Artix™

CA WSDM Integration Guide

Version 4.0, March 2006

Making Software Work Together™

Contents

CONTENTS

Preface

What is covered in this book

Artix supports integration with Computer Associates Web Services Distributed Management (CA WSDM). This guide explains how to integrate Artix with CA WSDM.

Who should read this book

This guide is aimed at system administrators using CA WSDM to manage distributed enterprise environments, and developers writing distributed enterprise applications. Administrators do not require detailed knowledge of the technology that is used to create distributed enterprise applications.

This book assumes that you already have a good working knowledge of CA WSDM. For more information on CA WSDM, see the Computer Associates website (http://www.ca.com).

Organization of this book

This book contains the following chapter:

- Chapter 1 explains the Artix integration with CA WSDM.

The Artix Library

The Artix documentation library is organized in the following sections:

- Getting Started
- Designing and Developing Artix Solutions
- Configuring and Deploying Artix Solutions
- Using Artix Services
- Integrating Artix Solutions
- Integrating with Enterprise Management Systems
- Reference Documentation

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.
- Getting Started with Artix describes basic Artix and WSDL concepts.
- Using Artix Designer describes how to use Artix Designer to build Artix solutions.
- Artix Technical Use Cases provides a number of step-by-step examples of building common Artix solutions.

Designing and Developing Artix Solutions

The books in this section go into greater depth about using Artix to solve real-world problems. They describe how Artix uses WSDL to define services, and how to use the Artix APIs to build new services. They include:

- Building Service-Oriented Architectures with Artix provides an overview of service-oriented architectures and describes how they can be implemented using Artix.
- Understanding 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 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.

Configuring and Deploying Artix Solutions

This section includes:

- Configuring and Deploying Artix Solutions discusses how to configure and deploy Artix-enabled systems, and provides examples of typical use cases.

Using Artix Services

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

- Artix Locator Guide discusses how to use the Artix locator.
- Artix Session Manager Guide discusses how to use 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 use the security features of Artix.

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 Enterprise Management Systems

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

- IBM Tivoli Integration Guide explains how to integrate Artix with IBM Tivoli.
- BMC Patrol Integration Guide explains how to integrate Artix with BMC Patrol.
- CA WSDM Integration Guide explains how to integrate Artix with CA WSDM.

Reference Documentation

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

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

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.

Artix Online Help

Artix Designer and the Artix Management Console 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

There are two ways that you can access the online help:

- Select **Help|Help Contents** from the menu bar. Sections on Artix Designer and the Artix Management Console appear in the contents panel of the Eclipse help browser.
- Press **F1** for context-sensitive help.

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

Artix Glossary

The Artix Glossary provides a comprehensive reference of Artix terminology. 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.

Additional Resources

The IONA Knowledge Base
(http://www.iona.com/support/knowledge_base/index.xml) contains helpful articles written by IONA experts about Artix and other products.

The IONA Update Center (http://www.iona.com/support/updates/index.xml) 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 (http://www.iona.com/support/index.xml).

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

Document Conventions

Typographical conventions

This book 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

This book 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**).

Artix CA WSDM Integration

Artix provides support for integration with Computer Associates Web Services Distributed Management (CA WSDM). This chapter provides an introduction, and shows how to configure CA WSDM integration in Artix applications.

In this chapter

This chapter includes the following sections:

Artix CA WSDM Observer

Overview

An Artix CA WSDM observer is a plug-in interceptor that integrates Artix with Computer Associates Web Services Distributed Management (WSDM) software. This section gives an architectural overview and lists the observed data.

Artix CA WSDM Observer

An Artix CA WSDM observer interceptor can sit on the client side or server side as shown in Figure 1.
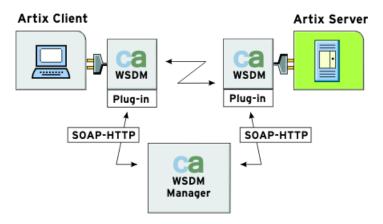

Figure 1: *CA WSDM Observer Architecture*

A CA WSDM observer operates as follows:

- Collects information about messages sent to observed services over any supported transports at both the server and client request interceptor level. It asynchronously reports this information to a CA WSDM service using SOAP over HTTP.

- Automatically registers all services it observes with CA WSDM by sending a service QName and a list of QNames of operations supported by a corresponding `portType` interface. This enables a CA WSDM operator to create *service groups*.

- Periodically polls a CA WSDM service for configuration updates. For example, CA WSDM transaction monitors can be enabled, which enable an operator to look at the raw input and output messages. The operator can check if it took an operation longer to complete its request, or if its request or response size was larger than expected.

Observed data

An Artix CA WSDM observer reports the following data to a CA WSDM service about any service operation:

- Operation name and namespace.
- Request and response size.
- Operation duration.
- Operation timestamp.
- Operation transaction identifier.
- Service port endpoint address.
- Client address (where the request came from).
- Request and response messages (if duration, request/response size monitors are enabled).
- User fault exception details.

Note: Some data may not be reported if it is not available at a request interceptor level for a given transport. For example, request and response size, or raw messages for CORBA services.

Configuring a CA WSDM Observer

Overview

You can enable an Artix CA WSDM Observer by adding a plug-in to your in `orb_plug-ins` list in your Artix server or client configuration scopes. All other CA WSDM configuration variables are optional. This section explains how to set all available options.

Setting the orb_plugins list

The CA WSDM Observer plug-in name is `ca_wsdm_observer`. To enable a CA WSDM observer, add this plug-in to your `orb_plugins` list in your server or client scope. For example:

```
# Artix configuration file

my_client_scope {
    orb_plugins = [ ..., "ca_wsdm_observer"];
    ...
};
```

Both client and server use the same endpoint, so specifying both gives more coarse-grained data. Typically, you would use either client or server only.

Specifying a minimum queue size

The minimum queue size specifies how many service request records must be available in a queue before a report is sent to a WSDM service. For example:

```
plugins:ca_wsdm_observer:min_queue_size = "6";
```

The default is 5. You should set this variable if your load is expected to be large. If this variable is too low, the observer may send reports too frequently, and if it is too high, the memory footprint may increase significantly.

Specifying a report wait time

The report wait time specifies how often reports should be sent in seconds. For example:

```
plugins:ca_wsdm_observer:report_wait_time = 10;
```

This variable is an alternative to min_queue_size, which instead specifies the frequency of reports on a time basis. This variable should be used with max_queue_size.

Specifying a maximum queue size

The maximum queue size specifies the maximum number of service request records that the observer queue can hold. For example:

```
plugins:ca_wsdm_observer:max_queue_size = "600";
```

The default is 500. New records are dropped when the queue size reaches this value. If report_wait_time is not set, this variable is ignored. In this case, reports are sent as soon as the queue size is equal to max_queue_size.

Automatically registering services

You can also specify whether the observer automatically registers observed services with a WSDM service. The default is:

```
plugins:ca_wsdm_observer:auto_register = "true";
```

If you have a large number of observed services, the runtime performance might be decreased because of equally large register service requests sent to a WSDM service.

You can set this variable to false and manually import service details from WSDL definitions into a WSDM console. However, this only works for SOAP-HTTP non-transient services. This is because WSDM can not import non-SOAP services described in WSDL, while Artix does not publish WSDL for transient services.

Specifying a handler type

A handler type specifies a value that identifies an Artix observer to a WSDM service. It should be above 200. The default is:

```
plugins:ca_wsdm_observer:handler_type = "217";
```

In addition, if you change the default, you must also update the following file with the new handler type:

```
WSDM-InstallDir/server/default/conf/WsdmSOMMA_Basic.properties
```

Entries in this file take a format of observertype.X=ArtixObserver, where X is the handler type value. The default entry is:

```
observertype.217=ArtixObserver
```

Specifying a configuration updates

To specify how often, in seconds, the observer should poll a WSDM service for configuration updates, use the following variable:

```
plugins:ca_wsdm_observer:config_poll_time
```

The default is 180 seconds (3 minutes). Configuration updates tell the observer whether transaction monitors have been enabled. If so, the observer copies input/output raw messages, and reports them to a WSDM service if duration or request/response size thresholds have been exceeded.

Further information

For a detailed example, see the CA WSDM demo in the following directory:

```
InstallDir\artix\Version\demos\integration\ca_wsdm
```

For more information on CA WSDM, see the Computer Associates website (http://www.ca.com).

For general information on Artix configuration, see Configuring and Deploying Artix Solutions.

Index

INDEX