



BEA Liquid Data for WebLogic®

Release Notes

Version: Liquid Data 8.5
Document Date: June 2005
Revised: June 2005

Copyright

Copyright © 2005 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, BEA JRockit, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

July 22, 2005 11:37 am

Contents

Release Notes

About this BEA Liquid Data for WebLogic Release.....	1
Product Limitations and Workarounds.....	2
References.....	14

Release Notes

BEA Liquid Data for WebLogic Version 8.5

Date: July 22, 2005

About this BEA Liquid Data for WebLogic Release

BEA Liquid Data for WebLogic 8.5 provides on-demand access to, and aggregation of, information in relational databases, Web services, Java function, XML files, and other types of disparate data. During development, Liquid Data is accessed as an application running under WebLogic Workshop. This is where data services, including query functions, are developed.

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

These Release Notes are subject to revision and are available at:

<http://edocs.bea.com/liquiddata/docs85/relnotes/index.html>

Items modified or added are so marked by date.

This document includes the following topics:

- [Product Limitations and Workarounds](#)
- [References](#)

Product Limitations and Workarounds

This section describes limitations in the current BEA Liquid Data for WebLogic 8.5 release and includes workarounds, where applicable.

The table below lists a CR (Change Request) number for each problem. Refer to this track number as limitations are addressed.

Please contact BEA customer support at <http://support.bea.com> for assistance in tracking unresolved problems.

Table 1 Known Product Limitations and Workarounds

Topic	Details
CR203394	Java functions return elements as untyped. [added: June 30, 2005]
Problem	An untyped XML stream is returned by the runtime Java function processor instead of returning a typed XML stream.
Platform	All
Workaround	<p>For Java functions that return an XMLBean object, add a wrapper read function in the physical data service file that validates the returned Java object by associating it with a type.</p> <p>Specifically, the user can make the imported external function private and then create a public read method that wraps the results of the external call with a <code>validate()</code>.</p> <p>Modify the physical data service function to be private by changing the kind attribute from <code>read</code> to <code>private</code>:</p> <pre>(::pragma function <f:function xmlns:f="urn:annotations.ld.bea.com" kind="private" nativeName="getEntryByCn"> <params> <param nativeType="com.customer.cr233742.AttrDocument\$Attr" /> </params> </f:function>::)</pre>

Topic	Details
	<pre data-bbox="552 354 1157 435">declare function f1:getEntryByCn(\$x1 as element(t1:attr)) as schema-element(t1:attr) external;</pre> <p data-bbox="512 447 1247 529">1. Create a wrapper function that validates the physical data service function with kind="read" so that it is exposed to the other data service. For example:</p> <pre data-bbox="512 545 1009 656">(::pragma function <f:function xmlns:f="urn:annotations.ld.bea.com" kind="read"> </f:function>::)</pre> <pre data-bbox="512 689 1193 826">declare function f1:wrappGetEntryByCn(\$x1 as element(t1:attr)) as schema-element(t1:attr) { validate{f1:getEntryByCn(\$x1)} };</pre>
CR203394	ROWTYPE input cursor not currently supported when turning a stored procedure into a data service.
Problem	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.
CR218040	The query engine cannot retrieve data from a Sybase data source when generated SQL contains functions which have dependencies on Java services.
Problem	<p data-bbox="512 1279 1228 1361">When using Sybase as a datasource, some SQL functions such as LTRIM and RTRIM work correctly in the database only when Java services are enabled by running the following in the SA role.</p> <pre data-bbox="552 1376 964 1400">sp_configure, "enable_java", 1</pre> <p data-bbox="512 1416 935 1440">The error message indicating this symptom is:</p> <p data-bbox="552 1456 1228 1536">Cannot run this command because Java services are not enabled. A user with System Administrator (SA) role must reconfigure the system to enable Java.</p>

Topic	Details
Platform	All platforms running Sybase.
Workaround	Enable Java services while in the SA role. Refer to Sybase documentation for further details.
CR231297	A cached data source being accessed using an <code>async()</code> function fails with an internal error.
Problem	Asynchronous access to a cached data source using the <code>fn-bea:async()</code> function may result in an internal error.
Platform	All
Workaround	Async access and caching are two tactics to improve latency of slow data sources. However, you should avoid using them together for the same source.
CR210000	The XQuery 1.0 <code>processing-instruction()</code> node test is not fully supported in the BEA XQuery processor.
Problem	Using <code>processing-instruction("name")</code> ignores the "name" and instead retrieves all processing-instruction nodes.
Platform	All
Workaround	None available
CR221145	WSDLs with multiple services may not work correctly with Liquid Data.
Problem	This is a limitation with the WebLogic web services stack.
Platform	All
Workaround	Split multiple services in the WSDL into multiple WSDL files.
CR214585	Erroneous results when using <code>fn:matches()</code> with a regular expression containing a caret (^).
Problem	The match beginning-of-line operator (^) in regular expressions produces erroneous results when used with <code>fn:matches()</code> .
Platform	All
Workaround	None available

Topic	Details
CR215251	Identifiers within two characters of the maximum length allowed by the database result in an error.
Problem	Some databases place limits on the length of identifiers (30 in the case of Sybase). Liquid Data 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.
Problem	The functionality to manage JDBC data sources and connection pools during metadata import is limited to the same level of functionality provided by via 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.
Problem	This version of Liquid Data 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".
Problem	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.
Problem	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 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	BEA DB2 JDBC XA driver returns the error XAER_RMERR when a local transaction read is followed by a global transaction read.
Problem	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: <code>AllowImplicitResultSetCloseForXA=false</code>
CR229758	Liquid Data Web service generation may fail to compile if the original was created in Workshop.
Problem	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 Liquid Data control, the project build will fail due to a duplicate schema element error.
Platform	All

Topic	Details
Workaround	<p>Assume that you have a Workshop-generated Web service named <code>getCustomerOrderByCustomerID</code>. A WSDL for this Web service with default <code>targetnamespace</code> would create the following schema elements:</p> <pre>{http://www.openuri.org/}getCustomerOrderByCustomerID</pre> <p>and</p> <pre>{http://www.openuri.org/}getCustomerOrderByCustomerIDResponse</pre> <p>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 Liquid Data control for this data service read function and generate a Web service from that control, a Web service function is created:</p> <pre>org.openuri.GetCustomerOrderByCustomerIDResponseDocument getCustomerOrderByCustomerID(org.openuri.GetCustomer OrderByCustomerIDDocument p0)</pre> <p>When compiling this Web service file, now workshop generates two identical schema elements each for</p> <p>When compiling this Web service file, Workshop now generates two identical schema elements each for:</p> <pre>{http://www.openuri.org/}getCustomerOrderByCustomerID</pre> <p>and</p> <pre>{http://www.openuri.org/}getCustomerOrderByCustomerIDResponse.</pre> <p>Thus the schema elements clash.</p> <p>If you modify either the target namespace of the final JWS or change the function name to a different one such as:</p> <pre>org.openuri.GetCustomerOrderByCustomerIDResponseDocument getCustomerOrderByCustomerIDNEW(org.openuri.GetCustomer OrderByCustomerIDDocument p0)</pre> <p>Then the types generated will be as follows and there will be no clash.</p>

Topic	Details
CR231733	<pre data-bbox="448 354 1170 604"> {http://www.openuri.org}getCustomerOrderByCustomerID {http://www.openuri.org}getCustomerOrderByCustomerIDResponse {http://www.openuri.org}getCustomerOrderByCustomerIDNEW and {http://www.openuri.org}getCustomerOrderByCustomerIDNEWResponse </pre> <p data-bbox="448 618 805 644">Thus the schema clash will be avoided.</p>
Problem	<p data-bbox="448 791 1170 930">If you have two schema files, one importing another, resulting in having an imported schema global element that is in a different namespace than the importing schema file's namespace, then the generated data service file will be invalid if the imported schema global element is the target element of the data service.</p> <p data-bbox="448 947 1170 1095">For example, say you have schema <code>s1.xsd</code> containing global element <code>{urn:schema:s1}E1</code> and schema <code>s2.xsd</code> containing global element <code>{urn:schema:s2}E2</code>. <code>s1.xsd</code> includes <code>s2.xsd</code>. If you associate the element <code>E2</code> by selecting it from the <code>s1.xsd</code> schema file, then the generated data service file will have an incorrect schema import statement.</p>
Platform	All
Workaround	<p data-bbox="448 1173 1170 1260">The statement in the problem description (above) can be corrected by changing the import file's namespace to point to the <code>urn:schema:s2</code> namespace instead of <code>urn:schema:s1</code> namespace.</p> <p data-bbox="448 1277 1170 1329">Alternatively, you can select <code>E2</code> by associating that element from the <code>s2.xsd</code> file instead of selecting it from <code>s1.xsd</code>.</p> <p data-bbox="448 1347 1170 1433">In the general case associate the data service file with the schema file that contains the element instead of selecting schema file that has the import statement.</p>
CR213916	BEA Informix JDBC driver does not return nullability information.

Topic	Details
Problem	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.
Problem	There are two limitations with this MSSQL VARIANT data type (sql_variant): <ol style="list-style-type: none"> 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.
Problem	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 the Source View syntax highlighting may become off by a character if certain sequences appear in the query.

Topic	Details
Problem	<p>If your query has any of the following patterns:</p> <ul style="list-style-type: none"> • Numeric XML character entity references, e.g., <code>&#20</code> • Alphabetic XML character entity references, e.g., <code>&amp</code>, <code>&apos</code>, <code>&gt</code>, <code>&lt</code>, and <code>&quot</code> • The <code>{ { and } }</code> tokens. • Escaped single and double quotes. <p>then syntax highlight may become off by one.</p>
Platform	All
Workaround	None available
CR219009	Liquid Data IDE Test View limitation when performing a SDO update.
Problem	<p>Through Liquid Data Test View Data you can accomplish typical read and update actions. For example, you can create new SDO records as part of update steps. However, when adding a new SDO record that has a nested structure, the template generated will only have first-level (i.e., no nested) elements.</p>
Platform	All
Workaround	You can add subsequent nested SDO records by right-mouse clicking on the container node.
CR221015	During metadata import the BEA Sybase JDBC driver is not showing all tables to which user has authorized access.
Problem	<p>During metadata import the BEA Sybase JDBC driver may not show all tables which have been granted access to the user.</p>
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 <code>dbo user</code> to see the full complement of available tables.
CR221956	Repeated builds of a Workshop application containing one or more Web service projects can run out of memory.
Problem	<p>If you have a Workshop application that has a large number of Web service files (<code>.jws</code>) in a Web service project, the build of such an application can cause a memory leak and, after certain number of repeated attempts, Workshop may run out of memory. This, in turn, can lead to various class loading errors.</p>

Topic	Details
Platform	All
Workaround	Obtain and apply the Workshop patch associated with CR226814. This patch corrects the memory leak problem described above.
CR202963	When using BEA Oracle JDBC driver with a TIMESTAMP values stored procedures gets truncated.
Problem	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 not supporting getBlob() call.
Problem	The Liquid Data 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.
Workaround	Use the BEA JDBC driver for Sybase databases when utilizing Sybase as the Liquid Data cache data source.
CR214730	SQL Server JDBC driver incorrectly renders the tinyint maximum value.
Problem	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, CR226171, CR226239	The Informix JDBC driver does not support standard JDBC syntax for specifying TIMESTAMP values.
Problem	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.

Topic	Details
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.
Problem	When using BEA JDBC driver for Oracle, retrieving UROWID returns an error, identified by the following message: <pre>[BEA] [Oracle JDBC Driver]Internal error: Net8 protocol error</pre>
Platforms	All platforms running Oracle with the BEA JDBC driver.
Workaround	Use the Oracle JDBC driver if your data contains UROWID column type.
212515	The Oracle stored procedure returning PL/SQL RECORD, BOOLEAN, or table with non-scalar element types is not supported.
Problem	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
Platforms	All platforms using Oracle.
Workaround	None available
CR202041	Metadata for SQL Server stored procedures returning CURSOR output cannot be created.
Problem	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.
Problem	When importing metadata from DB2, stored procedures returning CLOB data are not supported.

Topic	Details
Platforms	All platforms using DB2.
Workaround	None available.
CR202962	Oracle stored procedures containing CHAR or NCHAR as input can cause a server error.
Platform	All
Problem	<p>If you have Oracle stored procedures that use an INOUT parameter, you may get an error when you run a stored procedure using Liquid Data. The error appears as:</p> <pre>java.lang.RuntimeException: ORA-01460: unimplemented or unreasonable conversion requested ORA-06512: at "WIRELESS.SP_CHAR", line 17</pre>
Workaround	Modify your stored procedure call by reducing the size of the INOUT parameter using TRIM. See Listing 1 , below.

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://edocs.bea.com/liquiddata/docs85/index.html>

To access the [Liquid Data 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 Liquid Data 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 get it from the Adobe Web site at <http://www.adobe.com>.

