BEA AquaLogic Data Services Platform™

Release Notes

Note: Product documentation may be revised post-release and made available from the BEA e-docs site, listed below. Any major changes are noted in the online version of these notes.

`http://e-docs.bea.com/aldsp/docs21/index.html`

Version: 2.1
Document Date: June 2005
Revised: March 2006

Copyright

Restricted Rights Legend

Trademarks or Service Marks

March 16, 2006 4:12 pm

Contents

Release Notes

BEA AquaLogic Data Services Platform, Version 2.1

Most recent revision: March 16, 2006

About BEA AquaLogic Data Services Platform

Data Services Platform (DSP) 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 Data Services Platform mediator API, JDBC, SQL, or Data Services Platform controls — can invoke DSP 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: Data Services Platform was originally named Liquid Data. Some artifacts of the original name remain in the product, path, and components.

Revision Policy

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

`http://edocs.bea.com/aldsp/docs21/relnotes/index.html`

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

This document includes the following topics:

- What’s New in Data Services Platform 2.1
- Product Limitations and Workarounds
- References

What’s New in Data Services Platform 2.1

Version 2.1 adds a number of new features to Data Services Platform. These are described in Table 1.

Table 1 New and Enhanced Data Services Platform Functionality in DSP Release 2.1

Feature	Details
Audit framework	Provides uniform auditing across heterogeneous sources including: • Enabling read, update, and administration access for auditing through a configurable audit stream. • Configurable audit events via the Data Services Platform Console.
ADO.NET support	Allows for building read/write Microsoft .NET applications on top of DSP. Specifically: • Data services are exposed as ADO.NET services. • Automated updates are supported (that is, ADO diffgrams are propagated to backend sources). • Disconnection programming model is supported. **Note:** Any DSP schema (XML type) structure used for ADO.NET must confirm to ADO.NET’s more restrictive, database-centric schema model. For details see “Supporting ADO.NET Clients” in the *Application Developer’s Guide*.
Operational Monitoring	Detailed access metrics for SLA-monitoring, logging, or client logic. • Returned data can be configured to the desired level of detail. • Monitoring data can include execution times on data service level, queries, and result sizes into databases. • Monitoring data can be streamed to databases, log files, or the calling client application.
Catalog services	Catalog services provide metadata information for dynamic discovery of data services by client applications. This API: • Exposes details such as data service and function-level data services, functions, relationships, schemas, folder, and application dependencies.

Feature	Details
Data lineage graph	Improves maintainability by providing a graphical analysis of data service relationships. Features include: • Dependency view. • "Where used" view. • Adjustable granularity (per data service or per function). • Full integration with the Data Services Platform Console metadata browser.
Refactoring	Improves maintainability from staging to production by allowing data service developers to update dependencies when moving, renaming, or deleting data services, functions, or namespaces.
Externalizing physical source data configuration	Connection information for data sources in deployed data services can be changed to support operational staging.
Ad hoc query support	Allows entry and execution of on-the-fly query expressions in Test View.
Procedures	Exposes in data services stored procedures and Web service side-effecting functions that do not necessarily return data.
Generic SDO support	Provides the ability to dynamically invoke data service routines in support of frequently changing data source structure.
Performance enhancements	In addition to increased query and update speed, developers can now define optimizable queries. This includes: • Inverse functions that enable the pushdown of predicates using custom functions. • Updates of calculated fields in results. • A Query Plan View visual indicator for sub-optimal queries. • Enhanced performance through streamlining of result set computation. • Improved SQL pushdown for such constructs as if-then-else, subsequence, and outer-join.
Enhanced WebLogic server support	Both WebLogic Platform 8.1 SP4 and SP5 have been certified for the release. In addition, the data service client API can be utilized by WebLogic Server 9.*x* applications. For detailed information on supported hardware, operating systems, databases, and connectivity software see "Preparing to Install BEA AquaLogic Data Services Platform" in the DSP *Installation Guide*.

Product Limitations and Workarounds

Table 2 lists known, potentially encountered limitations associated with the current BEA AquaLogic Data Services Platform 2.1 release. 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.

Please contact BEA customer support at:

`http://support.bea.com`

for assistance in tracking unresolved issues.

Table 2 Known Product Limitations and Applicable Workarounds

Topic	Details
CR267820	**Metadata synchronization update preview shows "Problem in parsing XML fragment" error.**
Description	If your project contains Java function signatures referring to XMLBeans, then erroreous 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 rebuilt.
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."
CR266236, CR267279	**Samples Tutorial I and Samples Tutorial II documentation are not updated from the 2.0.1 level.**
Description	See above.
Platform	All.
Workaround	None.

Topic	**Details**
CR253530	**When importing DSP projects into Workshop if the Copy into Application Directory option is not selected, importing a DSP project not already located in the application results in resource not found errors.**
Description	DSP 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 Copy into Application Directory option when importing DSP projects.
CR235816	**A ClassCastException occurs when the DSP JDBC driver is invoked from DB Visualizer if ldjdbc.jar is not higher on the CLASSPATH.**
Description	The problem occurs when trying to retrieve data after `ldjdbc.jar` has been added via DB Visualizer -> Database ->Driver Manager.
Platform	All.
Workaround	Put `ldjdbc.jar` on dbvisualizer's class path ahead of weblogic.jar.
CR204243	**In creating a BigDecimal (basis for xs:decimal) from a long value (basis for xs:integer) sometimes the result are not always 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. 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 validate is active.**
Description	Sometimes the order of elements in a diffgram changes which can lead to datagraph validation failure.

Topic	**Details**
Platform	All using ADO.NET clients.
Workaround	If possible, turn off validation for the operation.
CR268183	**Workshop removes a necessary line when editing a DSP Control-generated JWS file.**
Description	After creating a DSP 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 in the JWS header is needed in order to generate ADO.NET-enabled WSDL files.
Platform	All.
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 edit the file in Workshop and in the process restore the line to the JWS file.
CR245418	**SDO does not support periods [.] in element names.**
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.
Platform	All.
Workaround	Avoid using periods in element names.
CR265706	**When no values are passed during an insert operation, the generated query contains a null, which leads to an exception on the database side.**

Topic	Details
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 appearsis that when metadata is created for database tables, the default value of the database setting is not retained in SDO. So, for example, if the value of the column is "not null" in the database and the SDO datagraph passes a does not have a value assigned to that column, the exception will occur.
Platform	All.
Workaround	Make sure that SDO properties for database columns that cannot be set to null are populated properly, either through client code or through server-side update override code.
CR266289	**SDO does not support periods in element names.**
Description	A more general limitation statement is that SDO does not support periods in schema paths, including leaf elements.
Platform	All.
Workaround	Avoid using periods in schema paths.
CR265950	**End point name changes do not take effect for operations in document-style Web services.**
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.
CR265230	**Documentation should make clear that the samples tutorial needs to be created on the ldplatform samples domain.**

Topic	Details
Description	Samples, sample tutorials, and the RTLApp sample application are all designed to be run on the ldplatform domain, located at: `<weblogic81>/samples/domains` This is where the sample data used in the various samples and applications is located.
Platform	All.
Workaround	N/A.
CR259692	**DSP Console should provide a means to block a user from running ad hoc queries.**
Description	Occasions may arise when it is necessary to block a particular user or user group from running ad hoc queries.
Platform	All.
Workaround	Users or user groups can be blocked from running ad hoc queries by securing the EJB method executeQuery() from the WebLogic Administration Console.
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 accordingly.
CR258884	**Security decisions are not audited.**
Description	XQuery functions supporting security are not tracked by the audit framework.
Platform	All.
Workaround	None.

Topic	Details
CR257878	**Use of Java keywords in schema elements and namespaces can 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.
CR256214	**Some base platforms 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 literals which 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	All using a base database platform.
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.**

Topic	Details

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, 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.

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.

Topic	Details
CR247416	**Running a Web service created 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 flag to the 8.1 SP5 domain is needed:
	`-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`

CR242938	**Multi-dimension soap arrays are not supported in RPC mode.**
Description	Currently the Web services wrapper provided by DSP only supports single-dimension arrays in RPC style Web services
Platform	All.
Workaround	None.
CR224815	**The initial invocation of a Web service call from the 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.
CR238736	**Wildcards may not work when specifying a catalog name.**

Topic	Details
Description	The JDBC driver may not support the use of wildcards (i.e., WIRE% for WIRELESS) as a means of identifying database catalogs. The result is that no catalog is found when a wildcard is used to specify an available database catalog.
Platform	All.
Workaround	Use the exact name, not wildcards, when specifying a catalog name.
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.**
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 will upon execution 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 while 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.

Topic	Details
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.
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 interferes with the ability to handle WSDL files with multiple data services.
Platform	All.
Workaround	Split multiple services into multiple WSDL files.
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.

Topic	Details
CR215251	**Identifiers within two characters of the maximum length allowed by the database may result in an error.**
Description	Some databases place limits on the length of identifiers (30 in the case of Sybase). Data Services Platform places single quotes around pushdown queries, effectively reducing the maximum identifier length by two characters (28 in the case of Sybase).
Platform	All platforms running Sybase (and possibly other) databases.
Workaround	Possible options include renaming the table or creating a view with a shorter name.
CR203174	**Some data sources may not appear when attempting to import metadata.**
Description	The 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	This version of Data Services Platform cannot invoke SOAP 1.2 based Web services.
Platform	All.
Workaround	None available.
CR222822	**Linux application built in Workshop may fail with the message "Error creating temporary file".**
Description	While building an application inside Workshop on Linux, you may get the above error when 1) the application includes a large number of schema files, and 2) when the Platform Installation is under a different user's name than the user running Workshop.
Platform	Linux.

Topic	Details
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 it's QName and the number of parameters (arity) of the function. If you set security policies on a function and changes the number of parameters to the function, then this function is treated as a new function and the previously set policies will no longer be in effect.
Platform	All.
Workaround	If a functions arity is changed, the user will have to 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	**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 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.**

Topic	Details
Description	The BEA Informix driver does not return information about table column nullability (that is, it is marked as unknown). Thus 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.**
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.
CR218449	**Data services cannot have multibyte namespace prefixes.**
Description	During metadata import, tables with multibyte names may create multibyte namespace prefixes based on the table name's first few characters. The resulting data service file will not be parsed correctly because the XMLBeans parser is unable to handle multibyte namespace prefix value.
Platform	All.
Workaround	None available.
CR211377	**In Source View syntax highlighting may become off by a character if certain sequences appear in the query.**

Topic	**Details**
Description	If your query has any of the following patterns: • Numeric XML character entity references, e.g., `&#20` • Alphabetic XML character entity references, e.g., `&amp`, `&apos`, `&gt`, `&lt`, and `&quot` • The `{{` and `}}` tokens. • Escaped single and double quotes. then syntax highlight may become off by one.
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 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().

Topic	Details
Platforms	All platforms running Sybase with the Sybase JDBC driver.
Workaround	Use the BEA JDBC driver for Sybase databases when utilizing Sybase as the 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	**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`

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 the 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.
CR265965	**Updating or deleting Oracle's 'CHAR'/'NCHAR' with trailing blanks failed with 'Optimistic locking failure' 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.
Platform	All using Oracle's non-XA JDBC driver.
Workaround	Where possible use BEA's Oracle JDBC driver (Type 4).
CR202962	**Oracle stored procedures containing CHAR or NCHAR as input can cause a Server error.**
Platform	All.

Topic	Details

Description	If you have Oracle stored procedures that use an INOUT parameter, you may get an error when you run a stored procedure using 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

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, now workshop generates two identical schema elements each for:

```
{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 will 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 DSP 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:

```
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 DSP 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 to 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 either be changed on server, database or column level.

Option 2

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

```
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 DSP engine now has to 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 ALDSP 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 get 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;
/
```

References

Documentation for this product is available on an Online Documentation CD-ROM that ships with the product, and is also available on the Web at the BEA e-docs documentation site at the following location:

`http://e-docs.bea.com/aldsp/docs21/index.html`

To access the 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 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://e-docs.bea.com`

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

`http://www.adobe.com`