

# Debugging and Logging

- [General CXF Logging](#)
- [Message Logging](#)
- [Old Message Logging](#)
- [ATOM Logging](#)

## Debugging Tools

### Eclipse IDE

See this [blog entry](#) for information on debugging web services using Eclipse. Note this is primarily for tracing/debugging source code; you will probably still want to use one of the tools below to capture network traffic, view SOAP requests and responses, etc.

### NetBeans IDE

NetBeans include a [debugger](#), profiler, and an HTTP monitor that can assist in troubleshooting SOA applications.

### tcpmonplus

[tcpmonplus](#) allows you to easily view messages as they go back and forth on the wire.

### WSMonitor

[WSMonitor](#) is another option to Tcpmon with slightly more functionality.

### NetSniffer

[NetSniffer](#) makes it possible to track the network traffic between arbitrary devices within a LAN segment.

### Wireshark

[Wireshark](#), a network packet analyzer, is useful for following the routing of SOAP messages. It can also help when you are getting an HTML error message from the server that your CXF client cannot normally process, by allowing you to see the non-SOAP error message.

### SOAP UI

[SOAP UI](#) can also be used for debugging. In addition to viewing messages, it allows you send messages and load test your services. It also has plugins for the [Eclipse IDE](#), [NetBeans IDE](#) and [IntelliJ IDEA](#).

## Other Helpful Tools

### WSDL Viewer

[WSDL Viewer](#) is a small tool to visualize web-services in a more intuitive way.

## SOAP Fault for debugging

### Stack trace in fault details

CXF supports the ability to put server stack trace information into the fault message fault details, if you enable the option of 'faultStackTraceEnabled'. It is useful for debugging if the soap fault message is not defined in the WSDL operation.

```
<jaxws:endpoint id="server" address="http://localhost:9002/TestMessage"
  wsdlURL="ship.wsdl"
  endpointName="s:TestSoapEndpoint"
  serviceName="s:TestService"
  xmlns:s="http://test" >
  <jaxws:properties>
    <entry key="faultStackTraceEnabled" value="true" />
  </jaxws:properties>
</jaxws:endpoint>
```

## Showing the cause exception message

CXF doesn't show the cause exception message in the fault message due to security consideration. However, this could potentially cause some trouble on the client side, as the client will not be able to see what the real cause of the exception is. You can let the CXF server return the fault message with the embedded cause exception message by enabling the option of 'exceptionMessageCauseEnabled' like this:

```
<jaxws:endpoint id="server" address="http://localhost:9002/TestMessage"
  wsdlURL="ship.wsdl"
  endpointName="s:TestSoapEndpoint"
  serviceName="s:TestService"
  xmlns:s="http://test" >
  <jaxws:properties>
    <entry key="exceptionMessageCauseEnabled" value="true" />
  </jaxws:properties>
</jaxws:endpoint>
```