



Release Notes

Version 2.1.7, July 2006

IONA, IONA Technologies, the IONA logo, Orbix, Orbix/E, Orbacus, Artix, Orchestrator, Mobile Orchestrator, Enterprise Integrator, Adaptive Runtime Technology, Transparent Enterprise Deployment, and Total Business Integration 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.

While the information in this publication is believed to be accurate, 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 © 2001–2006 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: 25-Jul-2006

Contents

Upgrading from 1.x Versions	1
New Features in Artix 2.1	3
New Features in Artix 2.1.3	6
New Features in Artix 2.1.4	6
New Features in Artix 2.1.5	7
New Features in Artix 2.1.7	7
Documentation Updates in Artix 2.1	8
Documentation Updates for Artix 2.1.4	9
Documentation Updates for Artix 2.1.5	11
Known Problems	12
Fixed Bugs in Artix 2.1	18
Fixed Bugs in Artix 2.1.3	20
Fixed Bugs in Artix 2.1.4	21
Fixed Bugs in Artix 2.1.5	22
Fixed Bugs in Artix 2.1.6	23
Fixed Bugs in Artix 2.1.7	24
Reporting Problems	25
Other Resources	25

CONTENTS

Artix 2.1.7 Release Notes

In this document

This document contains the following sections:

Upgrading from 1.x Versions	page 1
New Features in Artix 2.1	page 3
New Features in Artix 2.1.3	page 6
New Features in Artix 2.1.4	page 6
New Features in Artix 2.1.5	page 7
New Features in Artix 2.1.7	page 7
Documentation Updates in Artix 2.1	page 8
Documentation Updates for Artix 2.1.4	page 9
Documentation Updates for Artix 2.1.5	page 11
Known Problems	page 12
Fixed Bugs in Artix 2.1	page 18
Fixed Bugs in Artix 2.1.3	page 20
Fixed Bugs in Artix 2.1.4	page 21
Fixed Bugs in Artix 2.1.5	page 22
Fixed Bugs in Artix 2.1.6	page 23
Fixed Bugs in Artix 2.1.7	page 24
Reporting Problems	page 25
Other Resources	page 25

Upgrading from 1.x Versions

When upgrading from a 1.x version of Artix perform the following steps:

- Acquire and install new licences. If you have not received your new license please contact your IONA representative.
- Recreate any existing Artix 1.x projects. (The Artix 2.x and Artix 1.x project formats are not compatible.)
- Regenerate all code generated from WSDL.
- Due to changes in the way Artix generates C++ code for complex types containing elements with `minOccurs=0` or `maxOccurs>1`, you must change any Artix application code that explicitly uses `ElementListT`. Modify your code as shown by the following example:

```
// Existing code:
IT_Bus::ElementListT<SomeType, &SomeComplexType_x_qname, 0, 10>&
list = something.getx();
// Correct code:
IT_Vector<SomeType>& list = something.getx()
```

- Due to a change in the Artix C++ servant implementation, user code can no longer get the `Port` object directly from within the generated server implementation object. You must now use the `Current` object to get the `Port` object in Artix server code. The code for getting a port using the `Current` object is shown below:

```
void TestImpl::do_stuff() IT_THROW_DECL((IT_Bus::Exception))
{
    Current& current = get_bus()->get_current();
    Port& port = current.get_operation().get_port();
}
```

- Due to a change in the way attributes are mapped in C++, attribute getter and setter methods use pointers in 2.x. The code generated for attributes in 2.x resembles the following:

```
Int* Foo::geta();
Foo::seta(const Int*);
Foo::seta(const Int&);
```

As a result of this change, all applications that use `get()` to retrieve attribute values must be changed to check that the value is set and not a null pointer.

- Due to a change in how Artix C++ code is generated, the `to_string()` function is no longer included in the generated classes to support user defined types. User code that uses the `to_string()` method must now be modified to include `artix\to_string.h`. The new implementation of `to_string()` takes a pointer to the `anyType` to be printed and optionally the `QName` of the element to print. In addition, your application must be linked with `it_bus_xml.lib`.
- Check the relevant development guides for other changes to Artix generated code that will impact your applications.
- Recompile all applications that contain generated code or reference generated header files.

New Features in Artix 2.1

The following new features have been added for Artix 2.1:

- [SOAP Binding](#)
- [Fixed Binding](#)
- [IDL to WSDL Compiler](#)
- [Artix C++](#)
- [Artix Java](#)
- [Type Support](#)
- [Artix Designer](#)
- [Internationalization](#)
- [Hostname Publishing Configuration](#)
- [Logging](#)
- [Routing](#)
- [Schema Validation](#)
- [Lifecycle Service](#)
- [Reflection](#)

SOAP Binding

The SOAP binding has the following new features in 2.1:

- Artix now honors the `<soap:header>` element.
- Artix now honors the `parts` attribute of the `<soap:body>` element.
- Support for SOAP with attachments has been implemented.

- `minOccurs` and `MaxOccurs` attributes in `<sequence>` elements are now supported in the SOAP binding.

Fixed Binding

The fixed binding now supports fault messages.

IDL to WSDL Compiler

The IDL to WSDL compiler's default behavior has been changed for 2.1. The compiler now defaults to generating *Doc Style* messages. Doc style messages have parts that are wrappers of the inputs and outputs of an operation.

Artix C++

Artix C++ has been updated in the following ways:

- The API for dynamic port configuration has been changed.
- The API for adding SOAP headers to messages has been changed.
- The API for adding GIOP service contexts has been changed.

Artix Java

Artix Java has the following new features in 2.1:

- Artix reference support in the SOAP binding.

Type Support

Artix 2.1 has been updated to support the following XML Schema Types:

- `xsd:any`
- `xsd:list`
- `xsd:ID`
- `xsd:anyURI`
- `xsd:gYear`
- `xsd:gDay`
- `xsd:gMonth`
- `xsd:gYearMonth`
- `xsd:gMonthDay`
- `xsd:nonPositiveInteger`
- `xsd:nonNegativeInteger`
- `xsd:positiveInteger`
- `xsd:negativeInteger`
- `xsd:integer`
- `xsd:dateTime`

Artix Designer

The following new features have been added to Artix Designer:

- When **New/Workspace** is chosen from the Artix Designer menu, a **Fast Track** dialog is displayed which contains a set of predefined workspace templates. These templates can be used to create workspaces containing pre-configured deployment profiles, shared resources, collections, deployment bundles, and code generation settings.
- A custom workspace called “CustomTemplate” is included for creating a custom workspace containing any combination of deployment profiles, shared resources, collections, collection-specific resources, and deployment bundles.

Internationalization

Artix 2.1 has added the following i18n support:

- Internationalization for the fixed length record binding.
- Codeset conversion for transports that do not have their own concept of headers (for example, IBM Websphere MQ, BEA Tuxedo, and Tibco Rendezvous). This is implemented using an Artix message interceptor and WSDL port extensors.

Hostname Publishing Configuration

A new configuration variable, `plugins:wSDL_publish:hostname`, has been added to control how host names are published by the WSDL publishing plug-in. For full details, see *Deploying and Managing Artix Solutions*.

Logging

A new plug-in, `bus_response_monitor`, is shipped with Artix. The new plug-in can log response metrics for both Artix servers and clients. For more details, see *Deploying and Managing Artix Solutions*.

Routing

The schema for defining attribute base routes has been updated to reflect the new context mechanism used by Artix. For full details, see *Designing Artix Solutions*.

Schema Validation

The Artix schema validation tools can now validate your contracts for WS-I compliance. For more information, see *Designing Artix Solutions* and <http://www.ws-i.org>.

Lifecycle Service

Artix now comes with a plug-in lifecycle service that will allow Artix to control the number of services running in a bus. This is typically required when applications are dynamically creating services on demand and need to ensure that their applications do not bloat. This is particularly important when customers cannot manage these applications directly such as when they are using the router. The demo `servant_management/transient_servants` shows how you might use the lifecycle service to perform garbage collection of unused services.

Reflection

Artix reflection provides a way of representing Artix data types so that they are self-describing. Using the reflection API, you can employ recursive descent parsing to process any data type (whether built-in or user-defined), without knowing about the data type in advance.

The Artix reflection API is useful in those cases where you need to write general-purpose code to process Artix data types. If you are familiar with Java or CORBA, you probably recognize that Artix reflection offers functionality similar to that of Java reflection and CORBA `DynamicAny`.

New Features in Artix 2.1.3

Artix Java supports references using the CORBA binding.

New Features in Artix 2.1.4

The following features are new in Artix 2.1.4:

- [Java SOAP Headers](#)
- [Java Security Support](#)
- [Java Message Handler Support](#)
- [Reflection Type Support](#)
- [Logging](#)

Java SOAP Headers

Artix's Java API now supports the use of SOAP headers. For more information see *Developing Artix Applications in Java*.

Java Security Support

You do not need to generate Java classes for the standard security support for each client and service you develop. The Artix Java Security libraries are used to access WS-Security types, which are optionally used in WSDL definitions. For more information, see the *Artix Security Guide*.

Java Message Handler Support

Artix's Java APIs have preliminary support for developing message handlers using the JAX-RPC `Handler` and `MessageContext` interfaces. Typically, message handlers are used to provide some common function that will be applied to many client and server elements. For more information, see *Developing Artix Applications in Java*.

Reflection Type Support

Reflection support has been updated to support all C++ types except for `SoapEncArray`.

Logging

Artix Java applications can now be configured to write log messages to the `log4j` logfile.

New Features in Artix 2.1.5

The JMS transport's WSDL extensions now provide support for setting a number of JNDI connection properties. This enables Artix to work with a greater number of JMS providers.

New Features in Artix 2.1.7

Artix 2.1.7 can now be set to use an individual listening address on a multihomed host. To use this feature, set the following configuration variable to the appropriate listening IP address:

```
policies:http:server_address_mode_policy:local_hostname
```

For example, set:

```
policies:http:server_address_mode_policy:local_hostname =  
"10.2.4.101";
```

Documentation Updates in Artix 2.1

Note: The Artix documentation is updated regularly on the IONA Support Web Site. Please check there regularly for the latest documentation.

The following members of the Artix documentation set changed for Artix 2.1:

- [Artix Tutorial](#)
- [Deploying and Managing Artix Solutions](#)
- [Designing Artix Solutions](#)
- [Designing Artix Solutions from the Command Line](#)
- [Developing Artix Applications in C++](#)
- [Developing Artix Applications in Java](#)
- [Getting Started with Artix](#)
- [Learning About Artix](#)

Artix Tutorial

This book was deprecated and its content moved into *Learning About Artix*.

Deploying and Managing Artix Solutions

This book was reorganized to make locating information easier. In addition, the chapter on logging was augmented, more detailed information about the Artix standalone service was added, and information on the new 2.1 features was added.

Designing Artix Solutions

This book was expanded to include the use of the Artix command line tools in designing Artix contracts. In addition, it includes information on using the new features supported by the Artix design time tools.

Designing Artix Solutions from the Command Line

This book was deprecated and its content moved into *Designing Artix Solutions*.

Developing Artix Applications in C++

This book was updated to include:

- New dynamic port configuration APIs.
- New type mappings.
- Updated SOAP header API.
- Updated GIOP service context APIs.
- Usability updates to the Artix Reference documentation.

Developing Artix Applications in Java

This book was updated to include:

- Artix reference support.
- `xsd:any` support.
- New type mappings.
- Threading model information.
- Non-JAX-RPC client creation.
- Class loading.

Getting Started with Artix

This book was deprecated and its content moved into *Learning About Artix*.

Learning About Artix

This is a new book in the Artix library that replaces *Getting Started with Artix* and the *Artix Tutorial*.

Documentation Updates for Artix 2.1.4

Note: The Artix documentation is updated regularly on the IONA Support Web Site. Please check there regularly for the latest documentation.

The following members of the Artix documentation set changed for Artix 2.1.4:

- [Developing Artix Solutions](#)
- [Developing Artix Applications in C++](#)
- [Developing Artix Applications in Java](#)
- [Deploying and Managing Artix Solutions](#)
- [Artix Security Guide](#)

Developing Artix Solutions

Developing Artix Solutions was updated in the following ways:

- The XML binding information was updated.
- The chapter on creating bindings using Artix Designer reflects changes made to the wizards.
- The chapter on creating routes using Artix Designer reflects the new routing wizard added in 2.1.4.

Developing Artix Applications in C++

Developing Artix Applications in C++ has been updated to include details about:

- Reflection support for `IT_Bus::AnyURI`, `IT_Bus::Any`, `IT_Bus::AnyList` and `IT_Bus::AnyHolder`.
- Reflection support for the binary types `IT_Bus::Base64Binary` and `IT_Bus::HexBinary`.
- The WSDL-to-C++ mapping for `xsd:list`.

Developing Artix Applications in Java

Developing Artix Applications in Java was updated to include information on the following:

- Using the JAX-RPC `MessageContext` interface and the JAX-RPC `Handler` interface to develop message handlers in Java.
- Using the Artix-specific `IonaMessageContext` interface to create custom SOAP headers.
- Using the new type factory registration mechanism.
- Developing Java plug-ins.

Deploying and Managing Artix Solutions

Deploying and Managing Artix Solutions was updated to include information on the following:

- Details on configuring Artix Java applications to log Artix messages using `log4j`.
- The options that can be passed to the `artix_env` script.
- New variables in the `plugins:xml_log_stream` and in the `binding:artix` namespaces.

- Changes to the configuration of the Artix WSDL publishing plug-in.

Artix Security Guide

The *Artix Security Guide* includes information about programmatically changing security attributes from the `bus-security.xsd` schema using Java message contexts.

Documentation Updates for Artix 2.1.5

Note: The Artix documentation is updated regularly on the IONA Support Web Site. Please check there regularly for the latest documentation.

The following members of the Artix documentation set changed for Artix 2.1.5:

- [Developing Artix Applications in Java](#)
- [Deploying and Managing Artix Solutions](#)
- [Other Books](#)

Developing Artix Applications in Java

Developing Artix Applications in Java was updated to include information on the following:

- Getting a reference to a running bus.
- Updates to the configuration variables used in configuring Java interceptors.

Deploying and Managing Artix Solutions

Deploying and Managing Artix Solutions was updated to include information on the following:

- Updates on deploying Artix services using a generic Artix service container.
- An updated section on deploying the Artix standalone service to reflect that it is really a router.
- Miscellaneous bug fixes.

Other Books

All of the books in the Artix library have been updated to address bugs raised against the documentation.

Known Problems

The following are known problems in this release:

- [Installation](#)
- [Type Support](#)
- [Demos](#)
- [Artix Designer](#)
- [Tuxedo Plug-in](#)
- [Java](#)
- [JMS](#)
- [Security](#)
- [Imported Schema Serialization](#)
- [Secure Artix Configuration](#)
- [Artix Transformer](#)
- [Routing](#)
- [SOAP Binding Generation](#)
- [HTTP Transport](#)
- [Compiling on AIX](#)

Installation

The following are known issues with the installation of Artix 2.1:

- Artix 2.x cannot be installed in the same directory tree as an Artix 1.x installation. It is recommended that you completely remove any Artix 1.x installations from your system before installing Artix 2.x.
- Uninstalling Artix leaves behind a number of files that prevent successful re-installation into the same directory. Manually delete all files left behind by the uninstaller before attempting to reinstall.

Type Support

The following are known problems in Artix type support:

- Due to platform specific limits, floats and doubles are limited to the range supported by `FLT_MIN-FLT_MAX` and `DBL_MIN-DBL_MAX` as defined by the C++ compiler in `float.h`. This range may not completely reflect the range required by XML Schema. This also affects the Java runtime as it is also limited by range of the C++ runtime.

- When reading an `xsd:dateTime` using the Tibco binding, Artix creates a 0 UTC time zone offset instead of using local time.
- When writing an `xsd:dateTime` using the Tibco binding, Artix generates a warning if a local time is specified.

Demos

The following are known issues with the demos included with Artix:

- When running the `basic/oneWay` demo, you get a warning message that the JMS plug-in can not be loaded. You can ignore this message.
- On HP-UX, the JMS demos require loading the `jvm_manager` plug-in, which depends on the `libjvm.sl` JRE library. Loading `libjvm.sl` is problematic for the following reason documented on HP's website (http://www.hp.com/products1/unix/java/java2/sdkrte14/infolibrary/sdk_rnotes_1.4.2.02.html#libjvm).

After updating to the proper patch, simply calling `export`

```
LD_PRELOAD=jre_lib/lib/PA_RISC/server/libjvm.sl
```

enables you to run the JMS demos.

- When running the JMS demos, the following statement appears:

```
(IT_CORE:1) E - could not load plug-in jms: Using configuration
scope of "demos.oneWay", configuration variable
"plugins:jms:shlib_name" is not set or is set to an empty
string. Use "-ORBname" to specify appropriate configuration, if
necessary.
```

This statement can be ignored.

- On Linux, you must use the following command to source the demo environment scripts:

```
source ./env
```

- On Unix, to get the `\routing>tagged_tuxedo_soap_http` demos to work, you must run the following commands:

```
export IT_DOMAIN_NAME=tagged_tuxedo_soap_http
export IT_CONFIG_DOMAINS_DIR=../../etc
```

- On AIX, to run the `transports/soap_over_mq` demo, call `export EXTSHM=ON` before running the demo.

- On AIX, to run the Java client of the `routing/soap_mq_corba`, set `EXTSHM=ON` before running the client.
- On AIX, the Java version of the `basic_callback` demo fails with a `java.lang.ClassCastException` exception in Xerces code.
- On Windows, with Visual C++ version 6, if you run the `transient_servants` demo, and then kill the router with `Ctrl+C` as recommended in the README for that demo, the router fails with a stack trace.
- If you mix Java and C++ clients and servers, the `custom_interceptor` demo does not work. The C++ and Java versions of this demo were designed separately.
- The README file in `transactions/artix_client_orbix` server demo is incorrect. The file `run_corba_servers.bat` does not exist.

Artix Designer

The following are known issues when using Artix Designer:

- The script generated to start a C++ server is incorrect.
- The ability to use the workspace services when creating a deployment bundle is not functional.
- In this release, you must customize the custom template in one of several ways:
 - i. Using the **Template Settings** Dialog: When the **CustomTemplate** icon is selected in the **New Workspace** dialog, the Fast Track Details section of the dialog contains a **Template Settings** button. Choosing this button invokes a dialog to allow the customizing of the shared resources and collection-specific resources that can be added to the workspace when it is generated. It also includes a default deployment bundle for generating code for one of the specified resources. The service and port for the selected resource can also be specified.
 - ii. Editing `WorkspaceTemplate.xml`: This file is included in `InstallDir\artix\2.1\etc\xml`, along with a corresponding `WorkspaceTemplate.xsd` schema file that defines the structure of the template. It is recommended that you copy the `WorkspaceTemplate.xml` file and create a new template rather than modifying the original. The easiest way to customize the template is to

modify the attributes of an existing XML element within this sample file, or else copy the XML element and then modify the new XML elements' attributes to create another instance of it. For example, if you want to create two Deployment Profiles with the Workspace, you would copy the default `<profile>` element under the “profiles” section and paste it relative to the existing one. Then the new profile elements' attributes can be modified to make it unique. The schema will specify the required attributes that need to be set for each type of element:

```
<profiles>
<profile name="CustomProfile" description="" os_system="">
<artix artix_dir="" env_script=""/>
<code language=""/>
</profile>
<profile name="CustomProfile2" description="" os_system="">
<artix artix_dir="" env_script=""/>
<code language="Java"/>
</profile>
</profiles>
```

- If you generate C++ code multiple times in a single Designer session, the Designer does not release the file handle for generated makefiles. Therefore, close Designer before attempting to build the generated code.

Tuxedo Plug-in

Tuxedo does not allow you to have a space in the path name of any path referenced in the `tux.env` script. Use the short file name format for Windows environments. You can display short file names using `dir /x`.

Java

The following are known problems with Artix Java:

- Java clients do not support WSDL operations that have no `<output>` message. To make a one-way operation work with a Java client, you must edit the WSDL definition to include an `<output>` message that has no parts.
- `THREAD-LOCAL` should not be used when developing Artix Java server applications.

- On AIX, there is a possible conflict between the version of `xerces` and `xalan` in the AIX JDK and the version that is shipped with Artix. You may need to add the following flags when running Artix Java applications:
 - ◆ `$JDK_ENDORSED_DIRS`
 - ◆ `$JDK_BOOTSTRAP_CLASSPATH`
- Artix does not support the generation of WSDL from Java operations that return an array of a base type like `int`, `long`, or `String`.
- Artix Java applications do not support the use of C++ interceptors built using Visual C++ v7.1.
- A `log4j` initialization error is occasionally displayed when Artix Java applications are started. You can ignore this message. All logging systems will work as expected.

JMS

Artix C++ applications that use the Artix JMS transport do not support the use of Visual C++ v7.1. They must be built using Visual C++ v6.0.

Security

During startup, the security server displays the following messages on HP-UX:

```
/usr/lib/dld.sl: Unresolved symbol: oop_iterate__7oopDescFP10OpClosure (code) from
/install_dir/jre/lib/PA_RISC/server/libjvm.sl
/usr/lib/dld.sl: Unresolved symbol: _adjust_pointer__9MarkSweepSFPP7oopDescb (code) from
/install_dir/jre/lib/PA_RISC/server/libjvm.sl
/usr/lib/dld.sl: Unresolved symbol: do_oop_nv__16FilteringClosureFPP7oopDesc (code) from
/hp/install_dir/jre/lib/PA_RISC/server/libjvm.sl
/usr/lib/dld.sl: Unresolved symbol: oop_iterate__7oopDescFP10OpClosure (code) from
/install_dir/jre/lib/PA_RISC/server/libjvm.sl
/usr/lib/dld.sl: Unresolved symbol: _adjust_pointer__9MarkSweepSFPP7oopDescb (code) from
/install_dir/jre/lib/PA_RISC/server/libjvm.sl
/usr/lib/dld.sl: Unresolved symbol: do_oop_nv__16FilteringClosureFPP7oopDesc (code) from
/install_dir/jre/lib/PA_RISC/server/libjvm.sl
```

The messages occur due to dependency on the classic JVM, which is deprecated in JDK 1.4.x, and can be safely ignored.

Imported Schema Serialization

When you serialize WSDL with multiple levels of imported schemas using `WSDLDefinitions::write()`, the serialized WSDL contains types from the directly-imported schema, and import statements for the indirectly imported

schemas. If the import statements use relative file URLs and you are using the serialized WSDL from a directory other than the directory of the original WSDL, the WSDL parser will not be able to find the imports.

Secure Artix Configuration

On Windows systems, the secure Artix configuration file, `install_dir\artix\2.1\ext\domains\artix-secure.cfg` may contain entries for the property `plugins:is2_authorization:action_role_mapping` that have spaces. The spaces, if any must be replaced by `%20`. For example, the entry

```
plugins:is2_authorization:action_role_mapping="file://C:\Program
Files\IONA\artix\2.0\demos\security\full_security\etc";
```

must be changed to

```
plugins:is2_authorization:action_role_mapping="file://C:\Program%20Files\IONA\artix\2.0\demos\s
ecurity\full_security\etc";
```

Artix Transformer

The transformer does not support complex types derived from other complex types.

Routing

The following are known issues with Artix routing:

- Failover routing does work with the CORBA binding.
- Failover routing does not work with the pass-through feature of the router activated. Set `plugins:routing:use_pass_through=false` when using failover routes.

SOAP Binding Generation

The `wsdltosoap` tool does not recognize the `-use` flag. The tool always generates a literal SOAP binding. If you wish to use encoded, you must manually generate the SOAP binding.

HTTP Transport

HTTP transport proxies are not thread-safe in Artix. To avoid thread clashes, make sure you use one proxy per client thread.

Compiling on AIX

When linking C++ applications, Visual Studio versions 5 and 6 generate duplicate symbol warnings for functions that exist in multiple object files. You can safely ignore these warnings. They are generated because the IBM C++ compiler, in accordance with ISO/IEC 14882:1998(E) : Programming Languages - C++ (See section 3.5 - Program and linkage), give non-inline functions eternal linkages. To not receive the warnings you can pass the compiler the `-qstaticinline` flag which will generate internal linkages for non-inline functions. You could also specify the `-bhalt:5` flag when linking to suppress the warning messages.

Fixed Bugs in Artix 2.1

The following bugs were closed in Artix 2.1:

Bug #	Description
68524	Customer wants what they call “clean WSDL.” Clean means it should only contain business relevant definitions, not types information added by Artix for construction reasons.
68850	The same Artix switch should support authentication on the request level and the message level.
68886	Customer wants the ability to assign individual namespaces to imported schemas.
68898	<code>wsdltocpp</code> and <code>wsdltojava</code> should have consistent flag names and flag behavior.
68914	The <code>javatowsdl</code> tool only searches the classpath for the class to convert to WSDL.
68920	<code>.xsd</code> file problems involving anonymous nested choice inside a sequence.
68921	<code>wsdltocorba</code> cannot handle empty sequences.
68923	<code>wsdltocorba</code> and the CORBA binding need to support <code>xsd:dateTime</code> .

Bug #	Description
68929	Artix processes need to gracefully exit if the specified port number is already in use.
68934	The expression of arguments is not consistent between <code>idl</code> and <code>wsdltosoap</code> .
68950	Absolute file path not documented.
68958	Artix 2.0.3 <code>artix_env</code> uses incorrect <code>if</code> condition for verifying the Artix installed JRE.
68960	Artix Designer puts IONA proprietary entries in the WSDL when a new Service section is added.
68966	Some log messages are logged as warning messages but should be information messages.
68968	Messages are incorrectly appearing in the Artix switch log file.
68981	Defining an <code>xmlns</code> in a WSDL import causes a runtime error.
69004	The RPC-Element SOAP faults are not currently supported in Artix.
69008	New switch to pass a value for <code>corba:address</code> location added to the IDL to WSDL tool.
69009	An option to control the value of the <code>name</code> in the definitions needs to be added to the IDL to WSDL tool.
69010	An option to qualify names used for CORBA binding, service and port names needs to be added to the IDL to WSDL tool.
69016	Artix 2.x to be supported on Red Hat Linux AS 3.0.
69018	Reference support Java requested.
69031	Artix Java documentation should note required properties for running Artix client from J2EE app servers.
69032	Generated Artix Java client stub should have the <code>ServiceFactory</code> property set.

Bug #	Description
69037	Artix clients cannot handle fault responses containing a user defined exception from an Axis server.
69038	Support for programmatic endpoint setting in Artix Java client needed.
69052	Artix documentation needs to be updated to add Java-specific threading configuration details.
69059	<code>javatowsdl</code> generates incorrect classes when arrays or sequences are used.
69062	The location tag is case sensitive.
69063	Artix does not support one-to-many relations between service name and operation name for Tuxedo.
69091	<code>artix_env</code> incorrectly uses <code>-d</code> when <code>-e</code> should be used in testing the presence of the <code>jre</code> directory.
69097	<code>wSDL_publish</code> needs to publish the fully qualified host name.
69118	Artix logging to XML file includes the subsystem name.
69149	Artix SOAP reader gets an Out Invalid data when deserializing <code>xsd:long</code> type error.
69178	Null pointer exception raised when generating Java code for faults.

Fixed Bugs in Artix 2.1.3

The following bugs were closed in Artix 2.1.3:

Bug #	Description
69182	Artix serialization error with IMS transactions.
69183	Artix 2.1.2 generates C++ code that fails to compile.
69209	Artix 2.1.2 <code>wSDLtojava</code> tool changed code generation.

Fixed Bugs in Artix 2.1.4

The following bugs were closed in Artix 2.1.4:

Bug #	Description
68266	Enhance the ASP 5.1 ME IMS Adapter to allow conversational support over OTMA.
69006	It is extremely difficult to determine the service pack level being used.
69159	Associate WSDL and/or XSD files with Artix Designer.
69164	Artix 2.1 designer will not open an invalid WSDL file.
69174	C++ mapping error of nested complex types.
69188	Nilable values in a nested sequence do not get parsed properly.
69197	HTTP:Uninitialized security context overrides authentication with nulls.
69212	The Artix router writes out the <code>xsi:type</code> in the SOAP message. It should not do this.
69221	<code>it_tls_atli2.lib</code> is missing in the installation.
69222	The Artix router wrongly qualifies elements.
69224	Exception raised when setting proxy endpoint address.
69225	The Artix router cannot propagate user exceptions to a client.
69237	No package name is generated when <code>-p</code> is used.
69258	Update <code>ElementListT</code> to take a default element name and add <code>set_element_name()</code> to update the element name.
69266	Unqualified elements not correctly supported in Java and the Artix router.
69289	Java <code>UnsatisfiedLinkError</code> for <code>writeInteger</code> .

Fixed Bugs in Artix 2.1.5

The following bugs were closed in Artix 2.1.5:

Bug #	Description
69241	Binding editor creates superfluous <code>xmlformat:body</code> tags.
69263	Security plug-in generates <code>ContextException</code> warning when registering the security context.
69310	There does not seem to be a way to populate an instance of <code>IT_Bus::DateTime</code> with the system value.
69320	Web chains currently have no capacity to manage faults or exceptions that may be returned by a member of the chain.
69325	Cannot successfully retrieve <code>MQ_OUTGOING_MESSAGE_ATTRIBUTES</code> .
69330	XML generated using the Artix C++ <code>to_string()</code> has excessive copies of namespace declarations.
69336	Artix cannot handle the HTTP header <code>Cache-Control</code> field if it has more than one value.
69347	<code>IT_ContextAttributes::BusSecurity</code> is not cleaned up properly.
69354	The HTTP plug-in needs to support <code>ResponseTimeout</code> and <code>SendTimeout</code> through the context mechanism.
69359	Artix needs to support <code><soap:fault></code> name attributes.
69360	The router gets a schema not found error while loading WSDL.
69428	Prefix not bound error received while using <code><any/></code> .
69436	<code>wsdltojava</code> generates incorrect code for oneway operations.
69465	Artix Java interceptor should not have the user set context class loader.
69502	Artix does not support oneways.

Bug #	Description
69501	Artix server core dumps caused by receiving Alteon/WebLogic heartbeat.
69513	Bogus code generated from simple service.
69514	Artix MQ needs to support <code>MQFMT_IMS</code> format.

Fixed Bugs in Artix 2.1.6

The following bugs were closed in Artix 2.1.6:

Bug #	Description
69478	Artix 2.1.4 Java server crashes on Windows if an empty string is sent in an <code>xsd:datetime</code> type.
69542	IS2 LDAP adapter not thread safe.
69544	Artix MQ does not support oneway messages.
69564	<code>wsdltocpp/wsdltojava</code> compile failures for MQ <code>ims_var_string</code> and <code>ims</code> format values.
69571	Artix-Java: <code>MinOccur=0</code> causes "deserialize called too many times" in Artix Java.
69585	Artix-Java: Java message interceptor eats up the transport exception.
69630	Artix 2.1.5 Java server core-dumps when returning a sequence of strings.
69650	Artix fails to properly marshal subtype.
69653	<code>xsd:positiveInteger</code> not supported when writing SOAP message.
69668	Request for patch merges on IS2 for Zurich Insurance.
69672	The Artix proxy client core-dumps when it receives "service not available" message from the WebLogic container.

Bug #	Description
69678	Artix stops accepting new HTTPS requests if a previous handshake failed.
69688	IS2 remote token cache feature does not work with demo file adapter.
69745	Artix HTTPS failed on an SSL handshake on valid certificates.
69752	wsdltocorba generates invalid IDL.
69773	MQ oneway operation with <code>Transaction=internal</code> does not perform rollback on application exception.
69787	Derived type support breaks <code>datetime</code> type in Java.
SR275764	Issue when multiple Java interceptors are configured.

Fixed Bugs in Artix 2.1.7

The following bugs were closed in Artix 2.1.7:

Bug #	Description
69840	By default, Artix opened a listener on all network interfaces of a multihomed host. This needs to be configurable. (See “New Features in Artix 2.1.7” on page 7 to take advantage of this fix.
69996	Exception when running an HTTPS Artix 3.0.1 client against an Artix 2.1.6 server.
70051	MQ transport should not call MQINQ on remote queue for backout count threshold attribute.
70105	Artix Java server crashes on Solaris when receiving a malformed SOAP request (via HTTP listener).
70213	Artix MQ client proxy failed to release handle during proxy destruction.

Bug #	Description
70274	wsdltcorba produces java.lang.StackOverflowError with no further error message.

Reporting Problems

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

Other Resources

- [Artix TechZone](http://www.iona.com/devcenter/artix) (<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 today.
- [IONA University](http://www.iona.com/info/services/ps/) (<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/) (<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](mailto:security-alert@iona.com) (security-alert@iona.com): The mailing list provides security updates associated with all IONA products. To receive security updates from IONA, send mail to listserv@iona.com with no subject and the body text `subscribe security-alert youremail`.

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

- [IONA Security Advisory Archive](http://www.iona.com/support/docs/artix/security_advisories/index.xml) (http://www.iona.com/support/docs/artix/security_advisories/index.xml): This page archives all of the Artix-related security updates that have been sent to the the IONA security mailing list.

- [Online Documentation](http://www.iona.com/support/docs/index.xml) (<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/kb/index.jspa) (<http://www.iona.com/support/kb/index.jspa>): A database that contains practical advice on specific development issues, contributed by IONA developers, support specialists, and customers.