default `java` and `javac` commands, you must specify the path to a Sun JRE or JDK during Artix installation, or you must override the path set by the installer by using the JAVA_HOME environment variable. See “Override default JRE for Red Hat systems” on page 16 for further information.

Setting Up the Artix Command-line Environment

Setting Artix development environment for C++/JAX-RPC runtime

Before running command-line development tools for the C++/JAX-RPC runtime, and before running any Artix container, service, or service consumer, you must set up the command-line environment. To do so, use the following commands:

Windows

```
> cd ArtixInstallDir\artix_version\cxx_java\bin
> artix_env
```

UNIX

```
% cd ArtixInstallDir/artix_version/cxx_java/bin
% . ./artix_env
```

This script sets up several Artix-specific environment variables, appends the Artix `bin` directory to the system search path, and appends the Artix shared library directory to the shared library path.

Note: You do NOT need to run `artix_env` before starting Artix Designer. Artix Designer now sets up its own environment. However, see "First Run of artix_env Script for C++ Development" on page 45 for an exception.

Setting Artix development environment for Java JAX-WS runtime

There is a similar environment-setting script for using tools for the Java JAX-WS runtime.

Windows

```
> cd ArtixInstallDir\artix_version\java\bin
> artix_java_env
```

UNIX

```
% cd ArtixInstallDir/artix_version/java/bin
% . ./artix_java_env
```

First Run of artix_env Script for C++ Development

First run of artix_env script

Certain Artix-specific makefile settings are generated and set up the first time you run the `artix_env[.bat]` script.

Thus, even though it is not required to run `artix_env[.bat]` before starting Artix Designer, you must run `artix_env[.bat]` at least one time before Artix C++ development can proceed. This first run can be in a shell window that you close immediately afterward; that is, there is no need to start Artix Designer from the same shell prompt.

Setting up for Windows C++ Development

Windows C++ environment

Whether using Artix command-line tools or Artix Designer, you must set up Visual C++ environment variables and paths before running any Artix tools. There are two cases:

1. You allowed the Visual C++ installer to configure the global Windows environment.

   In this case, you are ready for Artix development with Visual C++ with no further configuration:

   i. For Artix command-line development, run `artix_env` at the command prompt.

   ii. Start Artix Designer from the icon placed in the Start menu.

2. You did not allow the Visual C++ installer to configure the global Windows environment (for example, if your PC has more than one C++ development environment).

   In this case, you must set the paths and environment variables for Visual C++ from a batch file, `vcvars32.bat`, provided by the Visual C++ installer:

   i. For Artix command-line development, run `vcvars32.bat` at the command prompt, and then run `artix_env.bat`.

   ii. For Artix Designer, run `vcvars32.bat` at the command prompt, then start Artix Designer from the same prompt by specifying the path to `eclipse.exe`.

   iii. As an alternative, you can set up a start script for Artix Designer, as described in the next section.

Start Script for Artix Designer

If your Visual C++ environment settings are not set in the global Windows environment, then you may find it convenient to create a start script for Artix Designer. Your start script replaces the Artix Designer icon set up by the Artix installer. Your start script should:

- Source the `vcvars32.bat` file.
- cd to the directory containing `eclipse.exe`.
- Start Artix Designer by invoking `eclipse.exe`.

Example 2 shows an example start script for Visual C++ 6.0:

Example 2: *Start script for Artix Designer in Windows with Visual C++*

```
@echo off
setlocal
call "C:\Program Files\Microsoft Visual Studio\vc98\bin\vcvars32.bat"
cd /d C:\IONA\artix_5.0\tools\eclipse
start .\eclipse.exe
endlocal
```

The following example start script is for Visual C++ .NET 2003:

```
@echo off
setlocal
call "C:\Program Files\Microsoft Visual Studio .NET 2003\Common7\Tools\vsvars32.bat"
cd /d C:\IONA\artix_5.0\tools\eclipse
start .\eclipse.exe
endlocal
```

If you installed Visual C++ in a non-default location, then adjust the `call` line as appropriate for your machine.

Setting the environment for Visual C++ 6.0

The default Artix for Windows installation presumes the compiler in use is Visual C++ 7.1. If you are using Visual C++ 6.0 as your compiler, you must run a one-time setup command to configure the runtime environment.

To set the environment to use Visual C++ 6.0, open a new command prompt session (that is, one in which you have not already run the `artix_env` script) and run the following:

```
> cd ArtixInstallDir\artix_5.0\cxx_java\bin
> artix_env -compiler vc60
```

Note: You only need to use the `-compiler` switch one time to specify your compiler version. Once the compiler version is set, you can run the `artix_env` script normally, without the switch.

Resetting the environment for Visual C++ 7.1

To reset the Artix runtime environment for Visual C++ .NET 2003 (7.1), run the following from a new command prompt:

```
> cd ArtixInstallDir\artix_5.0\cxx_java\bin
> artix_env -compiler vc71
```

Artix Designer Start Scripts and Log Files

Overview

Unlike previous releases, Artix Designer 5.0 does not start from a `start_eclipse` script or batch file.

For UNIX and Linux, start Artix Designer directly by running the Eclipse executable. For example:

```
/opt/iona/artix_5.0/tools/eclipse/eclipse &
```

For 64-bit UNIX and Linux, you can also run a 64-bit version of Eclipse, as in this example:

```
/opt/iona/artix_5.0/tools/eclipse_64/eclipse &
```

For Windows, start Artix Designer from its icon in the Start menu: **Start | (All) Programs | IONA | Artix 5.0 | Artix Designer**.

Exception: depending on how your Visual C++ environment variables are set, you may need to start Artix Designer with a batch file as described in "Setting up for Windows C++ Development" on page 46.

Log files

Artix Designer writes a log file named `ArtixDesigner.log` to the current directory when the Eclipse executable is started. The log file is only created in the event of a program fault or error. Artix Designer also writes a housekeeping file, `derby.log` to the current directory.

You may wish to control where these files are written by controlling the current directory at the time you start Artix Designer. In UNIX and Linux, you might create a start script like this example:

```
cd ~/logs
~/iona/artix_5.0/tools/eclipse/eclipse &
```

For Windows, the Artix Designer start menu icon makes the current directory the `eclipse` directory before starting `eclipse.exe`. For example:

```
    c:\IONA\artix_5.0\tools\eclipse
```

If you use a startup batch file for Visual C++ like Example 2 on page 47, be sure to include a line that changes to this directory before starting Eclipse, to keep the log files in the same place.

Artix Designer Workspace Considerations

Avoid spaces in path to workspace

When you first start Artix Designer, you are prompted for the location of a workspace directory to contain your project files.

The default workspace in Linux and UNIX systems is:

```
~/workspace
```

For example:

```
/home/login-name/workspace
```

For Windows, the default workspace is:

```
C:\Documents and Settings\login-name\workspace
```

It is best to override the Windows default to avoid the spaces in the pathname. You can specify a pathname with no spaces in the path, or specify the 8.3 version of space-containing directory components. For example, either of the following examples are good choices:

```
C:\EclipseWS\workspace
C:\Docume~1\login-name\workspace
```

You can use the `dir /x` command at the Windows command prompt to determine the 8.3 version of file and directory names.

LocalRepository directory created in workspace

When Artix Designer 5.0 creates the Eclipse workspace directory, it also creates a sub-directory named `LocalRepository`. This directory contains data used internally by Artix Designer but does not include any project-specific data. A new `LocalRepository` directory is created for you when you switch workspaces.

Setting up to Build Imported Samples in Artix Designer

Import Samples feature within Artix Designer

Artix Designer includes an **Import Samples** feature in the **Artix Designer** menu in Eclipse. The Artix samples were designed to run from the command line with Ant scripts and makefiles. The Import Samples feature lets you run a limited set of Artix samples from the Eclipse environment.

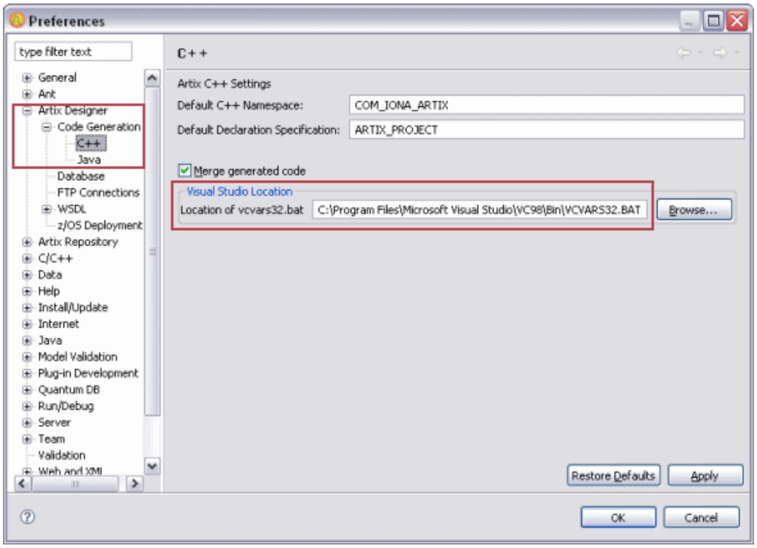
Setup step for Windows

To use the C++ samples with the Import Samples feature in Windows, you must set the path to your Visual C++ `vcvars32.bat` file in the **Eclipse Preferences** dialog.

> **Note:** The `vcvars32.bat` path setting for the **Import Samples** feature is distinct and separate from any environment settings you make to enable Visual C++ development in Artix as a whole. See "Setting up for Windows C++ Development" on page 46 for those instructions.
>
> The setting described in this section is *only* to enable the **Import Samples** feature for C++.

1. In Eclipse, invoke **Window | Preferences**.
2. Click the **Artix Designer** item in the left side menu.
3. Open the plus next to **Artix Designer** and navigate to **Code Generation | C++**.
4. In the **Visual Studio location** field, fill in the path to your `vcvars32.bat`, as appropriate for your version of Visual C++.

5. Click OK.



Artix C++ Settings

Default C++ Namespace: COM_IONA_ARTIX

Default Declaration Specification: ARTIX_PROJECT

☑ Merge generated code

Visual Studio Location

Location of vcvars32.bat C:\Program Files\Microsoft Visual Studio\VC98\Bin\VCVARS32.BAT

Installing Artix Designer into an Existing Eclipse Platform

Overview

By default, the Artix installer installs a new Eclipse framework, including the Artix Designer plug-ins, on your machine. However, you may want to use Artix Designer with an existing Eclipse platform.

Note: Artix Designer 5.0 must be used with Eclipse 3.2.2.

Eclipse prerequisites

To install and use the Artix plug-ins in your own instance of Eclipse, you must have:

- A licensed installation of Artix 5.0 on the same machine
- Eclipse 3.2.2, with the following features installed:
    - Java Development Tools (JDT) 3.2.1
    - C/C++ Development Tools (CDT) 3.1.1, if you plan to develop with C++
    - Graphical Modeling Framework (GMF) 1.0.2
    - Eclipse Modeling Framework (EMF) 2.2.1
    - EMF Service Data Objects (SDO) runtime 2.2.1
    - XML Schema Infoset Model (XSD) 2.2.1
    - Java EMF Model (JEM) 1.2.1
    - Graphical Editor Framework (GEF) runtime 3.2
    - Data Tools Platform (DTP) 1.0
    - Web Standard Tools (WST) runtime 1.5.2
    - SOA Tools Platform (STP) 0.6

Use Eclipse's **Help** | **About Eclipse SDK** | **Feature Details** button to confirm that you have the necessary Eclipse prerequisites.

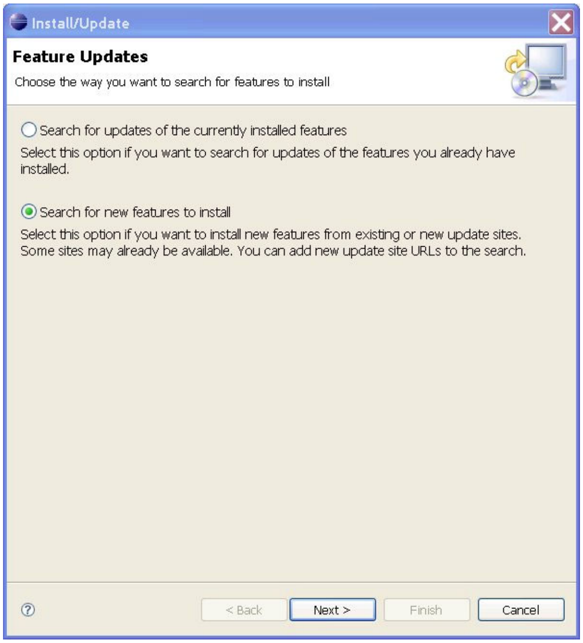
Using the Eclipse update mechanism

Use the Eclipse update mechanism to download and install the Artix plug-ins. This method ensures that the Artix plug-ins you use are the most up-to-date versions.

To add the Artix plug-ins to Eclipse, follow these steps:

1. In Eclipse, select **Help|Software Updates|Find and Install**. The Install/Update wizard launches.

Figure 1: *Feature Updates Panel*



Feature Updates

Choose the way you want to search for features to install

○ Search for updates of the currently installed features

Select this option if you want to search for updates of the features you already have installed.

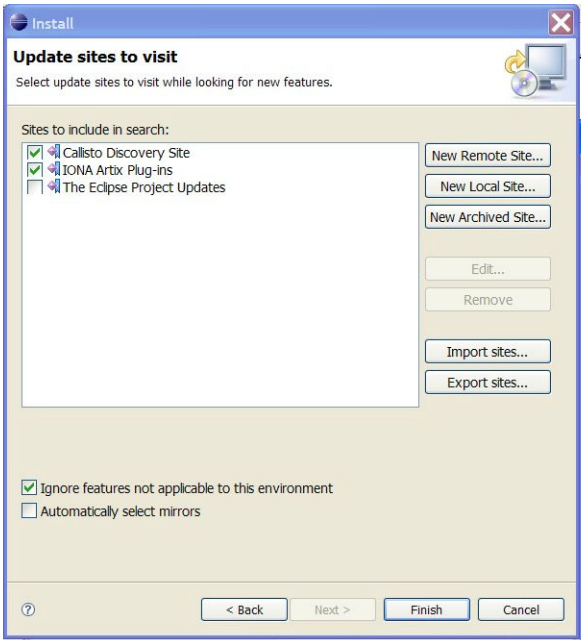
● Search for new features to install

Select this option if you want to install new features from existing or new update sites. Some sites may already be available. You can add new update site URLs to the search.

2. In the **Feature Updates** panel, select **Search for new features to install,** then click **Next**.
3. In the **Update Sites to Visit** panel, click the **New Remote Site** button.
4. Enter the following details in the **New Update Site** dialog box:
   - Name: `IONA Artix Plug-ins`
   - URL: `http://www.iona.com/downloads/artix/eclipse/5.0`
5. Click **OK**.

6. Select the Eclipse **Callisto Discovery Site** and **IONA Artix Plug-ins** check boxes, and unselect all other boxes in the **Sites to Include** section.
7. Check the **Ignore Features** checkbox and click **Finish**.

Figure 2: *Update Sites to Visit Panel*



Install

Update sites to visit

Select update sites to visit while looking for new features.

Sites to include in search:

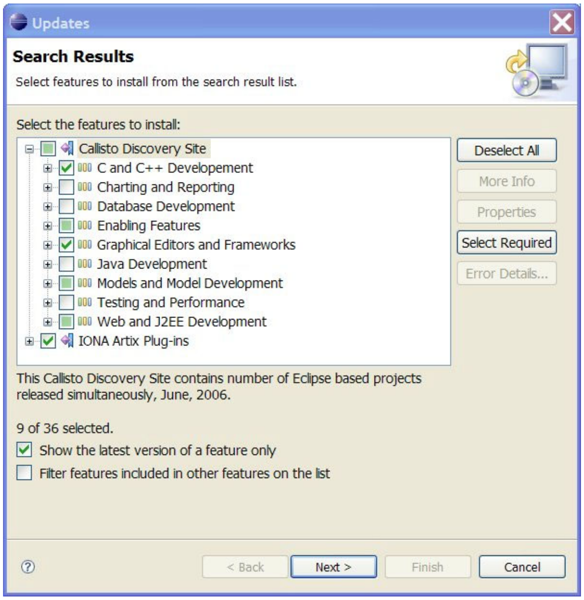
☑ Callisto Discovery Site
☑ IONA Artix Plug-ins
☐ The Eclipse Project Updates

☑ Ignore features not applicable to this environment
☐ Automatically select mirrors

8. Eclipse contacts the specified sites and returns with a list of available Eclipse plug-ins at those sites.
9. In the **Search Results** panel, select the check box beside the **IONA Artix Plug-ins** node.

10. Expand the **Callisto Discovery Site** node, and then click **Select Required**. This automatically selects any required Eclipse 3.2.2 plug-ins.

Figure 3: *Search Results Panel*



Updates

Search Results

Select features to install from the search result list.

Select the features to install:

- ▣ Callisto Discovery Site
  - ☑ C and C++ Developement
  - □ Charting and Reporting
  - □ Database Development
  - ▣ Enabling Features
  - ☑ Graphical Editors and Frameworks
  - □ Java Development
  - ▣ Models and Model Development
  - □ Testing and Performance
  - ▣ Web and J2EE Development
- ☑ IONA Artix Plug-ins

This Callisto Discovery Site contains number of Eclipse based projects released simultaneously, June, 2006.

9 of 36 selected.

☑ Show the latest version of a feature only

□ Filter features included in other features on the list

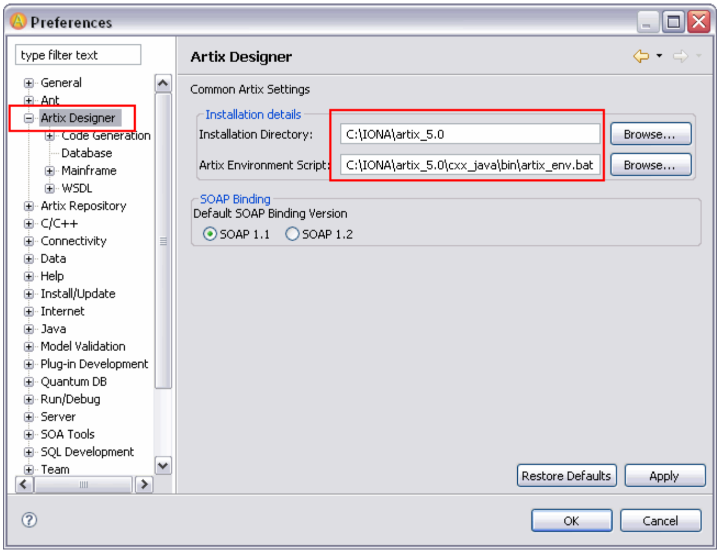
11. Click **Next**.
12. Accept the license agreement and click **Next**.
13. In the **Installation** panel, confirm the target installation location, and click **Finish** to begin the installation.

Setup with Artix Designer

When the Artix Designer plug-ins are downloaded and installed, you must set up the paths to your Artix installation in the Eclipse **Preferences** dialog:

1. In Eclipse, invoke **Window | Preferences**.
2. Click the **Artix Designer** item in the left side menu.
3. Fill in the path to your Artix 5.0 top-level installation directory and the path to the `artix_env[.bat]` script.
4. Click **OK**.

Figure 4: *Common Artix Settings in Eclipse Preferences*



Installation Directory: C:\IONA\artix_5.0

Artix Environment Script: C:\IONA\artix_5.0\cxx_java\bin\artix_env.bat

Default SOAP Binding Version

● SOAP 1.1 ○ SOAP 1.2

Running self-installed Artix Designer

To run the Artix Designer plug-ins in your own copy of Eclipse, you must have a licensed installation of Artix 5.0 on the same machine. The Artix libraries and sample files are used from the Artix installation.

You can have two or more instances of Eclipse on the same machine without conflict. There is no need to remove the Artix-installed instance of Eclipse if you prefer to use Artix Designer integrated in your own instance of Eclipse.

When you use Artix Designer in your own Eclipse instance, the same features are present as in the versions installed by the Artix installer. This includes the Artix perspectives, the Artix Designer menu, and the Artix additions to the help system, tutorials, and cheat sheets.

CHAPTER 3

Uninstalling Artix

This chapter describes how to uninstall Artix.

In this chapter

This chapter contains the following sections:

Uninstalling Artix Designer

Uninstalling from the Artix Eclipse platform

If you are running Artix Designer from the Eclipse platform that was installed along with Artix, Eclipse is removed when you uninstall Artix.

Uninstalling from an existing Eclipse platform

If you added the Artix Designer plug-ins to an existing Eclipse installation, you must delete the plug-ins manually to uninstall them.

To delete the Artix Designer plug-ins that you manually installed into Eclipse:

1. Shut down Eclipse.
2. Go to your `*EclipseInstallDir*/plugins` directory.
3. Delete all the plug-in folders whose names begin with `com.iona.bus`.
4. Restart Eclipse.

Uninstalling on Windows

Uninstalling Artix

To uninstall Artix from Windows:

1. From the Windows **Start** menu, select **(All) Programs|IONA|Artix 5.0|Uninstall IONA Artix 5.0**.
2. In the resulting dialog, click **Uninstall**.

As an alternative, you can run the following from a command prompt:

```
ArtixInstallDir\artix_version\uninstall\Uninstall_artix_version.exe
```

Note: Remember that after a silent installation, the next uninstallation is also run silently.

Uninstalling on UNIX

Uninstalling Artix

To uninstall Artix on UNIX, run the following script:

```
ArtixInstallDir/artix_version/uninstall/Uninstall_artix_version
```

Note: Remember that after a silent installation, the next uninstallation is also run silently.

Index