BEA SALT™

Release Notes

Version 2.0
Document Revised: August 31, 2007

Contents

BEA SALT 2.0 Release Notes

BEA SALT 2.0 Release Notes

The following topics are discussed in this section:

About This BEA SALT Release

BEA SALT (Service Architecture Leveraging Tuxedo) is the latest add-on to the Tuxedo product family. It provides a native Tuxedo Web service integration solution. BEA SALT allows external Web service applications to invoke native Tuxedo services (inbound), and conversely, allows Tuxedo applications to invoke external Web services (outbound).

The current BEA SALT 2.0 release incorporates significant enhancements based on the SALT 1.1 release.

What's New and Improved

BEA SALT Release 2.0 includes the following new features and enhancements:

- Outbound Support

  SALT Release 2.0 provides accessing external Web services capability from Tuxedo applications. You can easily extend Tuxedo integration capability without having an extensive understanding of Web services.

- Extended WS-* Standard Support
  - Web Service Security 1.0 and Web Service Security 1.1 (WSSE)

    Web Service Security version 1.0 and 1.1 are supported for inbound service authentications. Username Token and X.509 Token authentication are implemented in this release.

  - SOAP Message Transmission Optimization Mechanism (MTOM)

    SALT 2.0 supports SOAP message with XOP packages for CARRAY buffers or FLD_CARRAY fields.

- Customer Enhancements
  - Tuxedo TPFAIL Support for Web Services

    Provides enhancements to Tuxedo TPFAIL. User data returned with TPFAIL by Tuxedo services can be returned using SOAP fault messages.

  - Extensible Data Type Mapping and Message Conversion

    Provides customized Tuxedo buffer/SOAP message payload conversion that replaces SALT default data type mapping and message conversion.

  - Multiple Encoding Support

    Supports encoding (other than UTF-8) for SOAP messages. Provides straightforward encoding transmission between SOAP messages and Tuxedo STRING/MBSTRING/XML typed buffers.

Upgrade Considerations

For information on installing BEA SALT 2.0 on top of a previous SALT release, see the *BEA SALT Installation Guide*.

For information about migrating from BEA SALT 1.1 application, see Migrating from BEA SALT 1.1 Application in the BEA SALT *Administration Guide*.

BEA SALT Installation Prerequisites

Before installing BEA SALT 2.0, you must ensure that BEA Tuxedo 9.1 is installed.

Notes: For Tuxedo 9.1, rolling patch 070 or above is required.

For more information, see System Requirements in the BEA SALT *Installation Guide*.

BEA SALT Platform Support

BEA SALT 2.0 supported platforms are listed in Appendix A: BEA SALT 2.0 Supported Platforms in the BEA SALT *Installation Guide*.

Interoperability Considerations

BEA SALT 2.0 is compatible with, and fully supports, most industry-standard Web service development toolkits. For more information, see Interoperability Considerations in the BEA SALT *Administration Guide*.

Licensing Requirements

As a prerequisite, BEA Tuxedo 9.1 must be installed and operational with its license key file available. During installation, you are requested to select the BEA SALT license key directory in order to enable BEA SALT software. The BEA SALT license key is then appended to the BEA Tuxedo license key file.

Note: You can also enter your BEA SALT license key manually. For more information, see Post BEA SALT Installation in the BEA SALT *Installation Guide*.

Known and Resolved Issues

The following sections describe known problems in BEA SALT 2.0, as well as problems that were resolved in version 2.0. Entries include a description of the problem, and a workaround or solution where appropriate. Each problem is listed by the Change Request (CR) number. The CR number is provided to facilitate the tracking of these problems. A notation in the "**Fixed In**" column indicates that the problem has been resolved.

- GWWS Runtime
- Interoperability

- Documentation

GWWS Runtime

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR334161	**Problem: GWWS rejects non UTF-8 inbound SOAP request messages when SignBody WS-Security Policy is enabled.** When GWWS is configured with multiple encoding support, it can accept non UTF-8 encoded SOAP requests; however, the GWWS internally converts all non UTF-8 encoding messages into UTF-8 encoding messages for later operation. If a service requires <soap:Body> signature verification, the GWWS always verify the signature against the converted UTF-8 encoded <soap:Body> instead of the original <soap:Body> content. Thus the signature verification always failed.	2.0	
	Platform: All		
	Workaround: Web service client programs must initiate SOAP requests using UTF-8 encoding when the WS-Security Policy Assertion SignBody is enabled for the corresponding services.		
CR328329	**Problem: GWWS may reject valid SOAP requests if the target Tuxedo service consumes XML typed buffer as input and the input buffer is defined with "size" restriction in the Tuxedo Service Metadata definition.** GWWS automatically adds an additional '\0' to the end of the converted XML buffer. This additional byte may result the XML buffer length exceed the "size" value, hence reject by later Tuxedo buffer validation routine in the GWWS.	2.0	
	Platform: All		
	Workaround: Enlarge or remove the "size" restriction for XML typed buffer in the Tuxedo Service Metadata Definition.		

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR306710	**Problem: Tuxedo service may not receive the exact same non `UTF-8` encoding string as the string prepared in the SOAP request message.** If multiple encoding capability is turned on for the `GWWS`, and Web Service client programs written in Java send messages with non `UTF-8` encoding, `GWWS` may not send exact the same string value to the Tuxedo service. This is a general problem if different encoding conversion implementations are used. Java encoding implementation has slight difference from ICU encoding implementation (which is used by Tuxedo and SALT), hence an encoding string prepared by the Java program, after ICU "to `UTF-8`" and "from `UTF-8`" conversion, may not revert to the exact original string.	2.0	
	Platform: All		
	Workaround: None. Customers rarely use those characters. If some characters mapping are confirmed due to ICU bugs, please contact BEA Tuxedo Customer Support.		

CR Number	Description and Workaround or Solution	Found In	Fixed In

CR269765	**Problem: BEA SALT does not support multi-reference values in SOAP 1.1 messages.**
	GWWS cannot recognize SOAP 1.1 request messages with multi-reference values. See the following sample SALT un-supported soap 1.1 message segment:
	`<soap:Envelope ...>`
	`  ......`
	`  <soap:Body xmlns:e=... >`
	`  ......`
	`  <e:Books>`
	`  <e:Book>`
	`  <title>My Life and Work</title>`
	`  <author href="#henryford" />`
	`  </e:Book>`
	`  <e:Book>`
	`  <title>Today and Tomorrow</title>`
	`  <author href="#henryford" />`
	`  </e:Book>`
	`  </e:Books>`
	`  <author id="henryford">`
	`  <name>Henry Ford</name>`
	`  </author>`
	`  </soap:Body>`
	`</soap:Envelope>`
		1.1	2.0

	Platform: All		
	Solution: SOAP message with multi-reference values is supported in BEA SALT 2.0.		

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR283757	**Problem: `GWWS` does not reject SOAP request messages that containing invalid values of the following XML primitive data types:** **`xsd:byte, xsd:short, xsd:long, xsd:float, xsd:double`** The `GWWS` server tries to convert the invalid values of the above XML data types to valid C language primitive type values. For example, invalid `xsd:float` value "`1.23.45`" is converted to C float value "`1.23`".	1.1	2.0
	Platform: All		
	Solution: `GWWS` performs more strict data format validation against XML primitive numeric data type definitions.		

Interoperability

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR330363	**Problem: SALT multiple encoding feature does not interoperable with Microsoft .NET WCF 3.0 engine.** If SALT enables multiple encoding feature, when the inbound call Tuxedo service returns `MBSTRING` or `XML` typed buffer with non `UTF-8` encoding, the SOAP response message is encoded the same as the `MBSTING` or `XML` buffer. Such SOAP response message cannot be accepted by those Web Service client applications developed using Microsoft .NET WCF 3.0 engine.	2.0	
	Third-Party Web Service Toolkit: Microsoft .NET WCF 3.0		
	Workaround: Customers may need to develop custom encoder/decoder if the Tuxedo service may return non `UTF-8` typed buffers and `GWWS` multiple encoding feature is turned on. Alternatively, you may explicitly turn off the `GWWS` multiple encoding feature if you are aware all Tuxedo services in your Tuxedo domain never return non `UTF-8` buffers.		
CR296594	**Problem: SOAP fault response message cannot be accepted by Microsoft .NET 3.0 when the HTTP Content-Length exceeds 65536.** If the `GWWS` server returns a SOAP fault message when the HTTP Content-Length exceeds 65536, the .NET WCF 3.0 engine sends an exception to report the response is not well-formed. **Note:** If the `GWWS` server returns a normal SOAP message (non SOAP fault) when the HTTP Content-Length exceeds 65536, the .NET Web service engine can accept.	2.0	
	Third-Party Web Service Toolkit: Microsoft .NET WCF 3.0		
	Workaround: None. Avoid to return big buffer when invoking tpreturn() along with TPFAIL status code in the Tuxedo service.		

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR294785	**Problem: Apache Axis2/Java failes to handle Tuxedo FML32 TPFAIL response buffers that have field names with initial uppercase.** If a Tuxedo service returns TPFAIL with FML32 buffer, SALT maps each field as an XML segment in the SOAP fault detail, and the field name is used directly as the XML element tag name. If the FML32 buffer contains field names with initial letter uppercase, Axis2 may not recognize the SOAP fault messages that converted from this Tuxedo FML32 buffer.	2.0	
	Third-Party Web Service Toolkit: Apache Axis2/Java		
	Workaround: Modify the FML32 field name to avoid use initial uppercase name. Corresponding Tuxedo application also needs to be changed and re-compiled.		
CR306978	**Problem: Apache Axis2/Java does not recognize the SOAP with Attachment (SwA) featured WSDL file generated by BEA SALT.** If SwA featured WSDL file is generated by BEA SALT, Apache Axis2 `wsdl2java` utility generates Java stub code which is different from Apache Axis. Axis2 generated stub code cannot initiate a successful call to BEA SALT service.	2.0	
	Third-Party Web Service Toolkit: Apache Axis2/Java		
	Workaround: Use Apache Axis instead for SwA featured soap calls. MTOM is an alternative attachment format that supported by BEA SALT. You may also use MTOM feature with Apache Axis2/Java for CARRAY buffer stream.		

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR296221	**Problem: Apache Axis wsdl2java utility fails to compile the BEA SALT generated WSDL file if soap 1.2 binding with soap fault is defined in the WSDL file.**	2.0	
	Third-Party Web Service Toolkit: Apache Axis		
	Workaround: This is an Apache Axis bug, please refer to https://issues.apache.org/jira/browse/AXIS-2614. You may define SOAP version 1.1 for SALT WSDL if Apache Axis has to be used for Web Service client programming. Or you should manually re-compile Apache Axis classes using Apache Axis source code with the fix provide in the above URL link. You may also choose another third-party Web Service client toolkit for soap 1.2 binding with soap fault feature, such as BEA WebLogic 9.x Web Services, Apache Axis2, Microsoft .NET WCF 3.0, etc.		

CR Number	Description and Workaround or Solution	Found In	Fixed In

CR323477	**Problem: GWWS fails to call external Web Service applications built upon Microsoft .NET WCF 3.0 if asynchronous WS-Addressing feature is enabled.**
	BEA SALT supports WS-Addressing feature that conforms to WS-Addressing standard 200408 submission. While initiating an asynchronous outbound call, `GWWS` always defines a `<wsa:ReplyTo>` endpoint reference in the WS-Addressing soap header. See the following sample `<wsa:ReplyTo>` segment:
	`<wsa:ReplyTo>`
	`  <wsa:Address>`
	`  http://myhost:7102/?wsa_Msg_ID=uuid:B437A4F4-AF23-111E-FFFFFFAC1622FFFFFF9F0000-6BBE`
	`  </wsa:Address>`
	`</wsa:ReplyTo>`
	Host name "`myhost`" and port number "`7102`" in the above sample indicates the listening endpoint that is created by the `GWWS` which is used to accept asynchronous soap response messages for outbound calls.
	But Microsoft .NET WCF 3.0 does not recognize the `<wsa:ReplyTo>` endpoint in the request, and always returns the synchronous response through the request connection.
	`GWWS` then always encounters time out in receiving asynchronous response because Microsoft .NET WCF 3.0 never send the response to `GWWS` expected endpoint.
		2.0	

	Third-Party Web Service Toolkit: Microsoft .NET WCF 3.0		
	Workaround: None. You should disable WS-Addressing feature when initiating outbound call to external Web Service applications built upon Microsoft .NET WCF 3.0 . For more information about configuring WS-Addressing feature, see "Configuring Advanced Web Service Messaging Features" in the *BEA SALT Administration Guide*.		

Documentation

CR Number	Description and Workaround or Solution	Found In	Fixed In
CR304509	**Problem: BEA SALT 1.1 Documentation does not address very clear in using TMTRACE for GWWS tracing.**	1.1	2.0
	Platform: N/A		
	Solution: Please refer to "Tracing the GWWS Server" in the *BEA SALT Administration Guide* for the most up-to-date information.		

Where to Get Product Documentation

Documentation for this product is available from the BEA corporate Web site. From the BEA home page at http://www.bea.com, click on Product Documentation or go directly to the "e-docs" Product Documentation page at http://e-docs.bea.com.

To access the .PDF files, open the BEA SALT documentation home page, click the PDF files button and select the document you want to view or print. If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com.

Contacting BEA Customer Support

If you have any questions about this BEA SALT version, or if you have problems installing and running BEA SALT, contact BEA Customer Support through BEA WebSupport at www.bea.com.

You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes

- The name and version of the products you are using
- A description of the problem and the content of pertinent error messages

See Also

- *BEA SALT Product Overview*
- *BEA SALT Installation Guide*
- *BEA SALT Administration Guide*
- *BEA SALT Programming Web Services*
- *BEA SALT Reference Guide*