PROGRESS
SOFTWARE

Artix®

Release Notes

Version 4.2, May 2009

Updated: December 16, 2009

Contents

Artix 4.2

In this document

This document contains the following sections:

New Features

The following features have been added in IONA Artix 4.2:

- WSDLGen
- WS-RM Management
- SOAP 1.2 Security
- HP-UX Classic Runtime Support
- Artix Designer

WSDLGen

Artix 4.2 includes a new tool for generating code from WSDL.

WSDLGen is a template-driven code generation engine that can create full client and server code from even the most complex WSDL documents. See the Artix WSDLGen Guide for details.

WS-RM Management

Artix 4.2 supports the management of Web Services Reliable Messaging (WS-RM) using any Java Management Extensions (JMX) console. See the Managing Artix Solutions with JMX guide for details.

SOAP 1.2 Security

Support has been added for sending WS-Security headers over the SOAP 1.2 binding. See the Artix Security Guide for details.

HP-UX Classic Runtime Support

Artix runtime libraries now support applications built with the HP-UX Classic C++ runtime, as well as the already supported Standard runtime. For a full list of supported platforms and compilers, see the Artix Installation Guide.

Artix Designer

The following features are new in Artix Designer. See the online help for details:

WSDLGen support WSDLGen is the code generation engine used in Artix Designer 4.2.

Quick code generation You can now generate code quickly by right-clicking a WSDL file and choosing a generation option from the Artix Designer item on the pop-up menu.

Easier configuration of Artix services Configuring services such as Artix Security, the locator, and session manager has been simplified through the use of policies.

Documentation Updates

The following changes have been made to the Artix library in version 4.2.

- Artix Infocenter
- Using Artix Designer
- Artix Technical Use Cases
- Building Service Oriented Architectures with Artix
- Artix WSDLGen Guide
- Artix Security Guide
- Artix for CORBA

Artix Infocenter

The Artix Designer online help is now available from the IONA web site in Eclipse Infocenter format at http://infocenter.iona.com:8100.

Using Artix Designer

This book is now deprecated in favor of the Artix Infocenter.

Artix Technical Use Cases

This book now includes details on generating code using WSDLGen.

Building Service Oriented Architectures with Artix

This book has been reworked for greater clarity.

Artix WSDLGen Guide

This is a new book describing the WSDLGen code generation tool.

Artix Security Guide

The following changes have been made to the Artix Security Guide:

- In the “Partial Message Protection” chapter, the “Action Definitions” section includes a description of a new signature validation
- New support for transmitting security credentials in SOAP 1.2 headers is described in the “Overview of SOAP Security” section of the “Security for SOAP Bindings” chapter
- The “Managing Access Control Lists” chapter now describes the use of the wildcard character, *, in Access Control List (ACL) files

Artix for CORBA

The “Exposing a Web Service as a CORBA Service” now includes a description of how to enable the Java/CORBA binding in an Artix container.

Known Issues

The following are known issues in Artix 4.2:

- Artix Designer
- JRE on 64-bit Platforms
- Missing Classic Runtime Libraries on HP-UX
- No Uninstallation Script on AIX
- Daylight Savings Time

Artix Designer

The following are known issues with Artix Designer.

Installation directory on Windows Artix Designer fails to launch on Windows if Artix is installed in a directory beginning with the letter "u". This is an issue with Eclipse, which sees the `\u` character combination as indicating a Unicode character.

Empty start_eclipse script The Artix installer adds an empty `start_eclipse.bat` file to the *`InstallDir`*`\artix\4.2\bin` directory on Windows. You can ignore this file, as Artix Designer no longer requires a start script. You launch it by running the Eclipse executable in the *`InstallDir`*`\artix\4.2\eclipse` directory or from the Windows Start menu.

Generating a client in a database project fails When creating a database Web services project, you cannot use Artix Designer to generate a client to run against the database service, as described in the *Exposing a database as a Web service* cheat sheet. Use the `wsdltojava` command to generate the client instead.

Launching a secure client as a Java application fails When you create a Java client using a basic Web services project with security enabled, launching the client as a Java application from the Run window fails. Launch the client by selecting **Run** | **External Tools** | **External Tools** instead.

Preferences problem When editing or viewing settings in the Preferences window, selecting **Artix Repository** | **Connection Settings** throws an error that prevents you from applying any unsaved changes. The Artix Repository preferences are not relevant to Artix Designer.

Makefile problems Certain Artix-specific Makefile settings are generated the first time you run the `artix_env` script. Thus, even though it is not required to run `artix_env` before starting Artix Designer, you must run the script at least once before trying to build any Artix C++ code.

Running imported demos After importing a basic Artix demo into Artix Designer, you should run the start scripts by selecting **Run | External Tools | External Tools**, as described in the online help. To run the scripts from the *ProjectFolder*`/bin` directory, first add the following to the top of each script:

```
set SCRIPT_DIR=%~dp0
cd /d %SCRIPT_DIR%
```

Generated scripts on UNIX not executable The scripts that the Artix Designer code generator creates on UNIX platforms are not executable. You need to change their permissions using the `chmod` command.

Windows-style environment variable in UNIX start_client script When generating a client on UNIX platforms, the `start_client` script includes a Windows-style `%SCRIPT_DIR%` environment variable. You must change this to the correct UNIX format, `$SCRIPT_DIR`, for the script to run.

start_container for a database service fails on UNIX Before launching a container for a database service from the External Tools window, you must change the following line in the `start_container` script in *ProjectDir*`/.artix/containers/`*CodeGenDir*:

```
export CLASSPATH=PathToJdbcDriver;$CLASSPATH
```

to

```
CLASSPATH=PathToJdbcDriver;$CLASSPATH
export CLASSPATH
```

Alternatively, leave the script as is and launch it from a terminal window.

start_client fails where WSDL path contains spaces A `start_client` script that references a WSDL path containing spaces throws an error. In Java, you can work around this issue by launching the client as a Java application from the Run window. In C++, edit the `start_client` script in *`ProjectDir`*`/.artix/containers/`*`CodeGenDir`* by replacing all instances of `%20` with `%%20` in the WSDL URL. This was fixed in the Artix Designer 4.2.1 update release.

Problem applying policy changes In the Policy Editor, if you delete or add policies, the **Save** option remains grayed out in the **File** menu and you cannot save your changes. The only way to apply policy changes is edit some of the policy's properties and then save. This was fixed in the Artix Designer 4.2.1 update release.

Generating code from bindings or portTypes fails In Artix Tools, generating code based on bindings or portTypes causes code generation to fail.

JRE on 64-bit Platforms

The following are known Java Runtime Environment (JRE) issues on 64-bit platforms:

Linux and Solaris The Solaris and Linux versions of the Artix installer include a 32-bit JRE only. Users working on 64-bit versions of these platforms should download and install a 64-bit JRE.

Artix Security on Linux On 64-bit Linux, the Artix Security Service runs only with JRE 1.4.2_06. No other JRE is supported for this platform.

Missing Classic Runtime Libraries on HP-UX

Support for the HP-UX Classic C++ runtime is broken due to the absence of a number of libraries from the Artix 4.2 kit.

No Uninstallation Script on AIX

The Artix 4.2 installer does not create an uninstallation script when installing on AIX. As a workaround, you can uninstall Artix from AIX by removing its top-level directory, since Artix installation does not make changes to the PATH or global environment.

Daylight Savings Time

The new daylight savings time schedule in the US and Canada has no effect on how Artix behaves. For details of the OS, compiler and JVM versions on which Artix is certified to run, see the following Knowledge Base article:

http://www.iona.com/support/kb/entry.jspa?entryID=2532

Fixed Bugs

The following bugs have been fixed in Artix 4.2:

Table 1: *Bugs Fixed in Artix 4.2*

Bug #	Description
70562	Supporting `useMessageIDasCorrelationID` on the client side
70654	HTTP GET and POST support not properly implemented in Artix
70685	Artix IDL to WSDL mapping does not deal with null EPRs correctly
70803	`mustUnderstand=1` always added to `soap:header` coming from WSDL
70815	Revisit of feature request to log JMS header information to Artix log file
70827	Artix does not support multiple HTTP cookies
70867	WS-AT prepare/commit messages stop flowing after a Transformer crash
70880	Artix is core dumping when processing a sequence type
70885	Artix 4.x router silently modifies message before sending to destination
70895	Java DLL cannot handle renamed JAX-RPC operations

Table 1: *Bugs Fixed in Artix 4.2 (Continued)*

Bug #	Description
70898	Artix Security Guide refers to variables `check.kdc.running` and `check.kdc.principal` instead of `com.iona.isp.adapter.krb5.param.check.kdc.running and com.iona.isp.adapter.krb5.param.check.kdc.principal`
70902	Baltimore JCE provider conflict with Sun's JCE provider caused kerberos adapter not working
70903	If `check.kdc.running=true` in `is2.properties` and connection attempt to Kerberos server fails, then no information about the error cause is logged
70910	Strange behavior of router with a WSDL exposing two operations
70938	Artix incorrectly converts user exceptions to SOAPFaultExceptions when a custom handler is in the interceptor list
70939	Artix throws WSDL exception for valid WSDL containing `corba:const`
70942	Artix 4.1 server crashes when continuously receiving TCP handshakes with no data sent
70948	The operation `com.iona.jbus.messaging.ClientProxy.create` throws a NullPointer exception for certain EPRs
70952	Artix process hangs on Linux when initializing Java plug-ins
70955	Memory leak when routing requests from MQ series to, or from, another protocol
70964	Artix for Java does not have a way of getting the URL that a client posted a message to
70972	CORBA type mapping cannot reference types from another namespace

Table 1: *Bugs Fixed in Artix 4.2 (Continued)*

Bug #	Description
70981	Serializations symptoms (using HTTP) causing bottleneck in router Multi Instance threading model is leaky when called from the router Artix router leaks memory when handling custom Java transport
70984	`ws_coloc` will cause a VM crash for operations with more than one parameter
70996	Memory leak when routing one-way requests to/from MQ series from/to another protocol.
71013	Artix bus fails to initialize in multithreaded applications
71023	`XMUtils.toXML()` drops root element's namespace
71024	Artix C++ fails to unmarshal request using tagged binding and `minOccurs="1"` set for the message part
71026	Memory leak in Artix Java observed when `getRequestContext` called repeatedly
71046	Artix container crashes when attempting to use log4j log stream
71047	`CORBA::UNKNOWN` exception thrown without even dispatching the incoming request to the application-level code
71050	Crash when running inside JCA container and passing complex type
71052	Core dump due to unhandled JNI exception
71061	Artix J2EE will only register the last port in the `ejb_servants.properties` file when ports share a common service
71064	Latest Artix 4.1 cumulative patch breaks `MessageID/CorrelationID` pattern
71065	Artix 4.1: error in `oldjmswsdl_to_newjmswsdl.xsl` file

Table 1: *Bugs Fixed in Artix 4.2 (Continued)*

Bug #	Description
71066	Artix JVM core marshaling enum/union types
71075	In the router, implement a round-robin load balancer that can work in passthrough mode
71086	`close()` on socket returns EINVAL
71089	Need a solution to set the serviceID of a CORBA object for the CORBA binding programmatically in Artix Java
71091	Server getting write exception when using XML binding
SR 282042	Performance spike issue

Updating Artix Designer

To update Artix Designer to the latest version (Artix Designer 4.2.4):

1. In Artix Designer 4.2, select **Help|Software Updates|Find and Install**. The Install/Update wizard launches.
2. In the Feature Updates panel, select **Search for new features to install**, then click **Next**.
3. In the Update Sites to Visit panel, click the **New Remote Site** button.
4. Enter the following details in the **New Update Site** dialog box:
   - **Name**: IONA Artix Plug-ins
   - **URL**: `http://www.iona.com/downloads/artix/eclipse/4.2`
5. Click **OK**.
6. Select the **IONA Artix Plug-ins** checkbox and clear all other boxes in the Sites to Include section.
7. Check the **Ignore Features** checkbox and click **Finish**.

   Eclipse contacts the specified URL and returns with a list of available Eclipse plug-ins at that site.
8. In the Search Results panel, check the **Show the latest version** checkbox. Select the check boxes beside all the IONA Artix Plug-in entries, then click **Next**.

9. Accept the license agreement and click **Next**.
10. In the Installation panel, confirm the target installation location and click **Finish** to start the installation.

After downloading the updated plug-ins, relaunch Artix Designer.

We recommend that you select a new workspace after relaunching. If you want to continue using the same workspace, first delete the `LocalRepository` folder under the workspace folder in your file system. A new local repository will be created for you when you next create an Artix Designer project.

You should also check that your Artix Designer installation details were not overwritten by the update process by selecting **Window|Preferences** and then selecting **Artix Designer**.

Reporting Problems

Contact customer support at http://www.progress.com/support

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 developers, 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.
- Knowledge Base (`http://www.iona.com/support/kb/`): A database that contains practical advice on specific development issues, contributed by developers, support specialists, and customers.
- Online Documentation (`http://www.iona.com/support/docs/index.xml`): The latest updates to the Artix documentation are posted on-line.

Artix 4.2.1

In this document

This document contains the following sections:

For details on downloading and installing the Artix 4.2.1 rolled up patch, see the Artix Installation Guide.

Known Issues

As well as those listed for Artix 4.2, the following are known issues in Artix 4.2.1.

Artix Designer

The following issues apply to Artix Designer:

Java Applications on Linux

Generated Java clients and servers fail to launch on Linux due to an invalid WSDL URL in their launch configurations.

To workaround:

1. Select **Run | Run**.
2. In the Run dialog, expand the **Java Applications** node, and select the launch configuration for the client or server.
3. Click the **Arguments** tab.
4. In the Program Arguments field, locate the `-BUSservice_contract` argument.
5. In the WSDL URL, change `file://` to `file:/`.

Security and CORBA Bindings

When creating a basic Web services project based on a CORBA service and port, selecting the **Secure Communication** option does not adequately secure the application.

You should enable the Artix security service when generating from a CORBA binding.

Fixed Bugs

The following bugs have been fixed in Artix 4.2.1:

Table 1: *Bugs Fixed in Artix 4.2.1*

Bug #	Description
70562	Supporting `useMessageIDasCorrelationID` on the client side.
70858	Session manager prints a dependency on `locator_endpoint`.
71069	Artix Mainframe Designer should save previous details entered.
71145	Artix calling a secure Web service over an insecure proxy does not work. The Artix client is (incorrectly) communicating securely with the proxy server.
71159	FML error followed by a crash when running Tuxedo client.
71178	`wsdltojava` fails to generate code.
71201	Code generator `wsdltojava` produces Java code that does not compile from a valid XSD.
71211	Modify copybook generation to avoid exceeding level 48.
71212	Modify WSDL/CORBA validators to detect recursive types.
71213	COBOL writer is generating periods in column 73. This is invalid.
71214	Artix Designer crashes on reading customer WSDL with stack overflow.
71219	`wsdltojava` generates a `TypeFactory` that is too large to compile.
71233	Missing namespace declaration in server response.

Table 1: *Bugs Fixed in Artix 4.2.1 (Continued)*

Bug #	Description
71237	Artix loses some schema attribute values in the conversion from XML to Java.
71241	Artix generates SOAP request/reply messages with namespace prefix definitions for namespaces that are not used.
71258	Core dump in Artix router.
71259	Artix 4.2 conversion from XML to Java then back to XML generates wrong double values.
71273	Artix router throwing `SoapFaultExceptions` on initial invocation after disconnect (when using `locator_client` plug-in to resolve endpoints).
71274	Problem with WSDL-first projects that contain fault information when applying size edits.
71284	WSDLGen errors in Java method header of implementation class. Error in client sample when using WSDLGen.
71299	Artix generates SOAP request/reply messages with namespace prefix definitions for namespaces that are not used.
71310	Fails to advertise a Tuxedo service implemented as an Artix Java plug-in to the Tuxedo BBL.
71333	ISF server requires `iiop_tls`.
71335	SOAP message with bad `xsd:date` field not caught by Artix 4.2.
71368	Artix container deployment descriptor containing leading or trailing spaces fails to be deployed.
71374	`it_container` crash with core dump when shutting down.
71385	Artix 4.2 performance.
71386	User can not set `JMSCorrelationID` when making a request/reply invocation using a static `replyTo` queue.
71396	`wsdltojava` is not properly processing Java package name to exclude digits at the start.

Table 1: *Bugs Fixed in Artix 4.2.1 (Continued)*

Bug #	Description
71401	Artix SSL to support message encryption without `EstablishTrustInTarget` (enhancement).
71418	`wsdltocpp` mixed case code generation bug.
71451	`wsdltojava` produces uncompilable code when a `complexType` has a sequence with `minOccurs="0"` and `maxOccurs="unbounded"`.
71462	Multi-threading a Tuxedo server.
71467	Modify Artix 4.2 to allow 500 status code in the SOAP HTTP header to be returned.
71481	SOAP validation problem.
71493	`wsdl_publish` plug-in removes `name` attribute from `mime:part` element.
71496	MQ transport should improve the information put into the logs.
71506	Implementing a per-thread threading model in Java, expose C++ thread ID using a Java `ContextContainer`.
71519	`wsdltojava` seems to generate uncompilable code when there is a sequence of a sequence of a sequence in the `.xsd` file.
71526	Artix 4.2 does not enforce `"minoccurs=1"`.
71544	Problem with Artix 4 Baltimore CSSL libraries on Linux.
71572	`WSDLPublishContractResolver::resolve_url` always resolves to the primary network interface.
71585	Specifying a custom HTTP header name-value pair in the connection to the HTTP proxy.
71587	Memory leak in Java SOAP/HTTP server when a user fault is thrown.
71602	WSDLGen is populating new `java.math.BigInteger()` with strings.
71608	Artix login service URL can not be resolved using the Artix locator service.
71613	Wrong queue used by server making outgoing request.

Table 1: *Bugs Fixed in Artix 4.2.1 (Continued)*

Bug #	Description
71620	If an Artix Java server throws a `Caught exception while parsing the XML document containing message`, it leaks memory.
71621	Problem with message handlers and three-tier systems.
71623	C++ server request interceptor, setting WSEE user name/password, and security interceptor.
71647	Multi-threaded Artix client using HTTPS causes core dump on Solaris.
71651	Enhance the Artix security plug-in to use the `issue_external_token` API.
71656	Artix HTTPS does not send an EOF, potentially causing problems for non-Artix clients.
71659	MQ plug-in crashes container when user has insufficient privileges to the queue.
71673	When multiple instances of `it_container` are started simultaneously, and if a custom Java plug-in and the log4j plug-in are in the `orb_plugins` list, one or more JVM instances may crash.
71678	Artix showing high native memory usage, small leak discovered in JMS transport.
71695	WS-RM requires Internet access.
71707	JCA servant calls security APIs even for insecure services.
71734	Add a configuration setting to control whether WSSE plain text passwords are printed in the log.
71793	Crash in Artix client (`XMLParseException`) on loading and/or producing XML.
71815	Problem generating Java from WSDL/schema.
71821	Locator endpoints getting lost.
71823	Artix 4.2 Java with HTTPS leaks memory under a DDOS style attack (from a hardware load balancer).

Table 1: *Bugs Fixed in Artix 4.2.1 (Continued)*

Bug #	Description
71835	Remove nodes from the locator's sender list.
71847	ULOG errors reported when launching Tuxedo Java server from `it_container`.
71850	Race condition during `list_endpoints` call at random.
71865	XSD type with `xsd:totalDigits` results in `UnknownElementException`.
71886	Artix locator is holding stale endpoints.
71887	Artix 4.1 Artix HTTP client crashes if the server drops connection.
71896	Race condition in `IT_ATLI2_HTTP::HttpEndpointAdapterImpl::resolve_service()`.
71897	Race condition in `xercesc_2_4::UnixHTTPURLInputStream::UnixHTTPURLInputStream()`.
71900	`SIGSEGV` in `IT_ORB_ORBImpl::resolve_initial_references()`.
71902	Unexpected exception in `IT_Bus::MessagingClientOperation::do_invoke()`.
71903	Unexpected exception in `IT_Bus::DBReplicationMgr::init()`.
71904	Abort freeing memory in `IT_Bus::DBReplicationMgr::engage_request_forwarding()`.
71905	Locator exhibits unreclaimed memory growth for `list_endpoints()` and `registerEndpoint()` under certain conditions.
71910	Support for Actional management plug-in for Artix 4.2.
71914	Artix 4.2 SOAP fault messages not being serialized properly or processed properly.
71915	Logging of binary buffers is extremely slow.
71919	Locator downloads full application WSDL (and imported XSDs) during endpoint registration.
71923	Problem with Artix 4.2 server and client side processing of `xsi:type`.

Table 1: *Bugs Fixed in Artix 4.2.1 (Continued)*

Bug #	Description
71950	Bogus `transfer-encoding:chunked` header being set on outgoing callback invocation.
71953	Partial Message Protection in Artix 4.2 not working.
71955	SOAP with attachment supporting MIME type application octet-stream.
71958	Client proxy constructor is leaking memory.
71965	`wsdltocpp` generates code with errors.
71967	WebSphere crash—WAS core dumps.
71968	Inbound J2EE demo fails when deployed in WebSphere 6.1.
71972	JNI crash compressing large messages.
71977	`BinaryBuffer::allocate` out of memory error.
71996	`policies:soap:security:enforce_must_understand="false"` not obeyed by client binding for response.
72011	Locator endpoints lost under certain timing circumstances.
72022	`I18nClientInterceptor` can cause a Tuxedo server crash.
72038	Artix Designer problems while trying out fix for bug 71865 (XSD type with `xsd:totalDigits` results in `UnknownElementException`).
72055	Artix Designer code generation fails with `NullPointerException`.
72069	Stale endpoints are registered at the locator after restarting an endpoint.
72075	`SIGSEGV` can occur in `checkForCrossListenerPolicy` depending on the order of listener creation.
72084	Memory growth in HTTPS proxy.
72094	Locator replication across machines fails.
72105	Problem with Artix and HTTP chunking.

Table 1: *Bugs Fixed in Artix 4.2.1 (Continued)*

Bug #	Description
72106	Artix JAX-RPC applications are unable to send implicit header information if the header data is not wrapped in a new element.
72144	SSL problems when making Artix and MQ client invocations.
ART-10181	Java code generation fails in Artix Designer 4.2.1.
ART-10598	Artix 4.2 router crashes when certain tagged data is passed.
ART-10605	Problems in Java Proxies—`ResolveInitialReference()` and `createClient()`.
ART-10610	`wsdl_location` field in the deployment descriptor is ignored by the `getServiceWSDL()` API
ART-10631	Berkeley Database reports a panic state.
ART-10654	In a replicated environment, one of the Artix `itcontainer` instances exhibits linear growth in memory usage.
ART-10665	An Artix 5.1 JAX-WS client registers multiple instances of the same endpoint with a 4.2 locator.
ART-10689	Security token is not recognized after Artix 4 migration.

Configuration Changes

This section describes configuration changes that are necessary to use the fixes for specific bug fixes.

Bugs 71241 and 71299—Artix generates soap request/reply messages with namespace prefix definitions for unused namespaces

The following configuration change is necessary to use the fix for bugs 71241 and 71299. To have all the XML namespaces only at the root of the SOAP message, add the following configuration setting to your configuration scope.

```
plugins:soap:decl_namespaces_at_root="true";
```

The default is `false`, which is the behavior prior to this patch, and does not restrict the namespace declarations to any scope.

Bug 71401— Artix SSL to support message encryption without EstablishTrustInTarget

The product configuration changes to enable anonymous ciphersuite support are as follows:

1. In your Artix C++ HTTPS (or IIOP/TLS C++) configuration scopes, ensure that `EstablishTrustInClient` or `EstablishTrustInServer` are not supported for the endpoints that require anonymous ciphersuites. The following configuration enables anonymous Diffie-Hellman-based ciphersuites that provide confidentiality without authentication.

   The following example is for HTTPS, use `iiop_tls` instead of HTTPS for anonymous CORBA communications:

```
policies:https:target_secure_invocation_policy:requires =
  ["Confidentiality", "Integrity", "DetectReplay", "DetectMisordering"];
policies:https:target_secure_invocation_policy:supports =
  ["Confidentiality", "Integrity", "DetectReplay", "DetectMisordering"];
policies:https:client_secure_invocation_policy:requires =
  ["Confidentiality", "Integrity", "DetectReplay", "DetectMisordering"];
policies:https:client_secure_invocation_policy:supports =
  ["Confidentiality", "Integrity", "DetectReplay", "DetectMisordering"];

policies:https:mechanism_policy:ciphersuites =
  ["DH_ANON_EXPORT_WITH_RC4_40_MD5", "DH_ANON_WITH_RC4_128_MD5",
   "DH_ANON_EXPORT_WITH_DES40_CBC_SHA", "DH_ANON_WITH_DES_CBC_SHA",
   "DH_ANON_WITH_3DES_EDE_CBC_SHA"];

principal_sponsor:https:use_principal_sponsor = "false";
principal_sponsor:https:auth_method_id = "";
principal_sponsor:https:auth_method_data = [];
policies:https:trusted_ca_list_policy = "";
```

2. The available anonymous ciphersuites are listed in the above configuration fragment. Anonymous ciphersuites are explicitly disabled by default. You must explicitly specify them by populating `policies:https:mechanism_policy:ciphersuites` (or the `iiop_tls` equivalent) appropriately. In Artix, you can not mix anonymous ciphersuites and non-anonymous ciphersuites on a single endpoint.

   Note: To reset the list of available ciphersuites back to the internal default of all full strength, non-export, non-anonymous, non-null encryption ciphers, set the list to be empty:
   `policies:https:mechanism_policy:ciphersuites = [];`

3. As shown in the preceding example, there is no need for credential related information or trusted CA related information when using these ciphersuites.

   WARNING: These ciphersuites do not protect against man in the middle attacks (MITM). They provide confidentiality in a technical sense. However, because they do not authenticate the peer, it is possible for an undetected entity who has the capability of actively intercepting and controlling TCP communications between the two peers to setup a relay with separate SSL connections to the two parties. The undetected entity can then monitor their communication by interposing itself in the middle of their communication stream.

Bug 71418—wsdltocpp mixed case code generation bug

The following switch to `wsdltocpp` enables you to add words that are treated as reserved words:

```
[-user_reserved_words <word1[[:word2]...]>]
```

For example:

```
wsdltocpp -user_reserved_words SEC:MILLISEC -m UNIX:library test.xsd
```

Bug 71451—wsdltojava produces uncompilable code when a complexType has a sequence with minOccurs=0 and maxOccurs=unbounded

Previously, running `wsdltojava` produced type handlers. This is no longer the case. A new `-typehandlers` flag has been added, which can be used to generate handlers.

The `-notypehandlers` flag still exists. `-notypehandlers` and `-typehandlers` are mutually exclusive. An error is thrown if both are specified on the command line.

Bug 71526—Artix 4.2 does not enforce minoccurs=1

If the following error occurs from clients trying to connect to the locator service:

```
Expected element: node_id
```

The following setting should be added to your configuration file:

```
plugins:soap:sequence_validation = "false";
```

The error occurs when clients connect to the locator using a different WSDL interface than the one published by the locator. Use of this configuration setting is deprecated. Customers are advised to ensure locator clients are using the correct WSDL if they see this issue.

Bug 71572—WSDLPublishContractResolver::resolve_url always resolves to the primary network interface

The `plugins:wsdl_publish:hostname` variable has been enhanced to read a logical name or a virtual IP address. Standard literal values are `canonical`, `unqualified` and `ipaddress`. Now, any non-standard literal values are also published, other than the stripping of white spaces. This generates a header file that replaces the string `class SEC` with `class _SEC`.

Bug 71585—Specifying a custom HTTP header name-value pair in the connection to the HTTP proxy

Custom HTTP headers can be specified programatically using context containers. You can access the context, and set the required headers as `[name,value]` pairs. These are added as new HTTP headers by the Artix transport. The following code examples show how to do this in C++ and Java.

C++

Note: `transport_custom_headers_xsdTypes.h` is a new header.

```
#include <it_bus_pdk/context_attrs/transport_custom_headers_xsdTypes.h>

    // Get the Context Container
    //
    Bus* bus = Bus::create_reference();
    ContextRegistry* registry = bus->get_context_registry();
    ContextCurrent& current = registry->get_current();
    ContextContainer* request_contexts = current.request_contexts();

    AnyType* ctx =
       request_contexts->get_context(
           IT_ContextAttributes::TRANSPORT_CUSTOM_HEADERS, true );
    assert(ctx != 0);
    CustomHeaders* headers_ctx = dynamic_cast<CustomHeaders*>(ctx);
    if (headers_ctx != 0)
    {
        // Add the headers as [Name,Value] pairs
        //
        CustomHeader new_header;
        new_header.setname("MyHeader");
        new_header.setvalue("\"Myvalue\"");
        headers_ctx->push_back(new_header);
    }

    // make the invocation
    //

    // Obtain a pointer to the RequestContextContainer
    //
    ContextContainer* reply_container = current.reply_contexts();
    ctx = reply_container->get_context
           (IT_ContextAttributes::TRANSPORT_CUSTOM_HEADERS);
    assert(0 != ctx);
    headers_ctx = dynamic_cast<CustomHeaders*>(ctx);
    if (headers_ctx != 0)
    {
        ElementListT<CustomHeader>& headers =
            headers_ctx->getcustom_headers();
    }
```

Java

Note: `CustomHeaders` and `CustomHeader` are new classes added in this fix.

```
import com.iona.schemas.transports.header.CustomHeaders;
import com.iona.schemas.transports.header.CustomHeader;

    Bus bus = Bus.init(new String[0]);
    try {
        // Get the contexts
        ContextRegistry registry = bus.getContextRegistry();
        IonaMessageContext context =
            (IonaMessageContext)registry.getCurrent();
        CustomHeaders headers =
            (CustomHeaders)context.getRequestContext
               (ContextConstants.TRANSPORT_CUSTOM_HEADERS,true);
        if (headers != null) {
            CustomHeader header = new CustomHeader();
            header.setName("MyHeader");
            header.setValue("\"MyValue\"");
            ArrayList props = new ArrayList();
            props.add(header);
            if (props.size() != 0) {
                headers.setCustom_headers((CustomHeader[])props.toArray
                  (new CustomHeader[props.size()]));
            }
        }

        // make the invocation here.

        // Access the reply headers
        CustomHeaders reply_headers =

   (CustomHeaders)context.getReplyContext(ContextConstants.TRANSPORT_CUSTOM_HEADERS);
    } finally {
         bus.shutdown(true);
    }
```

Bug 71623—C++ server request interceptor, setting WSEE user name/password, and security interceptor

The `policies:asp:server_interception_point` configuration variable is used to control the point at which the Artix security interceptor is called.

By default, the interceptor is called at the `intercept_around_dispatch` phase. This default setting is suitable for most customers:

```
policies:asp:server_interception_point = "intercept_around_dispatch”;
```

However, some advanced users may wish to interpret incoming information on the wire and be able to set related information on the appropriate Artix security contexts before the Artix security interceptor is called. For example, if you wish to perform dynamic credential mapping, use the following setting:

```
policies:asp:server_interception_point = "intercept_pre_dispatch";
```

This advanced security setting can not be used in conjunction with the router when it is configured to use pass-thru mode. The Artix router in pass-thru mode skips the `intercept_pre_dispatch` interception point. This means that the Artix security interceptor only works with the Artix router when the secure application configured to use the default setting (`intercept_around_dispatch`).

Bug 71651—Enhance the Artix security plugin to use the API issue_external_token

The product configuration changes to enable support for `issue_external_token` are:

1. In the `etc/full_security.cfg` configuration file, in the `secure_artix.full_security.security_service` scope, replace

```
policies:external_token_issuer:client_certificate_constraints=[];
```

with

```
policies:external_token_issuer:client_certificate_constraints =
["%{CERT_CONSTRAINT_1}"];
```

2. In the etc/full_security.cfg configuration file, in the secure_artix.full_security.server configuration scope, add:

```
   policies:asp:enable_issue_external_token="true";
```

Bug 71678—Artix showing high native memory usage, small leak discovered in JMS transport

The messaging port maintains a number of pools for MessageReader, MessageWriter, SendMessageContext, and ReceiveMessageContext. In previous versions, these pools were unbounded. This fix introduces an new configuration variable to put an upper bound on the pool:

```
plugins:messaging_port:generic_pool_size
```

The default value is -1, which means the pool is unbounded. This is to keep the behavior backwards compatible. A value of 0 means no pooling is performed.

Bug 71968—Inbound J2EE demo fails when deployed in WebSphere 6.1

This bug fix adds the following Java configuration property:

```
EJBLookupUseAppServerClassLoader
```

The EJBLookupUseAppServerClassLoader configuration property controls whether or not the Artix J2EE Connector uses the application server's classloader to lookup EJB for an inbound invocation.

This value is a boolean, and the default is TRUE. If TRUE, the Artix J2EE Connector looks up the EJB using the application server classloader. Otherwise, the Artix J2EE Connector uses the .rar classloader.

For details on setting this property for JBoss, WebLogic, and WebSphere, see the Reference Information chapter in the *Artix for J2EE (JAX-RPC) Guide*.

Bug 71977—BinaryBuffer::allocate out of memory error

A new variable has been added for sharing of a pre-allocation buffer amongst threads:

```
policies:http:buffer:prealloc_shared
```

The default is `false`, which means that each thread pre-allocates its own buffer up front on first invocation. If set to `true`, this buffer is shared amongst threads. This means that it is up to the application to ensure that two invocations are not simultaneously active on this pre-allocated buffer.

Bug 72011—Locator Endpoints lost under certain timing circumstances

This bug fix adds the following configuration variable:

```
plugins:peer_manager:ping_on_failure = "...";
```

The default is `false`. If this variable is set to `true`, the receiver of a ping failure performs a reverse ping to verify the validity of the failure.

In general, the peer manager service on both sides (endpoint manager and locator service) ping each other as a health check. If this variable is set, the peer that sees the ping failure confirms the validity of the failure by performs a ping. If this reverse ping succeeds, the ping failure is spurious and can be ignored. If it does not fail, it is a genuine ping failure and the appropriate callback is notified.

The endpoint manager and locator service react differently to ping failures. For more details, see the *Artix Locator Guide*.

This kind of behavior would be useful in circumstances where the hardware clock malfunctions and creates unnecessary ping failure conditions (for example, reregistrations or removal of endpoints).

Bug 72144 / W901120032—SSL problems making Artix and MQ client invocations

The following system properties are supported by the Artix JAX-RPC runtime to enable control over the TLS setting for retrieving secure WSDL using `JavaHTTPSUrlConnection`. Previously, this could only be controlled using the standard JSSE system properties, such as `javax.net.ssl.trustStore`.

In certain scenarios, customers may already need to use these system properties, and require another set of values to control Artix JSSE usage when retrieving WSDL using built-in `java.net` URL functionality.

The new Artix JAX-RPC system properties associated with this enhancement apply to retrieving WSDL for Artix JAX-RPC applications only. These are set in the standard way—for example, as `-D`*`PropName`*`=`*`PropValue`* command-line options, or where applicable, using the Artix `jvm_options` configuration mechanism. Alternatively, these values can be read from configuration and set programmatically.

The new system properties are as follows:

- `com.iona.webservices.wsdl.processors.trustStorePath`
  This property is a pathname to a PEM format CA file list. Automatic stripping of descriptive text not wrapped with PEM Certificate tags is supported by the runtime.
- `com.iona.webservices.wsdl.processors.pkcs12Path`
  The path to a `PKCS#12` trust store. The password property described next is also needed.
- `com.iona.webservices.wsdl.processors.pkcs12PasswordFilePath`
  This property contains the path to a file that contains a password for a `pkcs#21` trust store. This password file should be protected appropriately by O/S permissions. The benefit of having the password in a separate file to the referencing configuration is that you can have more O/S ACL permissions control.
- `com.iona.webservices.wsdl.processors.pkcs12Password`
  This property directly contains the password for a `PKCS#21` trust store. Use of the password file property above is generally more secure and this property is most suitable for development and testing. However, if you embed passwords directly in configuration files, you must ensure that those configuration files have appropriately restrictive access.

- com.iona.webservices.wsdl.processors.disableHostnameVerification

  This defaults to true, for consistency with the C++ TLS stack, which does not enforce host name verification. To explicitly enable host name verification for WSDL retrieval in a JAX-RPC application, set this property to false.

  If this property is not set, whatever host name validation is in effect takes place (for example, the JDK implicit or explicitly set defaults).

Required Manual Updates

This section describes manual updates required to for J2EE interoperability with IBM WebSphere.

Updating the artix.rar file

Due to the updated it_wsdl.jar, you must run the following commands to update the artix.rar file used to deploy Artix in other containers:

```
cd $IT_PRODUCT_DIR/artix/4.2/etc/j2ee
ant ra.dd.10.notx
```

You should see the following:

```
$ ant ra.dd.10.notx
Buildfile: build.xml

check.environment:

ra.dd.10.notx:

prepare.rar.dd:
   [copy] Copying 1 file to $IT_PRODUCT_DIR/artix/4.2/etc/j2ee
   [copy] Copying 1 file to $IT_PRODUCT_DIR/lib/artix/j2ee/4.2
   [jar] Updating jar: $IT_PRODUCT_DIR/lib/artix/j2ee/4.2/artix.rar

BUILD SUCCESSFUL
```

Using artix.rar in WebSphere

When you install artix.rar in the WebSphere container, Artix expects the password to be DEFAULT in plain text, but wsadmin hashes the password during the artix.rar deployment. As a result, when the Artix J2EE Connector attempts to initialize, it will fail.

To verify that WebSphere has not encrypted the JAASLoginPassword during the deployment of artix.rar, perform the following steps:

1. Using the WebSphere Administrative Console, navigate to:

   Resource Adapters|ArtixConnector|J2C Connection Factories|ArtixConnectorCF
2. Click **CustomProperties**.
3. Click **JAASLoginPassword**. Verify that the value for the password is set to the string DEFAULT. If the value is already set to the string DEFAULT, no further steps are required. If the value for JAASLoginPassword is a hashed value, reset the value to DEFAULT, and save the property setting.
4. Restart the WebSphere container after saving the changes.

Tracking Patches

To determine what patch level has been installed, run the following command:

```
IT_PRODUCT_DIR/artix/4.2/bin/itartix_version
```