

---

# Celtix Enterprise 1.1 Release Notes

06 Mar 2007

## Table of Contents

Getting Started .....	1
New Features .....	2
Celtix Enterprise .....	2
Celtix Router .....	2
Celtix Container .....	3
Celtix Advanced Service Engine .....	3
Celtix Advanced Messaging .....	4
Documentation Updates .....	4
Known Issues .....	5
Installation .....	5
Celtix Enterprise Router .....	5
GUI Tools .....	6
Samples .....	6
Apache Incubator ServiceMix .....	8
Fixed Bugs .....	9
Celtix Advanced Messaging .....	9
Celtix Advanced Service Engine .....	10
Celtix Router .....	12
Enhancements .....	13
Celtix Advanced Messaging .....	13
Celtix Advanced Service Engine .....	13
Celtix Router .....	14

Celtix Enterprise is a certified open source ESB offered by IONA Technologies. It provides best of breed components to offer a complete SOA solution. For information on getting support for Celtix Enterprise go to <http://www.iona.com/products/celtix>.

## Getting Started

The first step to running any of the tools shipped with Celtix Enterprise is to run the **celtix\_env** script.

# New Features

## Celtix Enterprise

Celtix Enterprise has the following new features in 1.1:

- Improved binary installers

There are now binary installers for each of the products in the Celtix family. The installers are smaller and present the installer with only the options available for the product they have downloaded.

- Improved source installer

The source installer has been updated to use Maven 2.0. It provides a single target for building all of the projects incorporated into Celtix Enterprise. In addition, the individual projects can be built separately.

## Celtix Router

The Celtix Enterprise router has the following new features in 1.1:

- Operation-based routing

You can define routes that use specific operations of an endpoint as a source or target. This facilitates the construction of routes that are more fine grained than routes that have fully defined endpoints as sources or targets.

- Recipient List Routing Pattern

You can define a list of recipients for messages. The messages are routed to one or more of the recipients on the list based on a specified set of criteria.

- Routing Slip Routing Pattern

You can define routes that pass messages through a series of endpoints based on a specified set of criteria. This is a simplified form of service orchestration.

- Load Balancing

You can define routes that distribute requests across a number of endpoints that implement the same service. The default load balancing algorithm is round-robin. However, you can implement your own load balancing algorithm and plug it into the router.

## Celtix Container

The Celtix Enterprise container has the following enhancements in 1.1:

- Application naming

Applications deployed into the container now use the name of the WAR as their application name.

- Multiple Spring bean deployment

A user can now deploy multiple Spring beans to be deployed as a part of an application. Spring bean files must be placed in the `META-INF/spring` folder of the WAR.

- Drag and drop deployment

The Celtix Enterprise container now scans the container repository for newly added WARs. When a new WAR is found, the application it defines is automatically deployed into the container.

- Application monitoring

Using the Web Service management interface or the JMX management interface, you can get the state of an application in the Celtix Enterprise container. Applications are either `STARTED`, `STOPPED`, or `FAILED`.

## Celtix Advanced Service Engine

The Celtix Advanced Service Engine has the following new features in 1.1:

- Configuration

The configuration model used by Celtix Advanced Service Engine has been simplified. You no longer need to use the bean/property/value nesting. Instead configuration is specified using elements.

- Support for mixed style WSDL interfaces.
- Apache Incubator Qpid can now be used as an alternative JMS transport.

For information on setting this up see Chapter 4, Using Celtix Advanced Messaging as a JMS Replacement in *Using the Celtix Enterprise JMS Transport*.

- Support for JAX-WS `Handlers` when using Dispatch/Provider API's.
- Performance improvements.
- Bug fixes and other enhancements.

## Celtix Advanced Messaging

Celtix Advanced Messaging has the following new features in 1.1:

- JMS 1.1 compliance.
- Performance improvements.
- Bug fixes and other enhancements.

## Documentation Updates

- Writing WSDL Contracts

A new guide that provides an introduction to using WSDL to define service interfaces.

- Developing Applications Using JAX-WS

A new guide detailing how to develop services using the JAX-WS APIs. This guide covers using both a Java first and a WSDL first approach. It does not cover using the `Provider` interface.

- Developing Distributed Applications with Dynamic Languages

A new guide detailing how to develop services using Javascript.

- Using the Celtix Enterprise JMS Transport

A new guide detailing how to configure Celtix Enterprise to use JMS as a transport. This guide uses ActiveMQ as a reference for its examples. However, you are not limited to using ActiveMQ.

- Using the Celtix Enterprise Router

A new guide that describes the Celtix Enterprise router. It introduces how the router works and how to define basic routes.

- Celtix Enterprise Deployment Guide

A new guide providing information about how to deploy services into the Celtix Enterprise container. It also has some information regarding deployment into a servlet container.

# Known Issues

## Installation

The installation procedure has the following known issues:

- Welcome Page on \*NIX

On \*NIX systems where Mozilla or Netscape is not present, the Welcome Page will not display when the installer finishes running. You can manually load the Welcome page from the `docs` folder of the installation.

- Linux Installations with GCJ

There is a known problem with running the Linux installer (`.bin` extension) on a system with GCJ (GNU Compiler for Java) installed. It has been identified as an InstallAnywhere bug. When running the installer you may see the following output:

```
"Invocation of this Java Application has caused an InvocationTargetException. This
application will now exit. (LAX)

Stack Trace:
java.lang.NoClassDefFoundError: while resolving class: ZeroGe
    at java.lang.VMClassLoader.resolveClass(java.lang.Class)
    (/usr/lib64/libgcj.so.5.0.0) "
```

The solution is to install a Sun JVM and put that on your path. For example, on a machine with a JRE installed at `/opt/jdk1.5.0_03_64/jre` run the following:

```
export PATH=/opt/jdk1.5.0_03_64/jre/bin:$PATH
```

## Celtix Enterprise Router

The Celtix Enterprise router has the following known issues:

- Mule Download

You must download a number of the Mule runtime libraries from [www.mulesource.com](http://www.mulesource.com) before using the router. A script, **download\_mule** is provided with the installation. You must type **yes** when asked to accept the MuleSource Public License.

- POJO Mode

The following features are not supported:

- Proxification of WS-A EndpointReference objects passing through the router.
- Asynchronous operations.
- PAYLOAD Mode

The following features are not supported:

- SOAP message attachments
- Asynchronous operations.
- MESSAGE Mode

The following features are not supported:

- Asynchronous operations.
- REST Support

The router does not support services that use REST style interfaces. There is currently no way for the router to pass the HTTP verbs to endpoints.

## GUI Tools

The version of Eclipse SOA Tools Platform included with Celtix Enterprise has the following known issues:

- Code Generation

Do not generate more than one server, implementation, or client per project. Support for multiple code generations in a single project will be made available via the update site in the near future.

- Annotations

Using non-default attribute values when adding JAX-WS annotations can cause problems with the generated code and WSDL. We recommend using only the default values when creating annotations.

- Path Names

The GUI tools do not work properly when there are spaces in the workspace's path name.

## Samples

The samples included with Celtix Enterprise have the following known issues:

- Top-level README

The README in the samples folder contains instructions for setting a number of environment variables. These should be ignored. Celtix Enterprise includes a script, **celtix\_env**, that properly sets your environment.

- Building on Linux

On versions of Linux with a system wide Ant configuration file installed (/etc/ant.conf) you may see the following error when building the samples:

```
"Exception in thread main java.lang.ClassDefNotFoundError  
org.apache.tools.ant.Launch/Launcher"
```

If you get this error, use the **--noconfig** option when running Ant. This will bypass the system's ant configuration and ensure that the version of Ant installed with Celtix Enterprise is called.

```
ant --noconfig
```

- WS-RM Samples

The WS-Reliable Messaging demo in `service_creation/ws_rm` fails if the client is run a second time against the same server. This is due to a bug in the demo code whereby the `MessageLossSimulator` interceptor is installed on both the client and server side and results in loss of partial responses originated from the server.

To workaround, apply the following changes:

1. In `ws_rm.xml`, remove line 39:

```
<bean id="messageLoss" class="demo.ws_rm.common.MessageLossSimulator"/>
```

2. In `ws_rm.xml`, remove line 60:

```
<ref bean="messageLoss"/>
```

3. Insert the following line in `Client.java`, after line 42:

```
bus.getOutInterceptors().add(new demo.ws_rm.common.MessageLossSimulator());
```

- Service Creation HTTP Demos

Only some of the samples have been updated to reflect the new configuration system. For example, the `ws_addressing` sample has been updated, but not the `hello_world_https` sample. However, all demos continue to work as before.

## Apache Incubator ServiceMix

The version of Apache Incubator ServiceMix included with Celtix Enterprise has the following known issues:

- Start Script & SunOS

The Apache Incubator ServiceMix start script located in `$SERVICEMIX_HOME/bin/servicemix` is not compatible with SunOS `/bin/sh`, to workaround this issue we launch Apache Incubator ServiceMix using Bash in `$CELTIX_HOME/bin/servicemix_start.sh`. This requires that a Unix/Linux system have bash installed in `/bin`.

When `CELTIX_HOME/bin/servicemix_start.sh` is launched on a system without bash present in `/bin` the following warning will be displayed:

```
"The ServiceMix executable requires bash installed in /bin."
```

The script will then exit.

- Unpacking Files

Apache Incubator ServiceMix has a known issue with unpacking files placed in its deploy and install folders, this is documented in Apache Incubator ServiceMix's bug tracking system under SM-607. This issue may be observed when `CELTIX_HOME/bin/servicemix_deploy_sa` and/or `CELTIX_HOME/bin/servicemix_install_ca` scripts are executed. If this issue occurs the user may retry the script or they may edit the 'monitorInterval' value in `SERVICEMIX_HOME/conf/servicemix.xml` to a larger value.

## Fixed Bugs

### Celtix Advanced Messaging

JIRA #	Description
QPID-323	Message reference count not being incremented when message added to <code>UnacknowledgeMap</code>
QPID-321	Simplify logic to deal with setting <code>MessageListener</code> only after connection start has been called
QPID-317	Remove dependency on JMS from common and broker modules
QPID-315	<code>JMSMessageProperties</code> not all set when converting messages
QPID-313	Call to <code>attainState</code> in <code>makeBrokerConnection</code> can miss the notification of state change.
QPID-311	Dispatcher pause logic is not thread safe.
QPID-300	<code>FailoverException</code> is throw to clients
QPID-299	Messages not being correctly requeued when transacted session closed
QPID-294	Race condition in establishing client connection
QPID-293	Setting <code>MessageListener</code> after connection has started can cause messages to be "lost" on a internal delivery queue
QPID-282	<code>AMQConnection</code> does not support IP addresses
QPID-276	Potential race condition in <code>AMQChannel</code>
QPID-273	Remove unnecessary <code>Map</code> creation for each server side message
QPID-267	<code>AMQConnection#Stop</code> throws <code>NotImplementedException</code>
QPID-266	<code>PropertiesFileInitialContextFactory</code> doesn't lookup <code>PROVIDER_URL</code> correctly
QPID-233	<code>IllegalMonitorStateException</code> in <code>ConcurrentSelectorDeliveryManager</code>
QPID-231	<code>QueueSession.createSender()</code> should allow a null <code>Queue</code> name.
QPID-225	Queue Browser implementation: acknowledgements not handled properly for <code>CLIENT_ACKNOWLEDGEMENT</code>
QPID-222	Ensure pre-fetched messages received when in a transaction are correctly requeued

JIRA #	Description
QPID-214	Calling <code>recover</code> in <code>onMessage()</code> should stop Auto-ACK being sent for current message
QPID-212	Calling <code>close()</code> on a connection or session, any pending <code>receive()</code> should return null
QPID-211	<code>NullPointerException</code> in broker when message sent with no routingKey
QPID-209	<code>Message.acknowledge()</code> should only acknowledge messages consumed by the application
QPID-207	<code>Message.acknowledge()</code> should acknowledge received
QPID-206	Sending same <code>TextMessage</code> more than once causes text to be lost on second send
QPID-205	Do not allow subscription to temporary topic created on different session
QPID-202	<code>QueueSender</code> : incorrectly throwing <code>NullPointerException</code> when queue supplied to <code>send()</code>
QPID-201	<code>MapMessage</code> : wrong exception thrown when getting null as char
QPID-193	<code>AMQSession.createConsumer()</code> must be synchronous to avoid timing problems
QPID-192	<code>MapMessage.clearBody()</code> impl doesn't clear the map properly
QPID-189	<code>producer.sendMessage()</code> must not assume the <code>Message</code> argument is a Qpid implementation class
QPID-186	<code>AMQConnection</code> does not set error code when constructing a <code>JMSEException</code>
QPID-181	Message properties do not support JMS type conversion rules
QPID-179	<code>TextMessage</code> needs to be able to distinguish between a null payload and an empty string
QPID-177	<code>TopicSessionAdapter.createTextMessage(String)</code> ignores its argument
QPID-176	<code>TopicSession.createPublisher</code> does not handle null argument correctly

## Celtix Advanced Service Engine

JIRA #	Description
CXF-6	<code>jaxws-api</code> , <code>jsr181-api</code> should be optional in <code>cxf-common</code> pom

JIRA #	Description
CXF-19	<code>getPort(Class)</code> throws exception
CXF-29	<code>DatatypeProvider</code> not set
CXF-119	Empty request elements causes NPE
CXF-179	Move <code>Holder</code> code from <code>JAXWSMethodInvoker</code> to interceptors
CXF-184	parameter order should be take care of in runtime
CXF-188	invalid wsdl generated by <b>java2wsdl</b> tool
CXF-236	NPE thrown by <code>XmlStreamWriter</code> when calling <code>StaxUtils.copy()</code>
CXF-237	Problem with checkstyle organisation of imports
CXF-253	Server can not start when the port number is missing
CXF-255	<code>js_provider</code> sample problem
CXF-258	Server can not start when SEI has no <code>wsdlLocation</code> in the <code>WebService</code> annotation
CXF-263	<code>SOAPBinding</code> annotation can not placed on method with RPC style
CXF-265	Java2WSDL NEP when the SEI is DOC-LIT Wrapped style
CXF-269	CXF can not handle an empty reponse when using oneway operation.
CXF-289	Adding of Root Node in <code>XMLBinding</code> is done without a prefix.
CXF-290	wsdl space in path name problem on Windows
CXF-292	<code>CXFServlet</code> does not pass the wsdl packaged in war to the <code>EndpointImpl</code>
CXF-293	Defining a XML Binding with a <code>rootNode</code> results in Fault Response at the Client end.
CXF-295	Service Model can't support import <code>PortType</code> with different name-space
CXF-296	Create <code>BindingFactory</code> failure when the first ext element in wsdl binding part is not the protocol element
CXF-297	Checking in-bound message format should based on content-type, not by <code>mtom-setting</code> , <code>content-id</code> should be decoded by <code>URLDecoder</code> first
CXF-320	If add a user interceptor into <code>pre_stream</code> and <code>addBefore</code> to <code>SoapPreProtocolOutInterceptor</code> , CXF will throw an "Invalid ordering" exception
CXF-323	Improper synchronization of <code>getDefaultBus()</code> , <code>setDefaultBus()</code>
CXF-324	Dynamically generated WSDL for JAX-WS service is invalid
CXF-327	"Wrong number of arguments" exception if change <code>hello_world</code> samples wsdl

JIRA #	Description
CXF-336	JAX-WS <code>Handler</code> on client side not working
CXF-354	Printing Extension Elements in the wsdl doesn't have the proper indentation level.
CXF-355	<code>WrapperClassInInterceptor</code> throws NPE when handling a oneway operator
CXF-359	<code>wsdl2java</code> using wrong type (should use wrapper type, but use primitive type) for holder parameter when generating SEI class
CXF-370	<code>ResourceInjector</code> passes incorrect resource names
CXF-382	Java package to wsdl namespace mapping incorrect
CXF-390	Endpoint Bean provided configuration values do not override wsdl model
CXF-393	All JMS message properties are always copied from source to destination
CXF-394	HTTP binding expects '/' on each side of the brackets
CXF-398	XMLBinding Provider<Source> implementations can not receive Multipart/Related messages
CXF-405	<code>ws_addressing</code> sample fails
CXF-409	<code>samples/streamInterceptor</code> server fails to start
CXF-415	Decoupled Endpoint Observer should not insert <code>HttpRequest</code> and <code>HttpResponse</code> instances into the <code>Message</code> .
CXF-416	<code>ws_rm</code> and <code>streamInterceptor</code> samples fail
CXF-350	Add <code>SSL{Client Server}Policy</code> <code>CiphersuitesFilter</code> element

## Celtix Router

JIRA #	Description
TAN-140	Implement PAYLOAD mode routing for RPC-Lit style WSDL.
TAN-176	
TAN-156	Support faults in PAYLOAD mode.

## Enhancements

### Celtix Advanced Messaging

JIRA #	Description
QPID-320	Improve performance by remembering protocol version
QPID-271	Fanout Exchange not implemented
QPID-216	Implement no_local
QPID-215	Implement Custom JMSX Properties
QPID-200	set/get Destination not implemented in JMSMessage impl
QPID-180	PropertyFieldTable could be optimised

### Celtix Advanced Service Engine

JIRA #	Description
CXF-193	Support jax-ws handlers for dispatch/provider
CXF-147	support the wrapped and bare message at the same time
CXF-204	Remove Stream handlers
CXF-229	Lazy loading for attachments
CXF-231	Complete Simple Frontend
CXF-245	<code>SOAPHandlerInterceptor.handleMessage()</code> should be a no-op if no protocol handlers are registered.
CXF-249	Performance improvment by using <code>DOMXMLWriter</code>
CXF-261	Soap headers only work in doc/literal mode
CXF-266	Let servlet control can return list of services which are published by servlet for browsing
CXF-299	README should have links to bug tracking system
CXF-300	Create a demo to demonstrate stream interceptor
CXF-301	Refactor restful sample: remove the presence of wsdl from sample's client mode
CXF-319	Support for accessing message payload from logical handler when dispatch is used
CXF-332	Logical handler stops processing

JIRA #	Description
CXF-333	Logical handler throws <code>ProtocolException</code>
CXF-334	Logical handler's <code>close()</code> method is not properly called after completion
CXF-385	<code>XMLFaultOutInterceptor</code> does some redundant JAX-WS related fault detail processing.
CXF-389	Make <code>org.apache.cxf.jaxws.EndpointImpl</code> constructor generic

## Celtix Router

JIRA #	Description
TAN-97	Implement operation based routing.
TAN-181	Implement recipient list routing pattern.
TAN-182	Implement load balancing.
TAN-183	Routing management support
TAN-184	Container support
TAN-190	Implement message filter based routing.
TAN-263	Implement routing slip routing pattern.