Artix™

Installation Guide

Version 4.2, March 2007

Making Software Work Together™

IONA Technologies PLC and/or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this publication. Except as expressly provided in any written license agreement from IONA Technologies PLC, the furnishing of this publication does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Any rights not expressly granted herein are reserved.

IONA, IONA Technologies, the IONA logos, Orbix, Artix, Making Software Work Together, Adaptive Runtime Technology, Orbacus, IONA University, and IONA XMLBus are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries.

Java and J2EE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that appear herein are the property of their respective owners.

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.

COPYRIGHT NOTICE

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of IONA Technologies PLC. No third-party intellectual property right liability is assumed with respect to the use of the information contained herein. IONA Technologies PLC assumes no responsibility for errors or omissions contained in this publication. This publication and features described herein are subject to change without notice.

Copyright © 1999-2007 IONA Technologies PLC. All rights reserved.

All products or services mentioned in this publication are covered by the trademarks, service marks, or product names as designated by the companies that market those products.

Updated: April 10, 2007

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/4.2/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 patches

Artix 4.2 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 (1.4.2_13 or 1.5.0 for Artix runtime)
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) (1.4.2_13 or 1.5.0 for Artix runtime)
Red Hat Enterprise Linux AS 4.0 (64-bit)	x86_64		1.5.0_10 (32-bit and 64-bit) [3] (1.4.2_13 or 1.5.0 for Artix runtime)
Red Hat Enterprise Linux AS 3.0 (32-bit)	x86_32, x86_64	GCC 3.2.3	1.5.0_10 (32-bit) (1.4.2_13 or 1.5.0 for Artix runtime)
Red Hat Enterprise Linux AS 3.0 (64-bit)	x86_64		1.5.0_10 (32-bit and 64-bit) [3] (1.4.2_13 or 1.5.0 for Artix runtime)
SUSE Linux Enterprise Server 9 (32-bit)	x86_64	GCC 3.3.3	1.5.0_10 (32-bit) (1.4.2_13 or 1.5.0 for Artix runtime)
SUSE Linux Enterprise Server 9 (64-bit)	x86_64		1.5.0_10 (32-bit and 64-bit) [3] (1.4.2_13 or 1.5.0 for Artix runtime)
Solaris 10 (32-bit)	SPARC	Sun Studio 8, C++ 5.5, Sun Studio 11, C++ 5.8	1.5.0_09 (32-bit) (1.4.2_13 or 32-bit 1.5.0 for Artix runtime)
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	

Table 1: *Supported Platforms and Compilers*

Operating System	Hardware	C++ Compilers	JDK/JRE [1]
Solaris 10 (64-bit)	SPARC	Sun Studio 8, C++ 5.5, Sun Studio 11, C++ 5.8	1.5.0_09 (64-bit) (1.4.2_13 or 32-bit 1.5.0 for Artix runtime)
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	XLC 7.0	IBM JDK 1.5.0 (IBM JDK 1.4.2 or 1.5.0 for Artix runtime)
AIX 5.2		Visual Age 6.0.2 (32-bit)	
HP-UX 11i [4]	PA-RISC	aCC 3.56, aCC 3.73	HP JDK 5.0.05 (HP JDK 1.4.2.12 or 5.0.05 for Artix runtime)

1. Artix Designer 4.2 requires a Java 1.5.x JRE to run, but the Artix Java 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 both AMD64 (Opteron, Athlon 64) and Intel EMT64 (Xeon, Core 2) architectures.
3. When running 64-bit Linux on x86_64 64-bit hardware, you must have both 32-bit and 64-bit Sun JDKs installed and configured as described in "Setting up for 64-bit Linux on 64-bit Hardware" on page 51.
4. Starting with release 4.2, Artix runtime libraries support applications built with both the HP-UX Standard and Classic C++ runtimes.

Operating System Patch Requirements

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

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.
AIX 5.3	PowerPC	
AIX 5.2	PowerPC	Fix for IY57576

Table 2: *Required OS Patches (Continued)*

Operating System[1]	Hardware[2]	OS Patches; C++/Java Runtime Environment
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

Java JRE requirements for Artix Designer

Starting with Artix release 4.2, Artix Designer 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 requirement".

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 Artix installer offers to install this JRE for exclusive use by Artix, or allows you to specify the use of a previously installed system JRE or 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 40.

Note: The Artix installer offers to install a 32-bit JRE even when installing on 64-bit platforms. In most cases, this is not a conflict, but there is a special case described in "Setting up for 64-bit Linux on 64-bit Hardware" on page 51.

Java JDK requirement

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

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

A 1.5.0 JDK is recommended for command-line development.

The Artix Java 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 at the shell command line.

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 version 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 40.
- 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.

See also the special case settings described in "Setting up for 64-bit Linux on 64-bit Hardware" on page 51.

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++ demos, 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 44.

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 43.

JDK not needed for C++ only

If you will develop only in C++ (or in a language supported by Artix for z/OS), 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: in the case of 64-bit Linux running on x86_64 64-bit hardware, you must install two JDKs, even if you will only develop in C++. See "Setting up for 64-bit Linux on 64-bit Hardware" on page 51.

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.

Starting with Artix release 4.2, Artix Designer 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 3.2.1 environment, the Artix Designer plug-ins, and all necessary supporting plug-ins into the following directory:

```
ArtixInstallDir\artix\version\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 52.

Disk Space Requirements

Overview

This section lists the amount of permanent and temporary disk space required for different installations of Artix 4.2.

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, minimum, and runtime only installations. These entries include a dedicated JRE installed with Artix.

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

Installation Type	Windows	Linux	Solaris	AIX	HP-UX
Artix full installation	664	874	821	905	863
Artix custom installation with minimum options selected	400	523	538	742	801
Artix runtime only installation	261	431	506	808	726

Artix installer disk space

The temporary disk space used after unpacking the 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 unpacked Artix installer and JRE*

Installation Type	Windows	Linux	Solaris	AIX	HP-UX
Unpacked Artix installer files	442	469	432	317	291
Dedicated JRE installed with Artix	71	89	91	81	123

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, `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 demos and examples.

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 4.2, remember the following:

- Do not install Artix 4.2 under the same directory tree as an existing Artix installation. Either uninstall the existing version, or install Artix 4.2 under a separate directory structure.
- Do not install Artix 4.2 under the same directory tree as any other IONA product, except Orbix 6.3.1.
- If you are installing Artix 4.2 on the same machine as Orbix 6.3.1, first read "Installing Artix with Orbix" on page 57.

CHAPTER 2

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 4.2 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	Installation Package	Use with
Windows	`42WinXP.zip`	Windows XP and Server 2003
Linux	`42RHAS3.tar`	Red Hat Enterprise Linux AS 3.0, SuSE Linux Enterprise Server 9.0
Linux GCC 3.4	`42RHAS4.tar`	Red Hat Enterprise Linux AS 4.0
Solaris	`42Sun.tar`	Solaris 10, 9, 8
AIX	`42AIX52.tar`	AIX 5.3, 5.2
HP-UX	`42HP11i.tar`	HP-UX 11i

Download the package for your platform and extract its contents to a temporary directory on your hard drive.

Two installation packages for Linux

Starting with Artix 4.2, there are two Artix download kits for Linux, each appropriate for different supported Linux platforms.

GCC 3.4 introduced a change in the C++ Application Binary Interface (ABI), which is the set of runtime interfaces that determines binary compatibility. Because the ABI changed, applications built with GCC 3.4 or later cannot be mixed with libraries built with an earlier GCC version. Since Red Hat Enterprise Linux 4.0 now ships with GCC 3.4, it was necessary to build a separate GCC 3.4-only version of Artix for that release.

Installation issues

The following are known issues with the installation of Artix 4.2:

- Artix 4.2 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 4.2.
- When installing Artix 4.2 on Windows Server 2003, you must run the installer in Windows XP compatibility mode.
- When installing Artix 4.2 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 demo 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.exe
   ```

   UNIX

   ```
   ./artix.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` UNIX: `/opt/iona`	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. **Note:** If other IONA products are already installed on your machine, refer to "Installing Artix with other IONA products" on page 23.

Table 6: *Artix installation options*

Platform	Installation Option	Default	Notes
Windows only	Location for program shortcuts	The **Start | (All) Programs | IONA** menu for all users	You can select only one location. Some of the location options also allow you to check the "Set for all system users" checkbox. The default is to set up the shortcuts for the current user only.
All	Development versus runtime only	Development	Specify the runtime-only option when deploying an Artix service for testing or production on a system other than your development system.
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. **Note:** If you are running 64-bit Linux on 64-bit hardware, do not allow the Artix installer to install its 32-bit JRE. Instead, select your locally installed 64-bit JDK, and set the environment variables discussed in "Setting up for 64-bit Linux on 64-bit Hardware" on page 51.
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 37.
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.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.

Note: When using the console installation for UNIX systems, only Full and Runtime-only installation options are available.

WARNING: Console installation is only for UNIX systems. Do not use `-i console` when installing for Windows.

The Windows installer inadvertently run with `-i console` behaves like a silent installation with default options. In this case, the contents of `C:\IONA`, if any, are silently overwritten with a full installation of Artix.

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
`USER_INSTALL_DIR`	The directory where Artix will be installed on the user's machine
`SILENT_ACCEPT_LICENSE_AGREEMENT`	Set to `true` to accept the Artix license agreement.
`JAVA_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.
`INSTALLER_UI`	Set to `silent` for a silent installation

Table 7: *Properties File Variables (Continued)*

Variable	Description
USER_INPUT_SAVE_PROPERTIES_YES_NO	Set to No for a silent installation.
USER_INPUT_INSTALL_TYPE	Takes one of the following values: • Full Installation • Custom Installation • Runtime Installation
OPTIONAL_COMPONENT_LIST	Entries for this variable are only used when USER_INPUT_INSTALL_TYPE is set to Custom Installation. This variable is ignored otherwise. If used, this entry must be one long string containing a comma-separated list of values, with no spaces between entries. The valid values for this variable are shown in Table 8.

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

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

Table 8: *Valid values for OPTIONAL_COMPONENT_LIST*

Value	Description
artix.development.optional.eclipse	Installs an instance of the Eclipse framework with Artix plug-ins that enable Artix Designer.

Table 8: *Valid values for OPTIONAL_COMPONENT_LIST*

Value	Description
`artix.development.optional.ha`	Installs support for high availability service replication, which allows services to remain operational despite hardware or communication failures.
`artix.development.optional.locator`	Installs the Artix locator service, which allows clients to locate registered services independent of their deployed location.
`artix.development.optional.management`	Installs support for integrating Artix with Enterprise Management Systems from several vendors.
`artix.development.optional.mq`	Installs support for interoperability with WebSphere MQ message queues.
`artix.development.optional.routing`	Installs the Artix router service, which can be used as a bridge between different communication protocols.
`artix.development.optional.security`	Installs the IONA Security Framework, which includes: • Support for the WS-Security SOAP header format • Support for single sign on and mutual authentication • IONA Security Service (role based access control and authentication) • Plug-ins to support File Adapter, Netegrity, LDAP
`artix.development.optional.sm`	Installs the Artix session manager, which can be used to control the number of clients that can access a group of services concurrently.
`artix.development.optional.tibrv`	Installs support for interoperability with the TIBCO Rendezvous messaging transport.

Table 8: *Valid values for OPTIONAL_COMPONENT_LIST*

Value	Description
`artix.development.optional.tm`	Installs the Artix transaction manager, which supports interoperation with a CORBA OTS transaction system.
`artix.development.optional.tuxedo`	Installs support for interoperability with BEA Tuxedo middleware.

Example properties file

An example of a properties file is shown below:

```
USER_INSTALL_DIR=C:$/$IONA
USER_INPUT_INSTALL_TYPE=Custom Installation
OPTIONAL_COMPONENT_LIST=artix.development.optional.eclipse,artix
   .development.optional.locator,artix.development.optional.secu
   rity,artix.development.optional.ha,artix.development.optional
   .routing,artix.development.optional.tm,artix.development.opti
   onal.sm
JAVA_HOME=C\:\\Progra~1\\Java\\j2sdk1.5.0_11
SILENT_ACCEPT_LICENSE_AGREEMENT=true
USER_INPUT_SAVE_PROPERTIES_YES_NO=No
INSTALLER_UI=silent
```

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.exe -f your_properties_file
   ```

   UNIX

   ```
   ./artix.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.exe
```

UNIX

```
./artix.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 37.

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 38
- By appending the Artix license to an existing IONA product license (See "Merging Artix and Orbix licenses" on page 58)

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 4.2|License Installer**.

   UNIX

   Run the following script:

   ```
   ArtixInstallDir/artix/4.2/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
```

WARNING: If you have other licensed IONA products installed, setting the `IT_LICENSE_FILE` variable may cause your existing products to stop working. See "Merging Artix and Orbix licenses" on page 58.

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

You may need to set the JAVA_HOME environment variable before running Artix Designer, or before running the `artix_env.bat` script to set up your command-line development environment. The cases are as follows:

- If you opted to have the bundled JRE installed by the Artix installer, you do not need to set JAVA_HOME. In this case, the path to the bundled JRE is written in the `artix_env` script. The default path for Windows is `C:\IONA\artix\jre`.
- If, during Artix installation, you specified the path to an alternate JRE or JDK elsewhere on your system, and that pathname is free of spaces, then you do not need to set JAVA_HOME.
- If you specified an alternate JRE or JDK path during Artix installation, and the path to your JRE or JDK contains a space, then set JAVA_HOME.

The Artix installer writes the path to the JRE/JDK at the top of the `artix_env.bat` environment-setting script. The script checks whether JAVA_HOME is already set in the global environment, and uses that value instead.

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 long 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` environment-setting script. 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 15 for further information.

Special case for 64-bit Linux

For the case of 64-bit Linux installed on 64-bit x86_64 hardware, you must set three JAVA_HOME-related environment variables. See "Setting up for 64-bit Linux on 64-bit Hardware" on page 51.

Setting Up the Artix Command-line Environment

Setting the Artix runtime and development environment

Before you running Artix command-line development tools, 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\4.2\bin
> artix_env
```

UNIX

```
% cd ArtixInstallDir/artix/4.2/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: Starting with Artix 4.2, 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 43 for an exception.

Verifying the command-line environment

To verify that the Artix environment is correctly set up, run the following commands from a command prompt in which you have run `artix_env`:

Windows

```
cd %IT_ARTIX_BASE_DIR%
```

UNIX

```
cd $IT_ARTIX_BASE_DIR
```

Your current directory should change to the directory where you installed Artix.

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 enviroment (for example, if your PC has more than one 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 1 shows an example start script for Visual C++ 6.0:

Example 1: *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\4.2\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\4.2\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++ 7.1

The default Artix for Windows installation presumes the compiler in use is Visual C++ 6.0. If you are using Visual C++ .NET 2003 (7.1) as your compiler, you must run a one-time setup command to configure the runtime environment.

To set the environment to use Visual C++ 7.1, 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\4.2\bin
> artix_env -compiler vc71
```

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++ 6.0

To reset the Artix runtime environment for Visual C++ 6.0, run the following from a new command prompt:

```
> cd ArtixInstallDir\artix\4.2\bin
> artix_env -compiler vc60
```

Artix Designer Start Scripts and Log Files

Overview

Starting with Artix 4.2, Artix Designer is no longer started from a `start_eclipse` script or batch file.

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

```
~/iona/artix/4.2/eclipse/eclipse &
```

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

```
~/iona/artix/4.2/eclipse_64/eclipse &
```

For Windows, start Artix Designer from its icon in the Start menu: **Start | (All) Programs | IONA | Artix 4.2 | 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 44.

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 severe 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/4.2/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\4.2\eclipse
```

If you use a startup batch file for Visual C++ like Example 1 on page 45, 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 long file and directory names.

LocalRepository directory created in workspace

Starting with release 4.2, Artix Designer creates a directory named `LocalRepository` in your Eclipse workspace. This directory is created the first time you use the Artix Code Generation or Project Creation tools, and contains data used internally by Artix Designer.

If you change workspaces, a new `LocalRepository` directory is created when you use the tools listed above. No project-specific data is persisted in this directory, so there is no need for that same copy of the directory to follow Artix Designer in two or more workspaces. Artix Designer automatically creates a new copy of the directory when needed.

Setting up to Build Imported Demos in Artix Designer

Import Demos feature within Artix Designer

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

Setup step for Windows

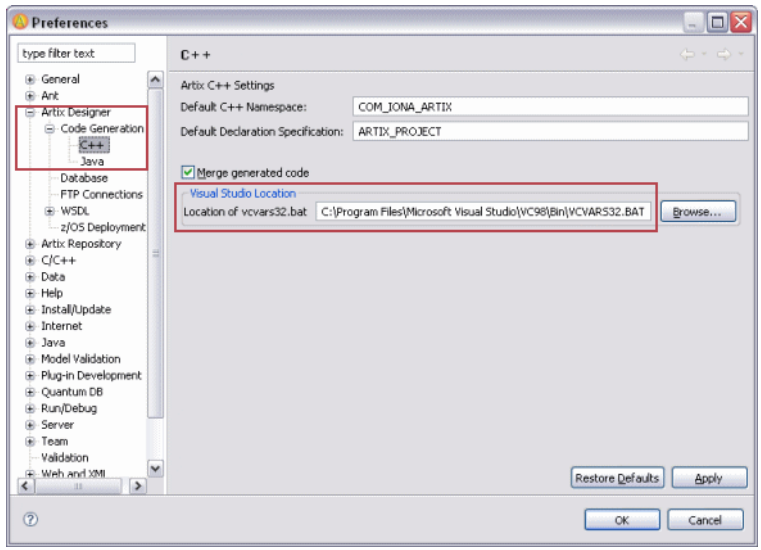
To use the C++ demos with the Import Demos feature in Windows, you must set the path to your Visual C++ `vcvars32.bat` file in the **Eclipse Preferences** dialog.

Note: The `vcvars32.bat` path setting for the **Import Demos** 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 44 for those instructions.

The setting described in this section is *only* to enable the **Import Demos** 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.


Preferences
type filter text
General
Ant
Artix Designer
Code Generation
C++
Java
Database
FTP Connections
WSDL
z/OS Deployment
Artix Repository
C/C++
Data
Help
Install/Update
Internet
Java
Model Validation
Plug-in Development
Quantum DB
Run/Debug
Server
Team
Validation
Web and XML
C++
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
Browse...
Restore Defaults
Apply
OK
Cancel

Setting up for 64-bit Linux on 64-bit Hardware

Overview

When running 64-bit Linux on x86_64 64-bit hardware, you must have both 32-bit and 64-bit Sun JDKs installed, and they must be identified with environment variables as described below.

This restriction is determined only by 64-bit Linux on 64-bit hardware. A 32-bit version of Linux follows 32-bit rules, even when installed on x86_64 hardware. This restriction applies equally to Java and C++ development, and equally to the development of 32-bit and 64-bit targets.

For a 64-bit JDK, use a 1.5.0.x version; for a 32-bit JDK, you can use 1.4.2 or 1.5.0. Sun JDK 1.6.x and non-Sun JDKs are not supported.

Environment variable settings

Set the JAVA_HOME_64 environment variable equal to the path to your 64-bit JDK's directory.

Set the JAVA_HOME_32 variable equal to the path to your 32-bit JDK's directory.

Then set JAVA_HOME to one of your JDKs, depending on whether you want a 32-bit or 64-bit runtime.

Setting the artix_env -bits switch

If you set JAVA_HOME to point to a 64-bit JDK, then there is one further setting. Run the `artix_env` script with the `-bits 64` argument, as follows:

```
> cd ArtixInstallDir/artix/4.2/bin
> ./artix_env -bits 64
```

Note: You only need to use the `-bits` switch one time to specify 64-bit development. Once the bit level is set, you can run the `artix_env` script without the switch to configure your shell environment.

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, onto your machine. However, you may want to use Artix Designer with an existing Eclipse platform.

Note: Artix Designer 4.2 must be used with Eclipse 3.2.

Eclipse prerequisites

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

- Eclipse 3.2.
- The Java Development Tools (JDT) plug-in.
- The C/C++ Development Tools (CDT) plug-in, if you plan to develop with C++.
- Eclipse Modeling Framework (EMF).
- EMF Service Data Objects (SDO).
- XML Schema Infoset Model (XSD).
- A licensed installation of Artix 4.2 on the same machine.

For each Eclipse plug-in listed above, no particular version number is required. Use the latest version appropriate for your Eclipse version.

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

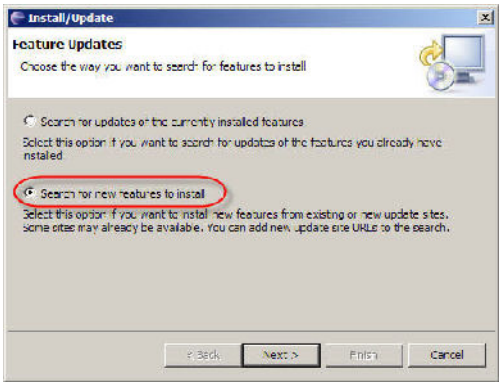
Using the Eclipse update mechanism

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:

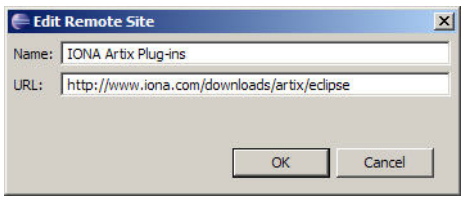
1. In Eclipse, select **Help|Software Updates|Find and Install**. The Install/Update wizard launches.

Figure 1: *The Feature Updates Panel of the Install/Update Wizard*



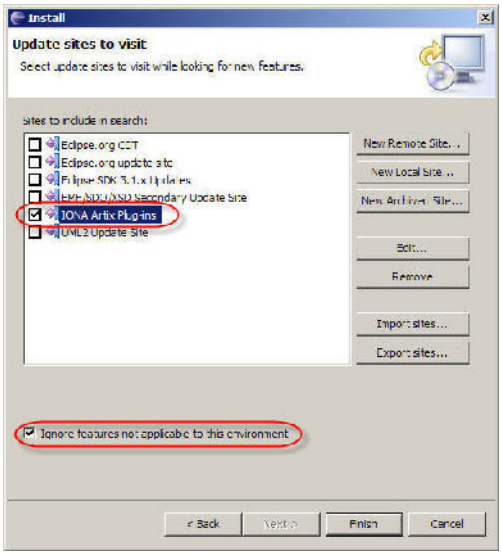
2. In the Feature Updates panel, select **Search for new features to install,** then click **Next**.
3. In the **Update Sites to Visit** panel, click the **New Remote Site** button.
4. Enter the following details in the New Update Site dialog box:
   - Name: **`IONA Artix Plug-ins`**
   - URL: **`http://www.iona.com/downloads/artix/eclipse`**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

Figure 2: *Edit Remote Site dialog*



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

Figure 3: *The Update Sites to Visit Panel*



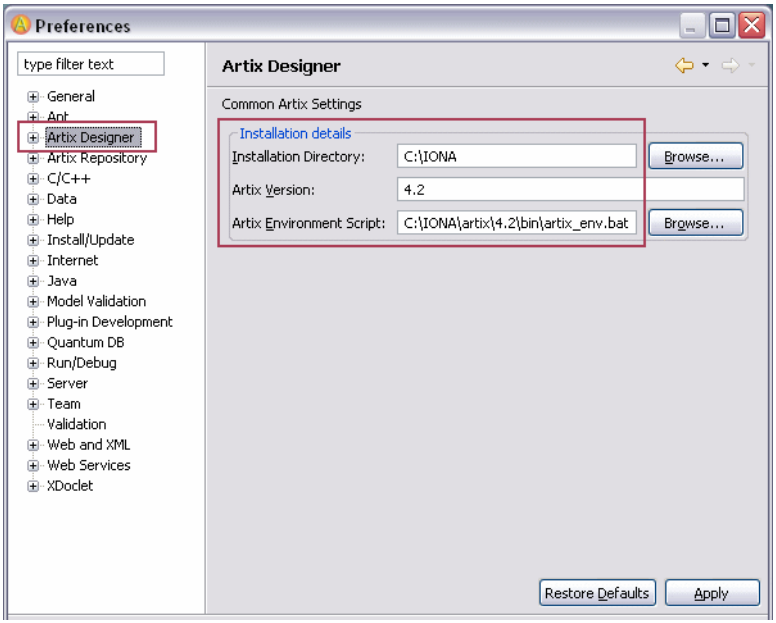
8. Eclipse contacts the specified URL and returns with a list of available Eclipse plug-ins at that site.
9. In the **Search Results** panel, check the **Show the latest version** checkbox. Select the check boxes beside all **IONA Artix Plug-ins** entries, then click **Next**.
10. Accept the license agreement and click **Next**.
11. In the **Installation** panel, confirm the target installation location and click **Finish** to begin the installation.

Setup with Artix Designer

Once 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 4.2 top-level installation directory, “4.2” for the version number, and the path to the `artix_env[.bat]` script.
4. Click OK.

Figure 4: *Common Artix Settings in Eclipse Preferences*



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 4.2 on the same machine. The Artix libraries and demo 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.

Installing Artix with Orbix

Overview

There are two possible reasons for installing Artix on the same machine as Orbix:

- To allow Orbix applications to use Artix functionality. For example, you may want to embed the Artix routing plug-in into an Orbix application.
- To enable Artix applications to use Orbix enterprise features, such as the Name Service.

Choosing an installation directory

There are two ways of installing Artix and Orbix on the same machine:

- Install both products in a common *ArtixInstallDir* directory, such as `C:\IONA`
- Install the products in separate directories

There are advantages and disadvantages to both approaches, as described in Table 9.

Table 9: *Installation directory comparison*

Destination	Advantages	Disadvantages
Common directory	Merges license files in the *ArtixInstallDir*/etc directory. Overwrites and updates Orbix library files in the *ArtixInstallDir*/bin directory.	Restricted to Artix 4.x and Orbix 6.3.1. Orbix must be installed first.
Separate directories	Can be used with Orbix 6.2 and earlier.	No merging of license files. Duplicates some libraries in each product's *ArtixInstallDir*/bin directory.

Merging Artix and Orbix licenses

If you install Artix into the same top-level directory as Orbix 6.3.1, the Artix installer automatically appends the Artix license to the existing Orbix license in the *ArtixInstallDir*/etc directory.

However, if you choose to install Artix and Orbix into separate directories, and you want your Orbix applications to access Artix functionality, you must merge the two license files using a text editor.

IONA recommends that you preserve your existing Artix and Orbix license files, and save the merged license file in a new location.

You then need to point to the location of the merged license in a

```
SET IT_LICENSE_FILE=
```

command in the *DomainName*_env file for your Orbix application.

Combining Artix and Orbix configuration files

By default, the Artix configuration file is installed in:

ArtixInstallDir/artix/*version*/etc/domains/artix.cfg

Once you have Artix and Orbix installed on the same machine, you need to reference the Artix configuration file from within the Orbix configuration file: You can do this in one of the following ways:

1. By adding the following `include` statement to the bottom of the Orbix configuration file:

```
artix {
  include "ArtixInstallDir/artix/version/etc/domains/
   artix.cfg";
};
```

2. By adding a new scope containing the entire contents of the `artix.cfg` file to the bottom of the Orbix configuration file, as follows:

```
artix {
  ContentsOfArtixConfigFile
};
```

3. By adding the following `include` statement to the bottom of the Orbix configuration file:

```
include "ArtixInstallDir/artix/version/etc/domains/
   artix.cfg";
```

Then open the `artix.cfg` file and add an `artix` scope around the entire contents of the file, as follows:

```
artix {
  ContentsOfArtixConfigFile
};
```

Running Orbix and Artix applications

Overview

If you combine your Artix and Orbix configuration files, take into account the implications described in this section when running Orbix or Artix applications.

Running an Orbix application in a pure Orbix environment

To run an Orbix application under a pure Orbix environment, use the *DomainName*`_env` file, which references the Orbix *DomainName*`.cfg` file.

As long as your application's `ORBname` does not include `artix` as the initial scope name, the Artix configuration information is ignored.

Running an Artix application in a pure Artix environment

To run an Artix application under a pure Artix environment, use the `artix_env` script, which references the `artix.cfg` file.

If you used the third approach to combining the Artix and Orbix configuration files, all Artix applications will need to run under a scope that starts with `artix`.

Running Artix/Orbix applications that use the other product's functionality

To run either an Artix or an Orbix application that references the other product's functionality, create an application-specific environment file that:

- References the location of the Orbix *DomainName*`.cfg` file
- Includes `artix` as the initial scope of the application's `ORBname`
- Sets the location of the merged license file (`IT_LICENSE_FILE`)
- Sets the `PATH` environment variable, including the Artix directories first, then the Orbix directories

The application reads all the Orbix configuration information first and then reads all of the Artix configuration information.

If the same configuration entry is encountered twice, the Artix value takes precedence because it is contained in the nested scope. (All the Orbix entries are at global scope, whereas all the Artix entries are within the `artix` scope.)

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 4.2|Uninstall IONA Artix 4.2**.
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