



BEA SALT™

Samples Guide

Contents

BEA SALT Sample Guide

- BEA SALT Sample Applications 1
 - Inbound Samples 2
 - Basic Sample: simpapp 2
 - Attachment Sample: attachapp 2
 - Custom Type Sample: custtypeapp 2
 - Data Type Sample: datatypeapp 3
 - Reliable Messaging Sample: wsrmappp 3
 - Security Sample: secsapp 3
 - Fault Processing: faultapp 3
 - WS-Security 1.1: wsseapp 4
 - Outbound Samples 4
 - Outbound Authentication Mapping: obauthmapp 4
 - Outbound Web Service: obwsapp 4

BEA SALT Sample Guide

This section includes the following topics:

- [BEA SALT Sample Applications](#)

BEA SALT Sample Applications

BEA SALT allows external Web service applications to invoke native Tuxedo services (*inbound*), and conversely, allows Tuxedo applications to invoke external Web services (*outbound*). There 10 bundled Web service sample applications that demonstrate how to invoke BEA SALT using BEA WebLogic, Apache Axis or Microsoft .NET toolkits.

After BEA SALT is installed, you can find the following sample applications for your reference in the BEA SALT sample directory (each sample contains a detailed `readme` file):

- [Inbound Samples](#)
 - [Basic Sample: simpapp](#)
 - [Attachment Sample: attachapp](#)
 - [Custom Type Sample: custtypeapp](#)
 - [Data Type Sample: datatypeapp](#)
 - [Reliable Messaging Sample: wsrmappp](#)
 - [Security Sample: secsapp](#)
 - [Fault Processing: faultapp](#)

- [WS-Security 1.1: wsseapp](#)
- **Outbound Samples**
 - [Outbound Authentication Mapping: obauthmapp](#)
 - [Outbound Web Service: obwsapp](#)

Note: UNIX samples: `$TUXDIR/samples/salt`

Windows samples: `%TUXDIR%\samples\salt`

Inbound Samples

Basic Sample: `simpapp`

The Basic Sample demonstrates how to export a simple Tuxedo service as a Web service. The Tuxedo `simpapp` sample is used as an existing application to be exported as a Web service.

This sample contains all needed files to configure and export the `simpserv` server TOUPPER service as a Web service. The Web service accepts a single string parameter and converts it to uppercase. The client calls the service, and then prints the returned string.

This sample will enable you to learn the basics of running and accessing the GWWS server and the Web Services it provides.

Applicable Client Program(s): BEA WebLogic, Apache Axis for Java, Microsoft .NET.

Attachment Sample: `attachapp`

The Attachment Sample demonstrates how to transport CARRAY buffer types as MIME attachments according to SwA Protocol (SOAP with Attachment) in a SALT Web service. The Tuxedo `simpapp` sample is used as an existing application to be exported as a Web service. This sample contains all needed files to configure and export the `simpserv` server TOUPPER service as a Web Service.

Applicable Client Program(s): BEA WebLogic.

Custom Type Sample: `custtypeapp`

The BEA SALT 2.0 Custom Type Plug-in Sample demonstrates how to use SALT 2.0 plug-in extension mechanisms to implement customized mapping rules between Tuxedo Custom Typed Buffers and XML documents.

Applicable Client Program(s): BEA WebLogic.

Data Type Sample: datatypeapp

The Date Type Sample demonstrates how Tuxedo typed buffer are used in BEA SALT. In this sample shows how the FML and VIEW buffers and their sub-fields are defined in the Tuxedo Service Metadata Repository, and represented in a WSDL document. The WSDL document file generation utility, `tmwsdlgen` is used in this sample.

A WebLogic client program is also provided in this sample to help you get familiar with Web service client programming. The Tuxedo application server is a simple echo service in which the FML/VIEW buffer are checked and return the input data.

Applicable Client Program(s): BEA WebLogic.

Reliable Messaging Sample: wsrmap

This ReliableMessaging Sample demonstrates how to use BEA SALT WS-Reliable Messaging support and asynchronous communication with WS-Addressing. The Tuxedo `bankapp` sample is used as the Tuxedo application service provider. A WebLogic Server Web service client and standalone java Web service client are also included in this sample.

For more detailed WebLogic Server reliable messaging usage information, see http://e-docs.bea.com/wls/docs91/webserv/advanced.html#reliable_messaging.

Applicable Client Program(s): BEA WebLogic.

Security Sample: secsapp

The Security Sample leverages the existing Tuxedo `xmlstockapp` sample in a stock price query scenario. The STOCKQUOTE service is exported as a Web service by the GWWS server.

BEA SALT uses SSL/HTTPS to secure transport and message. It also supports Tuxedo authentication with HTTP Basic Authentication. You will learn how to configure security transport and how to authenticate using two Tuxedo authentication patterns: application password and user authentication.

A client program can be developed from the code in the sample combined with stub codes generated from the WSDL document. The WSDL document file is generated using the SALT configuration file and the `tmwsdlgen` utility.

Applicable Client Program(s): BEA WebLogic.

Fault Processing: faultapp

FAULT Processing Sample demonstrates how SALT handles user data with FAULT.

WS-Security 1.1: wsseapp

BEA SALT 2.0 implements part of WS-Security 1.1. It support Username token and X.509 token authentication, as well inbound message signature verification. This sample demonstrates how to bind WS-Security policy to Web services exported by a Tuxedo domain via GWWS. Three types of security are applied to three Web services respectively in this sample:

- TOUPPER: Username token authentication.
- TOLOWER: Username token and X.509 token authentication. Message integrity secured by signed soap body.
- REVERT: X.509 token authentication. Message integrity secured by signed soap body

Mandatory ACL security is enforced in the Tuxedo domain. X.509 token is mapped to a Tuxedo user using the certificate common name as the user name.

Applicable Client Program(s): BEA WebLogic.

Outbound Samples

Outbound Authentication Mapping: obauthmapp

The BEA SALT 2.0 Outbound Authentication mapping Plug-in Sample demonstrates how to use SALT 2.0 plug-in extension mechanisms to implement the HTTP basic authentication mapping from Tuxedo user id and group id.

Outbound Web Service: obwsapp

This sample demonstrates how to develop a Tuxedo client program to invoke Web Services using the BEA SALT Outbound Web Service feature. You can deploy a simple WEB service "Calculator" with Axis2 in your environment. The "Calculator" service provide "add" operation that adds two input integer number and return the result.