Artix™ ESB

CA WSDM Integration Guide

Version 5.1, December 2007

Making Software Work Together™

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

IONA, IONA Technologies, the IONA logo, Orbix, High Performance Integration, Artix, FUSE, and Making Software Work Together are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries.

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

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

COPYRIGHT NOTICE

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

Copyright © 2001-2008 IONA Technologies PLC. All rights reserved.

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

Updated: January 29, 2008

Contents

Preface 5
- What is covered in this book 5
- Who should read this book 5
- Organization of this book 5
- The Artix Documentation Library 5

Chapter 1 Artix CA WSDM Integration 7
- Artix CA WSDM Observer 8
- Configuring a CA WSDM Observer 10

Index 13

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. It applies to Artix applications written using C++ and JAX-RPC (Java APIs for XML-Based Remote Procedure Call).

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 web site (http://www.ca.com).

Organization of this book

This book contains the following chapter:

- Chapter 1 introduces the architecture of the Artix integration with CA WSDM, and explains how to configure Artix for integration with CA WSDM.

The Artix Documentation Library

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

CHAPTER 1

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	page 8
Configuring a CA WSDM Observer	page 10

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.

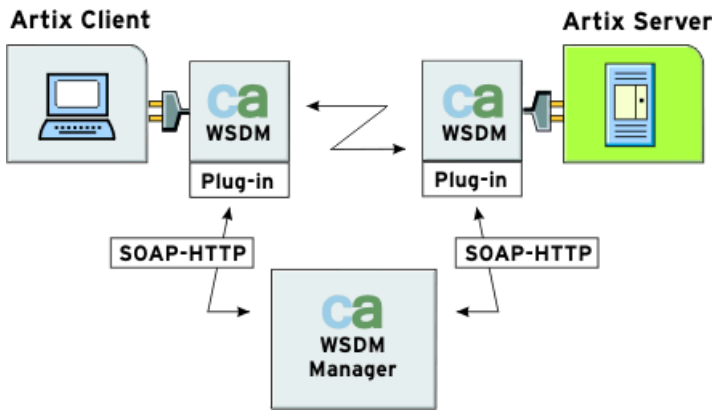

Figure 1: *CA WSDM Observer Architecture*

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 .cfg 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\Version\cxx_java\samples\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

A
architecture 8
Artix CA WSDM observer 8

C
CA WSDM 7
ca_wsdm_observer 10
client address 9
configuration updates 12

E
endpoint address 9

F
fault exception 9

H
handler type 12

M
maximum queue size 11
minimum queue size 10

O
observer queue 11
operation duration 9
operation name 9
operation timestamp 9
orb_plug-ins 10

P
plugins:ca_wsdm_observer:auto_register 11
plugins:ca_wsdm_observer:config_poll_time 12
plugins:ca_wsdm_observer:handler_type 12
plugins:ca_wsdm_observer:max_queue_size 11
plugins:ca_wsdm_observer:min_queue_size 10
plugins:ca_wsdm_observer:report_wait_time 11
portType 9

Q
QName 9

R
report wait time 11
request size 9
response size 9

S
service groups 9
service QName 9
SOAP over HTTP 8

T
transaction identifier 9