

# **Oracle® Service Architecture Leveraging Tuxedo (SALT)**

Samples Guide

10g Release 3 (10.3)

January 2009

**ORACLE®**

Copyright © 2006, 2009, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

## Oracle SALT Sample Guide

Oracle SALT Sample Directories .....	1
Oracle SALT Web Services Sample Applications .....	1
Inbound Samples .....	2
Basic Sample: simpapp .....	2
Attachment Sample: attachapp .....	2
Custom Type Sample: custtypeapp .....	3
Data Type Sample: datatypeapp .....	3
Reliable Messaging Sample: wsrmappp .....	3
Security Sample: secsapp .....	3
Fault Processing: faultapp .....	4
WS-Security 1.1: wsseapp .....	4
Outbound Samples .....	4
Outbound Authentication Mapping: obauthmapp .....	4
Outbound Web Service: obwsapp .....	4
Oracle SALT SCA Sample Applications .....	5
Basic Sample: simpapp .....	5
Other Uses .....	5
Advanced Sample: uBike .....	5
Other Uses .....	5
SCA Sample Using Web Services: calc client .....	6



# Oracle SALT Sample Guide

This section includes the following topics:

- [Oracle SALT Sample Directories](#)
- [Oracle SALT Web Services Sample Applications](#)
- [Oracle SALT SCA Sample Applications](#)

## Oracle SALT Sample Directories

After Oracle SALT is installed, you can find UNIX and Windows sample applications for your reference in the following Oracle SALT sample directories (each sample contains a detailed readme file):

- UNIX samples: `$TUXDIR/samples/salt`
- Windows samples: `%TUXDIR%\samples\salt`

## Oracle SALT Web Services Sample Applications

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

- [Inbound Samples](#)

- Basic Sample: `simpapp`
- Attachment Sample: `attachapp`
- Custom Type Sample: `custtypeapp`
- Data Type Sample: `datatypeapp`
- Reliable Messaging Sample: `wrmapp`
- Security Sample: `secsapp`
- Fault Processing: `faultapp`
- WS-Security 1.1: `wsseapp`
- Outbound Samples
  - Outbound Authentication Mapping: `obauthmapp`
  - Outbound Web Service: `obwsapp`

## 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):** Oracle 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):** Oracle WebLogic.

## Custom Type Sample: `custtypeapp`

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

**Applicable Client Program(s):** Oracle WebLogic.

## Data Type Sample: `datatypeapp`

The Date Type Sample demonstrates how Tuxedo typed buffer are used in Oracle 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):** Oracle WebLogic.

## Reliable Messaging Sample: `wrmapp`

This ReliableMessaging Sample demonstrates how to use Oracle 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://http://e-docs.bea.com/wls/docs103/webserv\\_adv\\_rpc/rm.html](http://http://e-docs.bea.com/wls/docs103/webserv_adv_rpc/rm.html).

**Applicable Client Program(s):** Oracle 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.

Oracle 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):** Oracle WebLogic.

## **Fault Processing: faultapp**

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

## **WS-Security 1.1: wsseapp**

Oracle SALT 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):** Oracle WebLogic.

## **Outbound Samples**

### **Outbound Authentication Mapping: obauthmapp**

The Oracle SALT Outbound Authentication mapping Plug-in Sample demonstrates how to use SALT 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 Oracle 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.



# Oracle SALT SCA Sample Applications

Three bundled SCA sample applications demonstrate how to develop applications using the SCA programming model, as well as configure the Oracle SALT SCA container.

- [Basic Sample: simpapp](#)
- [Advanced Sample: uBike](#)
- [SCA Sample Using Web Services: calc client](#)

## Basic Sample: simpapp

The Basic Sample demonstrates how to write a simple SCA application made up of a client program calling an SCA component via the Tuxedo infrastructure. It contains all the needed files to configure and deploy an SCA component hosted on a Tuxedo server, as well as the needed files to compile and configure an SCA client program to invoke the component. It represents an end-to-end application of SCA technology.

### Other Uses

The Basic Sample can invoke a regular Tuxedo ATMI service, or the SCA component may be invoked by a regular ATMI client. Also, the same SCA code can run without using `<binding.atmi>` in its SCDL configuration, demonstrating the flexibility of the setup.

## Advanced Sample: uBike

The Advanced Sample contains all the needed files to configure and deploy an SCA component hosted on a Tuxedo server, as well as the needed files to compile and configure an SCA client program to invoke the component. Data exchanged between client and component is of type `commonj:sdo::DataObject`, with the underlying transport being Tuxedo ATMI using STRING and FML32 Tuxedo buffers. It represents an end-to-end application of SCA and SDO technology.

### Other Uses

The Advanced Sample can invoke a regular Tuxedo ATMI service, or the SCA component can be invoked by a regular ATMI client. Also, the same SCA code may run without using `<binding.atmi>` in its SCDL configuration, demonstrating the flexibility of the setup.

## **SCA Sample Using Web Services: calc client**

The Web Services Sample demonstrates how to develop an SCA client program that invokes an external Web service. It contains all the needed files to configure Oracle SALT as needed by the runtime SCA configuration.