Artix™

Using the Artix Library

Version 4.2, March 2007

Making Software Work Together™

Updated: April 6, 2007

Contents

CHAPTER 1

Artix Library Overview

This chapter describes the contents of the Artix Library, how to get additional information, and the documentation conventions used.

In this chapter

This chapter includes the following topics

Artix Documentation Library

Overview

The Artix documentation library is organized into the following sections:

- Getting Started
- Designing Artix Solutions
- Configuring and Managing Artix Solutions
- Using Artix Services
- Integrating Artix Solutions
- Integrating with Management Systems
- Reference
- Artix Orchestration
- Artix Online Help

Getting Started

The books in this section provide you with a background for working with Artix. They describe many of the concepts and technologies used by Artix. They include:

- Release Notes contains release-specific information about Artix.
- Installation Guide describes the prerequisites for installing Artix and the procedures for installing Artix on supported systems.
- Using the Artix Library (this book) introduces the Artix documentation library, explains its conventions, and provides suggested reading paths.
- Getting Started with Artix describes basic Artix and WSDL concepts, and shows a simple example application.
- Artix Online Help Infocenter describes how to use the Artix Designer GUI tools to build Artix solutions.
- Artix Technical Use Cases provides a number of step-by-step examples of building common Artix solutions.

Designing Artix Solutions

The books in this section go into greater depth about using Artix to solve real-world problems. They describe how to build service-oriented architectures with Artix and how Artix uses WSDL to define services:

- Building SOA with Artix provides an overview of service-oriented architectures and describes how they can be implemented using Artix.
- Writing Artix Contracts describes the components of an Artix contract. Special attention is paid to the WSDL extensions used to define Artix-specific payload formats and transports.

Developing Artix Solutions

The books in this section how to use the Artix APIs to build new services:

- Developing Artix Applications in C++ discusses the technical aspects of programming applications using the C++ API.
- Developing Advanced Artix Plug-ins in C++ discusses the technical aspects of implementing advanced plug-ins (for example, interceptors) using the C++ API.
- Developing Artix Applications in Java discusses the technical aspects of programming applications using the Java API.
- Artix WSDLGen Guide discusses the technical aspects of generating C++ and Java code using scripting tools.

Configuring and Managing Artix Solutions

This section includes:

- Configuring and Deploying Artix Solutions explains how to set up your Artix environment and how to configure and deploy Artix services.
- Managing Artix Solutions with JMX explains how to monitor and manage an Artix runtime using Java Management Extensions.

For information on using third-party management systems, see Integrating with Management Systems.

Using Artix Services

The books in this section describe how to use the services provided with Artix:

- Artix Router Guide explains how to integrate services using the Artix router.
- Artix Locator Guide explains how clients can find services using the Artix locator.
- Artix Session Manager Guide explains how to manage client sessions using the Artix session manager.
- Artix Transactions Guide, C++ explains how to enable Artix C++ applications to participate in transacted operations.
- Artix Transactions Guide, Java explains how to enable Artix Java applications to participate in transacted operations.
- Artix Security Guide explains how to configure and develop secure Artix applications.

Integrating Artix Solutions

The books in this section describe how to integrate Artix solutions with other middleware technologies.

- Artix for CORBA provides information on using Artix in a CORBA environment.
- Artix for J2EE provides information on using Artix to integrate with J2EE applications.

For details on integrating with Microsoft's .NET technology, see the documentation for Artix Connect.

Integrating with Management Systems

The books in this section describe how to integrate Artix solutions with a range of third-party enterprise and SOA management systems. They include:

- IBM Tivoli Integration Guide explains how to integrate Artix with the IBM Tivoli enterprise management system.
- BMC Patrol Integration Guide explains how to integrate Artix with the BMC Patrol enterprise management system.
- CA-WSDM Integration Guide explains how to integrate Artix with the CA-WSDM SOA management system.
- AmberPoint Integration Guide explains how to integrate Artix with the AmberPoint SOA management system.

Reference

These books provide detailed reference information about specific Artix APIs, WSDL extensions, configuration variables, command-line tools, and terms. The reference documentation includes:

- Artix Command Line Reference
- Artix Configuration Reference
- Artix WSDL Extension Reference
- Artix Java API Reference
- Artix Security API Reference
- Artix C++ API Reference
- Artix .NET API Reference
- WSDLGen Java API Reference
- WSDLGen JavaScript API Reference
- Artix Glossary

Artix Orchestration

These books describe the Artix support for Business Process Execution Language (BPEL), which is available as an add-on to Artix. These books include:

- Artix Orchestration Release Notes
- Artix Orchestration Installation Guide
- Artix Orchestration Online Help
- Artix Orchestration Administration Console Help

Artix Glossary

The Artix Glossary is a comprehensive reference of Artix terms. It provides quick definitions of the main Artix components and concepts. All terms are defined in the context of the development and deployment of Web services using Artix.

Artix Online Help

Artix Designer and Artix Orchestration Designer include comprehensive online help, providing:

- Step-by-step instructions on how to perform important tasks
- A full search feature
- Context-sensitive help for each screen

You can access the online help the following different ways:

- Select **Help|Help Contents** from the menu bar. The help appears in the contents panel of the Eclipse help browser.
- Press **F1** for context-sensitive help.
- See the Artix Infocenter available online.

In addition, there are a number of cheat sheets that guide you through the most important functionality in Artix Designer and Artix Orchestration Designer. To access these, select **Help|Cheat Sheets**.

Getting the Latest Version

The latest updates to the Artix documentation can be found at http://www.iona.com/support/docs.

Compare the version dates on the web page for your product version with the date printed on the copyright page of the PDF edition of the book you are reading.

Searching the Artix Library

You can search the online documentation by using the **Search** box at the top right of the documentation home page:

http://www.iona.com/support/docs

To search a particular library version, browse to the required index page, and use the **Search** box at the top right, for example:

http://www.iona.com/support/docs/artix/4.0/index.xml

You can also search within a particular book. To search within a HTML version of a book, use the **Search** box at the top left of the page. To search within a PDF version of a book, in Adobe Acrobat, select **Edit|Find**, and enter your search text.

Additional Resources

The IONA Knowledge Base contains helpful articles written by IONA experts about Artix and other products.

The IONA Update Center contains the latest releases and patches for IONA products.

If you need help with this or any other IONA product, go to IONA Online Support.

Comments, corrections, and suggestions on IONA documentation can be sent to docs-support@iona.com .

Documentation Conventions

Overview

This section shows the typographical and keying conventions used by the Artix documentation library.

Typographical conventions

The Artix library uses the following typographical conventions:

`Fixed width`	Fixed width (Courier font) in normal text represents portions of code and literal names of items such as classes, functions, variables, and data structures. For example, text might refer to the `IT_Bus::AnyType` class. Constant width paragraphs represent code examples or information a system displays on the screen. For example: `#include <stdio.h>`
`Fixed width italic`	Fixed width italic words or characters in code and commands represent variable values you must supply, such as arguments to commands or path names for your particular system. For example: `% cd /users/`*`YourUserName`*
Italic	Italic words in normal text represent *emphasis* and introduce *new terms*.
Bold	Bold words in normal text represent graphical user interface components such as menu commands and dialog boxes. For example: the **User Preferences** dialog.

Keying Conventions

The Artix library uses the following keying conventions:

No prompt	When a command's format is the same for multiple platforms, the command prompt is not shown.
`%`	A percent sign represents the UNIX command shell prompt for a command that does not require root privileges.
`#`	A number sign represents the UNIX command shell prompt for a command that requires root privileges.
>	The notation > represents the MS-DOS or Windows command prompt.
`...` . . .	Horizontal or vertical ellipses in format and syntax descriptions indicate that material has been eliminated to simplify a discussion.
`[]`	Brackets enclose optional items in format and syntax descriptions.
`{}`	Braces enclose a list from which you must choose an item in format and syntax descriptions.
`|`	In format and syntax descriptions, a vertical bar separates items in a list of choices enclosed in `{}` (braces).
	In graphical user interface descriptions, a vertical bar separates menu commands (for example, select **File|Open**).

CHAPTER 2

Suggested Reading Paths

This chapter describes suggested reading paths for different types of Artix user.

In this chapter

This chapter includes the following topics

SOA Architects

Overview

This section describes a suggested reading path for SOA architects, and also includes suggestions for background reading.

SOA architect path

SOA architects should start with the following:

1. Building SOA with Artix presents an overview of SOA and ESBs, of how Artix fits into SOA, and of how Artix works.
2. Installation Guide. You must read the following sections about supported environments:
   i. *Supported Systems and Compilers*
   ii. *Java, Compiler, and Artix Designer Requirements*
3. Writing Artix Contracts includes the following information about basic WSDL concepts and how to write a service interface:
   i. *Introduction*. Overview of WSDL, the structure of a contract, and the steps involved in writing a service contract.
   ii. *Designing Logical Data Units*. How to create data types using XML Schema.
   iii. *Defining Logical Messages Used by a Service*. How to build the data types into the messages that a service will use to implement its operations.
   iv. *Defining Your Logical Interfaces*: How to create a service interface using the logical messages.
4. Writing Artix Contracts also includes information about creating specific bindings and endpoints.
   i. Read the relevant binding chapter that applies to your system (for example, SOAP, Fixed, or XML).
   ii. Read the relevant transport chapter that applies to your system (for example, HTTP, Tuxedo, or JMS).

5. Artix Infocenter explains how to use the Eclipse-based Artix Designer GUI tools to design WSDL service contracts and to generate C++ and Java implementation code.

Background reading

In addition, the following publications provide useful background information on Web services, XML, and WSDL:

- *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, by Eric Newcomer
- *Understanding SOA with Web Services*, by Eric Newcomer and Greg Lomow
- W3Schools online tutorials

  (see http://www.w3schools.com)
  - XML tutorial (http://www.w3schools.com/xml/default.asp)
  - XSD tutorial (http://www.w3schools.com/schema/default.asp)
  - XSLT tutorial (http://www.w3schools.com/xsl/default.asp)
- The W3C XML schema page

  (see www.w3.org/XML/Schema)
- THE W3C WSDL specification

  (see www.w3.org/TR/wsdl)

Administrators

Overview

This section describes a suggested reading path for Artix administrators.

Artix administrator path

Administrators can approach the Artix library as follows:

1. Installation Guide describes all the prerequisites and procedures for installing Artix on supported systems. You must read the following:
   i. *Supported Systems and Compilers*
   ii. *Java, Compiler, and Artix Designer Requirements*
   iii. *Installing Artix*
2. Configuring and Deploying Artix Solutions explains how to configure and deploy Artix services in an Artix runtime environment. It describes the Artix configuration file, how to find contracts that control your Artix services, and how to deploy Artix applications. You should start with the following chapters:
   i. *Getting Started* explains how to set up an Artix environment using the `artix_env` script, and Artix system environment variables.
   ii. *Artix Configuration* explains concepts such as Artix configuration files, scopes, namespaces, and variables.
   iii. *Artix Logging* explains how to configure Artix logging for services and Artix subsystems.
   iv. *Deploying Services in an Artix Container* explains how to deploy and manage C++ and Java services using an Artix container.
   v. *Deploying High Availability* explains how to configure and deploy high availability in Artix, which is based on Berkeley DB.
   vi. *Deploying Reliable Messaging* explains how to configure and deploy WS-RM and WS-Addressing for C++ and Java applications in an Artix runtime environment.

   vii. *Publishing WSDL Contracts* explains how to publish WSDL files that correspond to specific Web services, and enable clients to access the WSDL file and invoke on the service.
   viii. *Accessing Contracts and References* shows how to specify the location of WSDL contracts and references in a configuration file and on the command line.

3. Artix Configuration Reference provides a comprehensive reference for the all configuration variables in an Artix configuration domain.
4. Artix Security Guide provides detailed information on Artix security configuration and management.
5. Managing Artix solutions with JMX explains how to manage Artix systems using any JMX console (for example, MC4J, JConsole, and JMX HTTP Adaptor).
6. Administrator's using third-party management tools should see the following guides for information on how to integrate Artix with third-party enterprise management systems:
   - AmberPoint Integration Guide
   - BMC Patrol Integration Guide
   - CA WSDM Integration Guide
   - IBM Tivoli Integration Guide

Background reading

For background information on Web services, XML, and WSDL, see "Background reading" on page 15.

Service Developers

Overview

This section describes reading paths for the following developer use cases:

- "All developers"
- "Integration use case"
- "New development use cases" (C++ and Java)

All developers

All developers should read the following path:

1. Building SOA with Artix presents an overview of SOA and ESBs, of how Artix fits into SOA, and of how Artix works.
2. Installation Guide. You must read the following sections about supported environments:
   i. *Supported Systems and Compilers*
   ii. *Java, Compiler, and Artix Designer Requirements*
3. Writing Artix Contracts includes information about basic WSDL concepts and how to write a service interface.
   i. *Introduction*. Overview of WSDL, the structure of a contract, and the steps involved in writing a service contract.
   ii. *Designing Logical Data Units*. How to create data types using XML Schema
   iii. *Defining Logical Messages Used by a Service*. How to build the data types into the messages that a service will use to implement its operations.
   iv. *Defining Your Logical Interfaces*: How to create a service interface using the logical messages.
4. Artix Infocenter explains how to use the Eclipse-based Artix Designer GUI tools to design WSDL service contracts and to generate C++ and Java implementation code.

5. Configuring and Deploying Artix Solutions explains how to configure and deploy Artix services in an Artix runtime environment. You must read the following chapters:
    i. *Getting Started* explains how to set up an Artix environment using the `artix_env` script, and system environment variables.
    ii. *Artix Configuration* explains concepts such as Artix configuration files, scopes, namespaces, and variables.
    iii. *Artix Logging* explains how to configure Artix logging for services and Artix subsystems.

Integration use case

Follow this path if you are developing a service as a front-end for existing functionality.

1. Artix Technical Use Cases. Read the following chapter:
    i. *Web Service Enabling Backend Services*. Walks through the steps for the integration use case.
2. Router Guide. Read the following basic information about the router service, routes, and the tools used in making routes.
    i. *Introduction*. Overview of the router and how it is used.
    ii. *Compatibility of Ports and Operations*. Explains the requirements for routing between interfaces.
    iii. *Creating Routes Using Artix Tools*. Introduces the GUI and command line tools that can be used to create routes.
3. Writing Artix Contracts includes information about creating bindings and endpoints.
    i. Read the relevant binding chapters that apply to your system (for example, SOAP, Fixed, or XML).
    ii. Read the relevant transport chapters that apply to your system (for example, HTTP, Tuxedo, or JMS).
4. Router Guide. Read the following information about how to define the routes between the endpoints:
    i. *Creating a Basic Route*. This is required reading. It describes the minimum needed to create a route in Artix.
    ii. *Adding Operation-Based Rules to a Route*: This is optional. It expands on basic route design.

   iii. *Adding Attribute-Based Rules to a Route*. This is optional. It expands on basic route design.
   iv. *Adding Content-Based Rules to a Route*. This is optional. It describes how to create content based routes.
   v. *Linking Routes*. This is optional. It expands on previous chapters.
   vi. *Using Advanced Routing Features*. This is optional. It describes how to create routes for various advanced use cases such as load balancing and service fail-over.
   vii. *Deploying an Artix Router*. This is required reading. It describes how to deploy the router in an Artix runtime environment.

Advanced integration topics

In addition, you may wish to read the following:

- Artix for CORBA contains detailed information about using Artix to integrate with CORBA applications.
- Artix for J2EE contains detailed information about using Artix with J2EE applications.

New development use cases

Follow this path if you are developing new services.

Service consumer

Read the following if you are developing a service consumer:

1. Artix Technical Use Cases. Read the following chapter:
   i. *Building a Client for a Web Service*. Provides a walk through of the service consumer use case.
2. Writing Artix Contracts. Provides details about adding a binding and transport:
   i. Read the relevant binding chapter that applies to your system (for example, SOAP, Fixed, or XML).
   ii. Read the relevant transport chapter that applies to your system (for example, HTTP, Tuxedo, or JMS).

C++ development

For detailed information on developing a C++ service provider or consumer, read the following:

3. Developing Artix Applications in C++:
    i. *Getting Started with Artix Programming*. An overview of how a developer works in the Artix C++ development environment.
    ii. *Artix Programming Considerations*: *Operations and Parameters* section. An overview of how WSDL is mapped into C++.
    iii. *Server Programming*. The basics of developing an Artix C++ service.
    iv. *Client Programming*. The basics of developing an Artix C++ consumer.
    v. *Artix Programming Considerations*: *Compiling and Linking an Artix Application* section. What is needed to build Artix C++ applications.
    vi. *Artix Programming Considerations*: *Building Artix Stub Libraries on Windows* section. How to build Artix stub code into a Windows DLL.
    vii. *Artix Data Types*. Overview of how WSDL types are mapped into C++.
    viii. *Artix Programming Considerations*: *Exceptions* section. Overview of how to create and handle exceptions in Artix C++.
    ix. *Artix Programming Considerations: Locating Services with UDDI* section. How to use the UDDI interface as an alternate method of finding services.
    x. *Endpoint References and Callbacks*. Overview of EPRs and how to use them in implementing callbacks.
    xi. *Artix Contexts*. How to get information from the binding and transport layers of the runtime.
    xii. *Working with Transport Attributes*. How to extract a default set of transport information from the runtime.
    xiii. *Persistent Maps*. How to use persistent data for high availability.
    xiv. *Reflection*. How to determine the structure of an Artix data type without advance knowledge.

xv. *Default Servants*. How to write a scalable factory pattern.

xvi. *Artix Programming Considerations: Multi-Threading* section. Describes issues for multi-threaded Artix clients and servers.

Advanced C++ development

For detailed information on developing advanced C++ plugins, read the following:

4. Developing Advanced Artix Plug-Ins with C++. How to write custom interceptors and transport plugins.

Java development

For detailed information on developing a Java service provider or consumer, read the following:

5. Developing Artix Applications in Java:

   i. *The Artix Java Development Model*. An overview of the Artix Java development process. This includes a section on WSDL to Java mapping.

   ii. *Developing Artix Services*. The basics for developing and building a service in Artix.

   iii. *Developing Artix Consumers*. The basics for developing and building a consumer in Artix.

   iv. *Finding Contracts and References at Runtime*. How to use the Java APIs to locate contracts.

   v. *Things to Consider when Developing Artix Applications: Getting a Bus* section. How to get access to a bus reference from the runtime.

   vi. *Working with Artix Data Types*. Overview of XML schema to Java type mapping for most data types.

   vii. *Creating User-Defined Exceptions*. How to create and handle exceptions in Artix.

   viii. *Using Endpoint References*. Details of working with EPRs.

   ix. *Using Native XML*. How to develop Java applications that work with pure XML data.

   x. *Working with Artix Type Factories*. How to create type factories. Type factories are needed to work with features that follow.

xi. *Using Message Contexts*. Describes the Artix implementation and extension of the JAX-RPC `MessageContext` interface. This enables you to pass/receive information from the lower-levels of the Java runtime including the binding, transport, and handler layers.

xii. *Working with Transport Attributes*. Details about the Artix provided transport attribute types.

xiii. *Sending Message Headers*. Details about using the `MessageContext` to send message headers over SOAP and CORBA.

xiv. *Writing Handlers*. Overview of writing JAX-RPC handler objects. Handler objects are like interceptors in that work on a message as it passes through the Artix runtime.

xv. *Manipulating Messages in a Handler*. Details about altering the contents of requests or responses as they pass through the Handler objects in a message chain.

xvi. *Instrumenting a Service*. Details about adding JMX instrumentation to a service implementation.

xvii. *Using Persistent Datastores*. Details how to use persistent data in Artix Java applications.

xviii. *Using the Call Interface for Dynamic Invocations*. Details about writing clients that can invoke operations on a service for which it only has WSDL.

xix. *Using Substitution Groups*. Substitution groups are an advanced XML Schema construct.

xx. *Working with XML Schema anyTypes*. An `anyType` is the XML equivalent of a CORBA `Any`.

Advanced Java topics

For detailed information on developing Java advanced plugins (for example, transport plugins), read the following:

6. Developing Artix Applications in Java:
    i. *Using Artix Classloader Environments*
    ii. *Developing Plug-Ins*
    iii. *Developing Custom Artix Transports*
    iv. *Configuring Artix Plug-Ins*

Background reading

For background information on Web services, XML, and WSDL, see "Background reading" on page 15.