BEA SALT™

Product Overview

Version 2.0
Document Released: August 31, 2007

Contents

BEA SALT Overview

BEA SALT Overview

The following sections provide an overview to the BEA SALT product:

Understanding BEA SALT

BEA SALT (Service Architecture Leveraging Tuxedo) runs on top of Tuxedo. BEA SALT allows external Web service applications to invoke native Tuxedo services (*inbound*), and conversely, allows Tuxedo applications to invoke external Web services (*outbound*).

BEA SALT complies with standard Web service specifications (SOAP 1.1, SOAP 1.2, and WSDL 1.1), allowing BEA SALT to interoperate with other Web service products and development toolkits. Tuxedo applications can easily integrate with Web Services applications using BEA SALT.

What Are Web Services?

Web services are a set of functions packaged into a single entity made available to other systems on a network. They can be shared and used as a component of distributed Web-based applications. The network can be a corporate intranet or the Internet. Other systems, such as customer relationship management (CRM) systems, order-processing systems, and other existing back-end applications, can call these functions to request data or perform an operation. Because Web services rely on standard technologies which most systems provide, they are an excellent means for connecting distributed systems together.

The software industry has evolved toward loosely coupled service-oriented applications that interact dynamically over the Web. The applications break down the larger software system into smaller modular components, or shared services. These services can reside on different computers and can be implemented by vastly different technologies. They are packaged and made accessible using standard Web protocols, such as XML and HTTP.

Web services share the following properties that make them easily accessible from heterogeneous environments:

- Web services are accessed using widely supported Web protocols such as HTTP.
- Web services describe themselves using an XML-based description language.

  Web services communicate with clients (both end-user applications or other Web services) through simple XML messages that can be produced or parsed by virtually any programming environment or manually, if necessary.

Why Use BEA SALT?

BEA SALT is a native Tuxedo Web service integration solution. It reduces Tuxedo/Web Service integration costs and decreases conversion processes that may exist with other solutions for accessing Tuxedo services. It enables seamless connectivity between Tuxedo applications and external Web service applications.

BEA SALT allows existing Tuxedo services (inbound) to be easily exposed as Web services without additional programming tasks. It also allows you to create native Tuxedo applications that access external Web services (outbound) transparently.

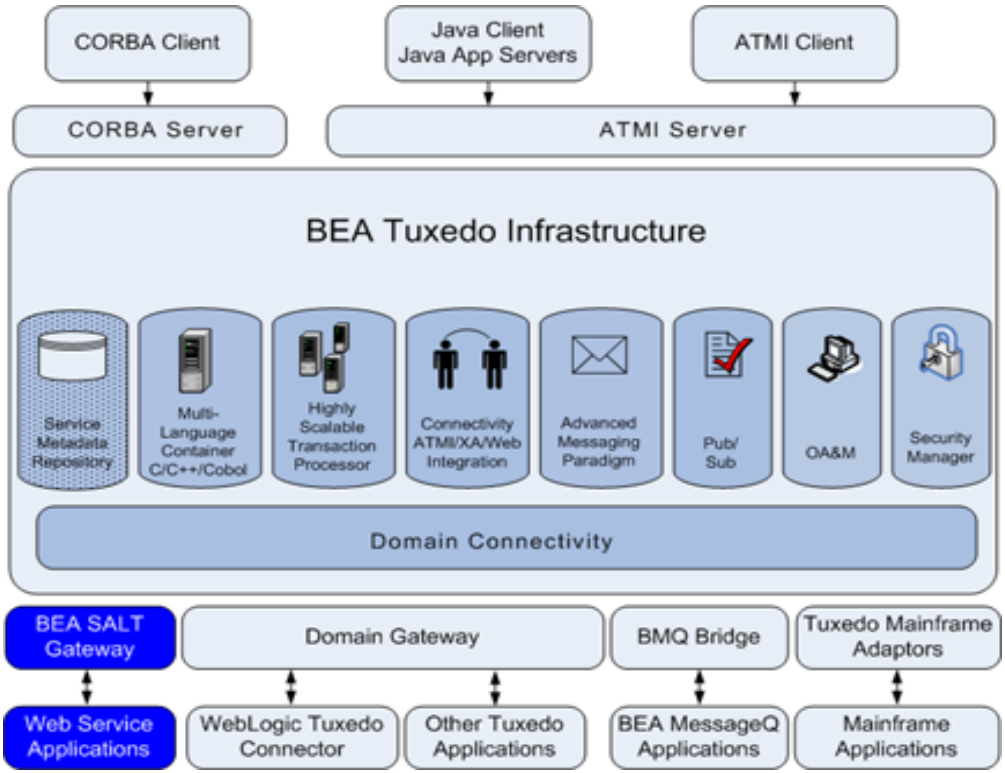
Major Web services benefits include:

- Interoperability among distributed applications that span diverse hardware and software platforms

- Easy, widespread access to applications using Web protocols
- A cross-platform, cross-language data model (XML) that facilitates developing heterogeneous distributed applications

Figure 1 illustrates how the BEA SALT gateway is used in the Tuxedo framework.

Figure 1 BEA SALT Gateway in Tuxedo Infrastructure



A Brief History of BEA SALT

BEA SALT is the latest add-on to the Tuxedo product family. Developed in 2006, BEA SALT is designed to provide a seamless Tuxedo solution of integrating Tuxedo applications and standard Web services application.

Release 1.1

Release 1.1 is the initial BEA SALT release. Made available in 2006, SALT 1.1 introduced the following major features:

- Inbound Service Support
  Permits Web service applications to invoke native Tuxedo services
- HTTP and HTTP over SSL Transport Support
- Asynchronous and Reliable Messaging Support

Release 2.0

The current BEA SALT 2.0 release incorporates significant enhancements based on SALT 1.1 release. SALT 2.0 introduces the following major features:

- Outbound Service Support
  Permits native Tuxedo applications to access external Web Services applications
- Extended WS-* Standards Support
- Improved Web Service Security.

What's New in SALT 2.0?

Outbound Support

The BEA SALT 2.0 release provides external Web service access capability from Tuxedo applications. You can easily extend Tuxedo integration capability without having an extensive understanding of Web services.

New WS-* Specification Support

- WS-Security 1.0 and WS-Security 1.1 Support (WSSE)

  WS-Security versions 1.0 and 1.1 are supported for inbound service authentication. Username Token and X.509 Token authentication are implemented in this release.

- SOAP Message Transmission Optimization Mechanism Support (MTOM)

  SALT 2.0 supports MTOM for inbound services. If a Tuxedo service uses CARRAY typed buffers or FML/32 typed buffers with FLD_CARRAY fields, the SALT 2.0 GWWS server

accepts SOAP Messages with XOP packages for CARRAY buffers or FLD_CARRAY fields.

Customer Enhancements

- Tuxedo TPFAIL Support for Web Services

  In SALT 1.1, if a Tuxedo service returns a TPFAIL containing important user data, the user data cannot be returned by Web service clients. SALT 2.0 provides TPFAIL enhancements. User data returned with TPFAIL by Tuxedo services can be returned via SOAP fault messages.

- Extensible Data Type Mapping and Message Conversion

  In SALT 1.1, all Tuxedo buffer types are tightly mapped with certain XML data types. SALT 2.0 allows you to associate a customized XML Schema definition with input and/or output typed buffer from any Tuxedo service. The generated SALT WSDL file incorporates the customized XML Schema definition for Web service client programming.

  In SALT 1.1, data conversion plug-in is only used for the custom typed buffer transformation. SALT 2.0 provides an extended message conversion plug-in mechanism. Customers can associate the input and/or output typed buffer from any Tuxedo service with a customized message conversion plug-in.

- Multiple Encoding Support

  In SALT 1.1, only UTF-8 encoded SOAP messages can be transmitted between GWWS servers and Web service applications. SALT 2.0 enhances encoding support capability. SALT 2.0 provides straightforward encoding transmission between SOAP messages and Tuxedo MBSTRING and XML typed buffers.

BEA SALT Feature Summary

BEA SALT consists of the following features:

- Configuration-Driven Deployment
- Data Type Mapping and Message Conversion
- Asynchronous and Reliable Messaging
- Security

Configuration-Driven Deployment

BEA SALT supports a configuration-driven deployment and management style. The BEA SALT deployment file is a single root XML file. The BEA SALT deployment file allows you to specify a list of GWWS processes and how they work for Web services.

Leveraging the Tuxedo Service Metadata Repository

Service contract information for all listed Tuxedo services must be made accessible via the Tuxedo Service Metadata Repository service provided by the local Tuxedo domain. BEA SALT leverages the Tuxedo Service Metadata Repository for mapping Tuxedo services and Web services.

Data Type Mapping and Message Conversion

BEA SALT provides a complete set of mapping rules for use between Tuxedo buffer/field types and XML primitive data types. The GWWS server provides automatic message transformation from Tuxedo buffers to SOAP messages (and vice versa), based on these mapping rules. SALT message conversion capability tremendously reduces programming effort when integrating Tuxedo applications with Web service applications. If BEA SALT pre-defined data type mapping rules are not sufficient, you can also develop customized conversion routines for any service to replace the default message conversion logic.

Asynchronous and Reliable Messaging

BEA SALT provides an asynchronous communication model that complies with the Web Service Addressing (WS-Addressing 1.0, Aug., 2004) specification. BEA SALT also supports reliable message delivery between Web service clients and Tuxedo servers conforming to the Web Service ReliableMessaging (WS-ReliableMessaging 1.0, Feb., 2005) specification.

The Web Service ReliableMessaging specification allows Web service instances to reliably communicate with the `GWWS` server in case of network failure or other scenarios. In particular, this feature provides for an interoperable protocol in which a message sent from a source endpoint to a destination endpoint is guaranteed with a defined delivery policy. The WS-Policy specification is supported to define the delivery policy for WS-ReliableMessaging.

Security

BEA SALT uses the Tuxedo security framework for inbound service authentication. Security is supported on three levels:

- Web Service Security Support
- HTTP Basic Authentication
- SSL Link-Level Security

Web Service Security Support

BEA SALT supports Web Service Security for inbound service authentications. User name tokens and X.509 tokens are supported to perform authentication based on the Tuxedo security framework.

HTTP Basic Authentication

BEA SALT supports HTTP Basic Authentication for both inbound and outbound services. For inbound services, the GWWS server supports Tuxedo user profiles passed from the Web service client via HTTP basic authentication protocol.

For outbound services, the GWWS server provides programming interfaces that allows customers to map Tuxedo user profiles to user profiles needed for outbound HTTP endpoints.

SSL Link-Level Security

The GWWS server handles both HTTP and HTTP/S requests. Certificate information for setting up an SSL connection can be specified in the BEA SALT Deployment file.

BEA SALT Components

BEA SALT consists of the following major components:

- BEA SALT Gateway (GWWS)
- WSDL Assistant Utilities

BEA SALT Gateway (GWWS)

The BEA SALT provided Tuxedo system server (GWWS), connects with other Web service applications via SOAP over HTTP/S protocol. The GWWS server acts as a Tuxedo gateway process and is managed in the same manner as general Tuxedo system servers. Each GWWS server has bi-directional (inbound/outbound) capability. The GWWS server:

- accepts SOAP requests from Web service applications and issue Tuxedo native calls to Tuxedo services.

- accepts Tuxedo ATMI requests and issues SOAP calls to Web Service applications.

You can have multiple GWWS instances in one Tuxedo domain. The same functionality for multiple GWWS instances is provided by specifying the same BEA SALT configuration to improve throughput and failover protection. You can also group multiple GWWS instances in different configuration files for different purposes.

When the GWWS server boots, it loads the specified SALT configuration file and Tuxedo service contract information from the Tuxedo Service Metadata Repository.

The GWWS server also acts as a simple HTTP Web server for WSDL document and XML Schema file download.

WSDL Assistant Utilities

The Web Services Description Language (WSDL) is an XML-based specification that describes a Web service. A WSDL document describes Web service operations, input and output parameters, and how a client application connects to the Web service. BEA SALT provides two utilities (`tmwsdlgen` and `wsdlcvt`) to map Tuxedo applications and Web Service WSDL descriptions.

WSDL Generator from Tuxedo Definitions

When using BEA SALT to publish Tuxedo services as Web services, you do not need to compose a WSDL document manually; it is automatically generated as part of the SALT Web service development process. The generated WSDL document can be integrated using Web service development tools, or can be published to a UDDI server.

There are two ways to obtain a WSDL document:

- Use `tmwsdlgen` (the WSDL document file generating utility).
- Download the GWWS server generated WSDL document via HTTP(S).

WSDL Convertor to Tuxedo Definitions

To support external Web Service applications, external WSDL documents need to be converted. The BEA SALT conversion utility, `wsdlcvt`, converts external WSDL documents to Tuxedo specific definition files (SALT Web Service Definition file, Tuxedo Service Metadata Repository Definition file and FML32 Field Table Definition file).

The SALT Web Service Definition file can be imported into a SALT Deployment file and utilized by a particular GWWS server. The Tuxedo Service Metadata Repository Definition file and

FML32 Field Table Definition file provide service interface descriptions for Tuxedo client programming.

BEA SALT Use Cases

The following sections describe the most common BEA SALT Web services use cases:

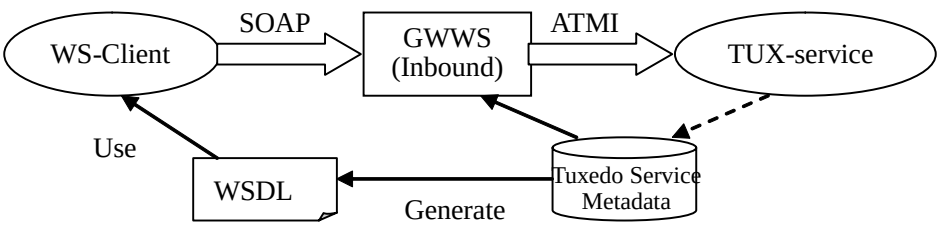
- Use Case 1: Exposing Native Tuxedo Services as Web Services
- Use Case 2: Invoking Web Services from Tuxedo Applications
- Use Case 3: Connecting Tuxedo Domains Using SOAP Protocol

Use Case 1: Exposing Native Tuxedo Services as Web Services

Native Tuxedo services can be exposed as Web services using standard Web service SOAP protocol. The GWWS server accepts SOAP requests through HTTP/S and then converts them into Tuxedo ATMI calls. SALT generates a WSDL document that describes the open standard Web service interfaces for Tuxedo services.The Tuxedo Service Metadata Repository is used to define Tuxedo service contract information. This is an “inbound” use case.

Figure 2 illustrates a generic inbound Web service call.

Figure 2 Exposing Tuxedo Services as Web Services (Inbound)



Use Case 2: Invoking Web Services from Tuxedo Applications

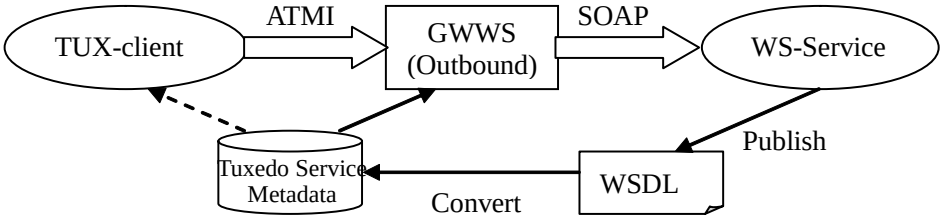
Web service applications can be imported into a Tuxedo domain, advertised as Tuxedo services through the GWWS server, and invoked from Tuxedo applications. SALT converts and maps each `wsdl:operation` as a particular Tuxedo service. The GWWS server advertises the mapped

services (called SALT proxy services), and accepts Tuxedo ATMI requests from Tuxedo applications.

The Tuxedo Service Metadata Repository is used to store converted Tuxedo service contract information and helps Tuxedo programmers understand what type of Tuxedo buffers are expected for the imported SALT proxy services. This is an “outbound” use case.

Figure 3 illustrates a generic outbound Web service call.

Figure 3 Invoking Web Services from Tuxedo Applications (Outbound)



Use Case 3: Connecting Tuxedo Domains Using SOAP Protocol

BEA SALT also allows you to connect two *different* Tuxedo domains using GWWS servers as an alternative to using /T domain. The GWWS server in the calling domain works in an outbound direction, the GWWS server in the receiving domain works in an inbound direction.

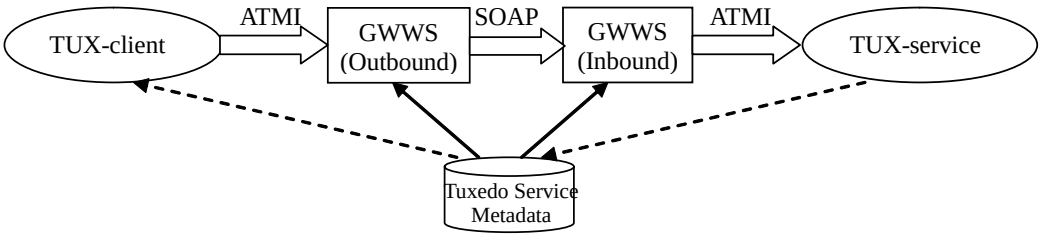
The receiving Tuxedo domain must propagate the Tuxedo service definition to the calling Tuxedo domain. This means that the calling domain Tuxedo Service Metadata Repository must contain the Tuxedo service definition file that runs in the receiving domain.

Note: This should be set up *manually*. The Tuxedo Service Metadata Repository infrastructure does not currently provide automatic propagation between Tuxedo domains.

The WSDL document is not required. BEA SALT provides simple configurations to allow two GWWS servers to work together for domain connectivity using SOAP protocol without needing to exchange WSDL documents.

Figure 4 illustrates how to use BEA SALT to connect two domains.

Figure 4 Connecting Two Tuxedo Domains with SOAP protocol



Two GWWS servers should not be used to create connections within the *same* Tuxedo domain, see Figure 5. Also, a single GWWS server cannot connect to itself, see Figure 6.

In either scenario, the GWWS server advertises the same Tuxedo services which are already advertised by other application servers. This might result in *dead-loop* service dispatching.

WARNING: It is strongly advised that you carefully plan and configure your BEA SALT application to avoid these scenarios.

Figure 5 Two GWWS Servers Making a Connection Within the Same Tuxedo Domain

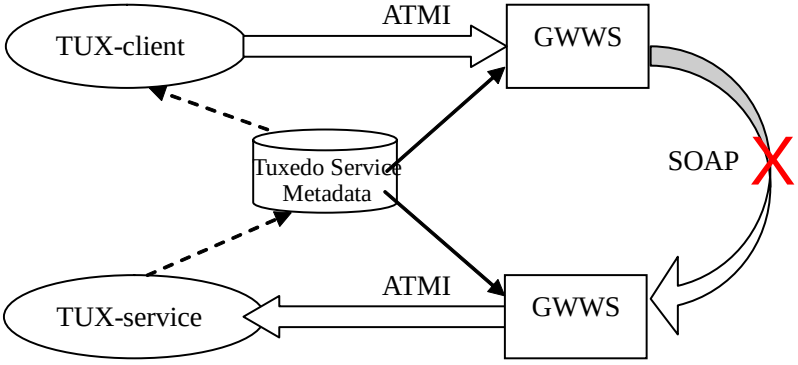
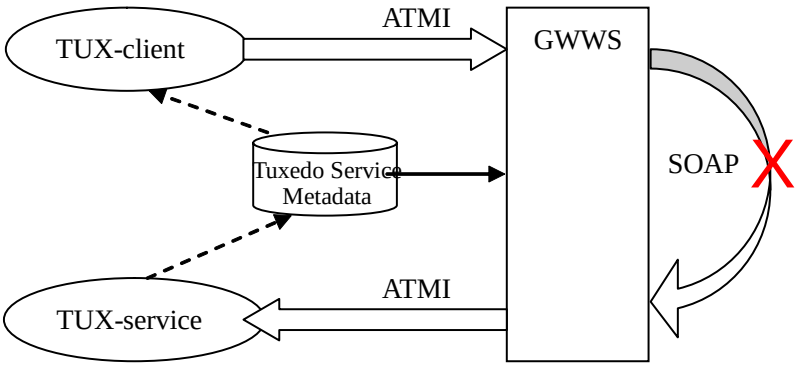


Figure 6 Single GWWS Server Making a Connection to Itself



Configuring Web Services with BEA SALT

The following steps are used typically when yo configure Web services using BEA SALT:

1. Configure Inbound Tuxedo Services
   a. Define Tuxedo application services using the Tuxedo Service Metadata Repository.
   b. Compose one or more BEA SALT Web Service Definition Files (WSDF).
2. Configure Outbound Web Services
   a. Convert an external WSDL document into Tuxedo the following components: SALT Web Service Definition file, Tuxedo Service Metadata Repository Definition file, and FML32 Field Table Definition file.
   b. Resolve potential naming conflicts for the auto-generated service names and FML32 field names.
3. Load all Tuxedo Service Definitions into the Tuxedo Service Metadata Repository using `tmloadrepos`.
4. Compose the BEA SALT Deployment File for both inbound and outbound services.
5. Add the `TMMETADATA` and `GWWS` servers to your Tuxedo `UBBCONFIG` file.
6. Boot the Tuxedo application.

Invoking Tuxedo Services Using Web Service Client Toolkits

1. Client end user downloads the WSDL document file from the GWWS server.
2. Client end user generates client-side stubcode from the WSDL document file with a SOAP development kit.
3. Generate client-side program.
4. Run the client to invoke the Web service with SOAP messages.

Invoking Web Services Using Tuxedo Programming Interfaces

1. Create a Tuxedo client/server program according to the generated Tuxedo Service Metadata Definition file. The client program can be written in any Tuxedo supported client-side programming language (C/C++, Java, COBOL, .NET, and so on).
2. Compile and deploy the Tuxedo client/server program.
3. Run the Tuxedo application to invoke the external Web service applications.

BEA SALT Supported Standards

BEA SALT supports the following Web service standards:

- Standards for transmitting data and Web service invocation calls between the Web service and the user of the Web service.
  - SOAP 1.1

    For more information, see SOAP 1.1 specification
    http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
  - SOAP 1.2

    For more information, see SOAP 1.2 specification
    http://www.w3.org/TR/soap12-part0/
  - SOAP with Attachment

    The SOAP with Attachment specification describes a standard way to associate a SOAP message with an attachment in its native format in a multipart MIME structure for transport. BEA SALT allows you to represent a MIME-attached Tuxedo CARRAY typed buffer.

For more information, see SOAP Messages with Attachments http://www.w3.org/TR/SOAP-attachments

  - MTOM

    The MTOM specification describes an optimized mechanism for SOAP message transmission. BEA SALT supports the optimized MIME Multipart/Related Serialization of the SOAP messages. It provides performance benefits for calling Tuxedo services that consume large CARRAY buffers.

    For more information, see SOAP Message Transmission Optimization Mechanism specification http://www.w3.org/TR/soap12-mtom/

- A standard for client applications to find a registered Web service and to register a Web service.
  - UDDI 2.0

    For more information, see UDDI 2.0 specification http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm

- Standards for describing the Web service to clients so they can invoke it.
  - WSDL 1.1

    For more information, see WSDL1.1 specification http://www.w3.org/TR/2001/NOTE-wsdl-20010315

  - WS-Policy

    For more information, see WS-Policy specification http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf and WS-PolicyAttachment specification http://specs.xmlsoap.org/ws/2004/09/policy/ws-policyattachment.pdf

- Standards for Web Service infrastructure.
  - WS-Addressing

    For more information, see WS-Addressing specification http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/

  - WS-ReliableMessaging

    For more information, see WS-ReliableMessaging specifications http://specs.xmlsoap.org/ws/2005/02/rm/ws-reliablemessaging.pdf and http://specs.xmlsoap.org/ws/2005/02/rm/WS-RMPolicy.pdf

- Standards for Web Service Security.

– WS-Security 1.0

  For more information, see WS-Security 1.0 specifications:

  http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

  http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

  http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf

– WS-Security 1.1

  For more information, see WS-Security 1.1 specifications:

  http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

  http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf

  http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf

What Next?

After becoming familiar with the BEA SALT Product Overview, refer to the following topics for installing, configuring, and running Web services using the SALT product:

- Install the BEA SALT product.

  For an explanation of how to install the product, refer to *BEA SALT Installation Guide*.

- Configure and administer the BEA SALT product.

  For an explanation of how to configure and administer the product, refer to *BEA SALT Administration Guide*.

- Program Web Services with the BEA SALT product.

  For an explanation of how to program with SALT refer to *BEA SALT Programming Web Services*.