

Oracle® Service Architecture Leveraging Tuxedo (SALT)

Release Notes

11g Release 1 (11.1.1.2)

April 2011

ORACLE®

Copyright © 2006, 2011, 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 11g Release 1 (11.1.1.2) Release Notes

About This Oracle SALT Release	2
Release 11g R1 (11.1.1.1.0)	3
Release 11g R1 (11.1.1.2.0)	4
Release 11g R1 (11.1.1.2.2)	4
Upgrade Considerations	5
Oracle SALT Installation Prerequisites	5
Oracle SALT Platform Support	5
Interoperability Considerations	5
Known Issues	5
GWWS Runtime	6
Interoperability	9
SCA Container	13
Web Application Server	15
Where to Get Product Documentation	15
Contacting Oracle Customer Support	16
See Also	16

Oracle SALT 11g Release 1 (11.1.1.2) Release Notes

Oracle SALT 11g Release 1 (11.1.1.2.2)
Date: December 2011

Table 1 Revision History

Revision Date	Summary of Change
December 2011	Updated “Oracle SALT 11gR1 Supported Platforms” in the Oracle SALT Installation Guide with: <ul style="list-style-type: none">• Oracle Solaris 11 (32-bit) on SPARC• Oracle Solaris 11 (64-bit) on SPARC
April 2011	Added Oracle SALT Web Application Server Support
February 2011	Updated “Oracle SALT 11gR1 Supported Platforms” in the Oracle SALT Installation Guide with: <ul style="list-style-type: none">• Oracle Linux 5.5 (64-bit) on ExaLogic X2-2• Oracle Solaris 11 Express x86 (64-bit) on Exalogic
November 2010	Added Oracle SALT SOCKS proxy support.

Table 1 Revision History

Revision Date	Summary of Change
June 2010	Updated “Oracle SALT 11gR1 Supported Platforms” in the Oracle SALT Installation Guide with: <ul style="list-style-type: none">• Novell Suse Linux ES 11 on 32-bit x86• Oracle Enterprise Linux 5.0 (32-bit) on x86• Oracle Enterprise Linux 5.0 (64-bit) on x86-64• Red Hat Linux Enterprise AS 5 (32-bit OS) on x86
March 2010	GA 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

- [Release 11g R1 \(11.1.1.1.0\)](#)
- [Release 11g R1 \(11.1.1.2.0\)](#)
- [Release 11g R1 \(11.1.1.2.2\)](#)

Release 11g R1 (11.1.1.1.0)

What's New and Improved

Oracle SALT Release 11g R1 (11.1.1.1.0) introduced the following features:

Python and Ruby Support

Python and Ruby support in SALT SCA provides a simple SCA client API for each language to perform SCA calls from Python or Ruby client programs, and an SCAHOST system server to allow access to Python or Ruby components. Python, Ruby or C++ components can now call or be called to/from other Python, Ruby or C++ components.

For more information, see [Oracle SALT SCA Programming](#) in the Oracle SALT Programming guide.

SCA Structure Support

Provides additional C++ structure functionality for improved performance.

For more information, see [Oracle SALT SCA Programming](#) in the Oracle SALT Programming guide.

Scatuxgen Metadata Generation Tool

The Scatuxgen Metadata Generation Tool parses C++ interfaces as used to develop SCA components for the SCA runtime functionality introduced in SALT 10gR3. It generates Oracle Tuxedo metadata repository interface data, and optionally a Web Services Definition File (WSDF) document.

For more information, see the [Oracle SALT Command Reference Guide](#).

WS-TX Support

Provides bi-directional transactional interoperability between Web Services and Oracle Tuxedo applications. Applications transparently make use of the GWWS system server transactional capabilities.

Outbound XA transactions are transparently propagated outside of an Oracle Tuxedo domain and mapped one-to-one to WS-TX transactions. Inbound WS-TX transactions are propagated into an Oracle Tuxedo domain and mapped to XA transactions.

For more information see, [WS-TX Support](#) in Oracle SALT Interoperability.

Release 11g R1 (11.1.1.2.0)

What's New and Improved

Oracle SALT 11g Release 11g R1 (11.1.1.2.0) introduced the following features:

SOCKS Proxy Support

SOCKS proxy support provides a configurable element associated to outbound endpoints and allows configuring a SOCKS server address that will proxy outbound connections according to the SOCKS V4 and V5 protocol.

Inbound support is not necessary as SOCKS proxies incoming connections without listening endpoints being aware of it. For more information, see the [Oracle SALT Deployment File Reference](#) in the Oracle SALT Reference Guide.

The `wsdlcvr` command is enhanced to connect to a SOCKS proxy server. For more information, see the [Oracle SALT Command Reference](#) in the Oracle Salt Reference Guide.

Release 11g R1 (11.1.1.2.2)

Oracle SALT 11g Release 1 (11.1.1.2.2) provides Apache and iPlanet plug-ins interfacing with:

- a CGI-like interface for writing dynamic HTML generating Tuxedo server
- a generic Oracle Tuxedo system server capable of running PHP scripts, Python WSGI applications (such as Django) and Ruby Rack applications (such as Rails). It also adds support for the PHP dynamic language in the SCA container.

What's New and Improved

Oracle SALT 11g Release 11g R1 (11.1.1.2.2) introduces the following features:

Web Server Plug-in/Web Gateway Interface Support

Web Server plug-ins in Apache 2 format and NSAPI format, for interfacing Apache HTTP server 2, Oracle HTTP Server or iPlanet Server and Oracle Tuxedo-based web applications. Support of a Web Gateway Interface in order to develop Oracle Tuxedo services that generate dynamic HTML pages.

Python, Ruby, and PHP Web Application Support

- PHP scripts supported directly,
- Python scripts supported via WSGI,

- Ruby scripts supported via Rack.
- PHP, Python and Ruby applications using most web frameworks such as Zend framework, Symfony, Django or Rails will run as-is in an Oracle Tuxedo environment.

PHP SCA Script Support

Support for PHP scripts as SCA client and SCA components, for SOA integration.

For more information, see the [Oracle SALT Administration Guide](#).

Upgrade Considerations

For information on installing Oracle SALT 11g Release 1 (11.1.1.2) on top of a previous SALT release, see the [Oracle SALT Installation Guide](#).

For information, see [Migrating from Oracle SALT 1.1 Application in the Oracle SALT Configuration Guide](#).

Oracle SALT Installation Prerequisites

Before installing Oracle SALT 11g Release 1 (11.1.1.2), you must ensure that Oracle Tuxedo 11g Release (11.1.1.2.0) is installed.

For more information, see [System Requirements](#) in the Oracle SALT Installation Guide.

Oracle SALT Platform Support

Oracle SALT 11g Release 1 (11.1.1.2) supported platforms are listed in [Appendix A: Oracle SALT 11g Release 1 \(11.1.1.2\) Supported Platforms](#) in the Oracle SALT Installation Guide.

Interoperability Considerations

Oracle SALT 11g Release 1 (11.1.1.2) 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 11g Release 1 (11.1.1.2). Entries include a description of the problem, and a workaround or solution where appropriate.

Each problem is listed by the Change Request (CR) or BugDB number.

- [GWWS Runtime](#)
- [Interoperability](#)
- [SCA Container](#)
- [Web Application Server](#)

GWWS Runtime

CR Number	Description and Workaround or Solution	Found In
CR334161	<p>Problem: GWWS rejects non UTF-8 inbound SOAP request messages when SignBody WS-Security Policy is enabled.</p> <p>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.</p> <p>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.</p>	2.0
	Platform: All	
	<p>Workaround:</p> <p>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.</p>	

CR Number	Description and Workaround or Solution	Found In
CR328329	<p>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.</p> <p>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.</p>	2.0
	Platform: All	
	<p>Workaround:</p> <p>Enlarge or remove the “size” restriction for XML typed buffer in the Tuxedo Service Metadata Definition.</p>	
CR306710	<p>Problem: Tuxedo service may not receive the exact same non UTF-8 encoding string as the string prepared in the SOAP request message.</p> <p>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.</p> <p>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.</p>	2.0
	Platform: All	
	<p>Workaround:</p> <p>None. Customers rarely use those characters. If some characters mapping are confirmed due to ICU bugs, please contact Oracle Tuxedo Customer Support.</p>	

CR Number	Description and Workaround or Solution	Found In
9281764	<p>Problem: WCF (.Net) C# clients may not handle SOAP 1.2 faults sent back by the GWWS gateway.</p> <p>When an Tuxedo service is exposed as a Web Service using the SOAP 1.2 style, GWWS may return faults such as service unavailable, problems during invocation, etc. When this happens, a C# client will receive an exception indicating that an invalid SOAP fault was received.</p> <p>This is due to the fact that GWWS does not specify an @xml:lang attribute in the /Fault/Reason/Text element returned.</p>	11gR1
	<p>Platform: All</p>	
	<p>Workaround:</p> <p>Code the C# client so that it receives the fault as an XML document which can then be parsed, as follows:</p> <pre>... catch (FaultException ex) { MessageFault mf = ex.CreateMessageFault(); XmlReader xr = mf.GetReaderAtDetailContents(); ... } --</pre>	

Interoperability

CR Number	Description and Workaround or Solution	Found In
CR330363	<p>Problem: SALT multiple encoding feature does not interoperable with Microsoft .NET WCF 3.0 engine.</p> <p>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 MBSTRING or XML buffer. Such SOAP response message cannot be accepted by those Web Service client applications developed using Microsoft .NET WCF 3.0 engine.</p>	2.0
	<p>Third-Party Web Service Toolkit: Microsoft .NET WCF 3.0</p>	
	<p>Workaround:</p> <p>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.</p> <p>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.</p>	
CR296594	<p>Problem: SOAP fault response message cannot be accepted by Microsoft .NET 3.0 when the HTTP Content-Length exceeds 65536.</p> <p>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.</p> <p>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.</p>	2.0
	<p>Third-Party Web Service Toolkit: Microsoft .NET WCF 3.0</p>	
	<p>Workaround:</p> <p>None. Avoid to return big buffer when invoking tpreturn () along with TPFail status code in the Tuxedo service.</p>	

CR Number	Description and Workaround or Solution	Found In
CR294785	<p>Problem: Apache Axis2/Java fails to handle Tuxedo FML32 TPFail response buffers that have field names with initial uppercase.</p> <p>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.</p> <p>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.</p>	2.0
	Third-Party Web Service Toolkit: Apache Axis2/Java	
	<p>Workaround:</p> <p>Modify the FML32 field name to avoid use initial uppercase name. Corresponding Tuxedo application also needs to be changed and re-compiled.</p>	
CR306978	<p>Problem: Apache Axis2/Java does not recognize the SOAP with Attachment (SwA) featured WSDL file generated by Oracle SALT.</p> <p>If SwA featured WSDL file is generated by Oracle SALT, Apache Axis2 <code>wsdl2java</code> utility generates Java stub code which is different from Apache Axis. Axis2 generated stub code cannot initiate a successful call to Oracle SALT service.</p>	2.0
	Third-Party Web Service Toolkit: Apache Axis2/Java	
	<p>Workaround:</p> <p>Use Apache Axis instead for SwA featured soap calls.</p> <p>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.</p>	

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	
	<p>Workaround:</p> <p>This is an Apache Axis bug, please refer to https://issues.apache.org/jira/browse/AXIS-2614.</p> <p>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.</p> <p>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.</p>	

CR Number	Description and Workaround or Solution	Found In
CR323477	<p>Problem: GWWS fails to call external Web Service applications built upon Microsoft .NET WCF 3.0 if asynchronous WS-Addressing feature is enabled.</p> <p>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:</p> <pre data-bbox="323 618 1042 812"><wsa:ReplyTo> <wsa:Address> http://myhost:7102/?wsa_Msg_ID=uuid:B437A4F4-AF2 3-111E-FFFFFFAC1622FFFFFFF9F0000-6BBE </wsa:Address> </wsa:ReplyTo></pre> <p>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.</p> <p>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.</p> <p>GWWS then always encounters time out in receiving asynchronous response because Microsoft .NET WCF 3.0 never send the response to GWWS expected endpoint.</p> <p>Third-Party Web Service Toolkit: Microsoft .NET WCF 3.0</p> <p>Workaround:</p> <p>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 <i>Oracle SALT Administration Guide</i>.</p>	2.0

SCA Container

CR Number	Description and Workaround or Solution	Found In
1. CR379052	<p>Problem: XmlHelper::save() loses content of XML document of mixed type</p> <p>In a schema where a type is defined as follows: <code><xsd:complexType name="myType" mixed="true"> ...</code> A document cannot be loaded by SDO XmlHelper, then saved as its original content. For example:</p> <pre>... 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> ...</pre>	10g R3
	Platform: All	
	<p>Workaround:</p> <p>These types of documents should be handled directly by the application code.</p>	

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), <code>ServiceInvocationException</code> has no place or mechanism to store this data. This may happen: <ul style="list-style-type: none"> • 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. 	10g R3
	Platform: All	
	Workaround: Use ATMI binding and /Domain feature to connect two Tuxedo domains.	
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.	10g R3
	Platform: All	
	Workaround: The services accessed may be configured as AUTOTRAN.	
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.	10g R3
	Platform: All	
	Workaround: To avoid this problem, you should not configure duplicated XML elements for ATMI binding without specifying target in the composite file.	

Web Application Server

BugDB Number	Description and Workaround or Solution	Found In
12332461	Problem: Shutdown of Oracle Tuxedo applications using <code>tmipcrm</code> cause Apache processes configured with <code>mod_tuxedo</code> to exit. When using <code>tmipcrm</code> to terminate an Oracle Tuxedo application, Apache or Oracle Http Server processes exit, possibly interrupting HTTP traffic.	11.1.1.2. 2
	Platform: All	
	Workaround: Always use <code>tmsshutdown</code> to stop the Oracle Tuxedo application. <code>tmipcrm</code> (or removal of Oracle Tuxedo IPC resources using an OS command such as <code>ipcrm</code>) should be used only as a last resort option. Oracle Tuxedo applications improperly shut down if using this method. Apache or Oracle Http Server processes exit because when used with <code>mod_tuxedo</code> become Oracle Tuxedo client programs.	
12395121	Problem: Ruby Rails application returns error with iPlanet plug-in. When configuring an Oracle Tuxedo Ruby Rails application and using iPlanet to connect to this application, Oracle Tuxedo reports an exception and stops processing the request as follows: <pre>082702.hostname!WEBHNDLR.22241.3081037488.0: ERROR: Ruby Exception calling application method: NoMethodError: 082702.hostname!WEBHNDLR.22241.3081037488.0: undefined method `chomp' for nil:NilClass</pre>	11.1.1.2. 2
	Platform: All	
	Workaround: This is a current limitation of the iPlanet Oracle Tuxedo plug-in. Please use the Apache or Oracle HTTP Server <code>mod_tuxedo</code> module.	

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 at:

<http://www.oracle.com/us/support/contact-068555.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 11g Release 1 Product Overview](#)
- [Oracle SALT 11g Release 1 Installation Guide](#)
- [Oracle SALT 11g Release 1 Administration Guide](#)
- [Oracle SALT 11g Release 1 Programming Guide](#)
- [Oracle SALT 11g Release 1 Reference Guide](#)