BEA AquaLogic Data Services Platform™

Release Notes

Note: Product documentation is subject to post-release revision. If so, it will be made available from the BEA edocs site, listed below. Any major changes are noted in the online version of these notes.

http://edocs.bea.com/aldsp/docs25/index.html

Version: 2.5
Document Date: June 2005
Revised: November 2006

Copyright

Restricted Rights Legend

Trademarks and Service Marks

Release Notes and Supported Configurations

BEA AquaLogic Data Services Platform, Version 2.5

Most recent revision: June 7, 2007

About BEA AquaLogic Data Services Platform

AquaLogic Data Services Platform™ provides read and write access to information in relational databases, Web services, Java functions, XML files, delimited files, and other types of disparate data.

Once developed, application developers — using access technologies such as the AquaLogic Data Services Platform mediator API, JDBC, SQL, or AquaLogic Data Services Platform controls — can invoke AquaLogic Data Services Platform functions as a means of providing their applications with access to integrated, updateable data from their enterprise.

Metadata, security, and cache management facilities are provided through the Data Services Platform Console (ldconsole).

Note: BEA AquaLogic Data Services Platform was originally named Liquid Data for WebLogic®. Some artifacts of the original name such as 'ld' remain in the product, path, and other artifacts.

Revision Policy

Release Notes are subject to revision between releases. The most recent version can always be found at:

```
http://edocs.bea.com/aldsp/docs25/relnotes/index.html
```

Items modified or added post-release are so marked and dated.

This document includes the following topics:

What's New in AquaLogic Data Services Platform 2.5

AquaLogic Data Services Platform version 2.5 adds a number of new features to AquaLogic Data Services Platform. Table 1 list many of these features.

Table 1 New and Improved Functionality in AquaLogic Data Services Platform Version 2.5

Feature	Details
Integration with AquaLogic Service Bus	Integration is provided to allow AquaLogic Service Bus users to leverage AquaLogic Data Services Platform for read/write database and other data source connectivity. Capabilities include: • Allowing custom AquaLogic Service Bus Transport to invoke data services directly. • Allowing data services to be invoked using RMI (thus avoiding Web services communication issues). • Preserving the security context during communication between services.
9.2 Client Support	Add support for WebLogic 9.2 clients in two areas: • Workshop controls • Mediator API
Excel Add-in	Provides on-demand query services and Excel reporting capabilities for moderately-sized data sets are involved. Specific features include: • Ability to invoke AquaLogic Data Services Platform-generated Web services within Microsoft Excel. • Data service data post-processing. • Secured Web service invocation. • Ability to schedule report refreshes.

Table 1 New and Improved Functionality in AquaLogic Data Services Platform Version 2.5

Feature	Details
JDBC Driver Enhancements	The AquaLogic Data Services Platform JDBC driver and associated tooling has been enhanced. Specifically: • Driver is more fully based on the 3.0 JDBC specification. • SQL-92 syntax is more fully supported. • Data streaming through the JDBC driver is more fully supported for large result sets. • The latest version of major reporting tools such as Crystal Reports, BusinessObjects, and Hyperion have been successfully tested.
Extending Tooling to Support SQL	A number of tooling enhancements have been made to provide enhanced support for SQL in two key areas: • Accessing SQL through data services. • Allowing SQL reporting tools to better access information under data service management. Specific new functionality includes: • Ability to publish data services as JDBC artifacts, with refactoring support. • Ability to create substitutions for generated SQL statements. This makes it possible for administrators to post-edit generated SQL for the limited purpose of providing database hints and so forth. • Standalone query plan viewer supporting both XQuery and SQL
JDBC Metadata Access Control	Allows use of the AquaLogic Data Services Platform console to secure metadata for: • SQL objects, which can be made invisible to the JDBC driver or • Data hiding, where the fields are visible, but the data is hidden to the unauthorized user.

Supported Configurations

As a WebLogic Workshop application, AquaLogic Data Services Platform generally supports the same platforms, operating systems, processor architecture, SDKs, and RDBMS systems as does supported servers, as described under "WebLogic Server and Client Support."

Note: For details on post-release testing of other operating systems, SQL databases, JDBC drivers, and Web services see the online version of these Release Notes:

`http://edocs.bea.com/aldsp/docs25/relnotes/relnotes.html`

Note: The minimum RAM recommendations are for supporting a single instance of WebLogic Server on which AquaLogic Data Services Platform is running. You may need more memory if you run two or more instances of WebLogic Server and AquaLogic Data Services Platform. Some systems use disk space as virtual RAM. Performance when running in virtual RAM may be markedly slower than when running in physical RAM.

The following abbreviations are used in this section:

Table 2 Testing Classifications

Classification	Meaning
TESTED	Tested.
SUPPORTED	Not specifically tested, but should work.
NS	Not supported.

WebLogic Server and Client Support

A supported version of WebLogic Platform 8.1 is needed for development and runtime (client). The following table (Table 3) details WebLogic server and client support.

Table 3 AquaLogic Data Services Platform 2.5 WebLogic Platform and Client Support

Component	Platform	JDK	Testing Status
Platform Server	WebLogic 8.1 SP4	Java HotSpot VM 1.4	SUPPORTED
		JRockit 1.4	SUPPORTED
	WebLogic 8.1 SP5	Java HotSpot VM 1.4	TESTED
		JRockit 1.4	TESTED
	Weblogic 8.1 SP6	Java HotSpot VM 1.4	TESTED (10/2006)
		JRockit 1.4	TESTED (10/2006)
	WebLogic 9.*x*	Java HotSpot VM 1.5	NS
		Jrockit R26.0.0	NS

Table 3 AquaLogic Data Services Platform 2.5 WebLogic Platform and Client Support

Component	Platform	JDK	Testing Status
Client	WebLogic 8.1 SP4	Sun JDK 1.4	SUPPORTED
		JRockit 1.4	SUPPORTED
	WebLogic 8.1 SP5	Sun JDK 1.4	TESTED
		JRockit 1.4	TESTED
	Weblogic 8.1 SP6	Java HotSpot VM 1.4	TESTED (10/2006)
		JRockit 1.4	TESTED (10/2006)
	WebLogic 9.*x*	Sun HotSpot VM 1.5	TESTED
		JRockit R26.0.0	TESTED

Supported Development and Client Configurations

For data service development a complete installation of a supported version of WebLogic Platform 8.1 is recommended. The following chart lists tasks associated with data service development and deployment and the minimal platform component installations.

(For the purposes of this table ALDSP stands for AquaLogic Data Services Platform; WLS stands for WebLogic server.)

Table 4 Supported Server/Platform Configurations for Development and Runtime

Activity	Minimal Configuration
ALDSP development under WLS 8.1 SP5 or SP6	ALDSP-enabled WLS 8.1
ALDSP development under WLS 8.1 SP4	ALDSP-enable WLS 8.1 + WebLogic Workshop
ALDSP client access	ALDSP-enabled WLS 8.1
ALDSP development and specific WebLogic Platform 8.1 components	ALDSP-enabled WLS + the same specific WebLogic Platform 8.1 components
RTLApp (sample ALDSP application)	WebLogic Platform 8.1

Supported Operating Systems and Hardware

The following table (Table 5) lists operating systems and/or hardware that are supported for this release. Testing Status indicates when testing has occured. Test criteria is defined in Table 2.

Platforms and versions in boldface (also with gray backgrounds on PDF pages) support both design time (data service development through WebLogic Workshop IDE) and runtime. Others support runtime only.

Platforms not specifically listed are not supported.

Table 5 Platform and Platform Version Support for AquaLogic Data Services Platform

Operating System	Platform	Version	Hardware	Testing Status
Microsoft Windows	**Windows Server 2000**	**2000, 2003**	**x86**	TESTED
	Windows XP	**XP**	**x86**	TESTED
UNIX	HP-UX	HP-UX 11i	PA-RISC	TESTED
	Sun Solaris	8, 9, 10	SPARC64	TESTED
		10	x86	TESTED (10/2006)
	IBM AIX	5.2, 5.3	pSeries	TESTED (10/2006)
Linux	**Red Hat Enterprise Linux**	**3.0, 4.0**	**x86**	TESTED
		4.0	64-bit Xeon/AMD64 (32-bit JVM)	TESTED
	Novell SuSE Linux Enterprise Server	9	x86, 64-bit Xeon/AMD64 (32-bit JVM)	TESTED (10/2006)

Supported Relational Database Management Systems

In general any thread-safe, transaction-callable relational database management system (RDBMS) accessible through JDBC should be available to AquaLogic Data Services Platform.

Several vendor-specific levels of support are provide, as shown in Table 6.

Table 6 Vender-specific Levels of RDBMS Support

Support Level	Description
Extended	The specific database version has been tested and found to be compatible with AquaLogic Data Services Platform. Extended support means that support is provided for stored procedures and some database functions. See Table 7 for full details on vendor and version support.
Tested	The specific database version has been tested and found to be compatible with AquaLogic Data Services Platform. Some operations may be pushed down to the database level. Tested support is provided for such vendors as Informix, MySQL, and Microsoft Access.
Generic	Although the particular database versions has not been tested, basic SQL read and write calls should work if the database supports JDBC standards. Generic support is sometimes referred to as *base support*.

Databases which have been tested with AquaLogic Data Services Platform 2.5 are listed in Table 7. Subsequent versions of particular DBMS systems should work to the same level of support, but have not been specifically verified.

Table 7 RDBMS Support in AquaLogic Data Services Platform 2.5

Database	Version(s)	RDBMS Support Level	Tested?	Supported?	Caching Support for Vendor
Oracle	10g	EXTENDED	Yes	Yes	Yes
	9i	EXTENDED	Yes	Yes	
	8.1.7	EXTENDED	Yes	Yes	
	RAC	EXTENDED	No	Yes	
IBM DB2	8.1	EXTENDED	Yes	Yes	Yes
Sybase	12.5 or prior	GENERIC	No	Yes	Yes
	12.5.2 and above	EXTENDED	Yes	Yes	
Microsoft SQL Server	SQL Server 7	EXTENDED	No	Yes	Yes
	2000	EXTENDED	Yes	Yes	
	SQL Server 2005	EXTENDED	No	Yes	

Table 7 RDBMS Support in AquaLogic Data Services Platform 2.5

Database	Version(s)	RDBMS Support Level	Tested?	Supported?	Caching Support for Vendor
Informix	IDS with Universal Data Option (9.2 and above)	GENERIC			No
	IDS 9.3	GENERIC	No	Yes	
	IDS 9.4	GENERIC	Yes	Yes	
			No	Yes	
Teradata	V2r4.1.3, v2r5.1.x v2r6.0	GENERIC GENERIC GENERIC	No	Yes	No
MS Access	2000	GENERIC	Yes	Yes	No
MySQL	4.1.14	GENERIC	Yes	Yes	No
	5	GENERIC	Yes	Yes	
Other DBMS systems	Any	GENERIC	No	No	No
PointBase	4.4[1]	EXTENDED	Yes	Yes[1]	No

1. PointBase Server is an all-Java DBMS product included with WebLogic Platform 8.1 distribution solely for evaluation purposes, either in the form of custom trial applications or through packaged sample applications provided with WebLogic Platform. Non-evaluation development or other use of the PointBase Server requires that a separate PointBase Server license be obtained directly from DataMirror. AquaLogic Data Services Platform does not support PointBase for production server deployments.

Supported Database-JDBC Driver Matrices

The following table (Table 8) describes results of testing AquaLogic Data Services Platform against various databases under different JDBC drivers. Both the BEA and native drivers were tested in XA and non-XA modes. Limitation encountered are noted as CR numbers in the Comments column.

Table 8 Matrices of Databases and JDBC Drivers Supported by AquaLogic Data Services Platform

Database	JDBC Driver Type	Data Source Type	Comments
Oracle	BEA non-XA	tables	CR199675
		stored procedures	CR202963 CR212515
	BEA XA	tables	CR199675
		stored procedures	CR202963 CR212515
	Oracle non-XA	tables	CR265965
		stored procedures	CR212515 CR202962
	Oracle XA	tables	
		stored procedures	
IBM DB2	BEA non-XA	tables	
		stored procedures	CR227440
	BEA XA	tables	
		stored procedures	CR227440
	DB2 non-XA	tables	
		stored procedures	CR227440
	DB2 XA	tables	CR227486
		stored procedures	CR227440
Sybase	BEA non-XA	tables	

Table 8 Matrices of Databases and JDBC Drivers Supported by AquaLogic Data Services Platform

Database	JDBC Driver Type	Data Source Type	Comments
		stored procedures	
	BEA XA	tables	
		stored procedures	
	Sybase non-XA	tables	CR223429
		stored procedures	
	Sybase XA	tables	CR223429
		stored procedures	
Informix	BEA non-XA	tables	CR211701
		stored procedures	not supported
	BEA XA	tables	
		stored procedures	not supported
	Native non-XA	tables	CR223486 CR226171
		stored procedures	not supported
	Native XA	tables	
		stored procedures	not supported
MS-SQL Server	BEA XA	tables	
		stored procedures	
	MS-SQL Server non-XA	tables	CR214730
		stored procedures	CR202041
	MS-SQL Server XA	tables	
		stored procedures	

Details related to these CR numbers can be found in "Product Limitations and Workarounds."

Configuring the AquaLogic Data Services Platform JDBC Driver for Reporting Tools

The following table (Table 9) describes the matrices for configuring the AquaLogic Data Services Platform JDBC driver for supported reporting applications.

Note: In cases where the reporting tool supports JDBC connectivity, use the AquaLogic Data Services Platform JDBC driver.

Table 9 Matrices of Required Connectivity Software for Reporting Applications Supported by the AquaLogic Data Services Platform JDBC Driver

Application and Version	JDBC Native	OpenLink ODBC/JDBC Lite Bridge
Crystal Reports XI	TESTED	
Business Objects XI, Release 2		TESTED
Hyperion BI 9+, Interactive Reporting		TESTED
MS Access 2000		TESTED
MS Excel		TESTED

Supported Web Services Standards

The following table (Table 10) lists Web services standards supported by AquaLogic Data Services Platform 2.5.

Table 10 Supported Web Services Standards

Web Services Standard	Version
SOAP	1.1
WSDL	1.1
JAX-RPC	1.0

Product Limitations and Workarounds

This section contains a table (Table 11) which lists known limitations associated with the current BEA AquaLogic Data Services Platform 2.5 release that users might encounter. Information regarding these limitations includes a CR (change request) number for each problem, applicable platform, a detailed description of the problem and workarounds, where applicable.

Limitations added to this list post general release availability are identified by date as well as CR number.

Please contact BEA customer support at:

`http://support.bea.com`

for assistance in tracking unresolved issues.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR265965 (revised Feb. 2, 2007)	**Updating or deleting Oracle's CHAR/NCHAR with trailing blanks failed using Oracle JDBC driver.**
Description	With Oracle's non-XA JDBC driver, CHAR and NCHAR columns can only be updated if the number of characters is 1024 or less. The following message typically appears if this problem occurs: `java.sql.SQLException: ORA-01461: can bind a LONG value only for insert into a LONG column`
Platform	All, using Oracle's non-XA JDBC driver.
Workaround	Whenever possible use BEA's Oracle JDBC driver (Type 4).
CR294861 (Oct. 22, 2006)	**An exception occurs if a timestamp with time zone is fetched before LONG_RAW.**
Platform	All using Oracle.
Description	When using an Oracle database, if a timestamp with a time zone is fetched before a LONG_RAW a "... Stream has already been closed" exception occurs. This is consistent with the way Oracle handles LONG_RAW fetched before blob and clob.
Workaround	Reverse the fetch order of timestamp with time zone and LONG_RAW when using Oracle.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR293244 (Sept. 30, 2006)	**When executing a SQL query containing a single parameter through the AquaLogic Data Services Platform JDBC driver, a "wrong parameter type" occurs.**
Platform	All.
Description	When a SQL query is executed, a parameter list containing one and only one parameter is treated as a table parameter instead of a normal parameter, resulting in an exception being returned.
Workaround	Create a SQL table with a single row contain the parameter value and pass that table parameter to the query. For detailed informations on table parameter s and creating a TableParameter class object see the topic "Table Parameter Support" in the Using SQL to Access Data Services chapter of *Application Developer's Guide*.
CR292297	**Importing two or more procedures that have the same return type into an existing data service may not succeed.**
Platform	All.
Description	An exception can occur when importing multiple data service procedures, if two or more of the procedures have the same return type.
Workaround	Import procedures having the same return type using separate import operations.
CR292585	**When publishing a WSDL generated from AquaLogic Data Services Platform to the AquaLogic Service Registry, multiple entries may appear for the same WSDL.** **Similarly, when configuring a Web service from the AquaLogic Service Registry through the Excel Add-in, multiple entries for the same WSDL may be seen.**
Platform	All.
Description	When a JWS is generated, by default the form-get and form-post flags are set to True. This setting in conjunction with the absence of a source directive (CR268183) cause multiple entries to appear.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Workaround	Using the Property Editor in WebLogic Workshop, set the form-get and form-post flags to False. Then, if the goal is to generate a WSDL for Excel Add-in use, the following entry should first be placed at the beginning of the comments in the JWS source file: `* @editor-info:link autogen-style="stateless" source="<jcxfilename>.jcx" autogen="true"`
CR286359	**When importing or synchronizing metadata with an Oracle database, the native width of float elements may be incorrectly calculated.**
Platform	All, using the BEA Oracle JDBC driver.
Description	When importing or synchronizing metadata with an Oracle data source, the width of float elements may be rendered incorrectly (15 instead of 6 or 9). This problem has been observed when using the WebLogic JDBC Oracle driver (version 3.0.5.0).
Workaround	Metadata synchronization using Oracle's native JDBC driver (Oracle.jdbc.driver.OracleDriver) did not exhibit this problem. Therefore, if the width of native float elements is an issue use the Oracle JDBC driver.
CR292429	**When working with DSP Transport in the AquaLogic Service Bus Console while using a multibyte character set (such as Japanese), unexpected characters may appear in editing fields.**
Platform	All.
Description	When using the AquaLogic Service Bus Console in conjunction with a DSP Transport Configuration, if the Request Encoding field or the Response Encoding field is set to utf-8 (the default value) unexpected characters may appear in editing fields when using multibyte character sets.
Workaround	When editing any field in a DSP Transport Business Service configuration, if the request encoding or response encoding should be any value other than the default (utf-8), the setting must be reset to their correct values before saving the configuration.
CR288104	**AquaLogic Data Services Platform functions accessed through a 9.2-level Data Service control via a Java Web service may not successfully execute.**
Platform	All.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Description	This problem can occur due to differences in the way Web service templates are generated under WebLogic 8.1 and Weblogic 9.2 The template that is generated by a WebLogic Service 9.2 test browser expects a qualified schema and therefore adds namespace prefixes to all elements in the schema. For details see: "CR288104 Details." By contrast, the schema associated with the WebLogic 8.1-level data service does not have elementFormDefault set, so it defaults to unqualified. However, the WebLogic 9.2 Web service sends a qualified xml fragment, causing an error when the schema is passed to the AquaLogic Data Services Platform server running under WebLogic 8.1.
Workaround	Users working with Weblogic 9.2-level Web services should replace the template generated by the WebLogic 8.1 test browser with the template generated by the WebLogic 9.2 test browser. Alternatively, generated code can be edited to include the XMLBean values in the method before passing it to the AquaLogic Data Services Platform control method.
CR292257	**Duplicate names are allow when mapping stored procedures to data services during metadata import.**
Platform	All.
Description	The SQL name mapping user interface may allow more than one stored procedure with the same name to be configured under the same schema. However, if more than one stored procedure with the same name is configured, the user will see unexpected results in the JDBC metadata API and on execution of the procedures via JDBC.
Workaround	In the Publish Data Service Functions for SQL Use wizard, manually rename stored procedures so that there are no duplicate names.
CR269337	**Undetected classname collisions in a Workshop application can cause compilation errors.**
Platform	All.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Description	This problem can occur when compiling schema files into SDOs using AquaLogic Data Services Platform. If an element or attribute has the same name (such as 'sequence') as an SDO reserved method (such as 'getSequence( )'), SDO will — through the XMLBean Java code generator — attempt to "uniqueify" the method by generating 'getSequence2', as an example. Here is a list of SDO keywords which could contribute to such name collision conditions: • `getBoolean, getByte, getChar, getDouble, getFloat, getInt, getLong, getShort, getBytes, getBigDecimal, getBigInteger, getDataObject, getDate, getString, getList, getSequence` • `setBoolean, setByte, setChar, setDouble, setFloat, setInt, setLong, setShort, setBytes, setBigDecimal, setBigInteger, setDataObject, setDate, setString, setList, setSequence`
Workaround	To use the uniqueified name: • Add a copy of `sdo.xsdconfig` to your schema project and • In your Java file correct the entry to the uniqueified name as in: `addrLine.setSequence2(1);`
CR291781	**Excel Add-in will initially attempt to reuse HTTP basic authentication login credentials for all Web services hosted on the same port.**
Platform	All.
Description	In cases where multiple Web services are configured in a single Excel worksheet, and these Web services are hosted by the same host machine and port number, the Excel Add-In will initially attempt to reuse the previously accepted username/password for these services. If the username/password is not valid for a particular Web service, a login dialog will be displayed. Subsequent Web service invocations during the same Excel session will use the correct login information for each service.
Workaround	No action has to be taken, unless it is not deemed acceptable that an attempt is made to authenticate a web service call initially with incorrect credentials. In such cases Web services requiring different credentials should be grouped in different servers OR Web services requiring different credentials hosted on the same host/port should be used on separate Excel worksheets.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR291062	**When using AquaLogic Data Services Platform with AquaLogic Service Bus, 'dsp' is not supported for inbound transport.**
Platform	All.
Description	A null pointer exception occurs if you choose dsp type when creating a proxy service in AquaLogic Service Bus. This is because the data service option erroneously appears in the list of available transport types for this kind of operation.
Workaround	Avoid selecting the dsp type from available options when creating a proxy service in AquaLogic Service Bus.
CR290239	**Several underlying WSDL Element definition attributes and Attribute definition attributes are not currently supported in the AquaLogic Data Services Platform Excel Add-in.**
Platform	All.
Description	The following Element definition attributes are currently unsupported: • `substitutionGroup` • `default` • `fixed` • `form` • `abstract` • `block` • `final` The following Attribute definition attributes are currently unsupported: • `default` • `fixed` • `form` • `abstract` • `block` • `final`
Workaround	Any WSDL containing the definition attributes listed above may not function as expected. If problems are encountered, remove these definition attributes from your WSDL.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR290041	**For data service files that are large and slow to compile, you may need to adjust the compiler.delay flag to improve response time during source editing.** **Consecutive key strokes during source editing are batched into a single compile event. If file compilation is slow, then you may want to adjust the time window for batching key strokes for compilation.**
Platform	All.
Description	When editing large files in Source View, editing during compilation may cause multiple compilations to occur.
Workaround	Adjust the compiler.delay environment variable (in milliseconds) to increase the delay between compilations when doing editing. This property should be specified in the `workshop.cfg` file as a '-D' parameter.
CR285121	**When creating a Data Service control under WebLogic 9.2, an error can occasionally occur if the editor of the Server View is open at the same time.**
Platform	All.
Description	When generating or editing a Data Service control on Weblogic 9.2, it is recommended that the editor of the Server View not be open.
Workaround	When generating or editing a Data Service control, if you need to test the JWS, use the: `Run as → Run on Server` option on the right click menu. This option deploys/undeploy the user component. If, in rare cases, the control fails to undeploy, then access the Web-based Weblogic administration console and perform the required operation from there. If the aforementioned editor is already open and you want to edit/generate a Data Service control, first close the editor and then restart WebLogic Workshop.
CR284995	**Exception when publishing a WSDL from a data service when an underlying schema does not have a targetNamespace defined.**

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Platform	All.
Description	An exception indicating that you cannot map a prefix when no namespace is present may occur when attempting to generate a WSDL file from a valid data service. This is due to the fact that one or more schemas that are depended upon by the data service do not have a targetNamespace defined.
Workaround	Make sure that all schemas underlying your data service have a proper targetNnamespace defined.
CR284834	**When accessing data from Informix, use the Informix native JDBC driver when possible.**
Platform	All using Informix database systems.
Description	The WebLogic Informix JDBC driver is less performant than the native Informix JDBC driver when accessing data through AquaLogic Data Services Platform.
Workaround	No workaround necessary, but for best performance with Informix data use its native JDBC driver.
CR284534 (updated: 7 June 2007)	**Unclear exception message when using a Data Service control with WebLogic 9.2.**
Platform	All.
Description	When using a Data Service control with Weblogic 9.2, if the `wls90interop.jar` that is required to be on the AquaLogic Data Services Platform Top level is not used correctly, a runtime exception may occur. The exception will appear similar to: `weblogic.rjvm.PeerGoneException: ; nested exception is: java.io.EOFException`

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Workaround	If you see such an exception, check the ALDSP 2.*x* server log. If an error such as the following appears: `java.io.InvalidClassException: javax.xml.namespace.QName; local class incompatible: stream classdesc serialVersionUID = 4418622981026545151, local class serial VersionUID = -9120448754896609940` then the cause of this error is likely due to one of the following: • the `wls90interop.jar` file is not in the ALDSP 2.*x* server classpath • the wls90interop.jar file is not listed first in the classpath.
CR279492	**When using a data service based on a Web service at runtime, a validation error may occur if form and/or elementFormDefault do not match. This happens because redefinition of the 'Form' attribute is not supported.**
Platform	All.
Description	At design time, if: • The elementFormDefault attribute in the primary schema does not match the elementFormDefault attribute in an imported or included schema, or • if the form attribute of an element does not match the elementFormDefault attribute in the primary schema. then, at runtime, validation errors will occur when accessing Web service-based data services.
Workaround	The elementFormDefault in the primary schema and any imported or included schemas should match prior to compilation. Also, the form attribute of the element should match the elementFormDefault in the primary schema.
CR288384	**The Data Lineage feature in AquaLogic Data Services Platform administration console requires the X11 graphical environment in Linux and UNIX environments.**
Platform	UNIX and Linux.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Description	The following error: "`java.lang.NoClassDefFoundError: sun/awt/X11GraphicsEnvironment`" may occur while accessing the Data Lineage feature of AquaLogic Data Services Platform Administration Console. The circumstances arise when the Administration Server hosting AquaLogic Data Services Platform Administration Console is running on a Linux or UNIX host with one of the following conditions (not an exhaustive list): 1. A headless environment is in use — for example, without monitor and/or X server. 2. With a monitor, but the user running the administration server is not the same user logged-in from the monitor, and therefore, does not have display permissions to the default display (:0.0).
Workaround	To resolve this issue, set the headless property to true: `-Djava.awt.headless=true` in the Weblogic Server startup script (`startWeblogic.sh`) in the section of the script where the server is started.
CR283262	**Unable to add criteria to SQL when using MS Excel with EasySoft or OpenLink.**
Platform	All.
Description	In Microsoft Query, adding a query criteria using the Add Criteria window, (Criteria →Add Criteria) throws an error message when accessing AquaLogic Data Services Platform data sources.
Workaround	To workaround this issue: 1. In the Microsoft Query window, select the Records menu option. If the Automatic Query option is checked, then clear this option. 2. Click View → Criteria. This will add a Criteria window to the query window into which you can enter appropriate criteria. 3. To execute query with the criteria added, click Records → Query Now.
CR270188	**Cannot use the WLS Administration Console to define a security policy for accessing the AquaLogic Data Services Platform Console.**

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details

Description	If a user defines a security policy in WebLogic admin console for AquaLogic Data Services Platform console (`ldconsole.ear`), only the user in the Administrator role can access/login to the AquaLogic Data Services Platform console. Other read-only roles (Operator, Monitor, Deployer) fail to access to AquaLogic Data Services Platform console, instead returning the error:
	`Authentication Denied`
	`The username or password has been refused by WebLogic Server.`

Platform	All.
Workaround	By default ldconsole is allowing access ONLY to administrator group. Only when security policy is defined, are other users allowed per defined security policies. The operation can be completed using the following steps: 1. Log in to the WLS console with admin credentials. 2. Created a new user (such as JACK_BLACK). 3. Assigned the user to the Monitors group. 4. Log out of the WLS console. 5. Log in to ldconsole in the Admin role. 6. Edit security settings in the AquaLogic Data Services Platform metadata browser to add a security policy allowing read access to users of group "Monitors". 7. Log in to the ldconsole page using the JACK_BLACK user credentials.
CR233032	**Under some conditions a data service project deployment fails with an error similar to:** **Deployment failed, Last Message from the server ...**
Description	Under some conditions the `weblogic-application.xml` can become corrupted when deploying a AquaLogic Data Services Platform-enabled application, leading to a AquaLogic Data Services Platform deployment failure.
Platform	All.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details

Workaround	A software patch is available at:
	`<weblogic_home>/liquiddata/platform_patches/81SP4/CR233032`
	Alternatively, performing a Clean operation before a project Build automatically repairs the file:
	`weblogic-application.xml`

CR267820	**Metadata synchronization update preview displays "Problem in parsing XML fragment" error.**
Description	If your project contains Java function signatures referring to XMLBeans, then erroneous results may be reported when doing a metadata update.
Platform	All.
Workaround	Delete the `<project.jar>` file in your application's library folder that was created when you last built your project. Then synchronize and rebuild.
CR264597	**String comparison operations involving MS-SQL may return incorrect results when the comparison operation is computed by MS-SQL.**
Description	See "CR264597 Details."
Platform	All.
Workaround	See "CR264597 Details."
CR253530	**When importing AquaLogic Data Services Platform projects into Workshop if the option Copy into Application Directory is not selected, importing a project not already located in the application results in 'resource not found' errors.**
Description	AquaLogic Data Services Platform requires that projects be contained in the application folder. This is handled automatically if the Copy into Application Directory option in Workshop is selected.
Platform	All.
Workaround	Select the Copy into Application Directory option when importing AquaLogic Data Services Platform Console projects.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR204243	**When casting xs:decimal from an xs:integer or xs:long, resulting values may not be precisely correct.**
Description	As above.
Platform	All.

Workaround	To avoid the possibility of an incorrect result use a string literal instead of an xs:integer literal. For example instead of:
	`xs:decimal( 9223372036854775807 )`
	use:
	`xs:decimal( "9223372036854775807" )`

CR260587	**An exception during an SDO update operation can occur if the order of elements in the client diffgram is changed and the Validate option is active.**
Description	Sometimes the order of elements in a diffgram changes, potentially leading to datagraph validation failure.
Platform	All, using ADO.NET clients.
Workaround	If possible, turn off validation for the operation.
CR268183	**Workshop removes a necessary line when editing an AquaLogic Data Services Platform Control-generated JWS file.**
Description	After creating a AquaLogic Data Services Platform control it may be necessary to edit the generated JWS file. For example, in the case of enabling Web service security, the user may need to modify the JWS property that specifies a Web service security file. If the JWS is edited in Workshop, a line in the header beginning: `* @editor-info:link autogen-style ...` is automatically removed. This line is needed in order to generate ADO.NET-enabled WSDL files. **Note:** This problem also affects the Excel Add-in.
Platform	All.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Workaround	Before editing a generated JWS file in Workshop open the file in a text editor and copy the line beginning: `* @editor-info:link autogen-style ...` Then, when editing the file in Workshop, restore the above line to the JWS file.
CR245418	**SDO does not support periods [.] in element names and schema paths.**
Description	SDO is dependent on XPath which supports indexing from 0. For example, SDO's Customer.0 equals XPath's Customer[1]. Since periods are already used for this type of notation, they cannot appear in element names. A more general statement of the problem is that SDO does not support periods in schema paths, including leaf elements.
Platform	All.
Workaround	Avoid using periods in element names and schema paths.
CR265706	**When no values are passed during an insert operation, the generated query contains a null, which can lead to an exception on the database side.**
Description	Default values for SDO properties are always null. When no values are passed during an insert operation, the generated SQL insert contains a null. This can lead to an exception of the following form on the database side: `java.sql.SQLException: ORA-01400: cannot insert NULL into ("COMMON"."CASE"."CREATE_USERNM")` The reason this exception occurs is that when metadata is created for database tables, the default value of the database setting is not captured in SDO. So, for example, if the value of the column is "not null" in the database and the SDO datagraph passed a null value to the database, the exception will occur.
Platform	All.
Workaround	Default values should be set in the SDO client.
CR265950	**End-point name changes do not take effect for operations in document style Web services.**

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Description	Document style Web services use input (call parameter) types to determine the Web service operation being invoked. The operation name is not included in the SOAP request. For this reason, overwriting the operation name as part of an end point change does not work for document style Web services.
Platform	All.
Workaround	None.
CR259356	**Obscure error message when changing Web service end-point settings if an incorrect operation name is used.**
Description	The following message: `"javax.xml.rpc.JAXRPCException: Unable to find operation 'null' in port 'AVP2Port'. Please check the WSDL"` most likely means that an invalid operation name has been selected.
Platform	All.
Workaround	Find correct operation name and change end-point settings accordingly.
CR258884	**Security decisions are not audited.**
Description	XQuery functions supporting security are not tracked by the audit framework.
Platform	All.
Workaround	None.
CR257878	**Use of Java keywords in schema elements and namespaces may cause name conflicts.**
Description	Schema elements and namespaces are converted into Java classes and packages as part of schema compilation process. This creates the potential for name conflicts with Java keywords and constructs.
Platform	All.
Workaround	Avoid as necessary using syntax which the JVM might recognize as a reserved word or construct.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR256214	**Some DBMS systems may not properly handle "pushed down" constants.**
Description	SQL statements sent to base (not specifically supported) database platforms use a "best guess" as to the syntax for string literal. Such formulations may not work in all cases. An example of this is MySQL which requires every backslash [\ ] to be escaped with another backslash. Such cases are not handled by SQL generation code and might result in invalid SQL being generated.
Platform	Database platforms for which only generic support is provided.
Workaround	There are two possible workarounds for this problem: • Convert constants to parameters by using an external variable instead of a constant. For example: `where $customer_id eq fn-bea:fence("CUSTOMER001")` • Properly escape the XQuery string literal according to the rules of the underlying database.
CR253085	**Invalid SQL syntax generated for MS Access when fn:lower-case() or fn:upper-case() is used.**

Description	SQL generation for fn:lower-case() and fn:upper-case() functions produces SQL statements that fail to execute against MS Access database.
	fn:lower-case() is translated into LOWER() and fn:upper-case( ) into UPPER( ).
	The problem is that MS Access doesn't support these functions (LOWER( ) and UPPER( )), hence such generated SQL statements cannot be executed.
	Example:
	`for $i in CUSTOMER()`
	`where lower-case(data($i/FIRST_NAME)) eq "john"`
	`return $i/CUSTOMER_ID`
	is translated into:
	`SELECT t1."CUSTOMER_ID" AS c1`
	`FROM "CUSTOMER" t1`
	`WHERE LOWER(t1."FIRST_NAME") = 'john'`

Platform	All running MS-Access.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details

Workaround	Apply fn-bea:fence() to the parameter of an fn:lower-case() or fn:upper-case() function to block SQL pushdown and force evaluation in the engine.
	Example:
	`for $i in CUSTOMER()`
	`where`
	`lower-case(fn-bea:fence(data($i/FIRST_NAME))) eq "john"`
	`return $i/CUSTOMER_ID`
	is translated into:
	`SELECT t1."CUSTOMER" AS c1, t1."FIRST_NAME"`
	`FROM "CUSTOMER" t1`
	The function can then be successfully executed by MS Access.

CR248407	**Metadata import wizard fails to detect in/out parameters.**
Description	In some situations associated with MSSQL and Sybase stored procedures, a resultset is returned which is not automatically detected.
Platform	All.
Workaround	First, manually build a schema that is mapped to the output of the resultset. Then, when importing metadata use the wizard, add a ROWSET and link it to the previously created schema.
CR247416	**Running a Web service derived from a Data Services Control from a remote machine yields the following exception:** **"Current server is the coordinator and transaction is not found."**
Description	Inter-domain transactions between WebLogic Server 8.1SP4 and 8.1SP5 require that a command-line be passed.
Platform	All.
Workaround	When using inter-domain transactions between 8.1 SP4 and 8.5 SP5, the following command-line argument needs to be passed to the 8.1 SP5 domain: `-Dweblogic.transaction.SecurityInteropMode=compatibility` More details on the JTA transaction can be found at: http://e-docs.bea.com/wls/docs81/ConsoleHelp/jta.html#1106135

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR242938	**Multi-dimension soap arrays are not supported in RPC mode.**
Description	The Web services wrapper provided by AquaLogic Data Services Platform only supports single-dimension arrays in RPC style Web services.
Platform	All.
Workaround	None.
CR224815	**The initial invocation of a Web service from an application server typically takes more time than subsequent calls. If the timeout value is less than the time required for the first call, the alternate expression (typically a timeout error) will be evaluated.**
Description	There is "startup overhead" the first time that a web service is invoked. The overhead can exceed the timeout threshold, leading to the specified timeout error.
Platform	All.
Workaround	When setting timeout on expressions that have a Web service invocation, set the timeout value to be greater than the measured amount of time required for the first invocation.
CR239369	**XQueries may generate invalid SQL for databases not supporting UPPER and LOWER (SQL-92).** **Also, empty input handling for base databases (databases not specifically support) as well as Oracle deviates from the XQuery specification when UPPER(null) or LOWER(null) is pushed down to the database level.**

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Description	There are two aspects to this problem: 1. XQueries containing upper-case() or lower-case() functions are pushed down for database processing as UPPER and LOWER. Some databases may not support these SQL-92 keywords, however. In such cases the generated SQL will be invalid and upon execution will fail. 2. Similarly, input handling by base databases (as well as Oracle databases) may not match the XQuery specification. The reason for this is that the XQuery specification requires that functions *return an empty string* if input is an empty sequence. However, when these functions are pushed down, they *return an empty sequence* instead. This happens because LOWER(NULL) is NULL in SQL.
Platform	All platforms running base databases and Oracle databases.
Workaround	Use the fn-bea:Fence() function to prevent pushdown of upper-case() or lower-case() functions to the database. Example: `lower-case(fn-bea:fence(...))`
CR237186	**SDO update sequence is not preserved.**
Description	SDO update does not preserve the sequence of update objects; instead it updates in alphabetical order (example: **a**ddress, **c**redit card, **c**ustomer). Since the insertion order is automatic, the update will fail unless the update elements match alphabetical order.
Platform	All.
Workaround	If the dependency order matches alphabetical order, the update will be successful.
CR207637	**An exception appears for XQuery functions accessing metadata derived from Microsoft SQL Server stored procedures containing xs:decimal.**
Description	When importing a stored procedure from Microsoft SQL Server, the BEA JDBC driver incorrectly maps SQL decimal type to schema integer (xs:int) type.
Platforms	All.
Workaround	During stored procedure import, change the data type from xs:int to xs:decimal. Alternatively, you can change the imported data service's metadata to specify the schema type for the affected column to be xs:decimal.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
CR203394	**ROWTYPE input cursor is not supported when creating a data service from a stored procedure.**
Description	Stored procedure IN and INOUT cursors containing ROWIDs are not currently supported for metadata import.
Platform	All.
Workaround	Avoid importing metadata on stored procedures which required use of IN or INOUT ROWID parameters.
CR221145	**WSDLs with multiple services are not supported.**
Description	A limitation of the WebLogic Web services stack affects the ability to handle WSDL files with multiple data services.
Platform	All.
Workaround	Split WSDLs with multiple data services into multiple WSDL files each with one data service.
CR214585	**Erroneous results may occur when using fn:matches() with a regular expression containing a caret (^).**
Description	The match beginning-of-line operator (^) in regular expressions produces erroneous results when used with fn:matches().
Platform	All.
Workaround	None available.
CR215251	**Identifiers within two characters of the maximum length allowed by the DBMS may result in an error.**
Description	Some DBMS systems place limits on the length of identifiers (30 in the case of Sybase). AquaLogic Data Services Platform places single quotes around queries being pushed to the database, effectively reducing the maximum identifier length by two characters (28 in the case of Sybase).
Platform	All platforms running Sybase (and possibly other) databases.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Workaround	Possible options include renaming the table or creating a view with a shorter name.
CR203174	**Some data sources may not appear when importing metadata.**
Description	Functionality to manage JDBC data sources and connection pools during metadata import is limited to the same level of functionality provided by BEA Workshop (Tools → WebLogic Server → DataSource Viewer).
Platform	All.
Workaround	For full functionality use the WebLogic Administration Console to manage JDBC data sources and connection pools.
CR209659	**SOAP 1.2 for Web services runtime is not supported.**
Description	SOAP 1.2-based Web services are not supported.
Platform	All.
Workaround	Where possible use a supported SOAP version.
CR222822	**Linux applications built in Workshop may fail with the message "Error creating temporary file".**
Description	While building an application inside Workshop on Linux, you may get an “Error creating temporary file” message when 1) the application includes a large number of schema files, and 2) the Platform Installation is under a different user's name than the user running Workshop.
Platform	Linux.
Workaround	Ensure that the Workshop instance is running under the same name as was used for Platform installation.
CR226019	**Access control policies associated with a data service function may disappear if the function's number of parameters is changed.**
Description	A data service function's signature is its QName and the number of parameters (arity) of the function. If you set security policies on a function and then change the number of parameters to the function, the function is treated as new and any previously set policies will no longer be in effect.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Platform	All.
Workaround	If a function's arity is changed, reapply security policies to that function.
CR227486	**The BEA JDBC XA driver for DB2 returns the error XAER_RMERR when a local transaction read is followed by a global transaction read.**
Description	Pertains to a known JDBC driver problem described in BEA CR229071.
Platform	All.
Workaround	For the 3.4 JDBC driver, set the driver property to: `AllowImplicitResultSetCloseForXA=false`
CR229758	**AquaLogic Data Services Platform Web service generation may fail to compile if the original Web service was created in Workshop.**
Description	If you have a data service created from a Workshop-originated Web service and then, subsequently, you turn your data service into a Web service via a AquaLogic Data Services Platform control, the project build will fail due to a duplicate schema element error.
Platform	All.
Workaround	See CR229758 Details.
CR213916	**BEA Informix JDBC driver does not return nullability information.**
Description	The BEA Informix driver does not return information about table column nullability (that is, it is marked as unknown). During metadata import the minOccurs of the elements corresponding to the columns in the generated XML schemas is set to 0.
Platform	All platforms running Informix.
Workaround	Modify the imported metadata files by changing the minoccurs value for the nullable columns from 0 to 1.
CR214983, CR211701, CR201821	**MSSQL VARIANT datatype has only limited support.**

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Description	There are two limitations with this MSSQL VARIANT data type (sql_variant): 1. For SQL_VARIANT data type update will fail. 2. You cannot read a null value for the SQL_VARIANT data type.
Platform	All.
Workaround	None available.
CR221015	**During metadata import the BEA Sybase JDBC driver may not display all tables to which user has authorized access.**
Description	During metadata import the BEA Sybase JDBC driver may not show all tables which have been granted access to the user.
Platform	All platforms accessing Sybase through the BEA Sybase JDBC driver.
Workaround	This is a BEA Sybase driver limitation. For the import purpose, you can change to `dbo user` to see the full complement of available tables.
CR202963	**When using BEA Oracle JDBC driver with a TIMESTAMP values, stored procedures are truncated.**
Description	When using the BEA's Oracle JDBC driver, if a stored procedure returns a TIMESTAMP value then the value gets truncated at the milliseconds level. For example, if the value was 1997-01-31 09:26:50.124 then the stored procedure will return a 1997-01-31 09:26:50.0 value.
Platforms	All platforms running Oracle with the BEA Oracle JDBC driver.
Workaround	Use the Oracle JDBC driver stored procedures that returning TIMESTAMP values.
CR223429, CR228802	**Sybase JDBC driver does not support a getBlob() call.**
Description	The AquaLogic Data Services Platform cache configuration does not work if using Sybase JDBC driver because the configuration implementation uses a getBlob() call on the JDBC driver. The Sybase JDBC driver does not support getBlob().
Platforms	All platforms running Sybase with the Sybase JDBC driver.

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Workaround	Use the BEA JDBC driver for Sybase databases when utilizing Sybase as the AquaLogic Data Services Platform cache data source.
CR214730	**SQL Server JDBC driver incorrectly renders the tinyint maximum value.**
Description	The SQL Server tinyint maximum value of 255 gets interpreted as -1 by the Microsoft SQL Server JDBC driver.
Platforms	All platforms running SQL Server with the SQL Server JDBC driver.
Workaround	Use the BEA JDBC driver for SQL Server.
CR223486, CR226239, CR226171	**The Informix JDBC driver does not support standard JDBC syntax for specifying TIMESTAMP values.**
Description	The Informix native driver doesn't support standard JDBC syntax for specifying TIMESTAMP values. For example: 1979-03-01 00:00:00.0 is not supported.
Platforms	All platforms running Informix with the Informix JDBC driver.
Workaround	Use the BEA JDBC driver for Informix.
CR199675	**The BEA JDBC driver for Oracle does not support UROWID column type for data retrieval.**
Description	When using BEA JDBC driver for Oracle, retrieving UROWID returns an error, identified by the following message: `[BEA][Oracle JDBC Driver]Internal error: Net8 protocol error`
Platforms	All platforms running Oracle with the BEA JDBC driver.
Workaround	Use the Oracle JDBC driver if your data contains UROWID column type.
CR212515	**The Oracle stored procedure returning PL/SQL RECORD, BOOLEAN, or table with non-scalar element types is not supported.**
Description	Oracle stored procedure limitations are detailed in the following currently-available document: http://www.stanford.edu/dept/itss/docs/oracle/9i/java.920/a96654/ref.htm#1007714

Table 11 Known Product Limitations and Possible Workarounds

Topic	Details
Platforms	All platforms using Oracle.
Workaround	None available
CR202041	**Metadata for SQL Server stored procedures returning CURSOR output cannot be created.**
Description	When importing metadata from SQL Server, stored procedures that return CURSOR output are not supported.
Platforms	All platforms using SQL Server.
Workaround	Modify the imported data service file to identify the correct data type for the cursor.
CR227440	**Metadata for DB2 stored procedures returning CLOB data cannot be created.**
Description	When importing metadata from DB2, stored procedures returning CLOB data are not supported.
Platforms	All platforms using DB2.
Workaround	None available.
CR202962	**Oracle stored procedures containing CHAR or NCHAR as input may cause a Server error.**
Platform	All.

Description	If you have Oracle stored procedures that use an INOUT parameter, you may get an error when you run a stored procedure using AquaLogic Data Services Platform. The error appears as:
	`java.lang.RuntimeException: ORA-01460: unimplemented or unreasonable conversion requested`
	`ORA-06512: at "WIRELESS.SP_CHAR", line 17`

Workaround	Modify your stored procedure call by reducing the size of the INOUT parameter using TRIM. See Sample code related to CR202962, in Listing 1.

Supplemental Release Note Documentation

This section contains code and other additional information related to previously described release notes.

CR229758 Details

Summary

AquaLogic Data Services Platform Web service generation may fail to compile if the original Web service was created in Workshop.

Scenario and Workaround

Assume that you have a Workshop-generated Web service named getCustomerOrderByCustomerID. A WSDL for this Web service with default targetnamespace would create the following schema elements:

```
{http://www.openuri.org/}getCustomerOrderByCustomerID
```

and

```
{http://www.openuri.org/}getCustomerOrderByCustomerIDResponse
```

When you import this WSDL, the same schema elements will be in the imported schema file and will be associated with the data service read function. When you build a Data Service control for this read function and generate a Web service from that control, a Web service function is created:

```
org.openuri.GetCustomerOrderByCustomerIDResponseDocument
getCustomerOrderByCustomerID(org.openuri.GetCustomerOrderByCustomerIDDo
cument p0)
```

When compiling this Web service file, WebLogic Workshop now generates two identical schema elements each:

```
{http://www.openuri.org/}getCustomerOrderByCustomerID
```

and

```
{http://www.openuri.org/}getCustomerOrderByCustomerIDResponse
```

If user had modified either the target namespace of this final JWS or changed the function name to a different one such as:

```
org.openuri.GetCustomerOrderByCustomerIDResponseDocument
getCustomerOrderByCustomerIDNEW(org.openuri.GetCustomerOrderByCustomerI
DDocument p0)
```

Then the types generated will be as follows:

- {http://www.openuri.org/}getCustomerOrderByCustomerID
- {http://www.openuri.org/}getCustomerOrderByCustomerIDResponse

and

- {http://www.openuri.org/}getCustomerOrderByCustomerIDNEW
- {http://www.openuri.org/}getCustomerOrderByCustomerIDNEWResponse

Thus the schema clash would be avoided.

CR264597 Details

Summary

String comparison operations involving MSSQL may return incorrect results when the comparison operation is computed by MSSQL.

Description

Depending on the database and server configuration, MSSQL Server may use case-insensitive collation for string comparison operations (this is the default configuration).

When generating SQL the AquaLogic Data Services Platform does not take database string collation into account. This can lead to different results being produced by expressions that were "pushed down" to a MSSQL database, as compared to their evaluation by the XQuery engine.

The following types of expressions are affected:

- string comparison operations
- string functions: fn:contains( ), starts-with( ), ends-with()
- order by clauses
- group by clauses.

For example, consider the following two-row, two-column table based on:

```
CUSTOMER(ID, FIRST_NAME)
```

ID	FIRST_NAME
1	John
2	john

The following XQuery might return different results depending whether it is evaluated by the database or not.

```
for $c in CUSTOMER()
where $c/FIRST_NAME eq "john"
return $c/ID
```

According to the XQuery semantics the query should return:

```
<ID>2</ID>
```

as only the second record matches the selection criteria.

However, when AquaLogic Data Services Platform pushes the query to the underlying MSSQL database the following SQL is generated:

```
SELECT t1."C_ID" AS c1
FROM "CUSTOMER" t1
WHERE t1."FIRST_NAME" = "john"
```

This might result in both records being returned by the MSSQL database (with case-insensitive string collation set):

```
<ID>1</ID>
<ID>2</ID>
```

Workaround

There are several workarounds to conforming with XQuery semantics for string comparisons when pushing computations down to MSSQL.

Option 1

Consider changing the collation setting that the database uses for string comparisons. See "SQL Server Collation Fundamentals" document located as of this writing at:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_da_1pwz.asp

Collation can change on a server, database or column level basis.

Option 2

Use the fn-bea:fence( ) function to block pushdown. In the above example, this would be rendered as:

```
for $c in CUSTOMER()
where fn-bea:fence(data($c/FIRST_NAME)) eq "john"
return $c/ID
```

Notice, however, that this approach may negatively impact performance since the AquaLogic Data Services Platform engine now must fetch and process the entire table.

To optimize performance, consider replicating comparison operation and pushing down one copy to be evaluated by the database while keeping the second copy on the AquaLogic Data Services Platform engine. The following query illustrates such an approach:

```
for $c in CUSTOMER()
where $c/FIRST_NAME eq "john"
where fn-bea:fence(data($c/FIRST_NAME)) eq "john"
return $c/ID
```

This query first limits the number of results that the XQuery engine must process and then applies the second selection to obtain the correct XQuery semantics.

CR202962 Listing

Listing 1 contains sample code for CR202962.

Listing 1 Sample code related to CR202962

```
CREATE OR REPLACE PROCEDURE WIRELESS.SP_CHAR
(P_CHAR_IN IN CHAR,
P_CHAR_OUT OUT CHAR,
P_CHAR_INOUT IN OUT CHAR,
P_ID_OUT OUT VARCHAR2 )
IS
TEMP VARCHAR2(10);
BEGIN
SELECT C_ID INTO P_ID_OUT
FROM WIRELESS.ALL_DATATYPES
WHERE C_CHAR = P_CHAR_IN;

SELECT C_CHAR INTO P_CHAR_OUT
FROM WIRELESS.ALL_DATATYPES
WHERE C_ID = '2';

SELECT C_ID INTO TEMP
FROM WIRELESS.ALL_DATATYPES
WHERE C_CHAR = P_CHAR_INOUT;

SELECT 'WORK' INTO P_CHAR_INOUT
FROM WIRELESS.ALL_DATATYPES
WHERE C_ID = TEMP;
END;
/
```

to adjust the size of `PCHAR_INOUT` using `TRIM` (see highlighted code)

```
CREATE OR REPLACE PROCEDURE WIRELESS.SP_CHAR
(P_CHAR_IN IN CHAR,
P_CHAR_OUT OUT CHAR,
P_CHAR_INOUT IN OUT CHAR,
P_ID_OUT OUT VARCHAR2 )
IS
TEMP VARCHAR2(10);
ACHAR CHAR(500);

BEGIN
ACHAR := trim(P_CHAR_INOUT);

SELECT C_ID INTO P_ID_OUT
FROM WIRELESS.ALL_DATATYPES
WHERE C_CHAR = P_CHAR_IN;

SELECT C_CHAR INTO P_CHAR_OUT
FROM WIRELESS.ALL_DATATYPES
WHERE C_ID = '2';

SELECT C_ID INTO TEMP
FROM WIRELESS.ALL_DATATYPES
WHERE C_CHAR = ACHAR;  // used to fail here

SELECT 'WORK' INTO P_CHAR_INOUT
FROM WIRELESS.ALL_DATATYPES
WHERE C_ID = TEMP;

END;
/
```

CR288104 Details

Here is the Java Web service schema generated under WebLogic 9.2, following by the same schema generated under WebLogic 8.1:

First is shown the generated Web service schema under WebLogic 9.2. Notice that the schema is fully namespace qualified:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/";
xmlns="http://a/b/c/d/e";
xmlns:m1="ld:DataServices/CustomerDB/CUSTOMER">
  <soapenv:Body>
<hello>
  <m1:CUSTOMER>
```

```
      <m1:CUSTOMER_ID>s</m1:CUSTOMER_ID>
      <m1:FIRST_NAME>s</m1:FIRST_NAME>
      <m1:LAST_NAME>s</m1:LAST_NAME>
      <m1:CUSTOMER_SINCE>2001-01-01</m1:CUSTOMER_SINCE>
      <m1:EMAIL_ADDRESS>s</m1:EMAIL_ADDRESS>
      <m1:TELEPHONE_NUMBER>s</m1:TELEPHONE_NUMBER>
      <m1:SSN>s</m1:SSN>
      <m1:BIRTH_DAY>2001-01-01</m1:BIRTH_DAY>
      <m1:DEFAULT_SHIP_METHOD>s</m1:DEFAULT_SHIP_METHOD>
      <m1:EMAIL_NOTIFICATION>1</m1:EMAIL_NOTIFICATION>
      <m1:NEWS_LETTTER>1</m1:NEWS_LETTTER>
      <m1:ONLINE_STATEMENT>1</m1:ONLINE_STATEMENT>
      <m1:LOGIN_ID>s</m1:LOGIN_ID>
    </m1:CUSTOMER>
  </hello>
```

Here is the generated Web service schema under WebLogic 8.1:

```
<getADDRESS xmlns="http://www.openuri.org/";
xmlns:cus="ld:DataServices/CustomerDB/CUSTOMER">
  <cus:CUSTOMER>
    <CUSTOMER_ID xmlns="">string</CUSTOMER_ID>
    <FIRST_NAME xmlns="">string</FIRST_NAME>
    <LAST_NAME xmlns="">string</LAST_NAME>
    <CUSTOMER_SINCE xmlns="">2008-09-29</CUSTOMER_SINCE>
    <EMAIL_ADDRESS xmlns="">string</EMAIL_ADDRESS>
    <TELEPHONE_NUMBER xmlns="">string</TELEPHONE_NUMBER>
    <!--Optional:-->
    <SSN xmlns="">string</SSN>
    <!--Optional:-->
    <BIRTH_DAY xmlns="">2014-09-19</BIRTH_DAY>
    <!--Optional:-->
    <DEFAULT_SHIP_METHOD xmlns="">string</DEFAULT_SHIP_METHOD>
    <!--Optional:-->
    <EMAIL_NOTIFICATION xmlns="">1</EMAIL_NOTIFICATION>
    <!--Optional:-->
    <NEWS_LETTTER xmlns="">1</NEWS_LETTTER>
    <!--Optional:-->
    <ONLINE_STATEMENT xmlns="">1</ONLINE_STATEMENT>
    <!--Optional:-->
    <LOGIN_ID xmlns="">string</LOGIN_ID>
  </cus:CUSTOMER>
</getADDRESS>
```

References

The most up-to-date product documentation is available from the BEA edocs documentation Web site at the following location:

http://edocs.bea.com/aldsp/docs25/index.html

To access the AquaLogic Data Services Platform documentation home page in a Web browser:

1. Click the PDF files button.
2. Select the document that you want to view or print.

Alternatively, from within a HTML document, click the View as PDF link on the top navigation bar to open a PDF version of the currently displayed document.

Information about BEA products including AquaLogic Data Services Platform can be found at:

http://dev2dev.bea.com

Documentation for all BEA products in both PDF and HTML format can be found at:

http://edocs.bea.com

If you do not have Adobe Acrobat Reader, you can obtain it from the Adobe Web site at:

http://www.adobe.com

Sample Domain

Samples, examples, sample tutorials, and the RTLApp sample application are designed to be run on the ldplatform domain, located at:

```
<weblogic81>/samples/domains
```

The ldplatform domain contains the data necessary for these samples to be developed and run.