

IBM Red Brick Warehouse



# RELEASE NOTES for Windows Platforms

*Version 6.20.TC6*



---

# Contents

<b>RELEASE NOTES</b>	<b>1</b>
Important release information	1
Customer support	1
New features in V6.20.xC5	1
New features in V6.20.xC2	1
New features in V6.20.xC1	2
Supported Windows platforms	4
ODBC Driver compatibility	5
JDBC Driver requirements	5
Significant operational considerations	6
Result data type for string functions	6
Red Brick Data Mine not supported	6
Performance of OLAP functions	6
Trailing spaces for CHAR columns preserved in query results	6
Additional memory allocated for large indexes	6
RANGE MOVE clause for ALTER SEGMENT	7
Installation and upgrade notes	7
For the first time	7
With an existing warehouse	7
Fixed issues in V6.20.x	8
Database administration issues	8
Query-related issues	11
Connectivity issues	13
Administrator tool issues	14
Documentation issues	15
Known issues	15
Database administration issues	16
Query-related issues	17
Connectivity issues	17
Administrator tool issues	19
Documentation notes	19
IBM Red Brick Warehouse library	19
Corrections to the Japanese documentation	20
Installation and Configuration Guide for Windows	20
Client Installation and Connectivity Guide	21
Administrator's Guide	21
Query Performance Guide	25
SQL Reference Guide	26
Table Management Utility Reference Guide	32
RISQL Entry Tool and RISQL Reporter User's Guide	36
Messages and Codes Reference Guide	36



---

# RELEASE NOTES

---

## Important release information

IBM Red Brick Warehouse Version 6.20.xC6 is a new release that contains maintenance fixes only.

The version string TC6 refers to the Windows platform. The strings UC6 and FC6 refer to the UNIX platforms (32-bit and 64-bit, respectively). Throughout these release notes, V6.20.xC6 refers to all platforms generically.

These release notes contain information about all of the Red Brick products, including the Client Connector Pack. There is no longer a separate release note for the client products.

Before you use V6.20.xC6, read the following sections:

- New features in V6.20.xC5
- New features in V6.20.xC2
- New features in V6.20.xC1
- Significant operational considerations
- Installation and upgrade notes
- Known issues
- Documentation notes
- UNIX operating-system parameters

---

## Customer support

If you have technical questions about IBM Red Brick Warehouse but cannot find the answer in the appropriate document, contact IBM Customer Support as follows:

**Telephone**      1-800-274-8184 or 1-913-492-2086  
(7 a.m. to 7 p.m. central time, Monday through Friday)

**Internet access**  
[www-3.ibm.com/software/data/informix/support/](http://www-3.ibm.com/software/data/informix/support/)

---

## New features in V6.20.xC5

IBM Red Brick Warehouse V6.20.xC5 includes the following enhancement:

- **Remote TMU support for alternative network connectivity on client machine**—for details, see page 32.

---

## New features in V6.20.xC2

IBM Red Brick Warehouse V6.20.xC2 includes the following enhancements:

- **Expression support added in the GROUP BY clause**—In addition to column names and aliases, you can use expressions in the GROUP BY clause. For details, see page 27.

- **String delimiter support in the TMU**—The TMU supports string delimiters, allowing data exported in delimited format from IBM DB2 UDB databases to be directly loaded into Red Brick tables. For details, see page 33.
- **REORG and SYNCH support added for Remote TMU**—You can use the rb\_ctmu program to run the following TMU operations:
  - LOAD DATA
  - UNLOAD
  - GENERATE CREATE TABLE
  - GENERATE LOAD DATA
  - REORG
  - SYNCH

For details about Remote TMU operations, see the *Table Management Utility Reference Guide*.

- **Creator qualifier for tables, views, and synonyms**—Users can create tables, synonyms, and views with an optional creator qualifier that enables third-party tools to work with Red Brick objects. For a list of SQL statements that support the creator qualifier, see page 26.

---

## New features in V6.20.xC1

IBM Red Brick Warehouse V6.20.xC1 contains a large number of new features and enhancements. The following list briefly identifies each feature and indicates which document to consult for more information.

### Query execution and optimization:

- **Query profiling and monitoring**—The Query Performance Monitor allows administrators, performance analysts, and capacity planners to profile a query as it executes, watch the profile develop, and save the profile so it can be used in before-and-after studies. The profile consists of detailed statistics captured at the operator level, such as the amount of processor time each operator uses, the number of physical read and write operations, and the number of spills to disk. (*Query Performance Guide, SQL Reference Guide*).
- **SQL OLAP functions**—The server supports all of the functions defined by the SQL standard's "Elementary OLAP" package, including RANK, DENSE\_RANK, NTILE, ROW\_NUMBER, RATIO\_TO\_REPORT, and the OLAP aggregation functions. Window partitions and window frames can be defined to accommodate a wide range of analytical business questions. Equivalent RISQL display functions continue to be supported (*SQL Reference Guide*).
- **Vista rewrite and Advisor extensions**—The precomputed view query rewrite system rewrites two additional classes of queries:
  - Queries that group by complex expressions
  - Queries that contain COUNT(DISTINCT), SUM(DISTINCT), MIN, or MAX functions whose input columns are defined in the precomputed view's GROUP BY clause and select list

The Advisor is aware of these rewrite capabilities and generates candidate views accordingly. To optimize candidate view analyses, this release also provides a configuration parameter and SET command for limiting the number of candidate views generated in a single analysis (*IBM Red Brick Vista User's Guide*).

- **TARGETjoin parallelism and segment elimination**—When TARGETjoin indexes are locally segmented, TARGETjoin queries benefit from parallelism in the join

phase of the query and unneeded segments can be eliminated from join processing (*SQL Reference Guide, Query Performance Guide*).

- **Local predicates in TARGETjoin queries**—When a query contains a local predicate on a TARGETjoined table, the predicate can be pushed down into the join operation (*SQL Reference Guide, Query Performance Guide*).
- **XML format for SQL EXPORT**—Query results can be exported to XML-format files. (*SQL Reference Guide*).

#### **TMU and PTMU enhancements** (*Table Management Utility Reference Guide*):

- **Backup and Restore operations**—The TMU performs full and incremental backups to disk, tape, and XBSA-compliant storage management systems. Backups can be performed when the database is available for read and write access. A detailed metadata history is maintained for all backups to support reliable and efficient restore operations.
- **Remote TMU**—DBAs can start LOAD DATA and other operations from a client machine, using local control files and input files. The PTMU runs on the remote server machine and returns its output files and discard files to the client.
- **XML load**—Tables can be loaded from files in XML format, based on paths in the TMU control file that map the data in the XML file to columns in the table.

#### **General administration features:**

- **Locally segmented indexes**—To facilitate the maintenance of time-cyclic databases, DBAs can segment TARGET and B-TREE indexes like the data even when the segmenting column of the table is not in the index. You create local indexes with the CREATE INDEX...SEGMENT LOCAL option and attach local indexes with the ALTER SEGMENT...RANGE LIKE SEGMENT option (*Administrator's Guide, SQL Reference Guide*).
- **Segment DDL enhancements** (*Administrator's Guide, SQL Reference Guide*)
  - Segment STAR indexes more easily with the CREATE STAR INDEX...SEGMENT LIKE DATA and ALTER SEGMENT ATTACH...RANGE LIKE SEGMENT options.
  - Release space after rows have been deleted with the ALTER SEGMENT...RELEASE STORAGE option.
  - Remove unused PSUs from the end of a segment with the ALTER SEGMENT...DROP LAST STORAGE option.
  - Change the position of a segment in a table without having to detach the segment with the ALTER SEGMENT...RANGE MOVE option.
  - Avoid unintentional invalidation of indexes; the INVALIDATE keyword is required for the ALTER SEGMENT...DETACH and RANGE MOVE options.
- **System catalog enhancements**—Improved catalog access time, memory usage, and use of database locks. A new system table, RBW\_VIEW\_REFERENCES, identifies the tables referenced in a view (*Administrator's Guide*).

**Client features:**

- **ODBC Driver support for scalar functions**—The Red Brick ODBC Driver supports additional numeric, datetime, and string scalar functions (*Client Installation and Connectivity Guide*).
- **Connectivity to OLE DB applications**—The Red Brick ODBC Driver supports Windows OLE DB applications through the Microsoft OLE DB-to-ODBC bridge. OLE DB-based applications can connect to Red Brick databases, including applications built with the following Microsoft products:
  - ActiveX Data Objects (ADO) and Active Server Pages (ASP)
  - Microsoft SQL Server Data Transformation Services (DTS) and Analysis Services
- **Certification of IBM HTTP Server with Red Brick JDBC**—The IBM HTTP Server is supported as the middle tier in a three-tier JDBC configuration (*Client Installation and Connectivity Guide*).
- **JDBC tools**—The Red Brick JDBC Driver installation includes two graphical tools: the Data Source Administrator and the JDBC Ping tool. These tools have been enhanced and documented for this release (*Client Installation and Connectivity Guide*).
- **Administrator tool enhancements**—Support for the new Segment DDL and local index features is available through the GUI, as well as the ability to start and stop the Performance Monitor daemon and set the maximum number of candidate views for Advisor candidate analyses.

---

## Supported Windows platforms

IBM Red Brick Warehouse V6.20.xC6 is supported on the following Windows platforms:

Operating system	Service packs
Windows NT 4.0	SP6A
Windows 2000 Server	SP4

The Windows components of Client Connector Pack V2.20.6 are supported on the following platforms:

Operating system	Service packs
Windows 98	
Windows NT 4.0	SP6A
Windows 2000 Professional	SP4
Windows XP Professional	SP1

IBM Red Brick Warehouse Administrator V2.20.5 is supported on the following platforms:

Operating system	Service packs
Windows NT 4.0	SP6A
Windows 2000 Professional	SP4
Windows XP Professional	SP1

Do not use earlier service packs than those listed.



---

## ODBC Driver compatibility

The following table describes the backward compatibility of UNIX and Windows Red Brick ODBC Drivers with IBM Red Brick Warehouse databases:

Windows ODBC Driver	UNIX ODBC Driver	Is compatible with:
V5.01 (Client Connector Pack V2.0.x)	V6.01	IBM Red Brick Warehouse V6.0.x, V6.10.x, V6.11.x, and V6.20.x databases
V5.02.x (Client Connector Pack V2.10.x and 2.11.x)	V6.02.x	
V5.62.x (Client Connector Pack V2.20.x)	V6.62.x	

If you use a V5.01/6.01 or earlier ODBC Driver with a V6.10.x or later database, you will not be able to define a column with the SERIAL data type.

The ODBC Drivers and ODBClib SDKs for UNIX and Linux platforms are installed from the server directory (RBW), not the Client Connector Pack directory (CCP). The Client Connector Pack also includes Windows versions of the RISQL Entry Tool, RISQL Reporter, and Client TMU applications. For installation instructions and more information about these components, see the *Client Installation and Connectivity Guide*.

For more information about the Red Brick ODBC Drivers on all platforms, see the *Client Installation and Connectivity Guide*.

---

## JDBC Driver requirements

IBM Red Brick Warehouse Version 6.20.xC6 supports JDK 1.2 and 1.3 (J2SE 1.2 and J2SE 1.3). Alternatively, you can install the Java Runtime Environment (JRE) Version 1.2 or 1.3. The JRE package is smaller and is sufficient for running Red Brick JDBC applications.

The following jar files are required for running the Red Brick JDBC Driver:

<b>redbrick.jar</b>	Red Brick JDBC Driver.
<b>jndi.jar</b>	The JNDI API; required only if you are working with Datasources.
<b>jdbc2_0-stdext.jar</b>	The JDBC 2.0 Standard Extension; required only if you are working with Datasources.
<b>fscontext.jar or ldap.jar</b>	File System Context Service Provider jar file or LDAP Service Provider jar file; required only if you are working with Datasources.
<b>providerutil.jar</b>	This file is required by fscontext.jar and ldap.jar (and is downloaded with these jar files).

Make sure the required jar files are set in your CLASSPATH environment variable. Note how these files are bundled for different Java versions:

- jndi.jar and ldap.jar are included in J2SE 1.3, not J2SE 1.2.
- jdbc2\_0-stdext.jar is not included in J2SE 1.2 or J2SE 1.3. This file can be downloaded separately from:

<http://java.sun.com/products/jdbc/download.html>

- The fscontext.jar is not included in J2SE 1.2 or J2SE 1.3, but it can be downloaded as part of the “JNDI 1.2.1 & More” package from:  
<http://java.sun.com/products/jndi/#DOWNLOAD12>

---

## Significant operational considerations

Before you use IBM Red Brick Warehouse V6.20.xC6, note the following operational considerations and changes.

### Result data type for string functions

The result data type for the following scalar string functions has changed from CHAR to VARCHAR. You might need to change your ODBC and JDBC applications to accept VARCHAR data returned from these string functions.

- CONCAT
- LTRIM
- RIGHT
- RTRIM
- SUBSTR
- SUBSTRB
- TRIM

### Red Brick Data Mine not supported

Data mining with the Red Brick Data Mine feature is not supported in Version 6.20. Models created with previous versions of IBM Red Brick Warehouse must be dropped before databases can be upgraded.

### Performance of OLAP functions

Queries that contain OLAP functions often benefit from automatic performance optimizations that are not implemented for equivalent RSQL functions. For new applications, IBM advises the use of OLAP functions in all queries. For existing applications, you should consider updating your queries to use OLAP functions at your convenience. For detailed information about OLAP and RSQL functions, see the *SQL Reference Guide*.

### Trailing spaces for CHAR columns preserved in query results

In V6.20.xC1, trailing spaces in fixed-width CHAR columns are preserved in query results. In previous releases of IBM Red Brick Warehouse, these trailing spaces were always trimmed. The new behavior is consistent with the ANSI SQL standard and applies to both ODBC and JDBC applications.

### Additional memory allocated for large indexes

Extra block cache space is now reserved for extremely large indexes. This extra reserved memory is allocated from the query memory limit and will only be used in order to scan large indexes. To have the same effective memory limit as previous releases, you must increase your query memory limit by at least 100KB.

A good indicator that there is not enough memory allocated, besides longer query response time, is increased spilling. Increasing the query memory limit should reduce spilling and return query response time to its previous level.

## RANGE MOVE clause for ALTER SEGMENT

A data segment must be empty before you can move its range with the ALTER SEGMENT...RANGE MOVE command.

To reuse a data segment for time-cyclic data, you must execute an ALTER SEGMENT...CLEAR command before the RANGE MOVE operation. For the procedure, see page 22.

---

## Installation and upgrade notes

The following sections contain notes on installing the products you have purchased and upgrading existing databases. After you have read these notes, see the *Installation and Configuration Guide* for standard installation and upgrade procedures.

### For the first time

If you are a new user, follow the procedure for a first-time installation described in the *Installation and Configuration Guide*. This procedure is designed for sites not currently using IBM Red Brick Warehouse.

### With an existing warehouse

If you are currently using an earlier version of the Red Brick server, read the information in the *Installation and Configuration Guide* under “Running Two Releases of IBM Red Brick Warehouse.” This section describes how to allow users to continue working with the current database and software while you install the new version in a separate directory. You can then test the new version to see how the changes will affect your warehouse implementation before you upgrade your production databases.

IBM recommends this procedure whenever the changes between releases are substantial or if an upgrade is required. A database cannot be used with an earlier release after it has been upgraded to a newer one.

#### Upgrade paths:

The V6.20.xC6 release requires an upgrade of existing databases to accommodate changes to the system catalog. You can upgrade any V6.0.x or V6.1x database directly to V6.20.xC6. To upgrade from V5.1x, you must upgrade to V6.0.x or V6.1x first.

#### Before upgrading, remove Data Mine models, drop backup data and version log:

1. The Red Brick Data Mine product introduced in Red Brick Warehouse V5.1 is no longer supported. Before upgrading a database that might contain data-mining models, run the following query:

```
select name from rbw_tables where type = 'MODEL';
```

If any models exist, use DROP MODEL statements to remove them before attempting the upgrade.

2. The TMU backup and restore features have been significantly changed and enhanced for this release. Before upgrading a database that contains a backup segment, drop the backup data first:

```
alter database drop backup data;
```

3. If you are upgrading a versioned database, the version log must be cleaned and dropped:

```
alter database clean version log;
alter database drop version log;
```

If you are not sure whether your database contains a backup segment or a version log segment, run the following query:

```
select name from rbw_segments where usage = 'BACKUP_DATA' or usage = 'VERSION_LOG';
```

4. Finally, make sure there are no pending ALTER TABLE operations or offline loads:

```
select name from rbw_segments where insynch = 'N';
```

After the upgrade, TMU backups taken with a server version prior to V6.20.xC1 will not be compatible with the V6.20.xC6 database. A new level 0 backup should be performed immediately.

### Selecting a code set during a Windows installation:

When you install IBM Red Brick Warehouse on Windows platforms and specify a locale, the default code set for the language and territory you choose appears in the list box. Other valid code sets can be selected from *above and below* the default selection. For example, if you select SimplifiedChinese and China as your language and territory, the default code set in the list box is MS936. The other valid choices are UTF-8 and MS54936, but only UTF-8 is displayed in the list. To select MS54936, click the up arrow next to the default selection.

### Administrator tool installation and compatibility:

Before installing IBM Red Brick Warehouse Administrator V2.20.5, be sure to uninstall the previous version of the tool. To do this, select the Uninstaller utility for IBM Red Brick Warehouse Administrator from the Programs menu.

You can use the current Administrator tool (V2.20.5) with IBM Red Brick Warehouse V6.0.x and later databases.

---

## Fixed issues in V6.20.x

This section describes issues reported by customers and fixed in IBM Red Brick Warehouse V6.20.xC2 through V6.20.xC6. The CSL number is the Customer Support Log number assigned to each case.

This section is organized as follows:

- Database administration issues
- Query-related issues
- Connectivity issues
- Administrator tool issues
- Documentation issues

### Database administration issues

- CSL #9345

An offline load resulted in a segmentation fault when the working space segment did not exist. In V6.20.xC2 and later versions, an appropriate error message is displayed when problems occur with the working space segment.

- CSL #15975  
CREATE TABLE, DROP VIEW, and other statements did not accept a table name qualified with the owner ID. See page 26 for details about support for owner qualifiers.
- CSL #16324  
When precomputed view maintenance failed for some reason, final load statistics for the detail table were not reported in either the log file or the RBW\_LOADINFO system table.
- CSL #324113  
When a non-existent table was referenced in a TMU control file, message 412 was issued. In V6.20.xC2 and later versions, a more appropriate message is used.
- CSL #326422  
When an SQL EXPORT statement contained a TO\_CHAR function and the client and server locales were set to different code pages, the exported file contained extra characters at the end of the TO\_CHAR column.
- CSL #328278  
CHECK TABLE reported different results for a table with multiple segments and a table with a single segment. The CHECK TABLE with the multiple segments reported no inconsistencies, but a segment was still damaged until the ALTER SEGMENT VERIFY command was run.
- CSL #330538, CSL #331159  
When a telnet connection was made to the TCP/IP port used for JDBC applications (server port + 1), the rbwapid daemon process terminated.
- CSL #331330, 940448  
New-format tables did not reuse deleted rows when new rows were inserted via SQL statements or TMU loads.
- CSL #333664  
When a user session was running a query or update against a temporary table and the session was terminated with an ALTER system command, the session's memory was not freed. When many sessions were terminated in this way, the lack of resource cleanup resulted in an operating-system read error.
- CSL #334799  
When the first segment of a table with a synonym was dropped, the server returned a fatal error.
- CSL #336623  
The PTMU discarded all the rows from an input file except the last one when the data contained illegal binary zeroes. In V6.20.xC2 and later versions, the presence of these characters in an input file results in an error and the load terminates.
- CSL #342268  
In Red Brick Warehouse Version 6.20.xC1, the performance of reading the system catalog could be degraded for a large catalog on an extremely busy system. This situation affected the speed of logins.
- CSL #342714  
The EXPORT command failed with a fatal error when the output was piped and parallelism was enabled.
- CSL #344619  
When a DML or DDL statement was executed on a table with the letters "gmk" or "rsq" in the table name, a segmentation fault (on UNIX) or General Protection Fault (on Windows) occurred.

- CSL #345406  
When an INSERT, UPDATE, or DELETE statement was executed, the STATE column of the DST\_COMMANDS system table showed the value EXECUTING instead of the number of rows inserted, updated, or deleted.
- CSL #357821  
A TMU backup of a large database failed when it tried to update the action log at the end of the operation.
- CSL #358885  
When the server and client were using different locales and the client locale's multibyte representation was larger than that of the server locale, incomplete character data was sometimes sent to the client.
- CSL #361625  
When a CHECK INDEX command was run on a large dimension table and INDEX\_TEMPSPACE\_MAXSPILLSIZE was set to a very high value, a segmentation fault (on UNIX) or General Protection Fault (on Windows) occurred.
- CSL #365262  
When an ALTER SYSTEM START PERFORMANCE MONITOR command was issued by a user without sufficient authority, subsequent attempts to start the performance monitor by authorized users also failed.
- CSL #365826  
When a pseudocolumn in a TMU control file referenced a decimal field type with a restricted date mask, the load failed with a fatal TMU error.
- CSL #367862  
An ALTER TABLE...ADD CONSTRAINT statement failed with a 10410 error when the foreign-key constraint to be added contained a CHAR column and the locale sort order was set to Binary.
- CSL #368288  
An ALTER SEGMENT...DETACH operation with the OVERRIDE FULLINDEXCHECK option sometimes failed during the index validation stage.
- CSL #370318  
An ALTER SEGMENT...ATTACH command for a VARCHAR composite-key index segment returned a fatal error.
- CSL #370980  
Queries and TMU UNLOAD operations returned inconsistent results when processing identical WHERE clause constraints on DATE and TIMESTAMP data. The UNLOAD processing has been fixed to match the query processing.
- CSL #371197  
Linguistic string comparisons returned incorrect results because trailing blanks were processed incorrectly.
- CSL #371893  
When a user attempted to log into RISQL (risql.exe) with an incorrect password or no password, an internal 132 error was sent to the active log file, as well as the standard error 145.
- CSL #373180  
When two tables were equi-joined over character columns, the joining column in the first table was larger than the joining column in the second, and the second joining column was a primary key, the query sometimes failed with an internal 132 error.

- CSL #373199  
An ALTER TABLE...ADD COLUMN...IN\_PLACE operation required excessive space and could not be resumed after space was added to the segment.
- CSL #374306  
OPTIMIZE ON loads in REPLACE mode behaved inconsistently in Versions 6.11 and 6.20 when indexes were in an INVALID state. Under the same conditions in Version 6.20.xC6, OPTIMIZE ON loads start building indexes in TURBO mode, then switch to OPTIMIZE ON when out-of-order rows are processed.
- CSL #375419  
The CHECK INDEX command incorrectly reported damaged segments for non-segmented B-TREE and TARGET indexes that had been upgraded from Version 6.1x to Version 6.20.
- CSL #376839  
Timing problems caused an internal error when the version log became full during a load with a large number of indexes.
- CSL #378888  
For Thai locales (and certain other locales), special characters that should have been sorted in combination with other characters were sorted incorrectly.
- CSL #380314  
On Windows platforms, when the target storage device or file system became full during an EXPORT operation, the EXPORT output file remained locked and could not be deleted unless the Red Brick service was stopped.
- CSL #382092  
A non-DBA user who created a segment with multiple PSUs could see only the first PSU when querying the RBW\_STORAGE system table.
- CSL #824224  
The TMU ignored options in the control file when there was no space between the option letters and their specific values. In V6.20.xC2 and later versions, control file options and their values must be separated with spaces.
- CSL #884642  
Message 10428 was not returned when an attempted RISQL connection exceeded the value of the MAX\_SERVERS parameter.
- CSL #987834  
If a TMU input record violates referential integrity for more than one table, the generic discard file can contain this record multiple times, one for each referenced table that does not have a discard file specified for it in the RI\_DISCARDFILE clause. In V6.20.xC6, the record is discarded only once in the generic discard file.

## Query-related issues

- CSL #I11557  
In a RISQL session, the SET COLUMN... NO REPEAT command did not repeat columns that had the same first few characters as the previous row, but different last characters. For example, rows with column values "css11" and "css11by" were grouped together under "css11," but "css11by" did not appear.
- CSL #I16928  
Queries with parallel fact-to-fact STARjoins sometimes returned incorrect results when the join predicate was on the leading common dimensions, the query constrained on the leading dimensions, and there were no constraints on the trailing uncommon dimensions. As a result, you might see more I/O operations

than in versions prior to Version 6.20.xC4. Depending on how many more rows will be fetched and how the rows are cached, the elapsed time of the query can also be longer. If you experience longer query response times, increase the number of parallel fetch tasks, if possible, and increase the value of the QUERYPROCS parameter to improve performance.

- CSL #I17087  
When a query contained a simple, single constraint on a column with a local TARGET index and the table had an empty segment that was not the first segment in the table, the query returned the wrong results. An example of a simple constraint is a column equal to a specific value.
- CSL #255283  
When a query executing in one RSQL session was cancelled by an ALTER SYSTEM CANCEL USER COMMAND in a second RSQL session, subsequent queries in the first session failed.
- CSL #336518  
A TARGETjoin was chosen over a STARjoin when a TARGET index was defined on a pre-loaded dimension table, but the query performed faster when the TARGET index was not available and a STARjoin was executed. In V6.20.xC5 and later versions, a STARjoin is chosen over a TARGETjoin when it is the better choice.
- CSL #342925  
A query that joined a fact table to the same instance of a dimension table across multiple foreign keys returned incorrect results when it used a STARjoin, but returned the correct results when a table scan was used.
- CSL #343277  
Shared memory was corrupted when two RSQL sessions were using the Query Performance Monitor concurrently and monitoring was turned off in both sessions.
- CSL #343491  
When a query used the LIKE predicate with a CONCAT function of all literal operands, the query returned incorrect results.
- CSL #344459  
Preliminary plans that included joins of dimension tables to non-related tables or subqueries might not have performed well. In V6.20.xC5 and later versions, you can set PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS to ON to exclude tables not joined by referential integrity relationships from a STARjoin preliminary plan. For details about this SET statement, see page 30.
- CSL #346786  
When the first table reference in the FROM clause was a subquery or a view that contained a CASE expression, the query returned incorrect results.
- CSL #348723  
If a TARGET index was defined for a column on which a query was constrained and the data type of the constraint did not match the data type of the indexed column, the query returned incorrect results.
- CSL #351761  
When a query contained the same subquery more than once and the subquery used a STARjoin or table scan plan, the query returned an internal error.
- CSL #352300  
A query failed with an internal error because two literals appearing in two different subqueries were treated equally.



- CSL #352919  
When a query contained multiple DISTINCT aggregation functions and the operand of one of the aggregations was in a GROUP BY expression, the query returned wrong results.
- CSL #364610, 368360  
Certain queries with very large IN lists ran successfully with Version 6.11 but failed with Version 6.2.
- CSL #367887  
When the QUERY MEMORY LIMIT was set to a very high value, certain queries encountered performance problems because of long compile times; as a result, other users were unable to log in because of system table locks.
- CSL #368175, 379400  
A query that contained pre-run subquery choose plans (for pre-running non-correlated and non-scalar subqueries) returned a fatal 132 error when Exchange operators (for parallelism) were added to the query plan.
- CSL #371958  
When a query selected from a view with a correlated subquery in the select list and an aggregate optimization was applied to a function in the select list, the query failed with an internal error.

## Connectivity issues

- CSL #310326  
An attempt to compile and run the sample ODBC program rb\_client with an invalid client locale caused a segmentation fault. In V6.20.xC2 and later versions, an invalid locale setting results in an appropriate error message.
- CSL #323898  
An attempt to create a SYSTEM type ODBC data source name (DSN) during installation of the Client Connector Pack resulted in a USER type DSN.
- CSL #334638  
When the application used the ODBC function SQLSetDescField with the SQL\_DESC\_ARRAY\_SIZE option (or function SQLSetStmtAttr with the SQL\_PARAMSET\_SIZE option or SQLParamOptions) to insert multiple rows, only one row was inserted. The Red Brick ODBC Driver now returns an error message to indicate that only one row at a time can be inserted.
- CSL #338050  
The *Client Installation and Connectivity Guide* implies that the Red Brick JDBC Driver cannot be installed on the same Windows machine as the IBM Red Brick Warehouse server. These products can be installed on the same machine; see page 21.
- CSL #344684  
When multiple ODBC threads were executing concurrently and many different error messages were generated at the same time, the Red Brick ODBC Driver would periodically hang with a very high CPU utilization.
- CSL #353789  
An enhancement to the rb\_ctmu program was requested to allow an alternative network path to be specified for sending data back to the client machine. In V6.20.xC5, a new **-b** option can be specified on the command line. For details, see page 32.  
During an unload operation, the rb\_ctmu program generated messages that were misleading and not as informative as the rb\_tmu or rb\_ptmu output. In

particular, the progress of the unload was not reported accurately by the rb\_ctmu, giving the impression that many more rows had been successfully unloaded to the client machine than was actually the case. In V6.20.xC5, the rb\_ctmu messages are more descriptive and accurately state the progress of the unload operation.

- CSL #357636  
A performance enhancement was requested for the rb\_ctmu program. In V6.20.xC5 and later versions, the performance of rb\_ctmu has been improved.

## Administrator tool issues

- CSL #15995  
The Show DDL option produced the GRANT and CREATE ROLE statements before the CREATE TABLE statements even though the DDL mode was set to Logical Order. When the generated DDL was executed, the GRANT and CREATE ROLE statements failed because the tables did not exist.
- CSL #313008  
The Size Wizard estimated a different size for the primary key index and a multicolumn B-TREE index than the size that dbsize estimated.
- CSL #322584  
In the ISQL window, the second SQL statement was ignored when the first statement included multiple comments.
- CSL #325585  
The Show DDL option produced only one CREATE TARGET INDEX statement for a table that has 19 TARGET indexes.
- CSL #328731  
The Show DDL option produced incorrect syntax for default values of date columns in CREATE TABLE statements.
- CSL #330772  
The labels of the radio buttons were incorrect on the Relationships Tab when a table name was double-clicked in the database tree of the Japanese Administrator Tool. **Referenced** should be on the left radio button and **Referencing** should be on the right.
- CSL #338021  
The Show DDL function at the Indexes level of the database tree did not produce CREATE statements for alternate STAR indexes on the selected table. In V2.20.5, all STAR indexes appear in the Show DDL output, including STAR indexes built over the primary-key columns of the table.
- CSL #338891  
The Alter Segment Wizard incorrectly generated the OF TABLE keywords for the ALTER SEGMENT statement after a table segment is first selected, the **Back** button is pressed, and then an index segment is selected.
- CSL #339857  
Queries from the ISQL window returned incorrect results if the column was created as a decimal or numeric column with a scale of 0. For example, the following query in the ISQL window resulted in a decimal point preceding the actual number of PSUs:  

```
select sum(npsus) from rbw_segments;
```
- CSL #354642  
The text of the **Save As** button on the Show DDL window of the Japanese version of the Administrator tool was truncated.

- CSL #835338, 947483  
The Print command failed to print multiple pages of database tree information; only the first page could be printed.
- CSL #994636  
When the output of a Show DDL command was used to create a new database, the tables were stored in the system catalog with mixed-case names instead of all uppercase.

## Documentation issues

- CSL #312551  
Segmenting by hash did not result in an even distribution of the data. For guidelines to segment by hash, see pages 22 and 30.
- CSL #334789  
The NODE\_NAME column was not documented in the description of DST\_SESSIONS. For details, see page 24.
- CSL #344447  
A documented workaround for displaying Japanese Gaiji characters no longer applied. This workaround has been removed from the release notes.
- CSL #345703, 348446, 370988  
The 6.2 Japanese versions of the *Administrator's Guide* and the *Table Management Utility Reference Guide* contain some translation errors. See page 20.
- CSL #355455  
The maximum setting of the EXPORT\_MAX\_FILE\_SIZE parameter was not documented. See page 30.
- CSL #357351  
The procedure for starting the rbwlogd daemon without stopping and restarting the rbwapid daemon was not clearly documented. See page 24.
- CSL #358950  
The description of the SQL EXPORT command in the *SQL Reference Guide* implies, incorrectly, that an EXPORT statement is always faster than a SELECT statement with the output directed to a file.
- CSL #362897  
The *Administrator's Guide* contained an incorrect cross-reference for information about the RB\_DSN environment variable. See page 21.
- CSL #369833  
An ALTER SEGMENT...MIGRATE statement failed because a macro was defined with the name "PSUS," which is a positional keyword. Positional keywords cannot be used as macro names, but this restriction was not documented.
- CSL #372044  
The interaction of the CONCAT function with the TRIM functions is not documented correctly in the *SQL Reference Guide*. See page 27.

---

## Known issues

This section describes known issues in IBM Red Brick Warehouse V6.20.xC6. This section is organized as follows:

- Database administration issues
- Query-related issues
- Connectivity issues

- Administrator tool issues

## Database administration issues

### Backups to Tivoli Storage Manager not supported on HP-UX platforms:

On HP-UX platforms, you cannot use the XBSA interface to run TMU backups to Tivoli Storage Manager (TSM).

### TMU UNLOAD not supported for tables with more than $2^{31} - 1$ rows:

A TMU UNLOAD operation can unload a maximum of 2,147,483,647 rows ( $2^{31} - 1$ ) but cannot unload any rows from a table that contains more than this number of rows. If the table you want to unload contains more than  $2^{31} - 1$  rows, the UNLOAD operation against the table will fail, regardless of the number of rows you try to unload. To unload rows from very large tables, use the SQL EXPORT command, as documented in the *SQL Reference Guide*.

### Leaving performance monitoring on in the Administrator tool can fill Performance DSTs:

If you are using the Administrator tool to profile queries, enter the SET PERFORMANCE MONITOR OFF command as soon as the queries have executed. If it is left on, the Query Performance Monitor profiles every catalog query that the Administrator tool submits to perform administrative functions, and these unnecessary statistics are stored in the Performance DSTs.

### Performance-monitoring session and command IDs are not displayed until query processing is complete:

Message 9107 is intended to help users identify the queries they are profiling by displaying a session ID and command ID for each query as soon as it is submitted. However, this message is not displayed until the first row of the result set is returned. There are two solutions to this problem:

- Query the DST\_PERFORMANCE\_COMMANDS table to obtain the session ID and command ID.
- Set unique “user comments” for each query you intend to profile.

### XML namespaces:

XML namespaces are not supported in TMU load operations:

- You cannot specify namespaces in XML paths inside TMU control files. For example, the following XML path will return a syntax error:

```
/Root/PU/myns:A/#PCDATA
```

where myns: is a namespace.

- XML namespace prefixes are allowed in XML input files, but they are ignored during parsing. For example, the tag <myns:title> is parsed as <title>.
- Two identical tags from different namespaces are assumed to be the same (whether they have default or explicitly named prefixes).  
For example, <ns1:title> and <ns2:title> are regarded as the same title, and <ns1:title> and <title> are regarded as the same title.

### Processing double-byte space characters:

East Asian languages, such as Chinese and Japanese, support a double-byte space character (for example, 0xa1a1 in Chinese). Using this double-byte space, as opposed to its single-byte ASCII counterpart (0x20), can cause unpredictable results with character-string comparisons, TRIM function processing, datetime string processing, and so on. Where possible, substitute double-byte spaces within strings with single-byte spaces, and trim any trailing double-byte spaces from the input file before you load the data.

#### **Double-byte characters in datetime input data for TMU loads:**

For datetime fields, the TMU does not recognize Traditional and Simplified Chinese double-byte characters, except for the double-byte designations of “a.m.” and “p.m.” If you are loading Chinese data into a date or time field, define the appropriate mask to suppress the double-byte data. Within date and time masks, each underscore character represents a single byte that you intend to mask out; therefore, you must specify the exact number of underscores to match the number of bytes you want to suppress. For example, a double-byte representation of a year value requires two underscores in the mask.

For detailed information about defining date and time masks, see Chapter 3 of the *Table Management Utility Reference Guide*.

#### **ALTER TABLE statement might run out of space:**

When VARCHAR columns are involved, it is not possible to calculate in advance the exact space requirements of an ALTER TABLE statement. IBM recommends that you always perform a backup of a table before altering it, in case the statement fails because it runs out of space.

## **Query-related issues**

#### **Reprojection of subqueries:**

Incorrect results can occur if a select-list subquery alias is projected from another subquery, or if the same expression containing only literals or correlation references is projected from two different subqueries. Errors can also result from a query that groups by a select-list subquery alias that is also projected from another subquery.

## **Connectivity issues**

#### **Red Brick ODBC Driver behavior with bulk inserts:**

The Red Brick ODBC Driver does not support bulk inserts. If `SQLParamOptions()` is called to set the bulk-insert array size, the ODBC driver returns an error. However, some applications, such as IBM DB2 Warehouse Manager, ignore this unexpected error and continue to send insert rows. The net effect is that only one row is inserted.

#### **Incomplete display of long ConnectionPoolDataSource names:**

In the Add and Edit dialog boxes for PooledDataSources in the JDBC Data Source Administrator tool (rbdsadmin.\*), long ConnectionPoolDataSource names are not fully displayed. The pop-up menu does not resize itself to accommodate long names.

**JDBC CONVERT function:**

The JDBC CONVERT function currently supports the following nonstandard type names: SQL\_CHAR, SQL\_VARCHAR, SQL\_REAL, SQL\_INTEGER, SQL\_FLOAT, SQL\_TIMESTAMP, SQL\_DECIMAL, SQL\_DATE, and SQL\_TIME. In the next major release, only standard types (without the SQL\_ prefix) will be supported.

**Microsoft SQL Server Data Transformation Services (DTS)—loading data with the TMU:**

DTS loads data into ODBC data sources using INSERT statements. These statements load one row at a time, and the performance is very poor. IBM recommends that DTS users run TMU LOAD DATA operations to load the data, then process the cube using Analysis Services.

**Microsoft SQL Server Analysis Services—incorrect DDL generated for Day option:**

Incorrect DDL is generated when the Day option is selected in the Create the Time Dimension Levels wizard (for example: Year, Month, Day) and the cube is processed with ROLAP data storage. To solve this problem, manually correct the DDL before it is executed.

**Processing cubes with Microsoft SQL Server Analysis Services:**

When you use Analysis Services to build a cube from tables in an IBM Red Brick Warehouse database, you can choose MOLAP (the default), HOLAP, or ROLAP for data storage. To use the ROLAP option, you must add the following parameter to your rbodbc32.ini file:

```
TXN_CAPABLE_ALL=MMC.EXE
```

where mmc.exe is the application program name for Microsoft SQL Server 2000 Analysis Services. When this parameter is set, SQLGetInfo(SQL\_TXN\_CAPABLE) returns SQL\_TC\_ALL for Analysis Services.

**TimeZone returns unexpected results with JDK 1.2:**

The JDBC 2.0 TimeZone function might return unexpected results when it is used in Java applications compiled with JDK 1.2.

**ODBC/JDBC Drivers incorrectly report query cancellation:**

The ODBC and JDBC drivers sometimes return a “Statement Cancelled” message despite the fact that the server has successfully executed the query. This timing problem only occurs when the cancel request is sent to the server immediately after the server has sent its OK status to the client and the client has yet to acknowledge that status.

**JDBC date string results:**

Because of a problem in the JavaSoft JDK, the JDBC Driver does not return an error for invalid date strings in getDate, getTimestamp, or getTime commands. For example, getDate(col('9999-12-33')) returns the date 10000-01-02 and getDate(col('1999-02-29')), a date that does not exist, returns the date 1999-03-01.

## Administrator tool issues

### Alter System wizard does not display list of users:

If you attempt to change the user priority via the Manage System wizard, the User Priority screen does not list any users. This problem arises when you use the following procedure:

1. From the Manage menu, select **System**.
2. Select **User Priority** and click **Next**.
3. Click the User drop down list.

At this time, there is no list of Users.

You can use the following procedure to change the user priority:

1. From the Manage menu, select **Users**.
2. Select **Alter User** and click **Next**.
3. Select a user to alter and click **Next**.

The Edit the Comment and Priority screen appears.

### Japanese character display in Advisor analyses:

When you use the Vista Advisor to run a Precomputed View Utilization Analysis, Japanese characters in user-defined object names might not be displayed correctly.

### Refresh All to see changes to the database:

After performing operations within the Administrator tool, you have to select **Refresh All** from the View menu in order to see the results of those operations. The tool does not update its view of the database automatically.

---

## Documentation notes

This section describes changes to the documentation and online help for IBM Red Brick Warehouse V6.20.

### IBM Red Brick Warehouse library

The Documentation CD contains all of the documents that have been updated for V6.20. Please note the following changes to the document set.

The following document is new for V6.20:

- *Query Performance Guide*

This document describes the determinants of query performance and shows how to tune a database for optimal query performance. Examples show how to monitor the performance and resource utilization of queries with Red Brick tools: EXPLAIN, SET STATS, the Dynamic Statistics Tables, and the Query Performance Monitor.

This document inherits Chapters 10 and 11 from the *Administrator's Guide*, as well as other information pertinent to query performance and tuning.

The following titles are no longer part of the document set:

- *Explaining EXPLAIN User's Guide*

The information in this document has been updated and moved into the *Query Performance Guide*.

- *TMU Backup and Restore User's Guide*

The information in this document has been updated and moved into the *Table Management Utility Reference Guide*.

## Corrections to the Japanese documentation

This section notes errors that occur only in the Japanese translation.

### Correct the maximum number of tables in Chapter 1 of the Administrator's Guide:

The maximum number of tables was changed from 32,767 (Version 6.1) to 32,764 (Version 6.2) in the English version of the document. This change was not picked up in the translation into Japanese for Version 6.2. The correct number is 32,764.

### Correct the cross-reference on page 8-3 of the Administrator's Guide:

For information on moving data among database servers, see the description of the copy management utility, *rb\_cm*, in the *Table Management Utility Reference Guide*.

### Correct the REORG syntax on page 6-11 of the Table Management Utility Reference Guide:

The REORG keyword should not be repeated at the beginning of the command.

## Installation and Configuration Guide for Windows

### Add the following text to "Shutting Down the Service" on page 3-11:

You cannot stop the Red Brick database service if active connections exist. If you attempt to do so with the **rbwservice** command, the following message appears:  
Cannot stop service service\_name: There are still connections to the server.

If you attempt to stop the Red Brick service from the Control Panel, the following message appears in a service message box:

Could not stop the service\_name service on host\_name.  
Error 2140: An internal Windows NT error occurred.

Browse the Windows Event Viewer to view the specific Red Brick service message. In either method of stopping the Red Brick service, the Windows Event Viewer shows the following message:

Cannot stop the service. There are still connections to the server.

### Remove the description of file *rbllib.dll* and add the following file descriptions to the table of directories and files in the *redbrick* directory on page A-3:

\bin\MSVCRT.dll	Shared library for Microsoft Visual C Run-Time
\bin\MSVCRT40.dll	Shared library for Microsoft Visual C++ 4.1 C run-time, for backward compatibility
\_UNODBC.log	File that contains information to uninstall the ODBC Driver



## Client Installation and Connectivity Guide

**The Important paragraph on page 1-5 should read as follows:**

The Client Connector Pack ODBC utilities are not required and must not be installed on a Windows machine where the IBM Red Brick Warehouse server is already installed. These utilities are installed by default as part of the server installation. If the Client Connector Pack version of ODBC is installed on the same computer, client tools might stop responding because of conflicts between the DLLs installed with the Client Connector Pack and equivalent files installed with the server.

This kind of conflict cannot arise with the JDBC Driver, which can safely be installed on the same Windows machine as the server, following the installation procedure in Chapter 4.

**Add the following note under “Data Type Conversion Functions” on page 2-37:**

In addition to SQL\_DATE, SQL\_TIME, and SQL\_TIMESTAMP, the CONVERT function supports the ODBC 3.x types SQL\_TYPE\_DATE, SQL\_TYPE\_TIME, and SQL\_TYPE\_TIMESTAMP.

**Add the following information to the “JNDI LDAP Server” section on page 6-5:**

The JDBC tools support the LDAP server convention for naming data sources, where name/value pairs are separated by commas, and each name and value is separated by the equals sign. For example:

```
cn=ds0  
cn=ds1, ds=MaryDS  
cn=ds2, ou=pool, ou=jdbc
```

If an LDAP data source name consists of more than one name/value pair, as shown in the second and third examples, the corresponding LDAP directories (ds=MaryDS and ou=pool, ou=jdbc) must exist in the LDAP server; otherwise, the Data Source Administrator returns an error.

## Administrator’s Guide

**On page 2-25, update the description of RB\_DSN:**

RB\_DSN: Defines a data source name (DSN). For more information, see the *RISQL Entry Tool and RISQL Reporter User’s Guide* and the *Table Management Utility Reference Guide*.

**On page 2-35, remove the second scenario about a French-speaking user connecting to a database with a German locale:**

This kind of configuration is not supported.

**Replace the first paragraph and first bullet on page 4-32 with the following text:**

To estimate the size of a STAR index using the dbsize utility, you need to know the following information:

- For each table referenced by a foreign key in the STAR index: The estimated maximum number of rows per segment (MAXROWS PER SEGMENT) and the maximum number of segments for that table (MAXSEGMENTS).

**Replace the first paragraph after the two bullets on page 4-33 with the following paragraph:**

The size of a TARGET index is affected by many factors, particularly by the skew in the numbers of rows for each key, which can cause the estimated size of a TARGET index to be inaccurate. Because domain MEDIUM uses compression, the dbsize estimates are less likely to be consistent with the actual index size than the estimates made for domain LARGE and SMALL. When a domain is not specified, dbsize estimates the size with all three domains and suggests the domain that results in the smallest index.

**Replace the fourth bullet under the section “Rules for Segmenting” on page 5-13 with the following bullet:**

- Any table can reside in multiple segments, with data distributed by the data values or by a hashing algorithm. Use the hashing algorithm under the following conditions to obtain a more even distribution of data among segments:
  - The number of segments is a prime number.
  - Only columns defined with the NOT NULL clause exist in the row.
  - Large columns (such as REAL and character) whose values do not vary much among the rows are near the beginning of the table.

**Replace the procedure “To move the range to the new time period” on pages 9-61 to 9-64 with the following procedure:**

1. If you want to save the data, use the TMU to unload the segment to a file or tape.
2. Clear the old data segment to enable the range to move.  

```
RISQL> alter segment s_1q00 clear;
```

The CLEAR operation removes the data rows in the segment and all the index entries that reference the data rows. If any precomputed views exist on the Sales table and maintenance is turned on, the CLEAR operation rebuilds them.

3. Rename the old data segment, which will hold data for the second quarter of 2002.  

```
RISQL> alter segment s_1q00 rename s_2q02;
```
4. Rename the old index segments.  

```
RISQL> alter segment star_1q00 rename star_2q02;  
RISQL> alter segment target_mkt_1q00 rename target_mkt_2q02;
```
5. Make any other needed changes to the table and index segments. For example, change the maximum size or path of a PSU, or add a new PSU to the segment.
6. If you have not preloaded the leading dimension table, modify the Period table by adding rows for the new quarter (second quarter of 2002).

**Tip:** Preload the leading dimension table with rows that you expect to use in the future.

The Period table must contain rows corresponding to the days in the new quarter. Otherwise, new rows inserted in the Sales table would be discarded due to referential integrity failures.



```

...
star_1q00      1  MIN  DEFAU      808 SALES_STAR_IDX  INDEX  NULL
star_2q00      1  DEFAU DEFAU      808 SALES_STAR_IDX  INDEX  NULL
star_3q00      1  DEFAU DEFAU      800 SALES_STAR_IDX  INDEX  NULL
star_4q00      1  DEFAU  MAX      776 SALES_STAR_IDX  INDEX  NULL
...
target_mkt_2q02 1   822  MAX      992 PERKEY_SALES_TIX INDEX  NULL
target_mkt_2q00 1   MIN  181      920 PERKEY_SALES_TIX INDEX  NULL
target_mkt_3q00 1   181  273      912 PERKEY_SALES_TIX INDEX  NULL
target_mkt_4q00 1   273  365      912 PERKEY_SALES_TIX INDEX  NULL
...

```

8. Move the newly renamed STAR index segment to the table.

```

RISQL> alter segment star_2q02 of index sales_star
range (821:899) move;

```

The star\_2q02 index segment is empty because the CLEAR operation in step 2 removed the index entries when the corresponding data rows were removed.

9. You can now use the TMU or SQL INSERT statements to populate the Sales table with data for the second quarter of 2002 and build the corresponding index entries.

**Replace the first two paragraphs at the top of page 9-99:**

**Restarting the log and administration daemons:** If the rbwlogd or rbwadmd daemon goes down while the rbwapid daemon is still running, you can restart each daemon separately. You do not have to stop and restart the rbwapid daemon. To restart rbwlogd or rbwadmd:

1. Make sure the RB\_CONFIG and RB\_HOST environment variables are set correctly in your environment.
2. Enter rbwlogd or rbwadmd on the command line:

```

% $RB_CONFIG/bin/rbwlogd
% $RB_CONFIG/bin/rbwadmd

```

The system uses the value of RB\_HOST as a parameter to these commands when it starts the daemons. This is useful for identifying which warehouse daemon the administration and log daemons belong to when there is more than one warehouse daemon running.

**Add the following note for stopping the Red Brick service to page 9-101:**

You cannot stop the Red Brick database service if active connections exist. For more information about stopping the Red Brick service, see "Installation and Configuration Guide for Windows" on page 20.

**Replace the maximum value for TUNE TMU\_BUFFERS on page B-19:**

The maximum value of 8208 is not correct for TUNE TMU\_BUFFERS. The maximum value should be 131071 on 32-bit platforms and 8388607 on 64-bit platforms.

**Change the description of NODE\_NAME in the DST\_COMMANDS and DST\_LOCKS tables on pages C-29 and C-34; add NODE\_NAME to DST\_SESSIONS on page C-35:**

NODE_NAME	Host name where the session is running.
-----------	---

**Change the description of the CACHE\_READS column in the DST\_COMMANDS table on page C-29:**

CACHE_READS	Number of times that a block was found in local buffer cache (avoiding a logical read request). This number is increased each time the session performs a read or write operation because the database server must read the data on the block in the cache before it can do the write.
-------------	--

## Query Performance Guide

**Add the following section to page 7-14 after the “Experiment” section:**

**Cumulative Statistics for Parallel Queries:** In general, the cumulative statistics for an operator represent the sum of the statistics of the operator’s child processes spawned for parallel processing. For example, the 5363 parent process in the “Totals by Process” sample statistics on page 7-13 has a CUM\_USER\_CPU\_TIME value of 56.26, which is the sum of the CUM\_USER\_CPU\_TIME values for the child processes 4875 through 4879.

However, the CUM\_ACTIVE\_TIME of a parent EXCHANGE represents the parent process only and does not include the active time of the child processes. The sample statistics show that the 5363 parent process has a CUM\_ACTIVE\_TIME value of 21.60 seconds, which is about the same value as some of the individual CUM\_ACTIVE\_TIME values of the child processes (21.41, 21.12, and so forth). Depending on the amount of concurrency, the CUM\_ACTIVE\_TIME value of the parent EXCHANGE can be lower than any of the child EXCHANGES.

**Add the following note for monitoring several query executions in one session to page 7-21:**

The CURRENT\_MEMORY\_USED and PEAK\_MEMORY\_USED columns of the DST\_PERFORMANCE\_OPSTATS table display the memory usage for each operator in a query. For the first execution of a query in a session, the Performance Monitor might show higher memory usage for certain operators (such as EXECUTE or STARjoin) than subsequent query executions. The extra memory is used for the duration of the session and is not specific to a single query.

**Change the descriptions of the following columns in the DST\_PERFORMANCE\_OPSTATS table starting on page 7-21:**

PARENT_OPERATOR_ID	Parent operator ID (NULL if there is no parent). A parent operator is any operator in a query plan that receives rows from an operator below it.
CACHE_READS	Number of blocks found in the Red Brick server’s local buffer cache. This number is increased for each block request, whether read or write. (The block has to be read before the write operation can be done.)

**Add the following notes on monitoring parallel correlated subqueries to page 7-21:**

For each operator in a correlated subquery, the statistics are aggregated for all instances of that operator; the instances may be within the same run of the subquery or in different runs. Therefore, there is only one entry in the

DST\_PERFORMANCE\_OPSTATS table for that operator, regardless of how many instances were created. Because the process ID is irrelevant in this case, it is shown as NULL. In the DST\_PERFORMANCE\_IOSTATS table, the number of logical reads/writes per PSU is reported as the total of the reads/writes per PSU performed by all instances of the operator.

## SQL Reference Guide

Add the following new section to page 2-9:

**Creator qualifier for tables, synonyms, and views:** You can create tables, synonyms, and views with an optional creator qualifier. The creator is also known as the owner.

The following CREATE statements allow the use of a creator qualifier:

- CREATE TABLE *creator.table\_name*
- CREATE TEMPORARY TABLE *creator.table\_name*
- CREATE VIEW *creator.view\_name*
- CREATE SYNONYM *creator.synonym\_name*

The value of the creator qualifier must match the current user name. For example, the following CREATE statements execute successfully if user1 executes them:

```
create table user1.tab1 (...;
create view user1.view1 as select... from user1.tab1...;
```

However, the following CREATE statements fail for user1:

```
create table user2.tab2 (...;
create view user2.view1 (...) as select... from user1.tab1...;
```

Once defined in a CREATE statement, the creator qualifier for an object can be referenced by any user in any of the following SQL statements:

- ALTER SEGMENT
- ALTER SYNONYM
- ALTER TABLE
- ALTER VIEW
- CHECK TABLE
- CREATE HIERARCHY
- CREATE INDEX
- CREATE SYNONYM
- CREATE TABLE
- CREATE VIEW
- DELETE
- DROP SYNONYM
- DROP TABLE
- DROP VIEW
- GRANT Privilege
- LOCK
- REVOKE Privilege
- SELECT
- UNLOCK

- UPDATE

**Replace the CONCAT Results section on page 5-36 with the following information:**

If each argument is a non-null character expression, the function concatenates the arguments and returns the concatenated string of characters; otherwise, it returns NULL.

The maximum length of the returned character string is the sum of the maximum byte lengths of its arguments. This length cannot exceed 1,024 bytes, as computed at compile time. The compile-time lengths of the arguments correspond to the column lengths defined in the CREATE TABLE statement.

For example, if you define a column as CHAR(1024) or VARCHAR(1024), then attempt to concatenate it with another expression that evaluates to one or more characters, the query returns an error because the maximum compile-time length of the result is 1,025 characters. To solve this problem, use the SUBSTR function to reduce the length of the character string:

```
RISQL> create table t1(col1 char(1024));
RISQL> insert into t1 values('b');
**INFORMATION** (209) Rows inserted: 1.
RISQL> select concat('a',col1) from t1;
**ERROR** (1626) CONCAT function result length 1025 is greater than the
maximum allowed length of 1024.
RISQL> select concat('a', substr(col1,1,1)) as col1 from t1;
COL1
ab
```

Note that the TRIM, LTRIM, and RTRIM functions cannot be used to the same effect as SUBSTR; these TRIM functions reposition blank characters and do not remove them.

**Replace the GROUP BY clause syntax diagram and its description on pages 7-40 and 7-41:**

#### **GROUP BY Clause:**

|—GROUP BY—*expression*—|

*expression* Uniquely specifies a grouping column in the result table. Typically, an expression is a column name, a column alias, or a compound expression associated with one or more column names. A column alias must be defined in the AS clause in the select list.

The GROUP BY clause divides the result table into groups of rows defined by the grouping columns. A group can contain one or more rows. If a group contains more than one row, each row has the same value in its grouping columns. NULL is treated as a distinct value in a column. The result table contains one row that summarizes each group.

The column used to group the results can be displayed in the result table. If results are grouped by a column that is not in the select list, the grouping column is not displayed in the result table.

If a query specification contains a GROUP BY clause, all columns in the select list that reference columns in the FROM clause must

either be arguments of set functions or be listed in the GROUP BY clause. In other words, an expression in the select list must be one of the following:

- A character or datetime literal or a numeric constant
- A column name or column alias specified in the GROUP BY clause or an expression constructed from column names or expressions specified exactly as in the GROUP BY clause
- A set function whose argument is a constant or an expression that references only column names of tables defined by the FROM clause
- A RSQL display function, OLAP function, or scalar function whose argument is an expression that references a constant; a grouping column (a column name or expression specified exactly as in the GROUP BY clause); or a set function that is valid for the query

The names of grouping columns can occur in the search condition of a HAVING clause.

If a query that contains a select-list subquery requires a GROUP BY clause, the correlation columns, if any, must be identified in the GROUP BY clause of the outer query.

**Add the following examples to the GROUP BY section on page 7-41:**

The following query groups the sales for each weekday in January 2000:

```
select extract(weekday from date) as weekday, sum(dollars) as jan2000_sale
from period natural join sales
where year = 2000 and month = 'JAN'
group by extract(weekday from date)
WEEKDAY      JAN2000_SALE
1             49443.60
2             44292.00
3             31556.00
4             32734.90
5             37303.90
6             33025.55
7             34555.60
```

The following example shows that an expression in the GROUP BY clause does not have to be specified in the select list:

```
select sum(dollars) as jan2000_sale
from period natural join sales
where year = 2000 and month = 'JAN'
group by extract(weekday from date)
JAN2000_SALE
49443.60
44292.00
31556.00
32734.90
37303.90
33025.55
34555.60
```

**Correct the restrictions for the ALTER SEGMENT... RANGE MOVE command on page 8-50:**

Add the following restriction:



- If you want to move the range of a table segment, the segment must be empty.

In the list of operations that the RANGE MOVE clause performs, remove the words “and clear” from the first operation “Detach and clear...” You must execute a separate ALTER SEGMENT CLEAR command before you can move the range.

**Replace the paragraphs following Figure 8-9 “Sales Table Segment Ranges” on pages 8-50 and 8-51 with the following text:**

Suppose you do not need the data in segment `daily_data2` and want to move it to a new range position at the end of the table. If 799 is the current maximum value, use the following SQL statements to empty the segment and move its range:

```
alter segment daily_data2 of table sales clear;
alter segment daily_data2 of table sales range (800:max) move;
```

If the Sales table has any precomputed views, the CLEAR operation rebuilds them.

The ALTER SEGMENT RANGE MOVE statement adjusts the range of the next-lower segment `daily_data1` to cover the range previously defined for segment `daily_data2`. The resulting ranges are as follows:

<code>daily_data1</code>	<code>daily_data3</code>	<code>daily_data4</code>	<code>daily_data2</code>
min:400	400:600	600:800	800:max

If you want to move the range position of the lowest segment `daily_data1`, use the following SQL statements:

```
alter segment daily_data1 of table sales clear;
alter segment daily_data1 of table sales range (800:max) move;
```

This RANGE MOVE operation extends the next-lowest segment `daily_data2` to cover the range. Using the original ranges in Figure 8-9, the new ranges are as follows:

<code>daily_data2</code>	<code>daily_data3</code>	<code>daily_data4</code>	<code>daily_data1</code>
min:400	400:600	600:800	800:max

**Correct the ALTER SYSTEM syntax on pages 8-74 and A-6:**

The keyword STATISTICS is incorrect. The correct syntax is ALTER SYSTEM CLEAR PERFORMANCE MONITOR. This command is correctly documented in Chapter 7 of the *Query Performance Guide*.

**Add the following paragraph to the descriptions of WITH FILLFACTOR on pages 2-18 and 8-84 and CHANGE FILLFACTOR on page 8-90:**

The valid range for this clause is 5 to 100. If you specify a value greater than 100, the database server stores the value 100 in the RBW\_COLUMNS system table.

**For the CREATE MACRO command, update the description of macro\_name on page 8-141:**

Macro and parameter names must be database identifiers *but cannot* be keywords used in SQL statements, such as TARGET, PSUS, and STORAGE.

**Note:** If you have a macro that uses a keyword name, consider dropping the macro and re-creating it with a new name.

**Add the following paragraph to the description of the SEGMENT BY HASH clause of the CREATE TABLE statement on page 8-171:**

The SEGMENT BY HASH clause creates a more even distribution of data among segments when the following conditions exist:

- The number of segments is a prime number
- Only columns defined with the NOT NULL clause exist in the row
- Large columns (such as REAL and character) whose values do not vary much among the rows are near the beginning of the table.

**Remove the following sentence from the EXPORT description on page 8-210:**

The export operation can move data from the database to an output file more efficiently than other methods because it bypasses the ODBC layer.

**Add the following DBA task authorizations to the list on page 8-225:**

- ACCESS\_ADVISOR\_INFO: Access information from the Vista Advisor tables.
- BACKUP\_DATABASE: Back up the database.
- IGNORE QUIESCE: Access the quiescent database.
- RESTORE\_DATABASE: Restore the database.
- TEMP\_RESOURCE: Create temporary tables.

**Update the description of SET EXPORT\_MAX\_FILE\_SIZE on page 9-22:**

The EXPORT\_MAX\_FILE\_SIZE parameter must be set to less than 2GB on 32-bit platforms and less than 4GB on 64-bit platforms. Therefore, the highest value you can enter on 32-bit platforms is 2097151K. The highest value you can enter on 64-bit platforms is 4194303K. When the parameter is set to 0, these limits apply.

**Add the SET PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS command to page 9-51:**

The SET PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS statement determines whether dimension tables that are not joined across primary-key foreign-key columns are included in STARjoin preliminary plans. The default setting is OFF, which allows these tables to be included in STARjoin preliminary plans.

This SET statement also applies to subqueries in the FROM clause; by default, subqueries have no referential integrity relationship to other joined tables.

►► SET PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS  OFF  ON ;

You might want to set PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS to ON when the following conditions exist:

- The query has a constraint on the dimension table that joins to the unrelated table or subquery and processing the preliminary plan is expensive.

The optimizer generates a preliminary plan for each constraint on a dimension table in a STARjoin plan. When you exclude subqueries and tables not joined by referential integrity relationships to the dimension table, performance of the query might improve because the optimizer can potentially avoid expensive joins when processing the preliminary plan.

- The join to the table does not significantly reduce the number of rows selected from the dimension table.

If the join between the dimension and the non-related tables in the preliminary plan is expensive compared to the STARjoin, exclusion of this join in the preliminary plan might improve performance. However, if you exclude a join that significantly reduces the number of rows selected from the dimension or outboard table, the STARjoin performance might degrade because it processes significantly more rows.

Although you can reset the `OPTION PREPLANS_HAVE_ONLY_PKFK_JOINS` parameter in the `rbw.config` file, which applies to all database server sessions, IBM recommends that you use this SET statement for specific queries during a session because this option can degrade performance.

**Note:** Carefully evaluate the selectivity of the joins that do not have a referential integrity relationship before you set `PREPLANS_HAVE_ONLY_PKFK_JOINS` because this option might degrade performance. When inclusion of these joins significantly reduces the number of rows selected from the dimension or outboard table, setting this statement to ON can degrade performance because more rows could participate in the STARjoin.

For example, the following query contains a join of the Product dimension table to the Brand outboard table. Assume that the following conditions exist:

- The Brand and Product tables do not have a primary-key to foreign-key relationship (otherwise, the optimizer would join them along the referential integrity columns even though the join predicate did not specify those columns).
- The join to the Brand table does not significantly reduce the number of rows selected from the Product table.

```
select extract(weekday from date) as Weekday, brand.flavor,
sum(dollars) as Jan2000_sale
from period natural join product p natural join brand natural join sales s
where year = 2000 and month = 'JAN'
and p.prodkey = s.prodkey and p.prodkey = brand.prodkey
and pkg_type = 'One-pound bag'
group by extract(weekday from date), brand.flavor;
```

With the default value (OFF) of this SET statement, the following excerpt from the EXPLAIN output shows that the second preliminary plan includes the join of the Brand and Product tables, and it is a hash join because an appropriate index does not exist to do the join.

```
...
Prelim: 1; Choose Plan [id : 2] {
BIT VECTOR SORT (ID: 3)
-- TABLE SCAN (ID: 4) Table: PERIOD, Predicate: ((PERIOD.MONTH) =
('JAN ')) && ((PERIOD.YEAR) = (2000))
}
Prelim: 2; Choose Plan [id : 2] {
BIT VECTOR SORT (ID: 5)
-- HASH 1-1 MATCH (ID: 6) Join type: InnerJoin;
---- TABLE SCAN (ID: 7) Table: P (PRODUCT), Predicate: (P.PKG_TYPE) =
('One-pound bag')
---- TABLE SCAN (ID: 8) Table: BRAND, Predicate: <none>
}
```

If you set `PREPLANS_HAVE_ONLY_PKFK_JOINS` to ON, the following excerpt from the EXPLAIN output shows that the second preliminary plan excludes the join of the Brand and Product tables.

```

...
Prelim: 1; Choose Plan [id : 3] {
BIT VECTOR SORT (ID: 4)
-- TABLE SCAN (ID: 5) Table: PERIOD, Predicate: ((PERIOD.MONTH) =
('JAN ') ) && ((PERIOD.YEAR) = (2000) )
}
Prelim: 2; Choose Plan [id : 3] {
BIT VECTOR SORT (ID: 6)
-- TABLE SCAN (ID: 7) Table: P (PRODUCT), Predicate: (P.PKG_TYPE) =
('One-pound bag ')
}

```

## Table Management Utility Reference Guide

Add the following option to the `rb_ctmu` syntax on page 2-15:

**-b *machine\_name*:** Optional. The *machine\_name* value specifies an alternative host name or IP address for the client machine, which the Driver TMU will use for connecting back to the client. Use this option when the client machine is configured with multiple networks and you want to specify a faster network path for remote TMU connectivity.

Add the following description of two new XML load parameters to Chapter 2:

When you load a table from multiple XML files, you can improve load performance by using parallel tasks for the parsing phase. To configure your system for parallel XML parsing, set the maximum number of parallel tasks and, optionally, define a spill area directory.

Define the maximum number of tasks as follows, using either a TUNE parameter in the `rbw.config` file or a SET command in the TMU control file:

```

▶▶—SET—XML_MAX_TASKS—max_tasks—;—————▶▶

```

```

▶▶—TUNE—XML_MAX_TASKS—max_tasks—————▶▶

```

*max\_tasks*      A positive integer that defines the maximum number of parallel tasks to allocate for parallel parsing of XML input files. The default is 1, which implies serial processing (as does a value of 0).

Serial processing involves parsing one XML file, then loading the rows from that file, then parsing the second file, and so on. Parallel processing, in this context, involves parsing the XML files in parallel, based on the `XML_MAX_TASKS` value, then loading the rows from the input files in sequence (load the rows from the first file first, then the rows from the second file, and so on). Note that the actual loading of rows is done serially, regardless of the degree of parallelism in the parsing phase. For XML loads, the bottleneck tends to be the parsing phase, so parallel parsing is likely to provide a significant performance gain.

The `XML_MAX_TASKS` parameter specifies the maximum number of parsing tasks that can be spawned. Each task can parse and load one XML input file, so the actual maximum number of tasks should equal the number of input files. For example, if `XML_MAX_TASKS` is set to 5 but there are only 3 input files, only 3 tasks will be active. Conversely, if `XML_MAX_TASKS` is set to 3 but there are 7 input files, only 3 of the tasks can be parsed concurrently. (The first process serves

as both the task for parsing the first file and the task for loading all of the rows.) IBM recommends that you set the XML\_MAX\_TASKS parameter to the number of CPUs on your system plus 1.

Optionally, you can define a temporary space directory for XML parsing, using either an OPTION parameter in the rbw.config file or a SET command in the TMU control file:

▶▶—SET—XML\_TEMP\_SPACE—'*dir\_path*' —;—▶▶

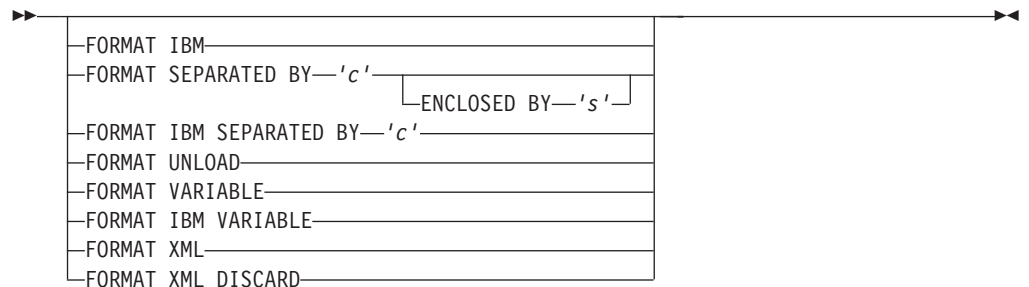
▶▶—OPTION—XML\_TEMP\_SPACE—*dir\_path*—▶▶

*dir\_path*            Name or pathname of a temporary spill directory on a local file system. The directory name in the SET command requires single quotes.

The XML\_TEMP\_SPACE directory is mainly used as a spill area during parallel parsing of XML files. If the specified directory becomes full, any in-process parsing operation is aborted; however, the TMU will attempt to re-parse the XML file in serial mode. If the XML\_TEMP\_SPACE parameter is not set, the QUERY\_TEMPSPACE\_DIRECTORY settings apply. The XML\_TEMP\_SPACE parameter does not have to be set for parallel XML parsing to be possible.

**Add the ENCLOSED BY specification to the Format clause on page 3-30:**

**format\_clause**



**FORMAT SEPARATED BY 'c'**

Specifies that fields in a data record are separated by the character *c*, which must be a single-character literal and must be different from the radix (decimal) point character. For example, data separated by an exclamation mark (!) is specified as follows:

format separated by '!'

Data separated by a tab character is specified as follows:

format separated by 'tab'

where *tab* represents the actual tab keystroke, which might appear as a blank space: ''

**ENCLOSED BY 's'**

Defines the string delimiter that encloses the character data in the input file. This specification applies only to delimited format data loads that use the FORMAT SEPARATED BY specification. If the input file contains string

delimiters, the ENCLOSED BY specification must be used; otherwise, the delimiter characters will be loaded as part of the data. String delimiters cannot be specified for loads in EBCDIC format (FORMAT IBM).

**Note:** The separator character (SEPARATED BY 'c') and the string delimiter (ENCLOSED BY 's') must be two different single-character literals. These characters can be single-byte or multi-byte, as long as they exist in the database-locale code set. If the characters used as separators and string delimiters in the input data cannot be expressed as characters in the database locale, the input data cannot be interpreted correctly.

When the ENCLOSED BY specification is used, columns in the input file that have the following data types must be enclosed by the specified delimiter character: CHAR, VARCHAR, DATE, DATETIME, and TIMESTAMP. String delimiters are supported for pseudocolumns that are defined with these data types. No other fields can contain delimiters. String delimiters must exist in pairs for each column; if a matching pair is not found, the row is discarded.

You cannot define any of the following characters as the string delimiter:

- A decimal point (.) or other radix point
- Binary zero
- A line-feed
- A carriage return
- A blank space
- A single quote

**Example:** Assume that the input file `aroma_product.txt` contains the following data:

```
1:00:"Veracruzano ":"No pkg"  
1:01:"Xalapa Lapa ":"No pkg"  
1:10:"Colombiano ":"No pkg"  
1:11:"Expresso X0 ":"No pkg"  
...
```

The TMU control file to load this data (`aroma_product.tmu`) might look like this:

```
load data  
inputfile 'aroma_product.txt'  
replace  
format separated by ':'  
enclosed by ''''  
discardfile 'product.discards'  
discards 1  
into table product (  
classkey integer external(2),  
prodkey integer external(2),  
prod_name char(30),  
pkg_type char(20)) ;
```

After `aroma_product.tmu` has been executed, the Product table would contain the following rows:

```
RISQL> select * from product;  
CLASSKEY  PRODKEY  PROD_NAME  PKG_TYPE  
          1         0 Veracruzano  No pkg
```

1	1 Xalapa Lapa	No pkg
1	10 Colombiano	No pkg
1	11 Espresso X0	No pkg

**Usage Notes:** TMU operations on character data, such as the application of string functions, apply to the data after it has been extracted from the delimiters that enclose it. Therefore, you must specify the starting and ending positions of substring operations, for example, with respect to the length of the extracted string.

When string-delimited data contains pairs of delimiter characters within character fields, these characters are loaded, as long as they are “escaped.” For example, assume that the string delimiter is a double quotation mark (") and a field in an input file contains the following string:

```
"Aroma ""Java"" beans"
```

The TMU will extract the first and sixth quotes as string delimiters and the second and fourth quotes as escape characters, then load the following string:

```
Aroma "Java" beans
```

However, if the input file contains the following string:

```
"Aroma "Java" beans"
```

the TMU will return an error because the quotes around the word Java are not escaped.

**Replace the four paragraphs under “Separated-Format Records” on page 3-128 with the following text:**

For separated-format records, the TMU determines the length of each field in the record by the separator character defined in the FORMAT clause of the LOAD DATA statement. The end of each record is indicated by either the newline character (if the RECORDLEN clause is not specified) or the record length (if the RECORDLEN clause is specified). The end of the file marks the end of the last record in the file. If records in separated format are longer than 8192 bytes, the RECORDLEN clause must be used.

Only character and external field types are allowed. Length values and POSITION keywords are ignored.

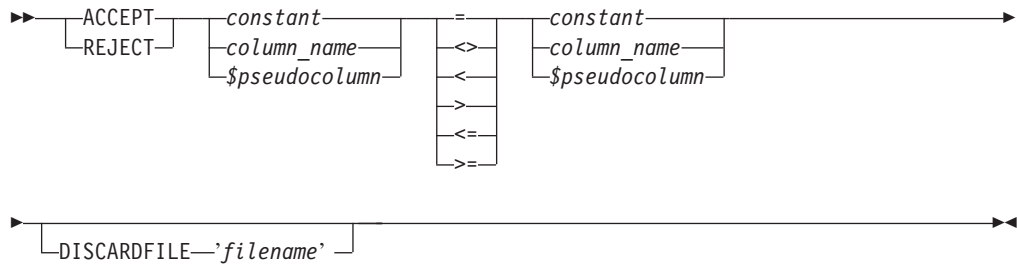
**Update the DISCARDFILE specification for the Optimize clause on page 3-60:**

Remove the last paragraph at the bottom of the page; discard files are supported for both OPTIMIZE ON and OPTIMIZE OFF loads. For OPTIMIZE ON loads, rows are discarded in fixed format. For OPTIMIZE OFF loads, rows are discarded in their original format.

**Add the DISCARDFILE specification to the Criteria clause syntax on page 3-90:**

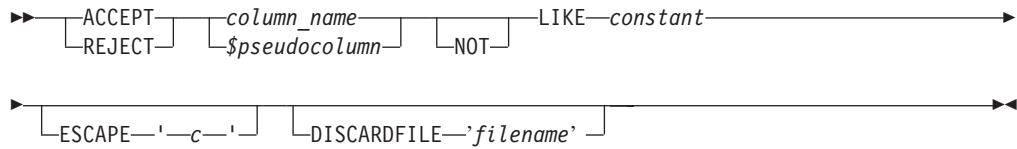
The syntax for *criteria\_clause* on a numeric or datetime column is as follows.

**criteria\_clause**



The syntax for *criteria\_clause* on a character column is as follows.

**criteria\_clause**



**DISCARDFILE 'filename'**

Specifies a single discard file for rejected records. Rows are discarded in their original format.

**Update the Task Authorization description for RESTORE operations on page 9-10:**

As well as RESTORE\_DATABASE authorization, users must have LOCK\_DATABASE authorization before they can perform a full database restore with the TMU. LOCK\_DATABASE authorization is not required for partial restores.

## RISQL Entry Tool and RISQL Reporter User's Guide

**Update the description of the -s (DSN) option on page 2-6:**

**-s dsn:** Data source name (DSN), as defined in the .odbc.ini file on UNIX systems and the odbc.ini file or the Registry on Windows platforms. A DSN can also be specified with the RB\_DSN environment variable. This option takes precedence over the -d database option.

**Update the bullet that describes DECIMAL format in "Changing the Format of Numeric Data" on page 3-19:**

If a REAL or FLOAT column contains a positive or negative value less than .005, the RISQL Entry Tool always displays it as 0.00, even when the SET COLUMN FORMAT DECIMAL command is used to increase the number of digits after the decimal point.

## Messages and Codes Reference Guide

The following messages were either added or changed in V6.20.xC2 and later versions:

---

**RBS168E** Value *value.16g* would overflow or underflow the column with type *data\_type*.



---

**RBS755E**    **DEFAULT or RANGE for TIMESTAMP column '*column\_name*' is not type-compatible.**

**Reason:** An invalid type DEFAULT or RANGE value was declared for a TIMESTAMP column.

**Response:** Ensure that the supplied value is type-compatible with the column.

---

**RBS760E**    **DEFAULT or RANGE for TIME column '*column\_name*' is not type-compatible.**

**Reason:** An invalid type DEFAULT or RANGE value was declared for a TIME column.

**Response:** Ensure that the supplied value is type-compatible with the column.

---

**RBS761E**    **DEFAULT or RANGE for DATE column '*column\_name*' is not type-compatible.**

**Reason:** An invalid type DEFAULT or RANGE value was declared for a DATE column.

**Response:** Ensure that the supplied value is type-compatible with the column.

---

**RBS1942E**    **The GROUP BY clause for the precomputed view is invalid.**

**Reason:** The GROUP BY clause can only contain a list of table columns. Column aliases and other expressions are not allowed.

**Response:** Correct and resubmit the SQL statement.

---

**RBS6416E**    **Cannot move the range of table segment *segment\_name* because it is not empty.**

**Reason:** RANGE MOVE is not valid for non-empty table segments.

**Response:** Delete the data in the segment or clear the segment and reissue the command.

---

**RBL8039E**    **Error accessing work segment *segment\_name*.**

**Reason:** An offline LOAD operation specified a work segment that could not be accessed. This may happen, if the work segment has corrupted or missing PSUs, or if there were I/O errors.

**Response:** Make sure that the PSUs for this work segment exist and are accessible.

---

**RBL8040E**    **String delimiter must be one character in length.**

**Reason:** The string delimiter must be one character in length. Load terminated.

**Response:** Correct the LOAD DATA statement.

---

---

**RBL8041E**    **RADIX POINT '*character*' cannot be the same as the ENCLOSED BY character.**

**Reason:** The RADIX POINT character must be different from the ENCLOSED BY character.

**Response:** Correct the error and resubmit the command.

---

**RBL8042E**    **Invalid ENCLOSED BY character.**

**Reason:** The ENCLOSED BY character is not supported.

**Response:** Correct the error and resubmit the command.

---

**RBL8043E**    **ENCLOSED BY character '*character*' cannot be the same as the SEPARATED BY character.**

**Reason:** The ENCLOSED BY character must be different from the SEPARATED BY character.

**Response:** Correct the error and resubmit the command.

---

**RBL8044W**    **Data does not start with ENCLOSED BY character. Row *number*. Column name: *column\_name*.**

**Reason:** Data content error. Record was placed in a discard file.

**Response:** Check the input data or the discard file.

---

**RBL8045W**    **Data does not end with ENCLOSED BY character or no escape character is included. Row *number*. Column name: *column\_name*.**

**Reason:** Data content error. Record was placed in a discard file.

**Response:** Check the input data or the discard file.

---

**RBL8046W**    **Encountered end of row before ENCLOSED BY character. Row *number*. Column name: *column\_name*.**

**Reason:** Data content error. Record was placed in a discard file.

**Response:** Check the input data or the discard file.

---

**RBL8731F**    **Invalid data in input row *Number of file file\_name*.**

**Reason:** This happens in the row conversion stage when invalid character (such as x00) was found in input data. Data load will terminate.

**Response:** Check the data contents.

---

---

**RBS8732E SET EXPORT\_MAX\_FILE\_SIZE value must be less than valueG.**

**Reason:** The value specified in the SET EXPORT\_MAX\_FILE\_SIZE command is too large. The value must be less than 2G on 32-bit platforms and less than 4G on 64-bit platforms. The highest value you can enter on 32-bit platforms is 2097151K, 2047M or 1G. The highest value you can enter on 64-bit platforms is 4194303K, 4095M or 3G.

**Response:** Specify a smaller value for the EXPORT\_MAX\_FILE\_SIZE parameter and submit the command again.

---

**RBO10117E Setting descriptor array size for bulk insert/fetch is not supported.**

**Reason:** SQLSetDescField with SQL\_DESC\_ARRAY\_SIZE or SQLSetStmtAttr with SQL\_ATTR\_PARAMSET\_SIZE is not supported.

**Response:** Insert or fetch one row at a time.

---

**RBO10194E Locale value 'locale\_name' is invalid.**

**Reason:** The locale setting is incorrect.

**Response:** Set a valid locale setting.

---

**RBC12002F Error while writing to file 'string'. OS error: string.**

**Reason:** The Client TMU encountered an error while writing to a file, possibly because the disk ran out of space. The corresponding OS error message is displayed.

**Response:** Correct the error and resubmit the command.

---

**RBC12014F Could not resolve machine name from RB\_HOST string. OS error: string.**

**Reason:** The Machine name is not resolved from the value of RB\_HOST. Corresponding OS error message is displayed.

**Response:** Check the rbw.config file for the correct RB\_HOST specification.

---

**RBC12017F Could not open file 'string' for read. OS error: string.**

**Reason:** File could not be opened for reading. The corresponding OS error message is displayed.

**Response:** Check the permissions, path and name of file and resubmit the command.

---

---

**RBC12019F Could not open file 'string' for write. OS error: string.**

**Reason:** File could not be opened for writing. The corresponding OS error message is displayed.

**Response:** Check the permissions, path and name of file and resubmit the command.

---

**RBC12025F Fatal error initiating connection to warehouse server. OS error: string.**

**Reason:** Could not make a connection to warehouse server. Warehouse server may not be up or not set up for remote TMU requests. The corresponding OS error message is displayed.

**Response:** Set up server for Remote TMU requests and bring up the server.

---

**RBC12026F Error in resolving connection details. OS error: string.**

**Reason:** Could not resolve the connection details for the warehouse server. The corresponding OS error message is displayed.

**Response:** Check the connection details specified.

---

**RBC12036F Select() call failed. OS error: string.**

**Reason:** Corresponding OS error message is displayed.

---

**RBC12048I Driver TMU client connection set to machine\_info.**

**Reason:** For a Remote TMU operation, the Driver TMU will connect back to the Client TMU through a different socket, as requested with the -b option in the rb\_ctmu command. The Driver TMU will use the IP address shown in the message to connect back to the client machine.

**Response:** None necessary.

---

**RBC12049E Client TMU and Driver TMU are incompatible.**

**Reason:** For a Remote TMU operation, the Client TMU and the Driver TMU must be compatible to communicate with each other effectively. This error indicates that either the client or the server program has significantly changed, making the programs incompatible for this operation.

**Response:** Make sure that you have compatible versions of the Client TMU and Driver TMU installed. For example, if you are using the Client TMU on Windows, you might have to upgrade to a later version of the Client Connector Pack. For more information about compatibility and upgrade issues, refer to the Release Notes.

---

**RBS12625E** Cannot create object '*name*'. Prefix '*user\_name*' does not match the user's login name: '*login\_name*'.

**Reason:** A table, view, or synonym was qualified with a username that is not the login name of the current user.

**Response:** Correct the error and resubmit the command. Either use the name of the user who is logged in or do not qualify the object name.

---

**RBS12626E** Owner of object '*name*' cannot be changed to '*user\_name*' during a RENAME operation.

**Reason:** An ALTER TABLE...RENAME operation on a table or synonym specified an owner qualifier that did not match the current owner of the object.

**Response:** Either use the correct username qualifier for the new name or do not qualify the object name.

---

**RBL13009W** Actual number of unloaded rows on the client machine might be less than the number reported by the rb\_ptmu.

**Reason:** When the Client TMU (rb\_ctmu) stops unloading rows because of an error, the number of rows successfully unloaded on the client machine might be less than the number of rows reported as unloaded by the TMU on the server machine. This situation can occur because of the network delay between the client and server TMU operations.

**Response:** Correct the error on the client machine and submit the unload command again.