

3.1 Migration Guide

Major Notes

- CXF 3.1 no longer supports Java 6. You must use Java 7 or Java 8.
- The JAX-WS/Simple frontend ServerFactoryBean will automatically call reset at the end of the create() call. This allows resources to be cleaned up and garbage collected sooner. However, it also prevents multiple calls to create() from sharing the same ServerInfo /EndpointInfo/etc... objects like they would we 3.0.x. That sharing has caused many problems in the past due to sharing of properties (like token caches) that are stored on those objects so the new behavior is more "correct", but it is different than previous versions so care must be taken while upgrading.
- The Karaf features.xml file for CXF 3.1 no longer will install spring or spring-dm when installing the "cxf" feature. If you require spring /spring-dm, you will need to install those features prior to installing the CXF feature.

Security changes

- The STS (Security Token Service) now issues tokens using the RSA-SHA256 signature algorithm by default (previously RSA-SHA1), and the SHA-256 digest algorithm (previously SHA-1).
- Some security configuration tags have been renamed from "ws-security.*" to "security.*", as they are now shared with (some of) the JAX-RS stack. The old tags will continue to work as before however without any change. See the [Security Configuration](#) page for more information.
- The SAML/XACML functionality previously available in the cxf-rt-security module is now in the cxf-rt-security-saml module.
- If you are explicitly specifying the SAML version in a SAML CallbackHandler, then this is changed in CXF 3.1 due to the migration to use OpenSAML 3.1. The version is now set on the SAMLCallback using a org.apache.wss4j.common.saml.bean.Version class. Previously there was a dependency on OpenSAML's SAMLVersion class.
- It is now possible to "plug in" custom WS-SecurityPolicy validators if you wish to change the default validation logic for a particular policy.

New Features

- The CXF JAX-WS code generator has a new option "seiSuper" that can be used to specify additional super interfaces for the SEI. This makes the code nonportable to other JAX-WS containers. The primary use would be to add AutoCloseable to the interface to allow use of the clients in Java7 try with resource blocks.
- New Metrics feature for collecting metrics about a CXF services. Codahale/DropWizard based collector included.
- New Throttling feature for easily throttling CXF services. Sample included that uses the Metrics component to help make the throttling decisions.
- New Logging feature for more advanced logging than the logging available in cxf-core
- New Metadata service for SAML SSO to allow you to publish SAML SSO metadata for your service provider.
- The "cxf" frontend to the JAX-WS code generator (-fe cxf) now generates code that is a bit more "Java7" friendly as the return type of the getPort(...) calls is a sub-interface of the SEI that also implements AutoCloseable, BindingProvider, and Client. Code that used to look like:

```
(AddNumbersPortType port = service.getAddNumbersPort());
((BindingProvider)port).getRequestContext()
    .put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, address);
port.addNumbers3(-1, 2);
((Closeable)port).close();
```

can be replaced with:

```
try (AddNumbersPortTypeProxy port = service.getAddNumbersPort()) {
    port.getRequestContext().put(BindingProvider.
ENDPOINT_ADDRESS_PROPERTY, address);
    port.addNumbers3(-1, 2);
}
```

Major Dependency Changes

- The Jetty based HTTP transport has been updated to support Jetty 9 as well as Jetty 8. However, support for Jetty 7 has been dropped.
- Due to the Jetty upgrade, support for running Jetty based endpoints in Karaf 2.3.x has been dropped.

- Support for using JAX-WS 2.1 based API jars has been removed. Java 7 (now required) includes JAX-WS 2.2 so this should not be an issue.
- WSS4J 2.1 is included, which in turn includes OpenSAML 3.0.