PROGRESS
SOFTWARE

Artix® ESB

Installation Guide

Version 5.5, December 2008

While the information in this publication is believed to be accurate Progress Software Corporation 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. Progress Software Corporation 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: November 5, 2009

Contents

List of Tables

Preface

What This Book Covers

This book discusses the prerequisites for installing Artix and the procedures for installing, configuring, updating, and uninstalling Artix.

Who Should Read This Book

This book is written for those responsible for installing Artix and assumes familiarity with:

- The hardware and software Artix interacts with.
- The user interfaces, graphical and character, used by Artix and by the hardware and software Artix interacts with.

How to Use This Book

This book contains the following chapters:

- Chapter 1, Prerequisites, discusses the things you should do and think about before installing Artix.
- Chapter 2, Installing Artix, describes various ways of installing Artix.
- Chapter 3, Configuring Artix, discusses the things you might have to do or think about after installing Artix.
- Chapter 4, Updating an Artix Installation, describes how the Artix Update tool can keep your Artix installation up-to-date.
- Chapter 5, Uninstalling Artix, describes how to uninstall Artix.

Artix Documentation Library

For information on the entire Artix Documentation Library, including organization, contents, conventions, and reading paths, see Using the Artix Library.

CHAPTER 1

Prerequisites

This chapter discusses what you should do and think about before installing Artix.

In this chapter

This chapter contains the following sections:

Pre-Installation Tasks

Overview

This section discusses what you should do before installing Artix.

Read the release notes

Before installing Artix:

- Visit the Artix 5.5 Product Documentation Web page.
- 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 Artix 5.5 Product Documentation Web page.

Pre-Installation Considerations

Overview

This section discusses what you should think about before installing Artix.

In this section

This section discusses the following topics:

Supported Platforms, C++ Compilers, JREs, and JDKs

Platforms and compilers

Artix 5.5 is supported on Windows, Linux, and UNIX.

Note: Artix Designer is supported on Windows, Linux, and Solaris only.

Table 1 shows the supported operating systems, hardware platforms, C++ compilers, JREs, and JDKs.

For the latest version of this information, see the Artix Platform Support Web page.

Table 1: *Supported Platforms and Compilers*

Operating System	Hardware[1]	C++ Compilers	JDK/JRE[2]
Windows XP SP2[3]	x86_32	Visual C++ 6.0 SP3 Visual Studio 2003, VC 7.1 Visual Studio 2005, VC 8.0	1.5.0_09 or higher 1.6.0_07 or higher
Windows Vista[3]	x86_32	Visual Studio 2003, VC 7.1 Visual Studio 2005, VC 8.0	1.5.0_09 or higher 1.6.0_07 or higher
Windows Server 2003[3]	x86_32	Visual Studio 2003, VC 7.1 Visual Studio 2005, VC 8.0	1.5.0_09 or higher 1.6.0_07 or higher
HP-UX 11i (11.23) (32-bit)[4,5]	PA-RISC	aCC 3.7x	HP Java 1.5.0_11 or higher HP Java 1.6.0_02 or higher
HP-UX 11i (11.23) (64-bit)[6]	Itanium	aCC 6.1x	HP Java 1.5.0_12 or higher HP Java 1.6.0_02 or higher
HP-UX 11i (11.31) (64-bit)[6]	Itanium	aCC 6.1.x	HP Java 1.5.0_12 or higher
Red Hat Enterprise Linux AS 5.0 (32-bit)[3]	x86_32	GCC 4.1.x	1.5.0_12 or higher 1.6.0_05 or higher
Red Hat Enterprise Linux AS 5.0 (64-bit)[3,7]	x86_64	GCC 4.1.x	1.5.0_12 or higher 1.6.0_10 or higher
Red Hat Enterprise Linux AS 4.0 (32-bit)[3,8]	x86_32	GCC 3.4.x	1.5.0_12 or higher 1.6.0_05 or higher

Table 1: *Supported Platforms and Compilers (Continued)*

Operating System	Hardware[1]	C++ Compilers	JDK/JRE[2]
Red Hat Enterprise Linux AS 4.0 (64-bit)[3,7]	x86_64	GCC 3.4.x	1.5.0_12 or higher 1.6.0_10 or higher
SuSE Linux Enterprise Server 10 (32-bit)[3]	x86_32	GCC 4.1.x	1.5.0_12 or higher 1.6.0_05 or higher
SuSE Linux Enterprise Server 10 (64-bit)[3,7]	x86_64	GCC 4.1.x	1.5.0_12 or higher 1.6.0_10 or higher
Solaris 10 (32-bit)	SPARC	Sun Studio 11 Sun Studio 12	1.5.0_12 or higher 1.6.0_07 or higher
Solaris 10 (64-bit)	SPARC	Sun Studio 11 Sun Studio 12	1.5.0_12 or higher 1.6.0_07 or higher
Solaris 10 (32-bit)	x86_32	Sun Studio 11	1.5.0_16 or higher 1.6.0_10 or higher
Solaris 10 (64-bit)	x86_64	Sun Studio 11	1.5.0_16 or higher 1.6.0_10 or higher
AIX 5.3 (32-bit)[9]	PowerPC	XL C++ 7	IBM Java 5.0 SR7 or higher IBM Java 6.0 SR1 or higher
AIX 6.1 (32-bit)[9,10]	PowerPC	XL C++ 7	IBM Java 5.0 SR7 or higher
AIX 6.1 (64-bit)[9,10]	PowerPC	N/A	IBM Java 6.0 SR1 or higher[11]

1. In the Hardware column, x86_32 indicates 32-bit Pentium architecture. x86_64 indicates any of the following: AMD64 (Opteron, Athlon 64), Intel Pentium Extreme Edition 64 bit, Intel EMT64 (Xeon, Core 2)
2. JDK/JRE versions are supported for both the JAX-WS and JAX-RPC runtimes. See "Java JRE and JDK Requirements" on page 18 for details.
3. Supported on VMware ESX/ESXi 3.5.0 for virtualization support.
4. For HP-UX 32-bit, the Artix 5.5 C++/JAX-RPC runtime libraries support applications built with both the HP-UX Standard and Classic C++ runtimes.
5. JAX-RPC development and runtime does not support Java 6 on this platform
6. For HP-UX 64-bit, the Artix 5.5 C++/JAX-RPC runtime libraries support applications built with the HP-UX Standard C++ runtime.
7. 64-bit Linux requires both a 32-bit and 64-bit JVM installed.
8. Due to a bug in the Eclipse SWT or RHEL GWT, the Artix Designer is not supported on this platform
9. Includes CRT runtime support for v7, v8, and v9.

10.JAX-WS development and runtime only supported on this platform. JAX-RPC is not supported.

11.Uses the JDK embedded in WebSphere Application Server 7.0

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 XP	x86_32	SP2
Windows Vista	x86_32	
Windows Server 2003	x86_32	
Red Hat Enterprise Linux Advanced Server 5.0	x86_32	
Red Hat Enterprise Linux Advanced Server 5.0	x86_64	
Red Hat Enterprise Linux Advanced Server 4.0	x86_32	
Red Hat Enterprise Linux Advanced Server 4.0	x86_64	
SUSE Linux Enterprise Server 10 (32-bit)	x86_64	
SUSE Linux Enterprise Server 10 (64-bit)	x86_64	
Solaris 10 (32-bit)	SPARC	
Solaris 10 (32-bit)	x86_32	

Table 2: *Required OS Patches*

Operating System[1]	Hardware[2]	OS Patches; C++/Java Runtime Environment
Solaris 10 (64-bit)	SPARC	
Solaris 10 (64-bit)	x86_64	
AIX 5.3 (32-bit)	PowerPC	April 2007 IBM C++ Runtime Environment Components for AIX PTF (ref. no. 4015708)
HP-UX 11i (11.23) (32-bit)	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)
HP-UX 11i (11.23) (64-bit)	Itanium	

[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.

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 12 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 "Configuring the Artix Visual C++ Environment" on page 51.

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 "Configuring the Artix C++ Development Environment" on page 50.

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.

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

Java JRE and JDK Requirements

Artix Designer Java requirements

Artix Designer 5.5 requires a Java 1.5.x JRE or JDK to run. Artix Designer 5.5 does not support Java 1.6.x.

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" on page 19. See also "Artix Designer Requirements" on page 21.

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. Alternatively, the installer also allows you to specify the location of a previously installed system JRE or JDK.

Note: Developing programs for the Artix 5.5 Java JAX-WS runtime requires a 1.5.x or higher 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 do not install the bundled JRE, 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 12.

If you specify the location of an existing JRE or JDK, you might also need to specify its location in the `JAVA_HOME` environment variable. See "Setting the JAVA_HOME Environment Variable" on page 46.

Java JDK requirements

You must install a JDK to use with Artix in the following 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.

As shown in Table 1 on page 12, a JDK of at least release level 1.5.x is required.

Specific Java suppliers supported

Artix supports the specific JRE and JDK versions listed in Table 1 on page 12. 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, see 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 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 location in a global `JAVA_HOME` environment variable, as described in "Setting the JAVA_HOME Environment Variable" on page 46.
- 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.

Artix Designer Requirements

Artix Designer 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.

Artix Designer 5.5 requires a Java 1.5.x JRE or JDK to run. Artix Designer 5.5 does not support Java 1.6.x. For more details, see "Java JRE and JDK Requirements" on page 18.

The Artix installer installs an Eclipse 3.3.2 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.

Interoperability

Overview

This section describes how Artix components interoperate with other Artix components and with third-party products.

Artix JAX-WS runtime

The Artix JAX-WS runtime interoperates with the following:

- Artix C++/JAX-RPC runtime
- .NET Framework 1.x/2.x runtime
- BEA AquaLogic Service Bus 2.6

Messaging

Artix supports the following messaging products:

- IBM WebSphere MQ 5.3 and 6.0
- 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 the following 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 (JAX-RPC) supports the following application servers:

- JBoss 4.0.1
- Apache Tomcat 5.5
- BEA WebLogic 8.1 SP3
- BEA WebLogic 9.0
- IBM WebSphere 5.1
- IBM WebSphere 6.0
- IBM WebSphere 6.1

The Artix J2EE Connector (JAX-WS) supports the following application servers:

- JBoss 4.0.1
- IBM WebSphere 6.1

The JAX-RPC and JAX-WS application server support applies to the operating systems that are supported by both Artix and the relevant application server.

Security

Artix supports the following security products and protocols:

- SiteMinder 4.6.1, 5.5
- Kerberos 5
- LDAP 3.0

Web services

Artix supports the following 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.

Artix supports .NET integration by interoperating with the Microsoft products listed in Table 3.

Table 3: *Microsoft Products that Interoperate with Artix*

Product Type	Product
Development environment	Visual Studio .NET 2003 and 2005
Runtime environment	.NET Framework 1.1
Operating system	Windows XP Windows Server 2003

Virtualization Support Requirements

Overview

Virtualization is the process of sharing the resources of a single computer across multiple environments, and thereby enabling one computer to do the job of many. Artix 5.5 provides virtualization support on selected platforms.

Supported platforms for virtualization

Artix 5.5 includes support for virtualization on the following supported platforms:

- Windows XP
- Windows Vista
- Windows 2003 Server
- Red Hat Enterprise Linux
- Suse Linux

Note: Each of the above platforms is supported on VMware ESX/ESXi 3.5.0.

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.

Disk Space Requirements

Overview

This section lists the amount of permanent and temporary disk space required for different installations of Artix 5.5.

Artix installation disk space

The disk space requirements for Artix depend on the installation options selected. Table 4 shows the approximate disk space in megabytes for full and runtime-only installations. These numbers include the bundled JRE optionally installed with Artix.

Table 4: *Disk Space Used by Artix Installations (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 the Artix installer package is shown in Table 5. 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 4.

Table 5: *Disk Space Used by the Artix Installer and JRE (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 4 and Table 5, 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
```

CHAPTER 2

Installing Artix

This chapter describes various ways of installing Artix.

In this chapter

This chapter contains the following sections:

Running the Artix Installer

Downloading an installation package

The Artix 5.5 installation package is available for download from the Product Download Center at http://www.iona.com/downloads/.

The following installation packages are available:

Table 6: *Artix Installation Packages*

Platform	Use with
Windows	Windows XP, Vista, and Server 2003.
Linux 32-bit	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
AIX	AIX 6.1, 5.3
HP-UX11i 32-bit	32-bit HP-UX 11i (11.23) on PA-RISC hardware
HP-UX11i 64-bit	64-bit HP-UX 11i (11.23) on Itanium hardware

Installer restrictions

The Artix 5.5 installer has the following restrictions:

- You must not install Artix 5.5 and any other IONA product in the same directory.

> **Note:** Although you could install particular pre-5.5 versions of Artix and particular versions of other IONA products in the same directory, this is *not* true of Artix 5.5.

- You must not load Artix libraries and Orbix libraries into the same process space.
- When installing Artix 5.5 on Windows Server 2003, you must run the installer in Windows XP compatibility mode.
- When installing Artix 5.5 on Windows platforms, you must 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 installing Artix 5.5 on UNIX using console mode, the only installation options available are Full and Runtime-Only.

Installation modes

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

- "Installing in GUI Mode" on page 32.
- "Installing in Console Mode" on page 34.
- "Installing in Silent Mode" on page 36.

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 on-screen instructions and respond to each prompt.

Use the information in Table 7 as a guide when selecting installation options as the installation proceeds.

Table 7: *Artix Installation Options*

Platform	Installation Option	Default	Notes

All	The **Choose Install Folder** panel lets you select the top-level directory for your Artix installation.	Windows:
		`C:\IONA\artix_5.5`
		UNIX:
		`/opt/iona/artix_5.5`
			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.

All	The **Choose Install Set** panel lets you select the components you wish to install.	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 9 on page 38 for help in deciding which options to install.

Table 7: *Artix Installation Options (Continued)*

Platform	Installation Option	Default	Notes
All	The **Choose Java Virtual** Machine panel lets you install or select a JVM.	Install a bundled JRE for use by Artix Designer.	Java JRE and JDK issues are discussed in "C++ Compiler Requirements" on page 17. The installer might 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.
Windows only	The **Choose Shortcut Location** panel lets you specify a 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.
All	The **Save Installation Properties** panel lets you save a properties file containing your 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 36.

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

Note: A full Artix license file, `licenses.txt`, is automatically installed by default in the *`ArtixInstallDir`*`/etc` folder of your product installation. A 30-day evaluation installation will also automatically install a license file.

Installing in Console Mode

Overview

UNIX users can run the Artix installer in console mode if no graphical-interface 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 on-screen instructions and respond to option prompts. Use the information in Table 7 on page 32 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 Example 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 9 on page 38 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] Development Tools
17-  |-[X] Artix Designer
18-  |-[X] Command-line Tools
19- [X] Third-party Integrations
20-  |-[X] EMS Integration
21-  |-[X] Amberpoint
22-  |-[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. Alternatively, 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 8:

Table 8: *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 8: *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 9, specify the Artix components you want to install. To specify a Full installation, you must list all group components in Table 9.
`ECLIPSE_SUPPORT`	Set to **`true`** to install Eclipse and the Artix Designer plug-ins[a]

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

The valid values for the `CHOSEN_INSTALL_FEATURE_LIST` variable are shown in Table 9.

Note: The codes in Table 9 are either group codes or individual feature codes. If you specify a group code, you specify all features in that group.

Table 9: *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.
`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 9: *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 9: *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.

Table 9: *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`[a]		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 `ECLIPSE_SUPPORT` variable to `true` to install Artix Designer.

Example properties file

An example of a properties file is shown below:

```
SILENT_ACCEPT_LICENSE_AGREEMENT=true
SET_PATH=
INSTALLER_UI=silent
USER_INSTALL_DIR=C\:\\IONA\\artix_5.5
ECLIPSE_SUPPORT=true
USER_INPUT_SAVE_PROPERTIES_YES_NO=No
CHOSEN_INSTALL_FEATURE_LIST=CXXCore,JavaCore,CXXLocator,CXXSecur
    ity,CXXHA,CXXRouter,CXXTX,CXXSM,EMS,Amber,CAWSDM,Third-p,CXXR
    T,CXXDev,JavaRT,JavaDev,JavaSecurity,JavaRouter,Develop,Desig
    ner,Command,ArtixLocator
JDK_HOME=
```

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 `-i silent` and `-f` options:

   Windows

   ```
   artix_platform.exe -i silent -f your_properties_file
   ```

   UNIX

   ```
   ./artix_platform.bin -i silent -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 -i silent
   ```

   UNIX

   ```
   ./artix_platform.bin -i silent
   ```

Note: A full Artix license file, `licenses.txt`, is automatically installed by default in the *`ArtixInstallDir`*`/etc` folder of your product installation. A 30-day evaluation installation will also automatically install a license file.

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 might appear to be hung. In fact, the silent uninstallation is proceeding silently. Control is returned to the dialog box when the uninstallation completes.

CHAPTER 3

Configuring Artix

This chapter discusses the things you might have to do or think about after installing Artix.

In this chapter

This chapter contains the following sections:

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, 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 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
```

To determine the 8.3 version of a file or directory name, use the `dir /x` command at a Windows command prompt.

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 20 for further information.

Configuring the Artix Command-Line Environment

Setting the Artix development environment for the 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 "Configuring the Artix C++ Development Environment" on page 50 for an exception to this.

Setting the Artix development environment for the 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
```

Configuring the Artix C++ Development Environment

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 once before Artix C++ development can proceed. This first run can be in a shell window that you close immediately afterward. Therefore, there is no need to start Artix Designer from the same shell prompt.

Configuring the Artix Visual C++ Environment

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`, supplied 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, and 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, you might 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:

1. Source the `vcvars32.bat` file.
2. cd to the directory containing `eclipse.exe`.
3. 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_5.5\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_5.5\tools\eclipse
start .\eclipse.exe
endlocal
```

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

Setting the environment for Visual C++ 6.0

The default Artix for Windows installation assumes that 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_5.5\cxx_java\bin
> artix_env -compiler vc60
```

Note: You only need to use the `-compiler` switch once to specify your compiler version. After 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_5.5\cxx_java\bin
> artix_env -compiler vc71
```

Starting and Logging Considerations for Artix Designer

Overview

Unlike previous releases, Artix Designer 5.5 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_5.5/tools/eclipse/eclipse &
```

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

```
/opt/iona/artix_5.5/tools/eclipse_64/eclipse &
```

For Windows, start Artix Designer from its icon in the Start menu: **Start | (All) Programs | IONA | Artix 5.5 | Artix Designer**.

Note: Depending on how your Visual C++ environment variables are set, you might need to start Artix Designer with a batch file as described in "Configuring the Artix Visual C++ Environment" on page 51.

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 might 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_5.5/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_5.5\tools\eclipse
```

If you use a startup batch file for Visual C++ like Example 2 on page 52, be sure to include a line that changes to this directory before starting Eclipse, to keep the log files in the same place.

Workspace Considerations for Artix Designer

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 5.5 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.

Configuring the Import Samples Feature of 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.

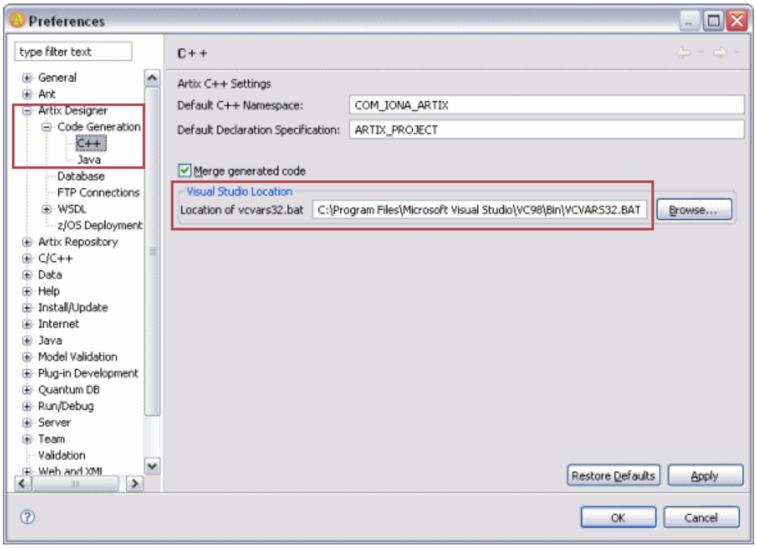
Setup step for Windows

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 "Configuring the Artix Visual C++ Environment" on page 51 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. Click 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

More Memory for Java Components of C++/JAX-RPC on AIX

On AIX, Java components within the Artix C++/JAX-RPC runtime might need additional heap space to function correctly. To increase the default heap space available to these components, set the following environment variable:

```
LDR_CNTRL=MAXDATA=0x80000000@USERREGS
```

For more information on this issue, visit the following Web site:

http://www.ibm.com/developerworks/eserver/articles/aix4java1.html

CHAPTER 4

Updating an Artix Installation

This chapter describes how the Artix update tool can keep your Artix installation up-to-date.

In this chapter

This chapter contains the following section:

Running the Update Tool

Overview

The Artix update tool can keep your Artix installation up-to-date. You can use it to do the following:

- Print a list of installed components and versions.
- List history of applied patches.
- Apply patches.
- Accept license agreements.
- Roll back patches.

Running an update

To run the update tool:

1. Change directory to `*InstallDir*\conf\bin` (where *InstallDir* represents the full path to your Artix 5.5 installation).
2. Enter the `update` command followed by any of the arguments described in "Update tool arguments" on page 62.

Update tool arguments

Strictly speaking, all arguments are optional. However, there is a range of arguments of which at least one must be specified with the `update` command.

Required arguments

At least one of the following arguments must be specified:

`-help`	Display usage table for the `update` command.
`-patchFile` *PatchFile*	Apply the patch contained in the specified *PatchFile* to your installation.
`-history`	Display patch history.
`-version`	Display component version information.
`-rollback` *Id*	Roll back the patch specified by *Id*.

Optional arguments

The `update` command may take any of the following optional arguments:

`-kitDir` *`Path`*	Specify the location of the kit.
`-nodiskcheck`	Disable disk space checking. Disk space checking is enabled by default.
`-acceptagreement`	Silently accept the license agreement.
`-nobackup`	Disable backup of files to be patched.
`-noupdaterpatch`	Disable patching of the update tool.
`-noversioncheck`	Disable patch version checking.

CHAPTER 5

Uninstalling Artix

This chapter describes how to uninstall Artix.

In this chapter

This chapter contains the following sections:

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