Oracle® Service Architecture Leveraging Tuxedo (SALT)

Release Notes

10*g* Release 3 (10.3)

January 2009

ORACLE®

Oracle SALT Release Notes, 10*g* Release 3 (10.3)

Contents

Oracle SALT 10*g* Release 3 (10.3) Release Notes

Oracle SALT 10*g* Release 3 (10.3) Release Notes

Table 1 Revision History

Revision Date	Summary of Change
January 15, 2009	Initial Release

The following topics are discussed in this section:

- About This Oracle SALT Release

- Upgrade Considerations

- Oracle SALT Installation Prerequisites

- Oracle SALT Platform Support

- Interoperability Considerations

- Known Issues

- Where to Get Product Documentation

- Contacting Oracle Customer Support

About This Oracle SALT Release

Oracle SALT (Service Architecture Leveraging Tuxedo) is the latest add-on to the Tuxedo product family. It provides a native Tuxedo Web service integration solution. Oracle 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 Oracle SALT 10*g* Release 3 (10.3) release incorporates significant enhancements based on the SALT 2.0 release.

What's New and Improved

Oracle SALT Release 10*g* Release 3 (10.3) includes the following new features:

Service Component Architecture (SCA) Programming

SCA provides a new programming model that aims at simplifying component re-use and seamless communications between components. The SALT 10*g* Release 3 (10.3) SCA container enables new programming model and leverages Tuxedo's most valued features, such as reliability, availability, scalability, and performance. SALT 10*g* Release 3 (10.3) introduces the following SCA features:

– Client-side binding for SCA invocations over ATMI and SOAP

– Server-side binding for serving SCA requests made over ATMI and SOAP

– Client-side binding for SCA invocations from Java environments

– Development and runtime tools: Commands to build and deploy SCA clients and servers as well as commands for runtime administration. For more information, see the SALT 10*g* Release 3 (10.3) Command Reference Guide.

– Authentication and authorization for SCA services

– Global transactions

– Thread-safe SCA/SDO clients and servers

– SCDL schema validation

– Support for simple data types

– Support for complex data types using SDO

– Automatic data transformation to/from Tuxedo buffer types

– Support for multi-byte characters using multiple encoding

For more information, see the SALT 10*g* Release 3 (10.3) Programming Guide.

Service Contract Discovery

Automatically discover service contract information at run time. The generated information can be put into metadata repository automatically or to a file which can then be loaded manually into the metadata repository using the `tmloadrepos` utility.

Access Log for All Incoming Requests

Assists Tuxedo client administrators to monitor application validity at runtime. You can record application high water client count, current client count, and named users.

Upgrade Considerations

For information on installing Oracle SALT 10*g* Release 3 (10.3) on top of a previous SALT release, see the *Oracle SALT Installation Guide*.

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

Oracle SALT Installation Prerequisites

Before installing Oracle SALT 10*g* Release 3 (10.3), you must ensure that Oracle Tuxedo 10*g* Release 3 (10.3) is installed.

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

Oracle SALT Platform Support

Oracle SALT 10*g* Release 3 (10.3) supported platforms are listed in Appendix A: Oracle SALT 10*g* Release 3 (10.3) Supported Platforms in the Oracle SALT *Installation Guide*.

Interoperability Considerations

Oracle SALT 10*g* Release 3 (10.3) is compatible with, and fully supports, most industry-standard Web service development toolkits. For more information, see Interoperability Considerations in the Oracle SALT *Interoperability Guide*.

Known Issues

The following sections describe known problems in Oracle SALT 10*g* Release 3 (10.3). 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.

- GWWS Runtime

- Interoperability

- SCA Container

GWWS Runtime

CR Number	Description and Workaround or Solution	Found 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.	

CR Number	Description and Workaround or Solution	Found In
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.	
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 Oracle Tuxedo Customer Support.	

Interoperability

CR Number	Description and Workaround or Solution	Found 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
CR294785	**Problem: Apache Axis2/Java fails 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 Oracle SALT.** If SwA featured WSDL file is generated by Oracle 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 Oracle 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 Oracle SALT. You may also use MTOM feature with Apache Axis2/Java for CARRAY buffer stream.	

CR Number	Description and Workaround or Solution	Found In
CR296221	**Problem: Apache Axis wsdl2java utility fails to compile the Oracle 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 Oracle WebLogic 9.x Web Services, Apache Axis2, Microsoft .NET WCF 3.0, etc.	

CR Number	Description and Workaround or Solution	Found 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.**
	
	Oracle 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-AF2`
	`3-111E-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 *Oracle SALT Administration Guide*.	

SCA Container

CR Number	Description and Workaround or Solution	Found In

1. CR379052	**Problem: XmlHelper::save() loses content of XML document of mixed type**
	
	In a schema where a type is defined as follows:
	
	`<xsd:complexType name="myType" mixed="true"> ...`
	
	A document cannot be loaded by SDO XmlHelper, then saved as its original content. For example:
	
	`...`
	
	`     Dear Mr.<name>John Smith</name>.`
	
	`     Your order <orderid>1032</orderid>`
	
	`     will be shipped on`
	`<shipdate>2001-07-13</shipdate>.`
	
	`...`
	
	can only be saved as:
	
	`...`
	
	`     <name>John Smith</name>`
	
	`     <orderid>1032</orderid>`
	
	`     <shipdate>2001-07-13</shipdate>`
	
	`...`
		10g R3

	Platform: All	
	Workaround: These types of documents should be handled directly by the application code.	

CR Number	Description and Workaround or Solution	Found In
2. CR379307	**Problem: Error data sent is ignored when Web service binding is used to connect two Tuxedo domains** The current implementation of Web services binding only returns the fault string when a SOAP fault occurs. For those services where fault detail may contain additional information or data (as handled by the GWWS SALT gateway), `ServiceInvocationException` has no place or mechanism to store this data. This may happen: • when attempting to invoke an existing Tuxedo service exposed as a Web service from an SCA component or SCA client. • only between two Tuxedo domains. **Platform:** All **Workaround:** Use ATMI binding and /Domain feature to connect two Tuxedo domains.	10g R3
3. CR383168	**Problem: JATMI binding does not support transaction.** The JATMI binding crashed when running with transaction because the JATMI container does not have sufficient transaction support for TSESSION. **Platform:** All **Workaround:** The services accessed may be configured as AUTOTRAN.	10g R3
4. CR383362	**Problem: JATMI reference binding does not check the presence of two different** `serviceType`**,** `inputBufferType`**,** `outputBufferType` **and** `errorBufferType` **elements without specifying a target attribute.** **Platform:** All **Workaround:** To avoid this problem, you should not configure duplicated XML elements for ATMI binding without specifying target in the composite file.	10g R3

Where to Get Product Documentation

Documentation for this product is available from the Oracle corporate Web site. From the Oracle home page at http://www.oracle.com.

To access the .PDF files, open the Oracle 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 Oracle Customer Support

If you have any questions about this Oracle SALT version, or if you have problems installing and running Oracle SALT, contact Oracle Customer Support through Oracle WebSupport at http://www.oracle.com/bea/support.html.

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

- Oracle SALT 10*g* Release 3 (10.3) Product Overview

- Oracle SALT 10*g* Release 3 (10.3) Installation Guide

- Oracle SALT 10*g* Release 3 (10.3) Administration Guide

- Oracle SALT 10*g* Release 3 (10.3) Programming Guide

- Oracle SALT 10*g* Release 3 (10.3) Reference Guide