IONA

Artix™

Release Notes

Version 4.0 March 2006

Making Software Work Together™

Contents

Artix 4.0

In this document

This document contains the following sections:

New Features

The following features have been added in Artix 4.0:

- WS-RM Support
- Native Support for JMS
- Artix Orchestration
- Database Services
- Transaction Enhancements
- Eclipse-based UI for Artix for z/OS
- Artix Management Console
- Dynamic Testing Tool
- JMX Instrumented Java and C++ Runtime and Services
- Artix Container
- WS-Addressing
- New FTP Transport
- QoS Features
- Security Features

WS-RM Support

Message delivery is now guaranteed across the extended enterprise thanks to support for the Web Services Reliable Messaging (WS-RM) standard. For more details, see:

- The Configuring and Deploying Artix Solutions guide
- The WS-Reliable Messaging demo (`demos\advanced\wsrm`)

Native Support for JMS

Artix now supports Java Message Server (JMS) out of the box. Thanks to the inclusion of the ActiveMQ open source JMS broker, users no longer need to build or license their own JMS product. For more details, see:

- The Understanding Artix Contracts guide
- The ActiveMQ home page (http://www.activemq.org/)
- The SOAP over JMS demo (`demos\transports\soap_over_jms`)

Artix Orchestration

Design, deploy, and debug Business Process Execution Language (BPEL) services with Artix 4.0 Orchestration. Please contact your IONA sales representative for more information on this feature.

Database Services

You can now expose your databases as Web services using Artix Designer or command line tools. For more details, see:

- The "Working with database projects" section of the Artix Designer help
- The Database Web Services demos (`demos\db_service\sql` and `\stored_procedure`)

Supported databases The following DBMS are supported:

- MySQL 4.0 and higher
- Oracle 8.1.7 and higher
- Sybase 12.5 and higher

JDBC details TThe table below shows the JDBC driver class and JDBC URL used for each database management system:

DBMS	JDBC Driver	JDBC URL
MySQL	`com.mysql.jdbc.Driver`	`jdbc:mysql://hostname:port/database`
Oracle	`oracle.jdbc.driver.OracleDriver`	`jdbc:oracle:thin:@hostname:port:database`
Sybase	`com.sybase.jdbc3.jdbc.SybDriver`	`jdbc:sybase:Tds:hostname:port/database`

Supported data types The following tables list the data types supported for each database system, how they are mapped to JDBC data types and their corresponding Java data types in Artix 4.0. Where no corresponding Java data type is given, the data type is not supported in Artix 4.0.

Table 1: *Supported MySQL Data Types*

MySQL Data Type	JDBC Data Type	Artix 4.0 Java Data Type
MySQL 4.0 and higher		
`BIT`	`BIT`	`boolean`
`BOOL`	`BIT`	`boolean`
`TINYINT`	`TINYINT`	`int`
`SMALLINT`	`SMALLINT`	`int`
`MEDIUMINT`	`INTEGER`	`int`
`MEDIUMINTUNSIGNED`	`INTEGER`	`int`
`INT`	`INTEGER`	`int`
`INT UNSIGNED`	`INTEGER`	`int`
`INTEGER`	`INTEGER`	`int`
`INTEGERUNSIGNED`	`INTEGER`	`int`
`BIGINT`	`BIGINT`	`java.math.BigDecimal`
`BIGINTUNSIGNED`	`BIGINT`	`java.math.BigDecimal`
`FLOAT`	`REAL`	`double`
`DOUBLE`	`DOUBLE`	`double`
`DECIMAL`	`DECIMAL`	`java.math.BigDecimal`

Table 1: *Supported MySQL Data Types (Continued)*

MySQL Data Type	JDBC Data Type	Artix 4.0 Java Data Type
NUMERIC	DECIMAL	java.math.BigDecimal
REAL	DOUBLE	double
DATE	DATE	java.util.Calendar
DATETIME	TIMESTAMP	java.util.Calendar
TIMESTAMP	TIMESTAMP	java.util.Calendar
TIME	TIME	java.util.Calendar
YEAR	DATE	java.util.Calendar
CHAR	CHAR	java.lang.String
VARCHAR	VARCHAR	java.lang.String
BLOB	LONGVARBINARY	byte[]
TINYBLOB	VARBINARY	byte[]
MEDIUMBLOB	LONGVARBINARY	byte[]
LONGBLOB	LONGVARBINARY	byte[]
TEXT	LONGVARCHAR	java.lang.String
TINYTEXT	LONGVARCHAR	java.lang.String
MEDIUMTEXT	LONGVARCHAR	java.lang.String
LONGTEXT	LONGVARCHAR	java.lang.String
ENUM	CHAR	——
SET	CHAR	——
MySQL 4.1.0 and higher		
BOOLEAN	BIT	boolean
MySQL 4.1.2 to 5.0.19		
BINARY	BINARY	byte[]

Table 2: *Supported Oracle Data Types*

Oracle Data Type	JDBC Data Type	Artix 4.0 Java Data Type
Oracle 8.1.7 and higher		
BFILE	—	—
BLOB	BLOB	byte[]
CHAR	CHAR	java.lang.String
CLOB	CLOB	java.lang.String
DATE	TIMESTAMP	java.util.Calendar
DOUBLE PRECISION	NUMERIC	java.math.BigDecimal
FLOAT(n)	NUMERIC	java.math.BigDecimal
INTEGER	NUMERIC	java.math.BigDecimal
LONG	LONGVARCHAR	java.lang.String
Long raw	LONGVARBINARY	---
NCHAR	CHAR	java.lang.String
NCLOB	CLOB	java.lang.String
NUMBER(p, s)	NUMERIC	java.math.BigDecimal
NUMBER	NUMERIC	java.math.BigDecimal
NVARCHAR2	VARCHAR	java.lang.String
RAW	—	—
REF	—	—
SMALLINT	NUMERIC	java.math.BigDecimal
Oracle9i and higher		
TIMESTAMP	TIMESTAMP	java.util.Calendar
TIMESTAMP WITH LOCAL TIME ZONE	—	—
TIMESTAMP WITH TIME ZONE	—	—

Table 2:   *Supported Oracle Data Types (Continued)*

Oracle Data Type	JDBC Data Type	Artix 4.0 Java Data Type
VARCHAR2	VARCHAR	java.lang.String
XMLType	—	—
Oracle10g only		
BINARY_FLOAT	—	—
BINARY_DOUBLE	—	—

Table 3:   *Supported Sybase Data Types*

Sybase Data Type	JDBC Data Type	Artix 4.0 Java Data Type
binary	BINARY	byte[]
bit	BIT	boolean
char	CHAR	java.lang.String
date	DATE	java.util.Calendar
datetime	TIMESTAMP	java.util.Calendar
decimal	DECIMAL	java.math.BigDecimal
float	FLOAT	double
image	LONGVARBINARY	byte[]
int	INTEGER	int
money	DECIMAL	java.math.BigDecimal
nchar	CHAR	java.lang.String
numeric	NUMERIC	java.math.BigDecimal
nvarchar	VARCHAR	java.lang.String
real	REAL	float
smalldatetime	TIMESTAMP	java.util.Calendar
smallint	SMALLINT	int

Table 3: *Supported Sybase Data Types (Continued)*

Sybase Data Type	JDBC Data Type	Artix 4.0 Java Data Type
smallmoney	DECIMAL	java.math.BigDecimal
sysname	VARCHAR	java.lang.String
text	LONGVARCHAR	java.lang.String
time	TIME	java.util.Calendar
timestamp	VARBINARY	byte[]
tinyint	TINYINT	int
unichar	CHAR	java.lang.String
univarchar	VARCHAR	java.lang.String
varbinary	VARBINARY	byte[]
varchar	VARCHAR	java.lang.String

Transaction Enhancements

Artix features a unique transactions API, designed to be compatible with a variety of different underlying transaction systems. The following transactions features are new in 4.0:

- XA support
- Recoverable WS-AT

XA support C++ developers can now expose Artix as an XA resource manager by registering the Artix XA switch with a third-party XA transaction manager.

Recoverable WS-AT The recoverable WS-AomicTransaction (WS-AT) transaction system is now layered on top of the OTS Encina transaction engine to give enterprise-level transaction support. The relevant OTS plug-ins are automatically loaded when WS-AT is enabled. For more details, see:

- The Artix Transactions Guide, C++ and Java
- The Artix for J2EE guide

Eclipse-based UI for Artix for z/OS

The GUI tool for Artix for z/OS has been rebuilt as an Eclipse plug-in and is fully integrated with the core Artix Designer tool. For more details, see the "Working with Artix Designer for z/OS" section of the Artix Designer online help.

Artix Management Console

A new Eclipse-based tool allows you to manage your Artix Web services. Accessible either from within Artix Designer or as a standalone application, this simple console takes full advantage of the Artix container functionality. For more details, see the Artix Management Console online help.

Dynamic Testing Tool

Artix Designer now includes a dynamic test client that uses WSDL contracts to allow users to test their web services. This test tool offers extensive type support over all Artix bindings and transports. For more details, see the "Testing a service" section of the Artix Designer online help.

JMX Instrumented Java and C++ Runtime and Services

You can now use Java Management Extensions (JMX) to monitor and manage key Artix runtime components, including bus, services, and ports, both locally and remotely. Using any JMX-compliant client you can view the status and configuration of any bus instance, stop or start a service, and change bus logging levels dynamically. You can also inspect interceptors within the interceptor chain of a selected bus.

The Artix locator and session manager services have also been instrumented and provide an additional set of properties on top of those common to all services.

For more details, see Configuring and Deploying Artix Solutions.

Artix Container

The following enhancements have been made to the Artix container:

- Artix runtime logging levels can be changed dynamically (see Configuring and Deploying Artix Solutions).
- Ability to gather performance metrics of services running in container (see the Artix Management Console online help).

- Container can be installed as a Windows service (see Configuring and Deploying Artix Solutions).
- Full service lifecycle means services can be deployed and removed dynamically (see the Developing Artix Applications in C++ and Java guides).

WS-Addressing

Artix 4.0 fully supports the latest version of WS-Addressing for improved interoperability. This replaces the proprietary addressing used in previous versions of Artix. We advise users to update their WSDL to take advantage of this standardized addressing type. For more details, see:

- The Developing Artix Applications in C++ and Java guides
- The Configuring and Deploying Artix Solutions guide
- "Migration Notes" on page 11

New FTP Transport

Artix now fully supports the File Transfer Protocol (FTP). For more details, see:

- The Understanding Artix Contracts guide
- The SOAP over FTP demo (`demos\transports\soap_over_ftp`)

QoS Features

The following quality of service features have been added:

- High availability
- Locator service improvements

High availability Artix now allows for a duplicate master resolution strategy and simplified HA configuration. See Configuring and Deploying Artix Solutions for more details.

Locator service improvements The following enhancements have been made to the locator service:

- A new plug-in removes the need to reference the locator in client-side code.
- The query functionality has been extended.

For more details, see:

- The Artix Locator Guide
- The Locator Query demo (`advanced\locator_query`)

Security Features

The following security features are new in Artix 4.0. See the Artix Security Guide for more details:

- Certificate revocation lists
- Artix bindings supported
- Authentication cache configuration settings
- Independent HTTPS policies
- Support for TLS load balancing with corbaloc lists

Certificate revocation lists Artix now allows you to attach a Certificate Revocation List (CRL) to your applications. A CRL is a list of certificates considered to be invalid by a certification authority. When you deploy a CRL file to a secure application, the application automatically rejects the certificates that appear in the list.

Artix bindings supported Artix bindings and transports are now supported in the communications channels between the Artix security plug-in and the Artix security service.

Authentication cache configuration settings Artix server processes that use the Artix security plug-in for authentication and authorization can specify a single authentication cache instance that is shared between multiple services registered with a single bus, or even services that are registered with separate bus instances.

Independent HTTPS policies In previous releases, the https plug-in used the `iiop_tls` plug-in policies. Setting the policy values for one plug-in affected the other plug-in. In Artix 4.0 the `https` and `iiop_tls` polices are independent.

Support for TLS load balancing with corbaloc lists Services connecting to a load balancing cluster of Artix security services can now specify the connection details using a `corbaloc` URL.

Migration Notes

Users upgrading from Artix 3.0.x to version 4.0, should note the following compatibility issues:

- Dropped platforms
- Binary Compatibility
- Security Compatibility
- Artix Designer
- Artix Services
- Endpoint References
- WSDL Compatibility—XML Binding
- WSDL Compatibility—JMS Transport

Dropped platforms

Artix no longer supports the following platforms:

- x86/Red Hat Linux 8
- x86/Red Hat Linux 9

For full platform support and system requirement details, see the "Installation Prerequisites" chapter of the Artix Installation Guide.

Binary Compatibility

Artix 4.0 is source compatible, but not binary compatible with Artix 3.0.x. You should regenerate Artix stub code and recompile application code after upgrading.

Security Compatibility

Artix developers and administrators should be aware of the following compatibility issues when migrating applications from Artix 3.0.x to Artix 4.0:

- Client authentication requirement
- Certificate constraints requirement
- Enable security flag defaults to true
- Enable authorization flag defaults to true
- Enable SSO flag deprecated
- Load balancing policy defaults to per-server

Client authentication requirement The Artix security service requires client authentication of TLS in Artix 4.0. All Artix services that have the Artix security plug-in loaded need to specify a client certificate in order to connect to the server. The client certificate can be specified using the following configurations:

```
principal_sponsor:use_principal_sponsor = "true";
principal_sponsor:auth_method_id = "pkcs12_file";
principal_sponsor:auth_method_data =
    ["filename=%{INSTALL_DIR}/%{PRODUCT_NAME}/%{PRODUCT_VERSION}/demos/security/certificates/tls/
    x509/certs/services/administrator.p12",
password_file=%{INSTALL_DIR}/%{PRODUCT_NAME}/%{PRODUCT_VERSION}/demos/security/certificates/tls/
    x509/certs/services/administrator.pwf}"];
```

Certificate constraints requirement Incoming requests to the security service are now subject to a further security check in that the received TLS credentials' peer cert chain is examined to ensure that it matches one of the constraints specified in a list. Two new configuration variables have been created to set the certificate constraints:

```
policies:security_server:client_certificate_constraints
policies:external_token_issuer:client_certificate_constraints
```

Enable security flag defaults to true The enable security flag, which can be set either via the `policies:asp:enable_security` configuration variable, or the `enableSecurity` bus-security port extensor, defaults to `true` in Artix 4.0. To revert to Artix 3.0.x behavior, set the following configuration variable at the appropriate bus scope:

```
policies:asp:enable_security = "false";
```

Enable authorization flag defaults to true The enable authorization flag, which can be set either via the `policies:asp:enable_authorization` configuration variable, or the `enableAuthorization` bus-security port extensor, defaults to `true` in Artix 4.0. To revert to Artix 3.0.x behavior, set the following configuration variable at the appropriate bus scope:

```
policies:asp:authorization = "false";
```

Enable SSO flag deprecated The enable single sign-on (SSO) flag, which could be set either via the `policies:asp:enabe_sso` configuration variable, or the `enableSSO` bus-security port extensor, is deprecated. This variable has no effect on the runtime behavior of Artix 3.0.x and 4.0 deployments.

Load balancing policy defaults to per-server The Artix security plug-in supports load balancing authentication and authorization requests made to the security service. In Artix 3.0.x, the default load balancing algorithm selects a new security service per request.

Artix 4.0 adds support for per-server load balancing, whereby a security service instance is selected by the load balancing mechanism at the first request from the security plug-ins to the server. Per-server load balancing is now the default load balancing policy.

To revert to 3.0.x behavior, set the following Artix configuration variable at the appropriate bus scope in the Artix configuration file:

```
policies:asp:load_balancing_policy = "per-request";
```

Artix Designer

You should be aware of the following when upgrading to Artix Designer 4.0:

- Project compatibility
- Dependency on CDT

Project compatibility When you import a project created in Artix Designer 3.0.x into an Artix Designer 4.0 workspace, you need to regenerate any code that the project contains.

Before you regenerate Java code, you should also delete the `.classpath` file from the root of the project folder, as it points to an Artix 3.0 installation. This file will be recreated with the correct values when you run the code generation configuration.

> **Note:** To preserve any edits that were made to the code, ensure that the **Merge generated code** checkbox is selected in the C++ and Java sections of the Artix Designer preferences panel. Also, be sure to use the same name for the code generation configuration as appears under the `ProjectName/outputs/applications` folder.

Dependency on CDT Artix Designer now requires the Eclipse C/C++ Development Tools (CDT), regardless of whether you plan to do any C++ development. The Artix installer includes the CDT plug-ins, but if you install Artix Designer into your own Eclipse platform, you must have CDT 3.0.2 or higher installed.

Artix Services

Artix 3.0.x applications can use the 4.0 versions of Artix services, such as the locator and session manager, without having to regenerate code. However, these services have been extended for Artix 4.0, so we recommend that you upgrade your applications to take advantage of the changes in Artix 4.0.

If you continue to use version 3.0.x applications, you need to configure them to use the old version of the `locator.wsdl` and `session-manager.wsdl`, which still ship with Artix 4.0. See the migration sections of the Artix Locator Guide and Artix Session Manager Guide for more details.

Endpoint References

All Artix services and tools now comply with the WS-Addressing standard for endpoint references. We advise you to migrate your code to the WS-Addressing APIs from the proprietary Artix `Reference` type APIs, which are now deprecated. All demos that previously used Artix `Reference` types now use WS-Addressing.

WSDL Compatibility—XML Binding

The target namespace for the XML binding has changed to `http://celtix.objectweb.org/bindings/xmlformat` to be compatible with Celtix. Artix 4.0 runtime can handle both the new and old XML namespaces.

WSDL Compatibility—JMS Transport

The JMS transport WSDL extensors have been changed to support interoperability with the Celtix open source Java ESB. See the `transports\soap_over_jms` demo for an example of the new WSDL.

In addition, we now provide JMS broker support out of the box. This means any Artix 3.0.x WSDL that uses a JMS transport does not work in Artix 4.0.

You need to upgrade your WSDL to comply with the latest WSDL extensors. We also recommend that you consider using the ActiveMQ JMS broker we ship as part of Artix 4.0 which removes any dependency on a third-party product.

You can update your WSDL by:

- Transforming the WSDL into the appropriate format using the XSLT stylesheets that we provide. See "Transforming the WSDL" on page 15.
- Redesigning the WSDL using Artix Designer, which has been upgraded to support the latest WSDL extensions.

- Hand editing the WSDL by consulting the WSDL examples in the demos.

Transforming the WSDL We provide the following XSLT stylesheets in the
`InstallDir\artix\4.0\etc\xslt\utilities\jms` directory:

- `oldjmswsdl_to_newjmswsdl.xsl`
- `anyjmsprovider_to_activemq.xsl`

You can transform the WSDL using these stylesheets and the `xslttransform`
script, which you can find in the `InstallDir/artix/4.0/bin` directory.

Task	Command
Convert 3.0-format WSDL to 4.0-format	`xslttransform -XSL oldjmswsdl_to_newjmswsdl.xsl -In` `OldFileName.wsdl -OUT NewFileName.wsdl`
Convert 3.0-format WSDL to 4.0-format and update JMS provider to ActiveMQ using default values	`xslttransform -XSL oldjmswsdl_to_newjmswsdl.xsl -In` `OldFileName.wsdl -OUT NewFileName.wsdl -param` `updateToActiveMQ true`
Convert 3.0-format WSDL to 4.0-format and update JMS provider to ActiveMQ using non-default values	`xslttransform -XSL oldjmswsdl_to_newjmswsdl.xsl -In` `OldFileName.wsdl -OUT NewFileName.wsdl -param` `updateToActiveMQ true -param userDefDestinationName` `JmsDestinationJndiName`
Convert 4.0-format WSDL using third-party JMS broker to use Active MQ	`xslttransform -XSL anyjmsprovider_to_activemq.xsl -In` `OldFileName.wsdl -OUT NewFileName.wsdl`

When updating the JMS provider to ActiveMQ using non-default values, you can
set the following parameters:

Parameter	Description
`userDefConnectionFactoryName`	Specifies the JNDI name bound to the JMS connection factory to use when connecting to the JMS destination.
`userDefDestinationName`	Specifies the JNDI name bound to the JMS destination to which Artix connects.
`userDefInitialFactory`	Specifies the initial factory setup class, which sets up the initial context based on the `jndi.properties` and WSDL configuration.
`userDefUrl`	Specifies the URL of the JNDI service where the connection information for the JMS destination is stored.

Documentation Updates

The following changes have been made to the Artix documentation for version 4.0:

- Using Artix Designer
- Building Service-Oriented Architectures with Artix
- Understanding Artix Contracts
- Developing Artix Applications in Java
- Configuring and Deploying Artix Solutions
- Artix Locator Guide
- Artix Session Manager Guide
- Artix Transactions Guide, C++ and Java
- Artix for J2EE
- BMC and IBM Tivoli Integration Guides
- Online Help
- Artix Glossary
- Artix Configuration Reference
- WSDL Extension Reference

Using Artix Designer

This new book aims to provide an overview of the functionality available in Artix Designer.

Building Service-Oriented Architectures with Artix

This new book discusses what makes a service-oriented architecture (SOA), the advantages of SOA to integration, and how Artix facilitates the deployment of an enterprise quality SOA.

Understanding Artix Contracts

This addition to the Artix library consists of material migrated from the *Designing Artix Solutions* guide, which is now deprecated. The following changes were made during the migration process:

- Added chapter on FTP transport
- Updated chapters on basic WSDL concepts
- Updated JMS chapter to reflect new WSDL extensions

- Updated Router chapter to discuss content based routing
- Updated XSLT chapter to clarify WSDL to XML mapping

Developing Artix Applications in Java

This book has been updated as follows:

- Highlights the use of the Artix container model
- Reference section updated to use the WS-Addressing `EndpointReference`
- Transaction section moved to *Artix Transactions Guides*
- Context section updated for clarity
- Updated type support
- New chapter on adding JMX instrumentation to a service

Configuring and Deploying Artix Solutions

This book contains the following new chapters:

- Deploying Reliable Messaging
- Publishing WSDL Contracts
- Managing the Artix Runtime with JMX

The following chapters have been updated:

- Artix Logging:
  - Message snoop logging
  - Rolling log file configuration
  - List of logging subsystems
  - Dynamic logging commands
- Deploying High Availability:
  - Simplified replication configuration
  - Minority master and random endpoint configuration.
  - Duplicate master resolution
- Deploying Services in an Artix Container:
  - Running a container as a Windows service
  - WS-A endpoint references

- Deploying an Artix Router
    - ♦ Defining routes in an Artix deployment descriptor
    - ♦ Router optimizations
- Accessing Contracts and References
    - ♦ WS-A endpoint references

The material on the locator and session manager has been removed and added to the new guides for these services

Artix Locator Guide

This is a new book on how to use the Artix locator service. It is based on material that existed in several books in the Artix 3.0 library.

Artix Session Manager Guide

This is a new book dealing with the Artix session manager service. It is based on material that existed in several books in the Artix 3.0 library and contains the following new material:

- Discussion about the session manager WSDL contract
- C++ and Java client updates based on WS-Addressing
- Migrating from Artix 3.0 to Artix 4.0
- Writing an Axis client for a session managed service
- Writing a .NET client for a session managed service
- Writing your own session management policy
- Use cases

Artix Transactions Guide, C++ and Java

These two new guides are based on material that existed in several books in the Artix 3.0 library. They explain how to program and configure Artix transactions in C++ and Java.

Artix for J2EE

The main additions to this book are:

- A new chapter on transactions, including global (XA) transactions
- Updates to the "Deploying Artix J2EE Connector" chapter—in particular, new information about using a classloader firewall

- New API implementation that facilitates transparent rebind and the use of the locator service

BMC and IBM Tivoli Integration Guides

Both books have been updated with information on using Artix Designer to generate Enterprise Management System files.

Online Help

The Artix Designer online help has been expanded to include the following new sections:

- Working with Artix for z/OS
- Using the Artix Management Console
- Exposing a database as a web service
- Deploying a service into a container
- Testing a service
- Launching Artix applications and containers

Artix Glossary

All the glossary material that previously existed in the Artix library has now been gathered together into a single book.

Artix Configuration Reference

This book contains new configuration variables for the following:

- Client Side High availability
- Database Environment
- FTP
- JMX
- Message Snoop
- Router
- Locator
- Web Services Addressing
- Web Services Reliable Messaging

WSDL Extension Reference

This new book is a reference to all of the Artix-specific WSDL extensions used in Artix contracts.

Known Issues

The following are known issues in Artix 4.0:

- 64-bit JVM
- Artix Container
- Artix Security
- Session Manager
- Artix for J2EE
- Artix Designer
- Demos

64-bit JVM

WARNING: The Solaris and Linux versions of the Artix installer installs 32-bit Java Virtual Machines (JVMs) only. Users working on 64-bit versions of these platforms should download and install a 64-bit JVM from:

```
http://java.sun.com/products/archive/j2se/1.4.2_04/index.html
```

Artix Container

If the Artix container cannot find `container.wsdl` in the expected location it crashes with no useful message. To identify whether the crash was caused by this problem, turn on logging in the container, try to start the container again, and look for the following in the log, just before the core-dump:

```
Service wsdl can not be read from
    InstallDir/artix/4.0/wsdl/container.wsdl
```

To workaround, either move the `container.wsdl` to the expected location or set the `bus:initial_contract:url:container` variable in the Artix configuration file to the actual location.

Artix Security

The following are known security issues in Artix 4.0:

- Multiple HTTPS endpoints in a single bus
- C++ server rejects SSO tokens
- Secure WSDL publish limitations
- Secure container limitations
- Startup errors
- SSL client cannot invoke against an Artix 2.1 server

Multiple HTTPS endpoints in a single bus If you specify HTTPS settings for multiple HTTPS endpoints hosted by a single Artix bus, there is a risk that incorrect TLS settings belonging to another endpoint could be used in error while accepting a new TLS connection or creating a new outbound TLS connection.

This is a security risk for customers who use endpoint level granularity for Artix HTTPS configuration. For more details, see the Artix Security Advisory Archive.

C++ server rejects SSO tokens A C++ server configured to accept SSO tokens rejects tokens that do not contain SAML assertion data. To workaround this issue, set the `plugins:gsp:accept_asserted_authorization_info` configuration variable to `false`. This forces the server to reauthenticate the delegated SSO token by contacting the security service.

Secure WSDL publish limitations The following limitations apply to the use of secure WSDL publish. See the Artix Security Guide for more details:

- You cannot specify separate keys and certificates for the `wsdl_publish` endpoint, except via bus-level configuration, which may be shared with other SSL endpoints serviced by the bus.
- If a `wsdl_publish` endpoint services a WSDL file containing an `import` element based on a URL, the URL must be in HTTPS format.

Secure container limitations The following limitations apply to the Artix container when it is deployed as a secure application. See Configuring and Deploying Artix Solutions for more details:

- The Artix container does not support mixed secure and insecure services. If the container is deployed as a secure application, its hosted services must also be secure.

- The it_container_admin application does not support the -host or -port flags to locate a secure container instance. Instead, applications must use the -container flag, specifying the URL for the target container instance.

Startup errors When the Artix security service starts, the following errors may be logged:

```
date time [host:port] (IT_BUS.JNI.JBUS:0) E - Failed to get Service
   IT_ISFX.AuthenticationServiceSOAPService
   xmlns="http://schemas.iona.com/idl/isfx_authn_service.idl"
date time [host:port] (IT_BUS.JNI.JBUS:0) E - Failed to get Service
   IT_ISFX.ServerAccessDecisionServiceSOAPService
   xmlns="http://schemas.iona.com/idl/isfx_server_access_decision.idl"
```

These errors are benign. The Artix security service will continue to function properly.

SSL client cannot invoke against an Artix 2.1 server An Artix 4.x client using SSL fails when invoked against an Artix 2.1 server. You need to upgrade the server to Artix 2.1.7 to fix this issue.

Session Manager

The WS-Addressing endpoint reference API provided by Artix Java does not allow you to query an endpoint for its service name.

Artix for J2EE

The following are known issues in Artix for J2EE:

- Deploying multiple J2EE Connectors to WebLogic
- J2CA inbound global transactions

Deploying multiple J2EE Connectors to WebLogic In order to deploy multiple instances of Artix J2EE Connector to a WebLogic server, you need to add a list of JAR files to the CLASSPATH in the WebLogic server start script. Please contact support@iona.com for more details.

J2CA inbound global transactions If your application needs to use J2CA inbound global transactions, you must add the following configuration variables to the J2EE ArtixConnector scope in the configuration file, where *PortNumber* is the value of a free port on your machine:

```
poa:j2ee_rm:direct_persistent="true";
poa:j2ee_rm:well_known_address:port="PortNumber";
```

By default the variables should be added to the j2ee.tx.xa scope in $IT_CONFIG_DOMAINS_DIR/artix.cfg

Artix Designer

The following are known issues in Artix Designer:

- Code generation
- Running JMS-based applications
- Service test
- Updating Artix Designer
- Importing demos

Code generation If you create a client and server code generation configuration, configuring the client and servers to use the locator or session manager services does not work.

Running JMS-based applications If you generate a client or server from a WSDL file that uses a JMS transport, the application will throw the following exception when you run it from the **Run|Run** menu option:

```
javax.xml.rpc.ServiceException: Could not create Dynamic Proxy,
    reason:com.iona.jbus.BusException: com.iona.jbus.ntv.NativeBusException: No transport factory
    available for port:  SoapPort
    at com.iona.jbus.jaxrpc.JBusService.getPort(JBusService.java:37)
    at com.iona.artix.jms.GreeterDemo.main(Unknown Source)
Exception in thread "main"
```

To work around this, you need to add the location of the Artix license file to the launch configuration's classpath, as follows:

1. From the menu bar, select **Run|Run**.
2. In the Run window, click the **Classpath** tab.
3. Select the **User Entries** node in the Classpath tree and click **Advanced.**

4. In the Advanced Options dialog, select **Add External Folder** and click **OK**.

5. In the Folder selection dialog, browse to the folder containing your Artix license file and click **OK**.

6. Click **Run** to run the application

Service test When creating a service launch configuration, if you choose to test a service running in the Management Console, you can only access a service running in the same workspace.

In addition, the dynamic service test client does not support the following XML Schema types:

- The `union` and `list` simple types
- Anonymous `element` types

Updating Artix Designer When you use the Eclipse Update Manager to update Artix Designer for the *first time*, you must follow the steps below:

1. From the **Help** menu, select **Software Updates|Find and Install**. The Install/Update wizard launches.

2. In the Feature Updates panel, select **Search for new features to install** and click **Next**.

3. In the Update Sites to visit panel, click the **New Remote Site** button

4. Select **IONA Artix Designer** from the Sites to include in search list and click **Finish**.

5. In the Search Results panel, select the version of the Artix Designer feature you want and then click **Next**.

6. Click **Finish** to begin the update.

For subsequent updates, follow the steps described in the "Updating Artix Designer" topic in the online help.

Importing demos When importing a demo into Artix Designer you are asked to select a language (C++ or Java). On importing, the demo is built in the specified language and the demo artifacts for both languages are imported into the project. Subsequently importing the same demo and specifying the other language has no effect. If you want to build the demo in that language you must do so manually.

Demos

There are known issues with the following Artix 4.0 demos:

- Content-based routing
- Session manager

Content-based routing The `routing.wsdl` file in the content-based routing demo (`routing\content_based`) fails validation with the `schemavalidator`. The WSDL is also flagged as invalid by Artix Designer. If you modify the WSDL so that is accepted by Artix Designer, it fails at runtime.

Session manager The Java client in the session manager demo (`advanced\session_management`) uses the deprecated proprietary Artix `Reference` type.

Fixed Bugs in Artix 4.0

The following bugs have been closed in Artix 4.0:

Bug #	Description
68705	ISF should allows trusted principals to be accepted by ISF.
68968	Artix 2.0.3 - these messages are incorrectly appearing in artix switch log file.
69134	Ability to tweak logging dynamically
69378	Proxies are not thread safe
69455	Artix clients should use session timeout in SAML messages for the cache timeout
69469	Artix demo suite should include a demo highlighting how to set up timeout values
69482	Improve locator lookup-by-name capabilities
69483	WSE 2.0 Complaint .NET Plugin
69510	Server acknowledges a client closing its connection with a warning logged on ATLI2 subsystem which should rather be logged as an INFO

Bug #	Description
69730	Artix rolling log feature needs to conform to the application standard
69737	Artix should provide ftp/tftp transport
69746	Artix needs to support the validation of schema constraint
69786	Improvements to JAX-RPC Soap message Handler approach
69840	By default Artix opens a listener on all network interfaces. This needs to be configurable.
69841	Artix Router coredumps with mismatched IDL/WSDL
69850	Architecture overview document needed
69873	HA Locator election for single instance
69915	`wsdltojava -plugin` *name* option needs additional documentation
69930	Artix allows incorrect SOAP message to be dispatched
69935	`HTTP_CLIENT_OUTGOING_CONTEXTS` not explicitly created by Artix
69944	Artix process stops taking requests after handshake error and `wsdl_publish` plugin is loaded
69963	Need to set JMS Pool Highwater Mark
69978	Artix processes don't handle wrong format cfg files very well
69983	SendTimeout/ReceiveTimeout values lost in the references returned by the locator
69996	Exception running HTTPS Artix 3.0.1 client against Artix 2.1.6 server
70001	Context cannot be found in the request interceptor
70009	JMS transport should automatically try reconnecting when Artix process starts up and initial connection attempt has failed.
70035	If the `principal_sponsor` method of setting certs is used over `at_http` an Artix SOAP to CORBA switch throws an exception
70046	Lack of Encina license causes Artix JCA connector to fail

Bug #	Description
70105	Artix Java server crashes on Solaris when receiving malformed SOAP request (via HTTP listener)
70106	Artix documentation should include a section listing different issues that may arise when compiling Artix generated IDL using CORBA 2.1 compliant ORBs
70120	Need JMS reply queues that are static for Artix
70150	`idltowsdl` generates illegal `nillable="true"` attribute in `corba:alias`
70155	Move configuration in `artix-secure.cfg` to global scope
70156	Using `javatowsdl` on J2EE EJB interface fails as a number of Java types commonly used in J2EE are currently not mapped to WSDL.
70161	Artix 3.0.2 `artix_env` script cannot be loaded inside of another script using `/bin/sh`
70165	Artix Designer needs to be enhanced so that when the Edit Port dialog is brought up, there are fields available for adding the username and password.
70168	Problem deploying a plug-in in the Artix Container (Artix 3.0.2)
70169	When you call `setType()` with strings, it sometimes works
70175	WSDL compile problems
70193	Artix needs to provide more detailed level of logging for baltimore SSL handshakes.
70200	Add capabilities in Artix to allow registration of the `it_container` as a Windows service
70205	Using `javatowsdl` on SBC EJB interface cannot map to WSDL a number of their Java types which are non-JavaBean
70229	If SSO is turned on the BusContext is not updated with the principal
70235	Error in the log "not been defined in the schema to be referenced"

Bug #	Description
70251	Customer wants clients in demo to have comments in them explaining that its not necessary to parse the arguments passed in via `argv`
70265	Personas documentation issues
70266	Artix 3.1 documentation issues
70268	Issues with Artix Getting Started document
70269	Issues with Artix Designing Solutions Document
70270	Artix documentation set should include an Architecture Guide
70274	`wsdltocorba` produces `java.lang.StackOverflowError` with no further error message
70277	Allow Artix users to configure JMS Temp Queue Pool parameters
70278	Artix to clean up JMS Temp Queues upon Timeout
70282	Artix 3.03 security/https demo needs to be updated to no longer use `.pem` certificates.
70287	In Artix 3.0.x, `wsdltocorba` should issue a warning when it detects recursions in the XML Schema types that cannot be mapped to IDL.
70296	Crash in `IT_Bus::WSDLPublishCallback::dispatch`
70298	Artix 3.0.x client has problem handling non-standard SOAP fault code from third-party server
70307	`wsdltocorba` does not support Mutual Recursive Structures
70314	`IT_Bus::LocalXMLResolver::get_resource()` core in mt client
70315	`IT_String::find` core in mt client
70316	`IT_Bus::String::compare()` core in mt client
70317	`IT_Bus::XMLResolverRegistryImpl::get_resource()` core in mt client
70321	Artix server spins when processing invalid SOAP request

Bug #	Description
70339	Writing an any with an undefined namespace prefix causes the JVM to core dump
70347	Artix does not support the default= attribute at this time. It does not handle WSDLs containing this attribute correctly for enums and fails to compile the generated Java code due to incorrect code generation.
70349	`wsdl2cpp` generate code missing some variable declaration
70356	Artix cannot read a WSDL from Windows from a UNIX file system
70375	WebSphere shuts down gracefully due to `com.iona.jbus.TransportException`
70378	Remove unnecessary import of `BusSecurityLevel` in document examples and demos
70385	When you have IDL with a long defined within a `struct`, and you run `idltowsdl` on it, the type information is incorrectly generated.
70390	`schemavalidator` giving false error messages
70392	Cleanup `artix-secure.cfg` from using `plugins:asp:security_type`
70393	*Artix Security Guide* mixes `plugins:asp:security_level` with `plugins:asp:security_type`
70396	Creating first element with inline `complexType` results in incorrect namespace prefixes
70397	Unable to add new operations or edit existing operations from GUI
70398	XSD file incorrectly declared erroneous
70399	`schemavalidator` reports null pointer exception on WSDL containing forward definitions of types
70400	Unable to import an XSD file into an XSD file
70401	Unable to enter relative path in Import wizard
70402	Unable to create new types or edit existing types from Designer

Bug #	Description
70404	Cannot configure HTTPS information programmatically on the server side
70406	Compiling generated Java code gives "incompatible type" error - (`boolean -vs- java.lang.Boolean`)
70408	`is2.properties.FILE` template contains undocumented property `com.iona.security.azmgr.PersistencePropertyFileName`
70411	no documentation of variable `password_retrieval_mechanism:inherit_from_parent`
70413	Need for `actionrolemapping.dtd` in the same directory as the `action_role_mapping` file is not documented
70415	Retrofit Artix 4.0 JCA connection caching mechanism for JMS to Artix 3.03
70434	Unable to validate first use of prefix within an XSD file
70439	Undocumented Java generic server variables in Artix and Orbix
70443	SOAP messages with "&" in `xsd:any` cause JVM crash
70446	SOAPFault raised by Artix Java server is wrapped by a JNI exception
70449	*Artix Security Guide*: request for a use case for setting up ACLs for multiple services running inside a container
70459	Artix 3.0.3 JMS transport mismatches responses if multiple concurrent clients uses the same permanent queue.
70463	Artix 3.0.3 (Java) `plugins:security:direct_persistence` not documented
70466	We need to document all logging subsystems available in Artix
70468	Artix 2.1.5 on Solaris gives a StackOverflowError exception when the customers WSDL is compiled using `wsdltocpp`.
70473	Artix fails to (un)marshal a user exception correctly
70475	Missing documentation of SOAP fault that is raised when authentication fails

Bug #	Description
70476	Artix Java documentation does not explain how to use `com.iona.jbus.FaultException` when working with other transports than IIOP
70487	Setting `authorizationRealm` in WSDL file is not being picked up at runtime
70491	Incorrect behavior in `schemavalidator`
n/a	Solaris C++ compiler support info wrong in online install guide

Reporting Problems

Contact customer support at http://www.iona.com/support/contact/.

Other Resources

If you need further help please use the following resources:

- Artix TechZone (`http://www.iona.com/devcenter/artix`) is a free online forum where IONA developers, your peers and other professionals come to share tips on Artix Web Services development. Visit the Artix TechZone today to start making the most of your Artix development experience.
- IONA University (`http://www.iona.com/info/services/ps/`) delivers practical and insightful courses that cover technical and product issues as well as standards-based best practices gleaned from real-world projects.
- IONA Professional Services (`http://www.iona.com/info/services/consulting/`) provide product expertise and consulting solutions that empower end-users, system integrators and software vendors with the knowledge to fully leverage IONA products. Together, IONA consultants and products equip you with a single platform for integrating and developing extremely reliable, scalable, and secure e-Business systems.

- **IONA Security Mailing List** (`security-alert@iona.com`): The mailing list provides security updates assocaited with all IONA products. To recieve security updates from IONA send mail to listserver@iona.com with no subject and the body text `subscribe security-alert` *youremail.*

  > **Note:** Please do not post queries to this e-mail alias; it has been set up only to notify you of security alerts.

- **Online Documentation** (`http://www.iona.com/support/docs/index.xml`): The latest updates to the Artix documentation are posted on-line.

Knowledge base articles (`http://www.iona.com/support/index.xml`): A database that contains practical advice on specific development issues, contributed by IONA developers, support specialists, and customers.