

**RELEASE NOTES FOR  
IBM Red Brick Warehouse  
for Windows Platforms**

**Date: 04/09/03**

**Version: 6.11.TC6**

**Table of Contents**

- I. Important Release Information
- II. Customer Support
- III. New Features in V6.11.xC4
- IV. New Features in V6.11.xC3
- V. New Features in V6.11.xC1 and V6.11.xC2
  - A. TMU Backup and Restore
  - B. ALTER TABLE Enhancements
  - C. Enhanced Parallelism for UNION, INTERSECT, and EXCEPT Queries
  - D. Optimizations for Aggregation Queries with WHEN and HAVING Clauses
  - E. TMU Aggregate Functions on Datetime Fields
  - F. EXPLAIN Documentation
  - G. Minor Features
- VI. New Features in V6.10.xC1
- VII. Significant Operational Considerations
  - A. Result Data Type for String Functions
  - B. Notes on Vista Aggregate Management
  - C. Additional Memory Allocated for Large Indexes
- VIII. Installation and Upgrade Notes
  - A. For the First Time
  - B. With an Existing Warehouse
  - C. Installation Using the Local System redbrick User
  - D. Supported File Systems
  - E. Windows NT 4.0 Service Packs
  - F. Windows 2000 Service Packs
- IX. Fixed Issues in V6.11.x
  - A. Database Administration Issues
  - B. Query-Related Issues
  - C. Connectivity Issues
  - D. Administrator Tool Issues
  - E. Documentation Issues
- X. Known Issues
  - A. Database Administration Issues
  - B. Query-Related Issues
  - C. Connectivity Issues
  - D. Administrator Tool Issues
- XI. Documentation Notes
  - A. TMU Backup and Restore User's Guide
  - B. Explaining EXPLAIN User's Guide
  - C. Vista User's Guide
  - D. Administrator's Guide
  - E. Installation and Configuration Guide for Windows Platforms
  - F. SQL Reference Guide
  - G. Table Management Utility Reference Guide
  - H. Messages and Codes Reference Guide

## I. Important Release Information

IBM Red Brick Warehouse Version 6.11.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, V6.11.xC6 refers to all platforms generically.

Before you install and use V6.11.xC6, read the following sections:

- New Features in V6.11.xC4
- New Features in V6.11.xC3
- New Features in V6.11.xC1 and V6.11.xC2
- New Features in V6.10.xC1
- Significant Operational Considerations
- Installation and Upgrade Notes
- Fixed Issues in V6.11.x
- Known Issues
- Documentation Notes

## II. 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** <http://www-3.ibm.com/software/data/informix/support/>

## III. New Features in V6.11.xC4

IBM Red Brick Warehouse V6.11.xC4 included the following enhancements to password security:

- Ability to restrict a password from containing the user's login ID.
- Ability to enforce a rule that a new password must be different from the previous password by more than the value of one or more numerics.
- Ability to enforce a rule that a password must be changed upon first login.

For details about these new security features, see "Administrator's Guide" on page 24.

## IV. New Features in V6.11.xC3

IBM Red Brick Warehouse V6.11.xC3 included the following new features:

- The Administrator tool (V2.11.3) supports the ALTER TABLE enhancements introduced in V6.11.xC1.
- The ODBC Driver supports the MONTHNAME function.

## V. New Features in V6.11.xC1 and V6.11.xC2

IBM Red Brick Warehouse V6.11.xC1 and V6.11.xC2 included significant new features and enhancements as well as maintenance fixes. (V6.11.xC2 was a re-release of V6.11.xC1 with additional fixes.)

### A. TMU Backup and Restore

The TMU supports full and incremental backup operations and full and partial restore operations. Incremental backups are based on information about changed blocks stored in a special backup segment. These backups can be used in conjunction with either full TMU backups or full backups performed with external tools. (For details, refer to the *TMU Backup and Restore User's Guide*, available in PDF format on the software CD-ROM.)

### B. ALTER TABLE Enhancements

The ALTER TABLE syntax has been extended to allow administrators to:

- Rename existing tables.
- Change a column originally declared as NOT NULL to nullable and vice versa.

For more information about these features, see page 32.

### C. Enhanced Parallelism for UNION, INTERSECT, and EXCEPT Queries

In most cases, parallel-query processing is induced by certain operators *within* query plans, such as table scans and STARjoins. The presence of the Exchange operator in the EXPLAIN output indicates that an operator, or series of operators, below the Exchange can benefit from parallel tasks allocated at run-time. In V6.11.xC1, queries that contain the set operators UNION, INTERSECT, and EXCEPT can benefit from parallelism at the level of the query plan, as well as parallelism at the operator level. The potential performance gains apply equally to queries that contain any combination of the three operators; in this context, such queries are referred to generically as "UNION queries."

#### PARALLEL\_SET\_OPERATION Parameter

If the configuration parameter TUNE PARALLEL\_SET\_OPERATION is set to ON, the query plans for each query expression in a UNION query can run concurrently rather than sequentially. Instead of waiting for the first query expression to complete its execution, the server starts processing some or all of the query expressions simultaneously. Consider the following abbreviated query:

```
select col1, col2, col3 from...
union
select col1, col2, col3 from...
union
select col1, col2, col3 from...;
```

Given enough parallel processes, the query plans chosen for the three query expressions (select...) will start executing at the same time. A detailed example of this kind of query and its EXPLAIN output is presented in the new document, *Explaining EXPLAIN*. Parallel set operations are represented in the output by an Exchange operator labeled "Generic Single Instance."

The `PARALLEL_SET_OPERATION` parameter is set to `OFF` by default because this special approach to parallelism will accelerate some `UNION` queries but potentially degrade the performance of others. (It has no effect on queries that do not contain set operators.) IBM recommends that you use the following `SET` command on a query-by-query basis to experiment with a variety of `UNION` queries and measure the performance in each case:

```
set parallel set_operation on;
```

From a query performance and resource allocation standpoint, this approach is safer than turning on the global `PARALLEL_SET_OPERATION` parameter in the `rbw.config` file.

### Tuning Recommendations

As with all parallel-query processing, the degree of parallelism per query is inhibited by the `QUERYPROCS` limit, the availability of multiple CPUs, and so on. Parallel set operations imply a redistribution of the available resources, not the automatic use of additional resources. When this feature is turned off and parallelism occurs only at the operator level, the total number of parallel tasks that are used and reused during execution is typically *higher*. However, a higher number of processes does not necessarily imply better performance and resource utilization. A lower degree of parallelism—applied to the operations that can benefit from it the most—is sometimes more efficient.

In particular, consider the following tuning recommendations:

- When the expressions on either side of a set operator are known to return very large intermediate result sets, setting `PARALLEL_SET_OPERATION` to `ON` can lead to reduced memory consumption and an overall improvement in query performance.

Every plan for every `UNION` query expression ends with a Merge Sort that passes results up to a single Sort 1-1 Match operator (except for `UNION ALL` queries). Large intermediate results computed in parallel during the sort phase may make more efficient use of limited parallelism than the underlying operators; therefore, it might be more optimal to run the sort operations in parallel than the underlying scans, joins, and so on.

- When `PARALLEL_SET_OPERATION` is set to `ON`, set `QUERYPROCS` to a generous number. If sufficient parallel tasks cannot be made available to the whole query, it is better to turn off this feature.

If possible, allocate enough parallel tasks to cover both the plan-level parallel set operations and the operator-level parallelism. The minimum requirement is at least one process per `UNION` query expression (one per “Generic Single Instance” Exchange operator). If the query could benefit from a high degree of parallelism at the operator level, make sure that the number of processes available is greater than the total number of query expressions. If the query contains large table scans or complex aggregations that would benefit from parallelism but no tasks are available, the performance of the whole query could suffer.

- The first (or leftmost) query expression in a `UNION` query is allocated parallel processes first; therefore, make sure the SQL is structured so that the most time-consuming query expression is at the top. (If plenty of parallel tasks are available for the whole query, the order of query expressions is not a factor.)

For a general discussion of parallel-query processing, see Chapter 11 of the *Administrator's Guide*.

## D. Optimizations for Aggregation Queries with WHEN and HAVING Clauses

In V5.1.x and later versions, simple aggregation queries are optimized when the aggregated column is indexed. In V6.11.xC1, this capability is extended to include queries that contain HAVING and WHEN clauses.

For example, the following COUNT(\*) query also contains a RANK function and a WHEN clause that constrains the ranked values. The EXPLAIN output shows that an “index-assisted aggregation” will be applied, using the index on the City column.

```
RISQL> explain select city, count(*) as cnt, rank(cnt) as rnk
> from store
> group by city
> when rnk = 1;
EXPLANATION
[
- EXECUTE (ID: 0) 1 Table locks (table, type): (STORE, Read_Only)
--- RISQL CALCULATE (ID: 1)
----- MERGE SORT (ID: 2) Distinct: FALSE
----- BTREE SCAN (ID: 3) Index assisted aggregation:GROUP BY , Count(*); Table:
STORE, Index: CITY_TGT_IDX ,Reverse order: FALSE; Start-stop predicate: <none> ;
Predicate: <none>
]
```

The following example contains a HAVING clause that constrains the results of the COUNT(\*). Again, the “index-assisted aggregation” can be applied, using the same index.

```
RISQL> explain select city, count(*) as cnt, rank(cnt) as rnk
> from store
> group by city
> having cnt > 1;
EXPLANATION
[
- EXECUTE (ID: 0) 1 Table locks (table, type): (STORE, Read_Only)
--- RISQL CALCULATE (ID: 1)
----- MERGE SORT (ID: 2) Distinct: FALSE
----- BTREE SCAN (ID: 3) Index assisted aggregation:GROUP BY , Count(*); Table:
STORE, Index: CITY_TGT_IDX ,Reverse order: FALSE; Start-stop predicate: <none> ;
Predicate: <none>
]
```

## E. TMU Aggregate Functions on Datetime Fields

The following aggregate functions can now be applied to DATE, TIME, and TIMESTAMP field types:

- MAX
- MAX\_NONULL
- MIN
- MIN\_NONULL

For example, the following LOAD DATA control file applies each aggregate function to a different DATE column:

```
load data inputfile 'aggdate.txt'
modify aggregate
into table aggdate (
a1 position(01:09) int external,
b1 position(10:19) date 'MM-DD-YYYY' MIN,
c1 position(22:31) date 'MM-DD-YYYY' MIN_NONULL NULLIF(9)='% ',
d1 position(34:43) date 'MM-DD-YYYY' MAX,
e1 position(46:55) date 'MM-DD-YYYY' MAX_NONULL NULLIF(9)='% ');
```

For more information about these functions and field types, refer to Chapter 3 of the *Table Management Utility Reference Guide*.

## F. EXPLAIN Documentation

The description of the EXPLAIN command in the *Administrator's Guide* is replaced with a new document, *Explaining EXPLAIN*, which is distributed in PDF format on the software CD-ROM. This document contains detailed discussions of both the text output and graphical EXPLAIN output for a wide range of queries.

## G. Minor Features

A complete list of fixed issues starts on page 11.

### PreparedStatement setAsciiStream and setCharacterStream Methods

The Red Brick JDBC Driver now supports setAsciiStream and setCharacterStream methods in the PreparedStatement class. For information about PreparedStatement objects, refer to Chapter 7 of the *Client Installation and Connectivity Guide*.

### Improved Error Handling for the TMU

When a fatal exception occurs during a load (such as a divide-by-zero error), the TMU displays an error message. In previous releases, the TMU exited without any message.

On 32-bit systems, the TMU warns users when high memory consumption might cause a load operation to fail. This situation can arise when the INDEX\_TEMPSPACE\_THRESHOLD parameter is set too high and the TMU\_MMAP\_LIMIT parameter is not set (which means that unlimited memory mapping can occur). When the combined memory allocation for the INDEX\_TEMPSPACE, TMU\_MMAP\_LIMIT, and TMU\_BUFFERS parameters exceeds 1.6 GB, warning message 8562 prompts users to solve the potential problem by adjusting these parameters.

### Schema Locks and the READER\_PRIORITY Option

In V6.11.xC1, table-level schema locks are always acquired when queries are executed, reducing contention for database-level locks on the system tables. For every table lock, there is a corresponding RS (read schema) or WS (write schema) lock, as shown in the DST\_LOCKS table and the output of the **showlocks** utility.

Schema locks are not affected by the setting of the READER\_PRIORITY option. For updated information about READER\_PRIORITY, see page 28.

## VI. New Features in V6.10.xC1

IBM Red Brick Warehouse V6.10.xC2 was an interim maintenance release; it did not contain any new features. IBM Red Brick Warehouse V6.10.xC1 contained a large number of new features and enhancements. The following list briefly identifies each feature and, where applicable, indicates which document to consult for more information.

### Precomputed View Maintenance

- Precomputed views are updated automatically when their detail tables are modified via DML commands or TMU load scripts. Maintenance is also supported through TMU REORG and ALTER SEGMENT CLEAR operations (*Vista User's Guide*).

### Query Execution and Optimization

- **Sampling**—Within the FROM clause of a query, you can request a representative sample of the data in any table, in terms of either a count or a percentage of its rows or blocks. This functionality can also be used to speed up Vista candidate analyses (*SQL Reference Guide*, *Vista User's Guide*).
- **Query rewrite for set operators**—Certain queries that contain UNION, INTERSECT, or EXCEPT operations are automatically rewritten and optimized.
- **Multiple DISTINCTs**—Multiple DISTINCT aggregate expressions can be specified in a query (*SQL Reference Guide*).

- **POSITION, POSITIONB, and TO\_CHAR functions**—These new scalar functions can be used to constrain character data by position and convert dates to character strings (*SQL Reference Guide*).
- **Query Optimizations**—Enhancements to the Red Brick query engine further improve the performance of complex queries. These include improved plan selection for queries involving multi-table joins, and query plan transformations that enable better utilization of join techniques and indexes for more efficient processing of queries that contain multiple blocks.
- **Delimited Format EXPORT**—SQL EXPORT now supports delimited format for exported data (*SQL Reference Guide*).

### TMU and PTMU Enhancements

- **Terse mode**—A new TMU parameter allows you to reduce the row message output during a load operation (*Table Management Utility Reference Guide*).
- **Parallel load with periodic commit**—The PTMU now supports “trickle-feed” load operations (*Table Management Utility Reference Guide*).
- **Parallel MODIFY/UPDATE loads**—The PTMU now supports loads in MODIFY and UPDATE modes (*Table Management Utility Reference Guide*).
- **Optimized index building for MODIFY loads** (*Table Management Utility Reference Guide*).
- **Optimized referential integrity checking** (*Table Management Utility Reference Guide*).

### General Administration Features

- **Versioning enhancements**—Versioning can now be used to support high recoverability during loads as well as to provide query-priority concurrency. To facilitate backups for versioned databases, new commands have been added to verify that the version log is cleaned while the database is available for concurrent query activity (*Administrator’s Guide*).
- **32-bit serial data type**—The serial data type can be used to generate and store unique values in a specific column (*SQL Reference Guide*).
- **Segment-level row counts** are now stored in the system catalog (*Administrator’s Guide*).
- **Updated Internationalization Library**—The server now supports a new internationalization library, which provides support for a greater variety of database locales (refer to the *locales.pdf* file in the RELNOTES directory on the software CD-ROM).
- **Licensing changes**—The Red Brick server is now enabled with a single license key and serial number. The Vista and PTMU products are bundled with the product and are no longer sold or enabled as options (*Installation and Configuration Guide*).

### New Server Platforms

IBM Red Brick Warehouse V6.10.x and later versions can be installed on the following new platforms:

- **Windows 2000 Server** (*Installation and Configuration Guide for Windows*)  
Windows 2000 Server was first supported on V6.0.3.

### Client Features

- **JDBC 2.0**—The Red Brick JDBC Driver now supports JDBC 2.0 functionality (*Client Installation and Connectivity Guide*).
- **ODBC 3.x**—The Red Brick ODBC Driver now supports ODBC 3.x functionality (*Client Installation and Connectivity Guide*).
- **Query timeout** is supported by the ODBC and JDBC Drivers (*Client Installation and Connectivity Guide, SQL Reference Guide*).
- **Administrator tool enhancements**—Support for new features is available through the GUI, as well as

improvements to the Show DDL function.

## VII. Significant Operational Considerations

Before you use IBM Red Brick Warehouse Version 6.11.xC6, read the platform-specific information for your operating system in the appendixes of the *Installation and Configuration Guide*. Also note the following operational considerations and changes.

### A. 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
- RTRIM
- SUBSTR
- SUBSTRB
- TRIM

### B. Notes on Vista Aggregate Management

Before using the Vista aggregate management features in V6.11.xC6, please read the following notes:

- The precomputed view maintenance feature is turned ON by default in the *rbw.config* file. If you use your own scripts to load and maintain aggregate tables *and* Vista precomputed view maintenance is turned ON, you run the risk of “double updates” to those tables when you update your detail tables.

If you have existing precomputed views built with a 6.0.x or earlier version of the server, their associated aggregate tables are not automatically maintained unless you turn on maintenance for those views. The Maintenance column of the RBW\_VIEWS system table is marked OFF for precomputed views defined in databases upgraded from a previous release. The Maintenance column must be manually set to ON or REBUILD for maintenance to occur. Use the ALTER VIEW...SET MAINTENANCE command to update the Maintenance column.

- If you modify an aggregate table manually, its precomputed view is marked invalid. This behavior is a change from the behavior in previous releases, when only changes to detail tables caused views to be marked invalid. In previous releases, users could turn off the auto-invalidation of precomputed views. This option has been removed.
- The syntax for setting the Valid/Invalid flag for precomputed views has been changed. The original SET commands are supported only for backward compatibility; the new ALTER VIEW and ALTER TABLE commands that replace the SET commands are documented in the *SQL Reference Guide*.
- SERIAL columns cannot be referenced in a CREATE VIEW statement for a precomputed view.
- Advisor candidate view analysis is now performed against a subset of the fact table rows by default. The subset is constructed by sampling 5 percent of the table.
- In V6.10.xC2 and later releases, the OPTION ADVISOR\_LOGGING configuration parameter is set to OFF by default. To log aggregate query activity and generate candidate views, you must change this parameter in the **rbw.config** file or use the SET ADVISOR LOGGING command. There is no change to the ADMIN ADVISOR\_LOGGING parameter, which creates the Advisor log file. This setting remains OFF by default.

For detailed information about Vista features, refer to the *Vista User's Guide*.



## C. 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 100 KB.

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.

## VIII. Installation and Upgrade Notes

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

### A. 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.

### B. 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 Red Brick Decision Server." 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 V6.11.xC6 release requires an upgrade of existing V5.1.x and V6.0.x databases to accommodate changes to the system catalog. You can upgrade any V5.1.x or later database directly to V6.11.xC6; however, to upgrade a V5.0.x database to V6.11.xC6, you must upgrade to V6.0 first.

### Administrator Tool Installation and Compatibility

There are no changes to the IBM Red Brick Warehouse Administrator tool for this release. The current version of the tool is V2.11.3. You can use V2.11.3 with IBM Red Brick Warehouse V6.0.x, V6.10.x, and V6.11.x. However, you cannot use Administrator V2.0.x and earlier with IBM Red Brick Warehouse V6.10.x and V6.11.x.

V2.11.3 runs on the following Windows operating systems:

- Windows 98
- Windows NT (Service Pack 6A)
- Windows 2000 Professional

Before installing V2.11.3, 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.

### **C. Installation Using the Local System redbrick User**

If the *redbrick* user is defined both on the local system and anywhere on the network domain, the installation process logs on to the network domain as a service using the *redbrick* user on the network domain. Because of this behavior, the password entered during the installation process should be that of the network *redbrick* user, not the local *redbrick* user. Also, ensure that the privileges to log on as a service and to act as part of the operating system are granted to the network *redbrick* user.

To install IBM Red Brick Warehouse using the local system *redbrick* user, make sure that there is no *redbrick* user present in the network domain.

### **D. Supported File Systems**

IBM Red Brick Warehouse for Windows is supported only on NTFS file systems. Do not use FAT file systems.

### **E. Windows NT 4.0 Service Packs**

IBM Red Brick Warehouse V6.11.xC6 is certified for use on Windows NT 4.0 Service Packs 5 and 6A. Do not use earlier Service Packs.

### **F. Windows 2000 Service Packs**

On Windows 2000 computers, IBM Red Brick Warehouse V6.11.xC6 is certified for use with Service Pack 2 only. Do not use Service Pack 1.

## IX. Fixed Issues in V6.11.x

This section lists and describes issues reported by customers that have been fixed in IBM Red Brick Warehouse V6.11.xC1 through V6.11.xC6. The prefix CSL followed by a number indicates the Customer Support Log number assigned to each issue.

This section is organized as follows:

- Database Administration Issues
- Query-Related Issues
- Connectivity Issues
- Administrator Tool Issues
- Documentation Issues

### A. Database Administration Issues

- CSL #10440

If the TMU failed with a 132 error during a fact table load because of a problem with the target table or a primary key index on a referenced table, the table in question was not identified. The name of the table is now included in the error message.

- CSL #12655, 13565, 16021

The TMU aggregate functions MAX and MIN could not operate on datetime field types. See page 5.

- CSL #12971, 15400, 16026, 18540, 834984, 978556, 982380

Tables could not be renamed without being dropped and re-created. See page 32.

- CSL #15113, 18400, 850016, 952789, 982382

Table columns could not be changed from NOT NULL to nullable and vice versa. See page 33.

- 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 #965863

When the INDEX\_TEMPSPACE\_THRESHOLD parameter was set too high and the TMU\_MMAP\_LIMIT parameter was not set, loads could fail without warning because of excessive memory consumption. See "Improved Error Handling for the TMU" on page 6.

- CSL #971114, 221787, 239250

When an EXPORT command was cancelled, its output files could not be deleted. If the ROWCOUNT setting was in effect, the TMU output file could not be created.

- 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 V6.11.xC5 and later versions, the record is discarded only once in the generic discard file.

- CSL #210487

When multiple DML commands were executed simultaneously, identical temporary space filenames were generated by different processes, causing an operating-system error.

- CSL #231718

A serial number with an invalid prefix was not detected during installation, and the **rbwapid** daemon failed to start.

- CSL #233647

The TMU generated an incorrect overflow error (message 331) when loading values into a DEC(4) column.

- CSL #232118

After a database was upgraded from V6.0 to V6.10, a load operation with precomputed view maintenance and AUTOROWGEN ON failed with an internal error.

- CSL #230681

The TMU failed to execute a control file when it contained multiple REORG statements that rebuild precomputed views.

- CSL #221787

The EXPORT command failed with an error when the name of the first directory after the drive letter began with a numeric character.

- CSL #240536

The EXPORT command returned incorrect results for SELECT DISTINCT queries.

- CSL #240537

When the database and client locales were set to different code sets whose maximum character lengths do not match, the EXPORT command did not return any data for character columns.

- CSL #241537

The EXPORT command produced DDL files in binary write mode (**wb**) instead of write mode (**w**).

- CSL #242570

The TMU returned a fatal error when the RECORDLEN clause was used and the input records were longer than 8192 bytes.

- CSL #236279

When a query was started while a REPLACE mode parallel load was in progress against the same table, the query completed after the load had finished but returned no rows.

- CSL #241926

The CHECK TABLE FIX command indicated that there were no inconsistencies on a table but the same command without the FIX option continued to report errors.

- CSL #246904

The TUNE TMU\_BUFFERS parameter accepted one block less than the documented maximum of 131,071 blocks.

- CSL #260438

The DDL produced by the TMU GENERATE CREATE TABLE command did not include the default value for a column when it was set to zero (0).

- CSL #252791

When a disk input file was used to load a table with the Parallel TMU (**rb\_ptmu**), the input data was randomly corrupted, causing intermittent warning messages and load failures.

- CSL #258393

On heavily used versioned databases, INSERT INTO...SELECT statements against temporary tables could take a long time to complete. In V6.11.xC5 and later versions, versioning performance is improved, particularly for temporary tables and operations that involve mixed blocking and versioning transactions.

- CSL #262061

Precomputed view maintenance failed with a fatal error when an aggregate table name was used in the control file for a REORG operation instead of the expected precomputed view name. In V6.11.xC5 and later versions, this scenario results in an appropriate error message.

- CSL #261982

After a database was upgraded from V5.x to V6.x, an ALTER TABLE ADD/DROP COLUMN command corrupted the target table if it was created in V5.x and contained 128 or more segments.

- CSL #265200

Logical database names registered in the **rbw.config** file were not recognized by TMU backup operations, causing such operations to fail with error 1298.

- CSL #266773

In rare cases, a DELETE operation on a base table caused an assertion failure during the referential integrity checking phase.

- CSL #269028

ALTER SEGMENT commands failed with a lock-conflict error after the associated table had been explicitly locked to prevent user access.

- CSL #269851

When a Parallel TMU MODIFY or UPDATE load modified the segmenting column of the table (and that column was not part of the primary key index), the updated rows were not moved to the appropriate segment.

- CSL #266773

A fatal error sometimes occurred when the DELETE\_REFCHECK operator was used to delete rows from a table with multiple referencing tables.

- CSL #263423

In the **rbw.config** file, the TUNE PARTITION\_PARALLEL\_AGGREGATION parameter displayed the ON option first, implying that ON is the default value. The default is OFF.

- CSL #275444, 314656, 318574

For MODIFY load operations with OPTIMIZE ON, when more than one input row had the same primary key value and that value did not exist in the table, only the last matching input row was loaded. The preceding matching rows were discarded as duplicates and no aggregate processing was performed on the new row. See page 38 for a description of the behavior in V6.11.xC5 and later versions.

- CSL #269581

TMU and DML modifications that involve incremental maintenance of a precomputed view failed with a fatal error when the view's underlying fact table contained a VARCHAR column.

- CSL #279387

There was a typographical error in EXPORT message 8755 in the Japanese message files.

- CSL #273123

When a TMU output or discard file exceeded the filesize resource limit, the appropriate error message could not be found and displayed.

- CSL #285307

Some enhancements to password security were requested. See page 26.

- CSL #293628

When the MAXROWS PER SEGMENT value was increased on a referenced table, an attempt to detach a segment from the referencing table using the OVERRIDE FULLINDEXCHECK option resulted in a fatal error.

- CSL #299348

The output of the CHECK INDEX...VERBOSE VALIDATE FULL command stopped at row number 499,971. In V6.11.xC5 and later versions, the output is segmented, with a maximum of 1,000,000 lines per report file.

- CSL #308224

When the INDEX\_TEMPSPACE\_THRESHOLD parameter was set to a high value, CHECK INDEX VALIDATE FULL statements either ran very slowly or failed with a swap-space error. See page 25.

- CSL #266773, 319399

When a DELETE statement was executed on a dimension table, but the rows were referenced so that no rows were deleted, an assertion could occur. In V6.11.xC5 and later versions, the DELETE operation returns a message to indicate that the rows were not deleted because they are referenced by another table.

- CSL #324113

When a non-existent table was referenced in a TMU control file, message 412 was issued. In V6.11.xC5 and later versions, a more appropriate message is used.

- 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 #334799

After the first segment of a table with a synonym was dropped, the server could return a fatal error for queries that referenced the synonym or any queries on RBW\_TABLES.

- CSL #336623

The PTMU discarded all the rows from an input file except the last one when the data contained illegal binary zeroes. In V6.11.xC6, the presence of these characters in an input file results in an error and the load terminates.

- CSL #338630

On Windows platforms, some rare error conditions in the ALTER DATABASE CREATE VERSION LOG statement caused a fatal error in the server.

- CSL #3428714

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.

## B. Query-Related Issues

- CSL #116928

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 V6.11.xC6. 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 #221831

The RISQL Entry Tool did not localize and display the decimal symbol correctly for databases with a Swedish locale.

- CSL #231754

After a database was upgraded from V5.1.7 to V6.0.4, a query no longer completed because of an inefficient preliminary plan.

- CSL #229330

Complex queries with many predicates could absorb large amounts of CPU and memory during compilation. The resources required to compile such queries have been dramatically reduced.

- CSL #249144

Queries with predicates involving the TO\_CHAR function returned incorrect results.

- CSL #249734

Queries and DML operations sometimes returned incorrect results when they used a multi-column index with constraints on more than one of its leading columns.

- CSL #252054

Queries sometimes returned incorrect results when a literal value was projected from an operand of an outer join. A literal result was sometimes NULL when it should not have been NULL, or vice versa, and the number of rows returned might have been incorrect if a predicate referenced such a literal.

- CSL #266116

Certain types of TARGET scans (those with empty blocks in lists of operands subject to OR processing) failed with an internal error.

- CSL #268908

Queries with predicates on TARGET-indexed LONGNUM columns that require complex internal processing produced wrong results.

- CSL #272855

In some cases, the server returned a fatal error when a subquery contained a complex expression in the select list, was grouped by the alias of this complex expression, and the query qualified for a set operation rewrite or a precomputed view rewrite.

- CSL #292480

A rewritten aggregate query with a subquery inside an IN predicate resulted in an assertion failure during Advisor candidate view generation.

- CSL #293030

When a subquery contained an aggregation function but no grouping or DISTINCT processing was required, the query returned a fatal error.

- CSL #301578

A precomputed view that was dropped and re-created was re-created as Valid instead of Invalid.

- CSL #306898

In some cases, a macro definition was expanded, causing the macro text to exceed the maximum length (1024 bytes).

- CSL #319399

If the QUERYPROCS parameter was set to a value greater than 10, execution of a parallel query could cause a fatal error and subsequent Red Brick Warehouse connections would fail if the client process was terminated by the user.

- CSL #320363, 338440

When a fact table was joined to dimension tables using a merge sort instead of a bit-vector sort, the query could return a fatal error.

- CSL #323587

In some cases, performance was degraded or incorrect results were returned when a query contained a UNION ALL, INTERSECT ALL, or EXCEPT ALL set operator and the set operation was rewritten.

- 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 #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 V6.11.xC6, a STARjoin is chosen over a TARGETjoin when it is the better choice.

- 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 V6.11.xC6, 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 more details about this SET statement, see page 34.

- 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. In V6.11.xC6, the value in the constraint is converted to the data type of the indexed column.

- 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 #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.

## **C. Connectivity Issues**

- CSL #236754

The ODBC Driver did not accept standard ODBC escape syntax for SQL\_DATE, SQL\_TIME, and SQL\_TIMESTAMP parameters.

- CSL #958220

Client tools would hang indefinitely when an incorrect but active ODBC port number was specified in a Red Brick DSN.

- CSL #233340, 241109, 246236, 224504

The ODBC Driver did not support the conversion of decimal data types to SQL\_C\_NUMERIC.

- CSL #268811

ODBC scalar functions inside queries generated by client tools returned syntax errors when a V2.1x Client Connector Pack was used with a V6.0.x server.

- CSL #224504

A Visual Basic application that was compatible with the ODBC Driver shipped with Client Connector Pack V1.0.6 returned an error when used with the driver shipped with V2.10.

- CSL #235233

The JDBC Driver failed to translate the Japanese character 0x92F7 in the MS932 code set.

- CSL #242973

When an application using the JDBC Driver prepared and executed a query multiple times, the process consumed a large amount of memory, resulting in slow performance or a StackOverflowException error.

- CSL #253699, 275303

The JDBC Driver could not be used with client locales other than US-ASCII, Japanese, and Chinese.

- CSL #262077

The Red Brick ODBC Driver returned a data type out-of-range error when a table with a SERIAL column was linked to a Microsoft Access database.

- CSL #287568

When the Red Brick server was running on a Windows platform and an application opened and closed a JDBC or ODBC statement without fetching any data, memory was not released.

- CSL #286979

As sessions timed out because of the OPTION IDLE\_TIMEOUT parameter setting, system memory was not released.

- CSL #298870

Previous versions of the release notes did not identify the Windows platforms supported by the Client Connector Pack. See the Release Notes for the Client Connector Pack.

- 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 #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.

## D. Administrator Tool Issues

- CSL #15743

In the Database tab, the Synonyms folder could not be expanded; no plus sign (+) was available.

- CSL #17204

When you right-clicked on the Constraints folder for a specific table in the Database tab, the Create option did not display the expected wizard.

- CSL #18106

The ISQL window failed to display rows when the result set formed a very wide table.

- CSL #22717

The Show DDL command produced incorrect range specifications for segmented STAR indexes. The MIN and MAX keywords were replaced with default segment names.

- CSL #993625

The Manage Tables wizard for altering a serial column generated invalid SQL.

- CSL #246769

The Manage Tables wizard for adding a column did not allow default values in uppercase.

- CSL #248624, 276623

The Show DDL command added the primary index RANGES keyword to the CREATE TABLE statement when no segmentation ranges were specified.

## E. Documentation Issues

- CSL #257758

The **-e** and **-f** options of the **rbwlogview** command do not display any output unless they are used in conjunction with either the **-a** option (active log file) or a named log file.

- CSL #9599

An enhancement to the EXPLAIN documentation was requested. See page 6.

- CSL #272027

The maximum number of rows that can be unloaded with the TMU ( $2^{31} - 1$ ) was not documented. See page 38.

- CSL #291488

An EXPORT command to a mapped drive on a Windows machine fails if the run-time user does not have write access to the drive; see page 34.

- CSL #293436

A known issue regarding the use of the ODBC SnapshotOnly option with Microsoft Access was not documented; see page 21.

- CSL #320528

The *Administrator's Guide* incorrectly states that the RK and RD lock types apply to versioned databases only. See page 28.

- CSL #334789

The NODE\_NAME column was not documented in the description of DST\_SESSIONS. For details, refer to page 30.

- CSL #344447

A documented workaround for displaying Japanese Gaiji characters no longer applied. This workaround has been removed from the release notes.

## X. Known Issues

This section describes known issues in IBM Red Brick Warehouse V6.11.xC6 and is organized as follows:

- Database Administration Issues
- Query-Related Issues
- Connectivity Issues
- Administrator Tool Issues

### A. Database Administration Issues

#### Version Log Must Be Cleaned Before a Partial Restore

The TMU does not wait for the version log to be cleaned before starting a partial restore; therefore, the vacuum cleaner could be cleaning the version log during the partial restore operation. This scenario could leave the database in an inconsistent state. To avoid this problem, users should clean the version log before starting a partial restore by issuing the following ALTER DATABASE command:

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

If the database contains any damaged segments, this command will not proceed. In this case, users should issue the following command:

```
alter database clean version log remove damaged segments;
```

#### 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, refer to 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.

#### Enabling Parallelism for a STARjoin

The ROWS\_PER\_JOIN\_TASK parameter enables parallelism for the STARjoin. If the number of join tasks is less than 1, then parallelism is not enabled for both the fetch and join phases. For more details, refer to “Enabling Parallelism for a STARjoin” in Chapter 11 of the *Administrator’s Guide*.

#### Incorrect Parallel Query Statistics Returned by the readacct Program

The **readacct** program does not report statistics correctly when queries run in parallel. IBM recommends using the DSTs or the SET STATS INFO command to return accurate information for parallel queries.

## 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*.

## B. 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.

## C. Connectivity Issues

### Microsoft Access Dynaset Mode

When the Microsoft Access query tool is used with the dynaset capability enabled, performance might be slow on certain tables with multi-column primary keys. If you encounter slow performance, disable the dynaset capability by changing the SnapshotOnly ODBC option setting to 1, which sets the query mode to SnapshotOnly. Note that results from SnapshotOnly mode queries, unlike results from dynaset mode queries, cannot be updated.

To change the setting in Microsoft Access 2.0, add the following line to the [ODBC] section of the WINDOWSMSACC20.INI file:

```
SnapshotOnly = 1
```

To change the setting in Microsoft Access 7.0 or later, use the Registry Editor to disable the dynaset capability by following these steps:

1. Start the Registry Editor by entering the following command at the MS-DOS prompt:

```
c:\> regedit
```

2. Navigate to the \Engines\ODBC folder. The location of this folder depends on your Windows operating system and Microsoft Access version. For example:

```
\HKEY_LOCAL_MACHINE\Software\Microsoft\Access\7.0\Jet\3.0\Engines\ODBC
```

or

```
\HKEY_LOCAL_MACHINE\Software\Microsoft\Jet\4.0\Engines\ODBC
```

3. Double-click **SnapshotOnly** and change its value to 1.

If the SnapshotOnly parameter does not exist in the ODBC folder, create it and modify it as follows:

1. From the Edit menu, select Edit, then New, then DWORD Value.
2. Enter **SnapshotOnly** for the DWORD name.
3. From the Edit menu, select Edit, then Modify.
4. Change Value data to 1 and click the OK button.

When linking tables from a Red Brick database with Microsoft Access 7.0, do not select any key columns. Doing so will cause poor performance on that table. For more information about configuring the ODBC settings for Microsoft Access, refer to the Microsoft Access Help topic "ODBC Settings."

## 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. If you encounter this problem, upgrade your application to JDK 1.3.

## 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 with the JavaSoft JDK, the JDBC driver will 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."

## Processing Cubes with Microsoft SQL Server 7.0 OLAP Manager

Before using the OLAP Manager to build a cube from tables in a Red Brick database, choose either MOLAP (the default) or HOLAP for data storage. The ROLAP option is not supported with Red Brick databases.

## D. Administrator Tool Issues

### Validate Hierarchy Function Uses Incorrect Query

The Validate function in the Create Hierarchy wizard runs a query that works only if the *to\_column* of the hierarchy does not allow NULL values. The query that should be used for validation is shown on page 24.

### Show DDL Function Generates Incomplete Foreign-Key Constraint

The Show DDL function generates only the first column name when a foreign-key reference consists of multiple columns.

### Show DDL Function Does Not Include Blank Default Values

The Show DDL function does not declare the default value for a column if the value was defined as one or more blanks (spaces). For example, the default value for column C2 is missing in the following Show DDL:

CREATE TABLE Statement

```
create table T1(  
  C1 CHAR(1) default '*',  
  C2 CHAR(1) default ' '  
);
```

Show DDL Output

```
create table T1(  
  C1 CHAR(1) default '*',  
  C2 CHAR(1)  
);
```

### 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.

## XI. Documentation Notes

The following documents are new for V6.11:

- *TMU Backup and Restore User's Guide*
- *Explaining EXPLAIN User's Guide*

These documents are available in PDF format only, on the software CD-ROM.

The following sections contain corrections to the information in the published manuals and descriptions of new features in V6.11.

### A. TMU Backup and Restore User's Guide

**Add the following note to the discussion of partial restores on page 48:**

After a damaged segment is restored with a TMU partial restore operation, the segment is still marked as "damaged" in the system catalog even though the physical PSUs of the segment are restored and intact. Users should mark the segment as "intact" by issuing the following command:

```
alter segment segment_name of table table_name force intact;
```

**Add the following paragraph to the "Valid Operations" section on page 14:**

If your database is versioned, you must clean the version log (ALTER DATABASE CLEAN VERSION LOG) before altering the backup segment with the MIGRATE TO option.

### B. Explaining EXPLAIN User's Guide

**Add the following parameter setting to the list on page 18:**

- ROWS\_PER\_SCAN\_TASK = 5000

### C. Vista User's Guide

**On page 2-20, the query for validating hierarchies works only if the *to\_column* of the hierarchy does not allow NULL values.**

The correct query is as follows:

```
select from_col, count(*) from
  (select from_column, to_column
   from table_name group by from_column, to_column)
 as dist_subq(from_col, to_col)
group by from_col
having count(*) <> 1;
```

If the hierarchy is valid, this query returns no rows.

### D. Administrator's Guide

**1. Update the "Database Limits" section on page 1-20:**

A segment can contain a maximum of **256** files, and a file can be a maximum of 2 gigabytes. Therefore, a segment can contain a maximum of **512** gigabytes.

**2. Fix the name of the .ini file on page 2-33:**

On Windows, the client locale is specified in the **rbodbc32.ini** file.



3. On page 2-35, remove the second scenario about a French-speaking user connecting to a database with a German locale; this configuration is not supported.

4. Replace the following sections on page 4-38:

#### THRESHOLD Value

The threshold specifies how much memory is used before spilling the intermediate results from an index build operation or CHECK INDEX VALIDATE FULL operation to temporary space on disk.

The default value for INDEX\_TEMPSPACE\_THRESHOLD is 10 megabytes (10M) to accommodate small systems. If you want to change this threshold value, use the SET INDEX TEMPSPACE statement with the THRESHOLD option and keep the default value in the `rbw.config` file.

The goal in selecting a threshold for INDEX\_TEMPSPACE is to select a large value relative to the system on which IBM Red Brick Warehouse is running but not so large a value that errors occur. Selecting too large a value—a value that exceeds available memory—might cause the operation to fail with an out-of-memory error. Selecting too small a value might cause poor performance because of time spent writing to disk. You want to estimate a reasonable threshold for both online and offline load operations and other index-building operations.

#### To estimate the threshold in Windows

1. Determine the amount of physical memory on your computer. The maximum physical memory on Windows is 2 gigabytes.
2. Select a value for INDEX\_TEMPSPACE\_THRESHOLD that is appropriate for the type of operation you are performing. ♦
  - If other users are not running queries concurrently while you load tables, build an index, or execute CHECK INDEX with the VALIDATE FULL option, IBM recommends that you set the threshold value to half of the physical memory size.
  - If you expect other users to query the database while you load tables, build an index, or execute CHECK INDEX with the VALIDATE FULL option, set the threshold value to no more than one-quarter of the physical memory on the computer.

Keep in mind that INDEX\_TEMPSPACE\_THRESHOLD is the amount of memory per user. Therefore, if you run CHECK INDEX with the VALIDATE FULL option while several users are running memory intensive operations at the same time, divide the threshold setting by the number of users.

For example, if two users run memory intensive operations while you run CHECK INDEX with the VALIDATE FULL, divide the threshold setting by three. If the physical memory size of your computer is 256 megabytes, one-quarter is 64 megabytes. Therefore, set the threshold to one-third of 64, which is 22:

```
SET INDEX TEMPSPACE THRESHOLD 22M
```

The value you enter is automatically rounded up to the nearest 8-kilobyte block (24 in this example). ♦

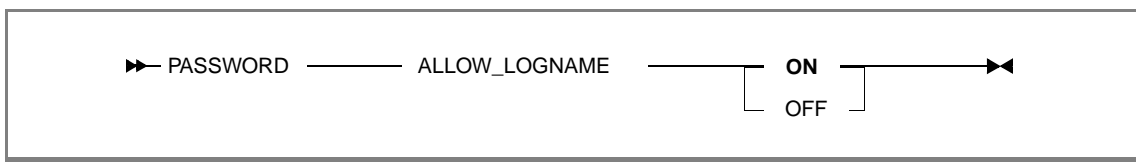
5. Add the following information to the “Administering Password Security” section on pages 7-27 and 7-33:

The following table lists the new password parameters and describes their functions.

PASSWORD Parameter	Description
ALLOW_LOGNAME	Allow password to include the user’s login name.
CHANGE_INITIAL	The user must change the initial password when the user logs in for the first time.
COMPLEX_NUMERIC	Changed password must differ from previous passwords by more than the value of a single numeric.

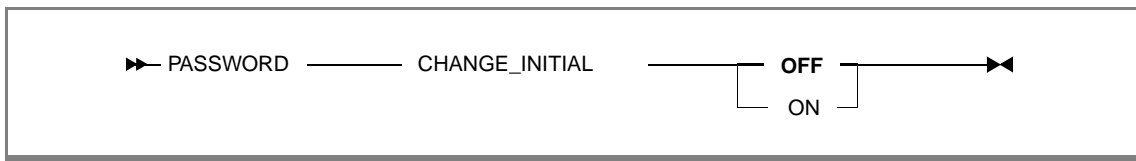
### Restricting Use of Login Name in Password

You can restrict users from using their login name as part of their passwords by setting the PASSWORD ALLOW\_LOGNAME parameter in the **rbw.config** file. Set this parameter with the following syntax:



### Requiring User to Change Initial Password at First Login

You can force users to change their initial passwords when they first log in by setting the PASSWORD CHANGE\_INITIAL parameter in the **rbw.config** file. Set this parameter with the following syntax:

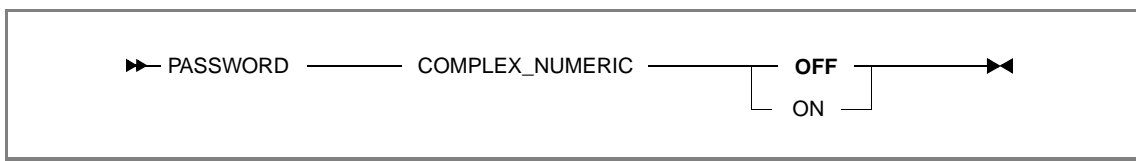


Change this parameter to ON if you want users to change their initial password right away.

### Enforcing Differences From Previous Passwords

You can prevent users from changing only the numeric portion of previous passwords by setting the PASSWORD COMPLEX\_NUMERIC and PASSWORD RESTRICT\_PREVIOUS parameters.

The PASSWORD COMPLEX\_NUMERIC parameter determines if changed passwords must differ from previous passwords by more than the value of one or more numerics. Set this parameter in the **rbw.config** file with the following syntax:



You must set this parameter in combination with the PASSWORD RESTRICT\_PREVIOUS parameter to ensure that users change more than just one or more numerics to another numeric in their previous passwords. If you do not specify the RESTRICT\_PREVIOUS parameter, it defaults to a value of 0, which indicates no restrictions for re-creating old passwords.

The PASSWORD COMPLEX\_NUMERIC restriction applies to the database administrator (and to users with the USER\_MANAGEMENT task authorization) only with respect to their own passwords. They can assign a user's previous password to the same user at any time.

## Examples

Suppose the **rbw.config** file contains the following entry:

```
PASSWORD ALLOW_LOGNAME OFF
```

Users must not use their login name in their passwords. Therefore, the following GRANT CONNECT statement returns an error message when issued by **owen**:

```
grant connect to owen with 'pw2owen1' ;  
** ERROR ** (1043) New password for user owen must not contain the login name.
```

Suppose the **rbw.config** file contains the following entries:

```
PASSWORD RESTRICT_PREVIOUS 4  
PASSWORD COMPLEX_NUMERIC ON
```

Suppose also that **user1** and **user2** execute the following GRANT CONNECT statements to create valid passwords:

```
grant connect to user1 with 'sample1' ;  
grant connect to user2 with 'other1samp2' ;
```

When users change their passwords, the passwords must differ by more than any numeric replaced with a different numeric. Therefore, the following GRANT CONNECT statements return error messages when issued by **user1** and **user2**, respectively, because only the numeric portion of the previous passwords was changed to form the new passwords:

```
grant connect to user1 with 'sample3' ;  
grant connect to user2 with 'other2samp4' ;
```

If the DBA executes the above GRANT statements, they complete successfully.

## 6. Add the following notes to the rbwlogview option descriptions on page 8-18:

The **-e** and **-f** options must be either preceded by the **-a** option or followed by a named log file. Otherwise, the **rbwlogview** command will not display any output.

**7. Replace the “Schema Locking” section on page 9-7:**

**READER\_PRIORITY Option**

The order in which locks are granted is controlled by the `READER_PRIORITY` option in the `rbw.config` file. By default, this option is set to `ON`, which gives priority to read locks at the expense of write locks. For example, if a user has a read lock on a table and another user requests a read lock on the same table, that request is granted immediately, even if a third user is requesting a write lock on the table.

Although this behavior is appropriate for most decision-support applications, heavy query loads can cause any concurrent DDL operations to be significantly delayed because each new request for a database read lock is given higher priority than requests for database write locks. And when queries are executed, they must acquire a database-level lock on the system tables, which is not released until the individual tables are locked.

For applications that support activities such as DDL operations and loads while the database is online, setting `READER_PRIORITY` to `OFF` disables query-priority locking. Operations get locks in the order for which they are applied, with no priority given to read locks for queries. Unless you routinely encounter problems with delayed DDL operations that must be performed while the database is online, IBM recommends that you leave `OPTION READER_PRIORITY` set to `ON`, giving queries priority over operations that write to the database.

**8. Replace the RK and RD rows in the table on page 9-8 with these rows:**

Lock Type	Definition	Used for This Type of Transaction
RK	Read-key	Read transaction that does not allow other transactions to change primary key values in the table.
RD	Read-data	Read transaction that does not allow any write transactions to the table.

**9. Update “Copying or Moving a Database” in Chapter 9:**

Using the `rb_cm` utility for migrating between different versions of IBM Red Brick Warehouse is supported only if the data is transferred in `TMU EXTERNAL` format.

**10. Update the Change column in the table on page 9-53.**

Delete the `MAPFILE` parameter.

Move the `DEFAULT_SEGMENT_SIZE` and `DEFAULT_PSU_EXTENDSIZE` parameters from page 9-53 to page 9-54.

**11. Add the following information to page 9-63.**

**The rbwservice Utility**

The `rbwservice` utility is used in scripts to control the Red Brick Service in the following ways:

- Start the service
- Stop the service
- Create the service
- Delete the service
- Query the service

The syntax to invoke the `rbwservice` utility is as follows:

```
rbwservice [\\machine] {-t <type>} {-h <host>} [-p <password>] -{<option>...}
```

You must maintain the order of the arguments as given above. The server returns an error if you specify an argument in the incorrect order.

There are five possible arguments:

- machine

The machine on which the service exists or will exist. If no machine name is given, the local machine is used.

- -t <type>

The service type.

- RBW - Red Brick Warehouse Service
- CM - Red Brick's Copy Management Service

- -h <host>

The Red Brick host name

- -p <password>

The Password of the REDBRICK user account. Use the REDBRICK user account to create the service. It is only required for the create option. If you do not provide the password, the SYSTEM local account is used.

- <option>

The following five options are possible.

- Create
- Delete
- Start
- Stop
- Query

To display the **rbwservice** syntax, enter **rbwservice** with no options at the system prompt.

## 12. Update the “TEMPSPACE” section on page 10-12:

On Windows platforms, if no temporary-space directories are defined, the following temporary file path is specified for the default directory:

- The path specified by %TMP%.
- The path specified by %TEMP%, if %TMP% is not defined.
- The Windows directory, if both %TMP% and %TEMP% are not defined.

## 13. Change the following section on pages 10-58 and 10-59.

### Example

The following example shows the output from EXPLAIN for a simple query against the **Sales** and **Promotion** tables from the **Aroma** database:

```
RISQL> explain select sales.promokey, dollars
>from promotion, sales
>where sales.promokey = promotion.promokey;

EXPLANATION
[
- EXECUTE (ID: 0) 2 Table locks (table, type): (PROMOTION, Read_Only), (SALES,
Read_Only)
--- EXCHANGE (ID: 1) Exchange type: Table Scan
---- TABLE SCAN (ID: 2) Table: SALES, Predicate: <none>
]
RISQL>
```

In this example, the report shows read-only locks on the **Promotion** and **Sales** tables. A scan of the **Sales** table is used because all of the information requested can be read from that one table.

**14. Correct the default value for MAX\_SERVERS on page B-10:**

Default: **10** (base 10 integer)

**15. Remove the following sentence from the READER\_PRIORITY description on page B-12:**

Schema locks can be acquired at the table level, instead of relying on a database-level lock on the system tables.

**16. On page B-15, add the File Element OPTION PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS to the Configuration Information table:**

<p>OPTION PREPLANS_HAVE_ONLY_PKFK_JOINS</p>	<p>Determines whether tables that are not joined by primary-key to foreign-key join predicates are included in STARjoin preliminary plans. When set to <b>ON</b>, only tables that are joined by their referential integrity relationship (primary-key to foreign-key) are included in the preliminary plan of a STARjoin.</p> <p>This option also applies to subqueries in the FROM clause; by default, subqueries have no referential integrity relationship to other joined tables.</p> <p>Default: <b>OFF</b></p>
---	---

For information on the SET statement for this option, refer to page 34.

**17. Replace the maximum value for TUNE TMU\_BUFFERS on page B-16:**

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.

**18. Correct the default value for INDEX\_TEMPSPACE\_THRESHOLD on page B-18:**

The default threshold for index-building operations is 10 megabytes, as documented on page 2-20 of the *Table Management Utility Reference Guide*. The maximum value for this parameter is 2 gigabytes.

**19. On page B-25, change the minimum value for ADVISOR\_LOG\_MAXSIZE to 10 kilobytes (10,240 bytes).**

**20. On page C-19, change the column type for MAXSIZE\_ROWS to DECIMAL(18, 0) in the RBW\_TABLES table.**

**21. Change the description of NODE\_NAME in the DST\_COMMANDS table on page C-26 and the DST\_LOCKS table on page C-31; add NODE\_NAME to the DST\_SESSIONS table on page C-32:**

<p>NODE_NAME</p>	<p>Host name where the session is running.</p>
------------------	--

**E. Installation and Configuration Guide for Windows Platforms**

**1. Add the following information to step 3 under “Removing Red Brick Decision Server” on page 4-4.**

On installation, two services are created if they do not exist. One is a service for Red Brick Decision Server and the other is a service for the copy management utility. On uninstallation, these services are not removed as they may be used by another installation. If you want to completely remove a Red Brick installation including these two services, you must perform the following steps:

1. Remove the service entry from the registry using the **rbwservice** utility.
2. Set the RB\_HOST and RB\_CONFIG environment variables.
3. Specify the **-t** and **-h** parameters.
4. Query the service to ensure that it is running.

```
C:\WINNT\Profiles\YOURSERVICE\Desktop>rbwservice -t rbw -h YOURSERVICE -query
Interrogating service RedBrickServiceYOURSERVICE ...
Red Brick Service RedBrickServiceYOURSERVICE is running.
```

5. Stop the service.

```
C:\WINNT\Profiles\YOURSERVICE\Desktop>rbwservice -t rbw -h YOURSERVICE -stop
Stopping service RedBrickServiceYOURSERVICE ...
Stopped service RedBrickServiceYOURSERVICE
```

6. Delete the service.

```
C:\WINNT\Profiles\YOURSERVICE\Desktop>rbwservice -t rbw -h YOURSERVICE -delete
Deleting Service RedBrickServiceYOURSERVICE ...
Deleted service RedBrickServiceYOURSERVICE
```

7. Stop the copy management utility.

```
C:\WINNT\Profiles\YOURSERVICE\Desktop>rbwservice -t cm -h YOURSERVICE -stop
Stopping Service RedBrickService-CopyManagement ...
Stopped service RedBrickService-CopyManagement
```

8. Delete the copy management utility.

```
C:\WINNT\Profiles\YOURSERVICE\Desktop>rbwservice -t cm -h YOURSERVICE -delete
Deleting Service RedBrickService-CopyManagement ...
Deleted service RedBrickService-CopyManagement
```

9. Uninstall the software.

**Important:** You must use the **rbwservice stop** command and remove the service entry before you uninstall the software.

For more information about the **rbwservice** utility, see page 28.

## 2. Add the following file descriptions to the table of directories and files on page A-3.

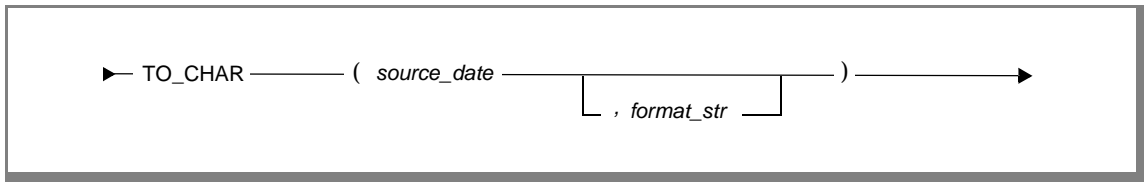
\bin\MSVCRT.dll	Shared library for Microsoft Visual C Run-Time
\bin\MSVCRT40.dll	Shared library for Microsoft Visual C++ 4.1 C run-time, for backward compatibility
\bin\Rbwping.cnt	Contents file for RBPing online help
\_UNODBC.log	File that contains information to uninstall the ODBC Driver

## F. SQL Reference Guide

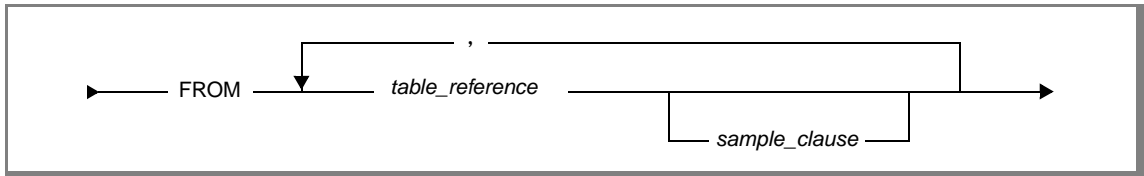
1. Replace the Usage Notes paragraph for the REAL data type on page 2-27:

Values of the real data type are single-precision and range between approximately 1.17549435E-38F and 1.E37 (minimum range; the maximum range depends on the host platform).

2. Make the following changes to the TO\_CHAR syntax diagram on page 5-69.



3. Make the following change to the FROM clause syntax diagram on page 7-23.



Each individual table reference can have a sample clause.

4. Replace the description of *override\_fullindexcheck\_specification* on page 8-32 with the following description:

Detaches (or clears) a data segment from the specified table without performing a potentially time-consuming full-index scan to remove index entries for the table segment. When an index segment is segmented exactly the same as the table segment, you can use this clause to name the corresponding index segments to remove the index entries by clearing the index segment rather than scanning the index. This option saves time when you are detaching a data segment from a very large fact table (more than a billion rows).

The Red Brick server does a complete check that the index segment corresponds to the data segment and issues an error message if it does not.

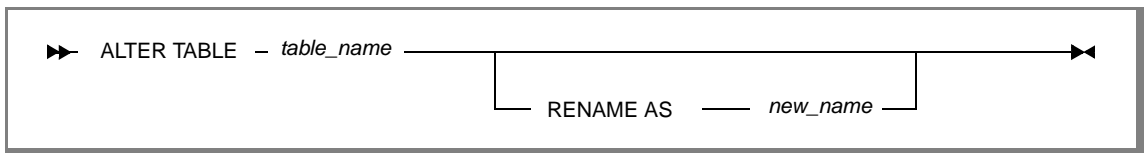
5. Remove the following information from page 8-39:

In the description of the ALTER SEGMENT OPTICAL command, the paragraph that begins, "If you are using HP OmniStorage 2.2 software..." no longer applies.

6. Make the following changes to ALTER TABLE on pages 8-66 and 8-67:

On page 8-66, the DEFAULT keyword and the values that follow it in the ALTER TABLE...ADD COLUMN syntax diagram are not required. On page 8-67, the description of the DEFAULT keyword should state that a default value is required for all data types except the SERIAL data type. Default values are not allowed for SERIAL columns.

7. Add the following RENAME AS option to the ALTER TABLE syntax on page 8-61:



## RENAME AS

Renames the base table specified by *table\_name* with *new\_name*. The new name cannot be the name of any existing temporary, permanent, or derived table or the name of a synonym.

Synonyms can be renamed, and synonyms for tables remain valid after their tables have been renamed.

The following objects cannot be renamed:



- System tables
- Dynamic Statistics Tables (DSTs)
- Model tables
- Derived tables such as views or aggregates
- Tables referenced by views. All views defined on a table must be dropped before the table can be renamed.

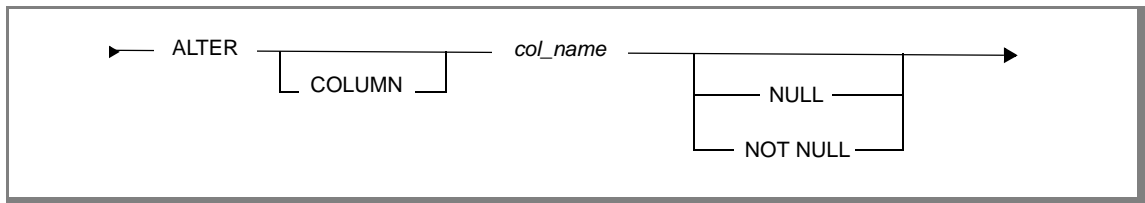
Tables in a versioned database cannot be renamed if the query revision is frozen.

Users must have permission to rename tables, which depends on their permission to perform other DDL tasks. For example, a user who has permission to create temporary tables but not permanent tables has permission to rename temporary tables but not permanent tables.

When you rename a table:

- All primary-key and foreign-key constraints continue to be honored. Constraint names and index names are not automatically updated to reflect the new table name. The output of queries against the RBW\_RELATIONSHIPS, RBW\_CONSTRAINTS, and RBW\_INDEXES tables might be misleading due to the default system-generated names for constraints and primary key indexes; these names have the original name of the table embedded in them.
- Macro text is not updated dynamically; therefore, macros referencing renamed tables will return errors or incorrect results.

**8. Add the following options to the ALTER TABLE ALTER COLUMN syntax on page 8-70:**



**NULL | NOT NULL**

Changes the specified column from “not nullable” to “nullable” and vice versa. The specified table must be a base table. You cannot alter columns to NULL or NOT NULL if they belong to:

- System tables
- DSTs
- Temporary tables
- Model tables
- Synonyms
- Derived tables, such as views or aggregate tables

The user must have the usual DDL permissions in order to alter a column to NULL or NOT NULL.

Columns cannot be changed to NULL or NOT NULL if:

- The database is versioned and the query revision is frozen.
- Precomputed views reference the column to be altered.

**Restricted ALTER COLUMN...NULL Operations**

If any of the following conditions exist, an ALTER COLUMN NULL operation fails with an error. The column must remain NOT NULL if it is:

- Declared UNIQUE.
- A SERIAL column.
- A segmenting column.
- A member of a primary key.
- A member of a foreign key.
- A member of a multi-column B-TREE index.
- The “From” column in a hierarchy.

### Restricted ALTER COLUMN...NOT NULL Operations

Columns cannot be changed to NOT NULL if they already contain NULL values. Therefore, when the ALTER COLUMN NOT NULL statement is issued, the whole table must be checked with an index scan or a table scan. If any NULL values are found in the specified column, the operation fails with an error. Since the scan might be time-consuming, IBM recommends that you verify the contents of the column with a query before issuing the ALTER COLUMN statement.

#### 9. Add the following paragraph to the descriptions of WITH FILLFACTOR on pages 2-19 and 8-67, and to CHANGE FILLFACTOR on page 8-73:

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.

#### 10. Add the following information to the EXPORT description on page 8-189:

If you intend to use the SQL EXPORT command to export data from a Windows server to a mapped drive on another Windows computer, you must install IBM Red Brick Warehouse as a local user who has permission to write to that drive. (This user must also have the group membership and privileges described on page 1-10 of the *Installation and Configuration Guide for Windows Platforms*.) The System user will not be able to export data to a mapped drive on a Windows machine.

If you have already installed IBM Red Brick Warehouse as the System user, a workaround is to change the Red Brick service to “run as” a local user with the appropriate privileges and write permissions. To make this change on a Windows NT computer, select Services from the Windows Control Panel, then select the Red Brick service from the list and click Startup. In the Startup dialog box, change “Log On As System Account” to “Log On As This Account: *username*.” On Windows 2000 computers, you can make the same change by selecting Administrative Tools, then Services from the Control Panel.

#### 11. Add the following section to page 9-46

### SET PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS

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.

### Syntax

The following syntax diagram shows how to construct a SET PREPLANS\_HAVE\_ONLY\_PKFK\_JOINS statement.



## Usage Notes

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.

**Important:** 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.

## 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
')
}

```

## 12. Replace the description of the **THRESHOLD** parameter of the **SET INDEX TEMPSPACE** statement on page 9-34 with the following description:

Specifies the amount of memory used before the intermediate results from index-building or CHECK INDEX...VALIDATE FULL operations are written to disk. For operations involving multiple indexes, this threshold value is allocated equally among the indexes being built. The default value is 10 megabytes (10M).

The size can be specified as kilobytes (K) or megabytes (M) by appending `K` or `M` to the number. Note that no space is allowed between the number and the unit identifier: for example, `1024K`, `500M`.

Before you use the `THRESHOLD` option, note the following guidelines:

- The threshold value should never exceed the physical memory size on the computer.
- IBM recommends that you set the threshold value from one-quarter to one-half of the physical memory on the computer, depending if other users are running queries concurrently:
  - If other users are not running queries concurrently while you load tables, build an index, or execute CHECK INDEX with the VALIDATE FULL option, set the threshold value to half of the physical memory size.
  - If you expect other users to query the database while you load tables, build an index, or execute CHECK INDEX with the VALIDATE FULL option, set the threshold value to no more than one-quarter of the physical memory on the computer.

Keep in mind that `INDEX_TEMPSPACE_THRESHOLD` is the amount of memory per user. Therefore, if you run CHECK INDEX with the VALIDATE FULL option while several users are running memory intensive operations at the same time, divide the threshold setting by the number of users.

For example, if two users run memory intensive operations while you run CHECK INDEX with the

VALIDATE FULL, divide the threshold setting by three. If the physical memory size of your computer is 256 megabytes, one-quarter is 64 megabytes. Therefore, set the threshold to one-third of 64, which is 22:

```
SET INDEX TEMPSPACE THRESHOLD 22M
```

The value you enter is automatically rounded up to the nearest 8-kilobyte block (24 in this example).

The threshold value must be specified before the corresponding MAXSPILLSIZE value is specified; it must precede the MAXSPILLSIZE entry in the **rbw.config** file.

A value of 0 causes files to be written to disk after the first 200 index entries.

There is no THRESHOLD parameter for query-processing operations.

### 13. Update page 9-35:

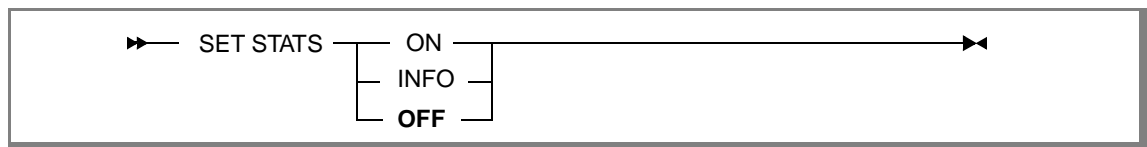
On Windows platforms, if no temporary-space directories are defined, the following temporary file path is specified for the default directory:

- The path specified by %TMP%.
- The path specified by %TEMP%, if %TMP% is not defined.
- The Windows directory, if both %TMP% and %TEMP% are not defined.

## G. Table Management Utility Reference Guide

### 1. Add the SET STATS command to Chapter 2.

The SET STATS statement turns on statistics reporting for the current TMU session. The following syntax diagram shows how to construct a SET STATS statement.



**ON** Returns a summary of statistical information for each load operation.

**INFO** The INFO setting returns the same statistics as the ON setting, along with additional information about the load operation, such as server messages generated during precomputed view maintenance.

**OFF** Turns off statistics reporting. The default setting is OFF.

### 2. Correct the statement about tape support on page 3-117.

On page 3-117, the guide incorrectly states that "Tape input is supported only on Windows." Tape input files are supported only on UNIX platforms, as stated on page 1-4.

### 3. Make the following changes to the OPTIMIZE ON/OFF description on pages 3-58 to 3-59:

**OPTIMIZE OFF, ON** Turns optimize mode on or off. The default behavior is OFF. This setting overrides the global optimize mode setting in the **rbw.config** file for this LOAD DATA statement. In a TMU control file containing multiple LOAD DATA statements, an Optimize clause applies only to the LOAD DATA statement that contains it. Any LOAD DATA statement that does not contain an Optimize clause uses the optimize setting in the **rbw.config** file. If OPTIMIZE OFF mode is used (non-optimize mode), indexes are updated when each input row is inserted into the data file, which provides better performance when the data being loaded contains many duplicate rows.

If OPTIMIZE ON mode is used (optimize mode), the index entries can be inserted as a batch operation. The batch operation is faster because it requires fewer I/O operations. OPTIMIZE ON loads automatically switch to OPTIMIZE OFF when the mode is MODIFY or MODIFY AGGREGATE and a regular column (not a pseudocolumn) is specified in the Criteria clause.

**DISCARDFILE** 'filename' Optional. File into which duplicate records are discarded. A duplicate record contains a value that is the same as the value in an existing row for a column that is declared UNIQUE in the CREATE TABLE statement and is indexed. This definition does not apply to MODIFY and MODIFY AGGREGATE loads, in which case such “duplicates” are not discarded but used to update existing rows and re-aggregate existing rows. These records are only discarded if there is an error during the update or re-aggregation operation.

### 4. Add the following limit to page 4-4:

A TMU UNLOAD operation can unload a maximum of 2,147,483,647 rows ( $2^{31} - 1$ ). If the table you want to unload contains more rows, break the operation into two separate unloads and apply constraints. Alternatively, use the SQL EXPORT command with a SELECT \* query against the table.

### 5. Update the REORG syntax diagram on page 6-12.

In the REORG syntax diagram, the INDEX clause must follow the INCLUDE PRECOMPUTED VIEWS clause. Also, change the EXCLUDES keyword on page 6-13 to read EXCLUDE INDEXES.

After the DISCARDS option on page 6-12, add the ROWMESSAGES option, as documented on page 3-55 of the *Table Management Utility Reference Guide*.

### 6. For information about backup and restore operations, refer to the TMU Backup and Restore User's Guide.

## H. Messages and Codes Reference Guide

The following messages were either added or changed in V6.11.x. Messages for the new TMU backup and restore feature are included in the *TMU Backup and Restore User's Guide*.

**RBS017E** Operation cannot be completed because an aggregate table is locked.

Cause: The ALTER VIEW <view\_name> SET VALID command needs a write lock on the aggregate table(s) to set the validity. However, the aggregate table(s) is locked by some other process, possibly, as part of an update to the detail table, preventing this operation from completing.

Response: Ensure that no other process holds a lock on the aggregate table, and re-attempt the operation.

- RBS072E** Invalid segmentation method for an index.
- Cause: A CREATE INDEX or CREATE TABLE statement was issued with a segmentation scheme for the index that is not supported.
- Response: Correct the error and resubmit the command.
- RBA991E** New password for user `user_name` is not unique within the past `pwd_previous` passwords for this user account.
- Cause: The new password specified for the user matches a password that has been used on this account within the number of previous passwords specified in the PASSWORD RESTRICT\_PREVIOUS parameter in the `rbw.config` file. The password may not be an exact match, but sufficiently close, if PASSWORD COMPLEX\_NUMERIC parameter is specified in `rbw.config` file.
- Response: Specify a password for the user account that does not match or is sufficiently different from one of the previously used passwords.
- RBA1043E** New password for user `user_name` must not contain the login name.
- Cause: The password specified for the user contains the login name of the user. This is not allowed by the password security policy.
- Response: Specify a password that does not contain the login name.
- RBA1044W** The password for user `user_name` must be changed during this session. Otherwise the account will be disabled.
- Cause: The password security policy specifies that the initial password for the user must be changed on the next logon. The warning reminds the user to change the password. Otherwise the account is disabled.
- Response: The password must be changed for the current user during the current session using the GRANT CONNECT command for the same user. The account is disabled until then. If the password is not changed and the account is disabled, the DBA can issue another password to the user.
- RBA1932W** The precomputed view `viewname` has been marked invalid.
- Cause: A table has been updated via an INSERT, DELETE, UPDATE, or load operation, thus invalidating all the precomputed views defined on the table.
- Response: Update the precomputed tables to match the new data in the detail table. After the precomputed tables have been updated to match the data in the detail table, you can then mark the precomputed views valid with either the ALTER VIEW `<view_name>` SET VALID command or the ALTER TABLE `<table_name>` SET PRECOMPUTED VIEWS VALID command.

**RBA1934W** Failed to invalidate all precomputed views.

**Cause:** A REORG operation deleted rows from the table being reorganized, which invalidated all precomputed views defined on the table. However, an error occurred that prevented REORG from automatically marking all of the precomputed views invalid. The most likely error is that REORG was unable to obtain the necessary database locks.

**Response:** Mark all precomputed views invalid with the ALTER TABLE <table\_name> SET PRECOMPUTED VIEWS INVALID command. Alternatively, update all precomputed views, even if they are marked valid, to reflect the rows deleted from the detail table. Then mark all precomputed views valid with the ALTER TABLE <table\_name> SET PRECOMPUTED VIEWS VALID command.

**RBL8562W** TMU Process memory allocation may exceed beyond 1.6 GB. The loader may terminate. Memory allocation estimation- Index Mmapp: mmap\_mem.2fMB, TMU Buffers: buffer\_mem.2fMB, Index Threshold: index\_mem.2fMB.

**Cause:** On WIN NT the process memory limited is @ 2 GB. The above message indicates that the memory allocation may exceed beyond 1.6 GB. The loader may terminate for want of more memory.

**Response:** Take into consideration the memory allocated for Index Mmapping, TMU buffers and Index Threshold. Adjust these parameters.

**RBL8769E** Unable to allocate IPC semaphores for precomputed view maintenance.

**Cause:** Precomputed view maintenance was ON when the LOAD or REORG operation started, but the TMU was unable to allocate necessary IPC semaphores for the precomputed view maintenance operation.

**Response:** Ensure that the system has sufficient IPC semaphore resources available, and resubmit the command.

**RBO10187E** Too few application parameters bound.

**RBO10188E** Could not initialize mutex (errno = os\_error\_number).

**RBO10189E** Could not lock mutex (errno = os\_error\_number).

**RBO10190E** Could not unlock mutex (errno = os\_error\_number).

**RBO10191E** Could not get thread-specific data (errno = os\_error\_number).

**RBO10192E** Could not set thread-specific data (errno = os\_error\_number).

**RBO10193E** The query timeout functionality is disabled for this application.

**Cause:** The query timeout functionality is not available to this application because the configuration parameter DISABLE\_QUERY\_TIMEOUT is enabled.

**Response:** If query timeout is needed, remove the name of this application from the DISABLE\_QUERY\_TIMEOUT option in the rbdcb32.ini file and run the application again.



**RBO10428F** RBWAPID could not start server: see server logs for details.  
Cause: RBWAPID either could not start a new session for the connection or could not fork a new server process.  
Response: Wait a while and then try again. On NT, try reducing the STACK\_SIZE config parameter, if it is specified.

**RBO10699E** Query timed out after *number* seconds.  
Cause: Application set the query timeout value, and the query execution exceeded the timeout value specified.  
Response: Increase the query timeout value.

**RBS12109I** The number of rows for segment *segname* in system table (*sysnum*) does not match actual number of rows found (*actnum*). The segment validation proceeds only if FIX option is specified.

**RBS12368I** The index segment range is invalid. Index must be recreated.  
Cause: The index has segment with invalid segment ranges. Keys may have been inserted in wrong segments. This can produce incorrect answers.  
Response: Drop the index and recreate the index.

**RBS12369I** Segment contains key values outside the range. Index must be reorganized.  
Cause: Key values have been inserted in incorrect segment. This may be due to a segment having invalid range.  
Response: Reorganize the index. If one of the segments of the index also has an invalid range, the index must be recreated.

**RBS12610E** New name defined by ALTER TABLE RENAME already exists.  
Cause: A base table, system table, synonym, view, or another temporary table with the same name as that specified in the ALTER TABLE RENAME syntax already exists.  
Response: Choose another table name and resubmit the statement.

**RBS12611I** Cannot rename table *table\_name* because it is referenced by the following:  
Cause: The named table cannot be renamed because it is referenced in at least one view. This informational message is always followed by one or more occurrences of message 673, listing the views that make use of this table.  
Response: To rename the table, first drop all the views that reference the table, rename the table, then re-create the views.

**RBS12612E** Column '*column\_name*' of table '*table\_name*' already allows NULLs.  
Cause: The user ran an ALTER COLUMN NULL statement on a column that already allows NULLs.  
Response: Make sure that the column was defined NOT NULL.

- RBS12614E** Cannot allow NULLs in column *column\_name* because it is referenced by a hierarchy.
- Cause: The specification for the named column cannot be changed to allow NULLs because it is the 'FROM' column of at least one hierarchy.
- Response: If it is necessary for ALTER COLUMN NULL to run, all hierarchies that reference this column in the FROM definition must first be dropped and then re-created.
- RBS12615E** Cannot allow NULLs in column *column\_name* because it is a segmenting column.
- Cause: An attempt was made to allow null values in the column, but this is not possible because it is a segmenting column.
- Response: Do not attempt this command on the segmenting column of a table.
- RBS12616E** Column *column\_name* of table '*table\_name*' is already defined as NOT NULL.
- Cause: The user ran an ALTER COLUMN NOT NULL statement on a column that was already defined as NOT NULL.
- RBS12617E** Cannot alter column *column\_name* of table '*table\_name*' to NOT NULL: this column has NULL values.
- Cause: An attempt was made to disallow null values in the column, but this is not possible because the column already contains NULL values.
- Response: Drop or modify the rows that contain NULL values in the column and submit the command again.
- RBS12618E** Cannot rename aggregate table '*table\_name*'.
- Cause: The ALTER TABLE RENAME command was issued on an aggregate table. This operation is not allowed on aggregate tables.
- RBS12619E** Cannot alter column *column\_name* of table '*table\_name*' because it is referenced by precomputed views.
- Cause: The ALTER COLUMN NULL/NOT NULL command was issued on a column that is referenced by a precomputed view, which is not allowed.
- Response: Drop all precomputed views referencing this column and retry the statement.
- RBS12620E** Failed to alter column *column\_name* of table '*table\_name*' to NULL: not enough space to create a null block of the B-TREE index on this column.
- Cause: An ALTER COLUMN NULL command was issued on a column, but there is not enough space on index segment 0 to allocate a null block for this column's B-TREE index.
- Response: Increase the index segment size and retry the statement.
- RBS12621E** Failed to alter column *column\_name* of table '*table\_name*' to NOT NULL: failed to remove the null block of the B-TREE index defined on this column.
- Cause: An ALTER COLUMN NOT NULL command was issued on a column but failed. An error occurred when the null block of this column's B-TREE index was removed.

**RBS12622E** NULL values are not allowed for SERIAL column *column\_name*.  
Cause: The user issued an ALTER COLUMN NULL command on a SERIAL column, but SERIAL columns cannot contain NULL values.  
Response: The statement is not allowed.

**RBS12623E** Cannot alter the column: it belongs to aggregate table '*table\_name*'.  
Cause: An ALTER COLUMN NULL/NOT NULL command was issued on an aggregate table. This operation is not allowed on aggregate tables.

**RBJ11706E** I/O error occurred while reading from stream object.  
Cause: I/O error occurred while reading from the specified Reader stream object.  
Response: Please make sure that the specified Reader stream object is available for read.

**RBJ11707E** The specified length: *stream length* is greater than what is available in stream object.  
Cause: The length parameter is greater than the number of characters in the Reader stream object.  
Response: Please make sure that the length specified is the same as the number of characters in stream object.

**RBJ11708W** The specified length: *stream length* is less than the number of characters in stream object.  
Cause: The length parameter is less than what is available in the Reader stream object.  
Response: Please make sure that the length specified is the same as the number of characters in stream object.