

Retek[®] Data Warehouse[™] 11.0

Performance Tuning Guide

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA
888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000
Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom
Switchboard:
+44 (0)20 7563 4600
Sales Enquiries:
+44 (0)20 7563 46 46
Fax:
+44 (0)20 7563 46 10

The software described in this documentation is furnished under a license agreement, is the confidential information of Retek Inc., and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403, and the copyright notice may not be removed without the consent of Retek Inc.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

Retek® Data Warehouse™ is a trademark of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2004 Retek Inc. All rights reserved.

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method Contact Information

E-mail support@retек.com

Internet (ROCS) rocs.retek.com
Retek's secure client Web site to update and view issues

Phone +1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835

Mail Retek Customer Support
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403

When contacting Customer Support, please provide:

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step-by-step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.

Contents

General performance considerations.....	1
Overview.....	1
Architecture.....	2
Best practice data warehouse modeling.....	3
Aggregation.....	4
Referential integrity.....	4
Operating system tuning.....	5
Database tuning suggestions for all database types.....	5
Fragmentation.....	5
Indexing.....	5
Statistics.....	5
Database tuning suggestions for Oracle only.....	6
Init.ora.....	6
Oracle partitioning.....	6
Oracle compression.....	6
RETL tuning.....	6
Data movement – extraction, transformation, and load.....	6
Application tuning.....	6
Batch scheduling.....	6
RDW compression.....	7
RETL configuration.....	7
Parallelism.....	7
Data purging strategy.....	7
Truncate vs. delete logic for retailers using an Oracle database.....	8
MicroStrategy tuning.....	8
Architecture.....	8
Monitoring statistics.....	8
Workload management.....	8
SQL generation and VLDB settings.....	9
Thread management.....	9
Partitioning.....	9
References.....	9

General performance considerations

Overview

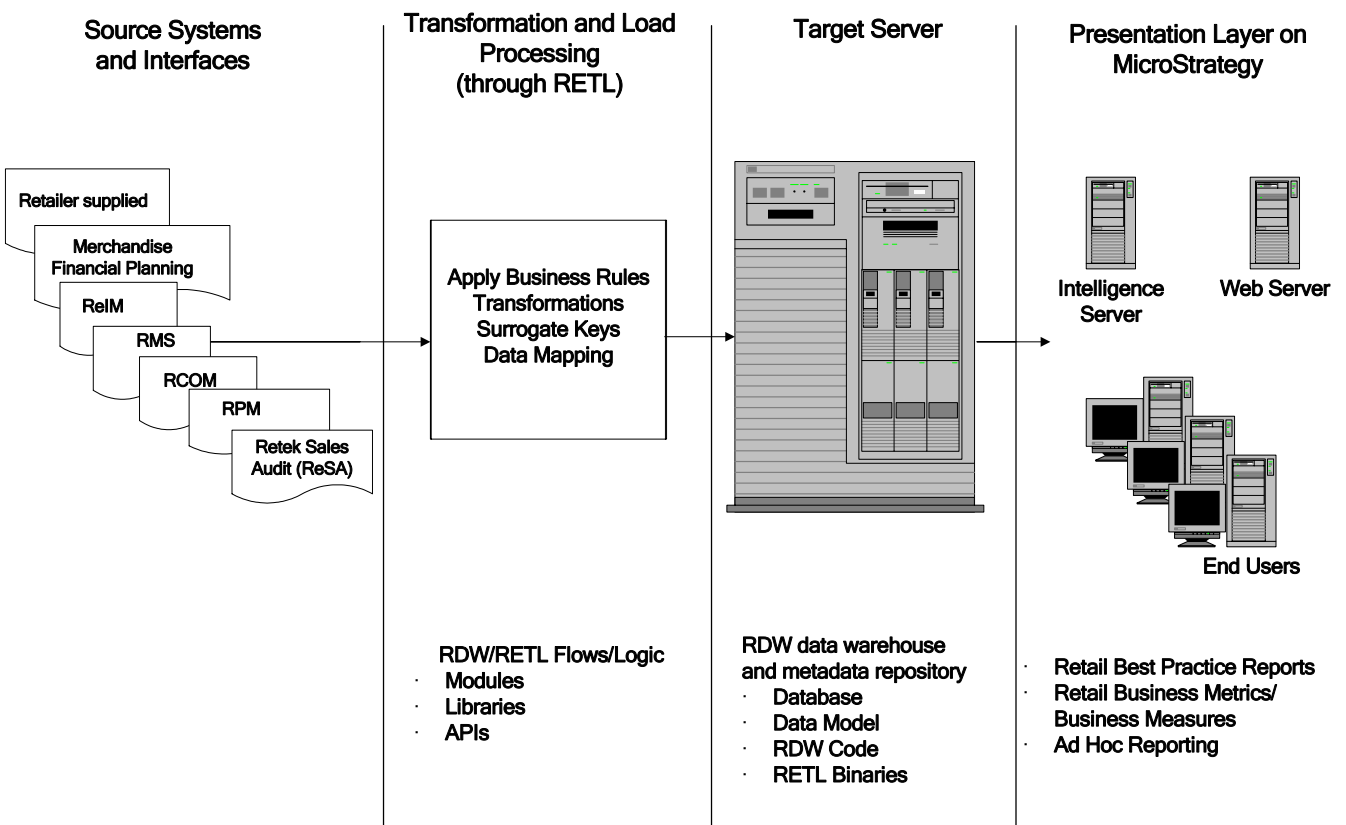
The performance considerations for the Retek Data Warehouse (RDW) begins with understanding the architecture and understanding end users expected response times well enough to plan and maintain the production environment. By understanding the architecture, you will know what layer of the application will need to be adjusted to meet performance goals. For instance, tuning may be needed at the operating system, database, RETL, application, or MicroStrategy layers. Adequate capacity planning for all of the layers is a key step when planning a production size data warehouse.

Monitoring and capturing baseline statistics as well as knowing what has changed in the environment over time is an important process. Retek recommends keeping environment change logs.

To begin tuning the RDW application starts with understanding the layers.

Architecture

The RDW architecture is comprised of a source system and a target system. RDW incorporates optional interfaces with a number of Retek applications, including the Retek Merchandising System (RMS), which can serve as the primary source for the data warehouse. The data warehouse can also operate as a standalone product and be fed from other legacy systems. Regardless of whether a Retek or a non-Retek application is used as the source system, the programs that extract data from the source system reside on the source database located on the source server. The data warehouse server or RDW server is considered the target server. For optimal performance on your production environments, RMS and RDW should run on separate servers. Please note that the programs that are used to extract data from a source system are *not* packaged with RDW. The extraction programs are packaged with the Retek applications that serve as the source systems to RDW.



Best practice data warehouse modeling

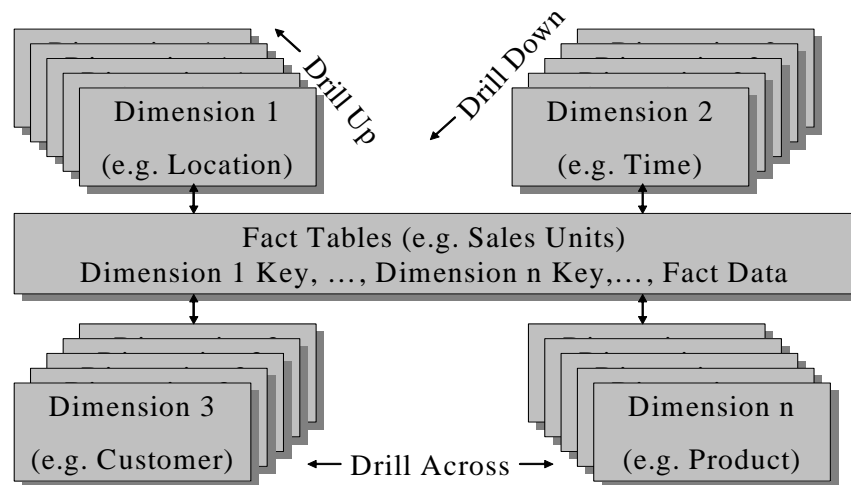
RDW has been developed using a dimensional modeling approach or star schema. The dimensional model is simple to understand, easier to navigate when compared to 3rd normal form, allows for high performance browsing across attributes in a dimension and processing data is more efficient. A star schema optimizes performance by keeping queries simple and providing fast response time.

A fact table starts with the set of base measurements, events, or transactions. A fact table is then surrounded by a set of dimension tables, which is commonly referred to as a star schema. The dimensional model is constructed to reflect the business needs and is used to answer the important business questions, such as,

- How do actual sales this period compare to the current plan?
- What is the retail value of inventory on hand and how does it compare to the same period last year?

RDW facts are the transactions that occur in your data warehouse's source systems, such as RMS. You might want to analyze sales transaction facts, or inventory stock count facts at stores or warehouses, or inventory movement facts.

Facts have little meaning by themselves because they are usually just values, for example: 6 sales at a store, 15 items left at a warehouse, or 300 items transferred. What gives fact data true meaning in RDW is the intersection of dimensions in which facts exist. In other words, 6 sales on Wednesday at store B, or 15 dishwashers in stock last Monday at the Chicago warehouse, or 300 blouses transferred during the last week in February from the St. Louis warehouse to the Denver warehouse. Dimension data, therefore, exists in the data warehouse to serve as reference data to facts.



Aggregation

A standard method used to optimize reporting performance in a data warehouse environment is the creation of aggregate fact tables. These aggregations are sometimes also referred to as 'summaries' or 'roll-ups'. A general rule for providing the quickest response time for an end-user query in a dimensional data warehouse is to hit the smallest (most highly summarized) fact table available.

RDW out of the box can be described as a sparsely aggregated data warehouse. For example in the base product, sales data is captured from the RMS at the item-location-day level and physically stored in the data warehouse. To improve response time for end-user queries, sales data is summarized and also stored at the following aggregation levels:

- Item-location-Gregorian month (if the retailer is using the Gregorian time calendar)
- Item-location-week
- Subclass-location-day
- Subclass-location-Gregorian month (if the retailer is using the Gregorian time calendar)
- Subclass-location-week
- Department-location-day
- Department-location-week

Additional aggregation levels can be added to improve response times for popular end-user queries. For example, if it is observed that many users query the data warehouse for sales data at the department by month level and response time is slower than desired, a new aggregate can be built at the department-location-month level and added to the project.

One place to begin defining requirements for aggregate tables is to make use of the features of MicroStrategy, which gathers statistics on actual performance of front-end reports. Using MicroStrategy reporting tools, it is possible to determine what reports are run most frequently and which reports run the longest. These reports can be considered candidates for tuning. This list of reports should be verified with the business users to ensure that they are in fact the best place to start. Other reports of a higher priority might be rarely used because of slow performance!

Whenever possible, the process of determining what, if any, aggregations should be added to RDW should involve analysis of empirical data from the existing implementation. Technical specialists who understand data warehouse performance tuning within the database and the MicroStrategy tools should perform this analysis. Additional aggregate tables should be added *only* after performance with the existing schema has been optimized.

A general rule when determining an aggregation layer is to use a ratio of 1:20 while keeping in mind your business requirements. For example, the goal is to aggregate or rollup approximately 20 records into one at a higher level (i.e., week, month, department, and so on) to see the benefit.

Referential integrity

RDW relies on the referential integrity already checked in the transactional or source systems (for example, RMS). By not repeating the integrity checks within the data warehouse this saves processing time and burdening resources unnecessarily.

Operating system tuning

The operating system components that are important to note for RDW are memory, kernel parameters, network, and RAID.

Memory and kernel parameter settings are important components when using RETL for data movement. For optimal performance, refer to the *Retek Extract Transform and Load Performance Tuning Guide* for guidelines.

Since RDW moves large amounts of data between servers, a gigabit or better network speed is recommended for best performance.

Considering the ETL process as well as the ad hoc query performance of RDW, RAID 10 (mirror then stripe) is recommended for optimal RDW performance and high availability.

Refer to the *Retek Software Performance Tuning Guide* for additional information and guidelines.

Database tuning suggestions for all database types

Carefully read through the *Retek Software Performance Tuning Guide* to set up the RDW database and its objects. Because of their importance, a few guidelines from that document are repeated here.

Fragmentation

The RDW architecture can cause dimension and matrix tables to become fragmented because all updates are treated as deletes then inserts. If direct load SQL*LOADER is being used (the recommended method) all inserts are above the high water mark of the table. A regular maintenance program is recommended to regain space, this maintenance can include export/import or alter table move.

Indexing

The SQL in MicroStrategy can be influenced but not directly changed. Creating indexes on the fact or dimension tables is a method for tuning a report or SQL statement generated by MicroStrategy. Create indexes conservatively because they will increase the time required for loading the data.

Statistics

Data warehouse tables have significant data volume added during each batch cycle and should be analyzed more often than an OLTP environment. The temp tables and indexes created by the batch processes are analyzed as part of the program logic. All custom batch programs should follow this recommended standard.

Database tuning suggestions for Oracle only

Init.ora

Pay careful attention to the Oracle initialization parameters that allocate the memory for the Shared Global Area and Program Global Area. Some of the important parameters are `DB_CACHE_SIZE`, `SHARED_POOL`, `DB_FILE_MULTIBLOCK_READ_COUNT`, `SORT_AREA_SIZE`, `HASH_AREA_SIZE` and `LOG_BUFFER`.

Oracle partitioning

Create all large fact tables with the proper partition keys to utilize Oracle's table partitioning. The number of partitions for partitioned tables depends upon the volume and purging strategy for the functional area. Typical RDW partitioning is based on range partitioning.

Oracle compression

Retek has not tested Oracle's table compression. The compression at the table level can lower the disk requirements for historical fact tables, but also adds overhead when loading the data. Refer to the Oracle documentation for the specifics on this feature.

RETL tuning

Data movement – extraction, transformation, and load

The data is moved from source systems into RDW with Retek Extract Transform and Load (RETL). RETL utilized parallel processing with parallel reads/writes, transformations and loads. For more information on the RETL technology, see the *Retek Extract Transform and Load Performance Tuning Guide*.

Application tuning

There are two parts to application tuning in RDW. First is the nightly batch processing and second is the front-end user query response time. Refer to the section on MicroStrategy Tuning to optimize the front-end user query response time. To optimize the nightly batch windows, the following should be taken into consideration.

Batch scheduling

For optimal performance, Retek recommends running the batch operation when front-end users are not on the system. This allows all system resources to be allocated to the batch process.

Optimize the RDW batch schedule by running jobs in parallel if there are no dependencies on other jobs. Refer to the *RDW Operations Guide* for job schedule dependencies.

RDW compression

Data warehouses are often very large. Storing this amount of uncompressed data is impractical from a disk storage perspective, both in the cost to store the rows and in the cost to perform backups and other database maintenance operations. RDW compression refers to storing physical data that only reflects changes to the underlying data source and filling in the gaps between actual data records through the use of database views.

For example, when querying sales data, a sales record either exists (a sale occurred) or a record does not exist (no sale occurred). Sales data is considered non-compressed. However, when querying for on-hand inventory, even if no change occurred to the inventory on the date desired, a valid value is still required. One way to resolve this discrepancy is to store a record *every* day for *every* valid item-location combination to show the inventory position. Another method, compression, allows for the storage of only changes to the inventory position. RDW creates an inventory position record only for days that have inventory position changes; but if the inventory doesn't change then a record will not be created in RDW. Inventory position is compressed in RDW. For a list of tables that are compressed and for more information on compression, see the *RDW Operations Guide*.

A general rule of thumb when creating RDW compressed tables is if over 60% of records change then it will be better not to compress the table. There is some overhead for maintaining compressed tables. For instance, if you are partitioning these tables then you need to run a seed process on the first day of the partition. The seeding process can take several hours depend on the volume of data.

RETL configuration

Configure RETL to use direct load instead of conventional load.

Currently the base RDW scripts are using the RETL database read and write operators to move data from a temp table to a target table. If a significant amount of data needs to be moved, it is recommended to convert the code to use a database INSERT/SELECT instead of the RETL operators. If you are using this method, verify your database rollback segment is large enough so that it can accommodate a large volume of inserts.

Parallelism

No programs in RDW are threaded. Threading allows a user to run multiple instances of a program, creating parallelism. Parallelism allows large volumes of data to be processed quickly. In order for RDW programs to be run in parallel, the base code would have to be modified.

Although no RDW load programs are threaded, a program that extracts sales data from RMS, `slsildmex.ksh`, is threaded. This program is packaged with RMS.

Data purging strategy

Currently there are no data purging scripts included in the base RDW product because data and the history needs varies from customer to customer. In order to speed up the performance for both batch and front-end queries, all fact tables need to purge data periodically.

Truncate vs. delete logic for retailers using an Oracle database

Often RDW processing deletes all the data from the target tables. The current standard has been to use DELETE FROM table logic but if the tables have large amount of data then it will take longer to delete data for an Oracle implementation compared to using TRUNCATE. The best approach for an Oracle implementation is to use the TRUNCATE table option, which is fast and resets the HWM (high water mark). In order to use the TRUNCATE statement, the batch user needs to have the truncate system privilege to the DM (datamart) tables. The other alternative is to make the batch user the same as the DM user.

MicroStrategy tuning

RDW has a variety of reports, ranging from the very simple report to very complex, analytical report. The expected response time for a complex report could be considerably longer than a simple report. To optimize response time, end users can narrow their search criteria (i.e., request for less historical data or request for less products or locations to be analyzed per report).

Architecture

The MicroStrategy platform is implemented on a 4-tier architecture. For more bandwidth you can cluster your Intelligence Servers or your Web Servers if needed. In a typical system, MicroStrategy recommends a 1:1 ratio of Intelligence Servers to Web Servers, which ensures that users optimize resources on both sides.

As always, the above sizing options should only be used as a guideline. Users should purchase enough hardware to allow room for growth as the user base, project size, and system complexity increase.

Monitoring statistics

The MicroStrategy Intelligence Server Enterprise Manager tool allows you to track and optimize report usage, display user behavior, and identify system bottlenecks. The Administrator tool allows you to set up caching on the middle tier, load balancing, and set up query governing thresholds.

The toolset has the ability to limit the number of users, prioritize users, or limit users' access to the system.

Workload management

MicroStrategy has the ability to cache reports, schedule reports or create reports which stores data temporarily in temp tables in the database. By scheduling processing you can spread the workload to non-peak processing times.

SQL generation and VLDB settings

The MicroStrategy engine generates SQL for each query. VLDB drivers adjust the way MicroStrategy generates the SQL used to extract information from the database. VLDB drivers can be modified to tune SQL so that it is optimal for a specific database and/or environment. VLDB settings are divided into nine categories and consist of over 90 parameters that can be modified. MicroStrategy provides VLDB settings for each database it supports.

Some VLDB settings have been changed for RDW to optimize database performance and are included in the metadata. See “Appendix D – Technical considerations” in the RDW User Guide for a list of these changes. The default VLDB settings for MicroStrategy are listed in Default VLDB Properties for Specific Databases in the *MicroStrategy 7: System Administrator Guide*.

Thread management

MicroStrategy Web uses a set of ASPs that communicate with the MicroStrategy Intelligence Server. By optimizing the ASP threads, this increases the ability to process more requests at the same time. The thread pool (ProcessorThreadMax setting) and the queue pool (RequestQueueMax setting) can be optimized for better performance. The queue pool is the number of ASP requests that are waiting to be processed by the ASP threads at any given time. Modifying these two parameters can improve web server response time by allowing the maximum possible number of ASP requests to be processed concurrently.

Partitioning

Oracle table partitioning allows a large fact table to be split into several smaller tables. The appropriate partitioning strategy can both significantly improve query response time and decrease the time required to load tables. These advantages must be weighed against the increased database maintenance that is required in a partitioned environment.

The following two methods are available for partitioning in a MicroStrategy 7 environment:

- Warehouse partition mapping manages partitioning using mapping tables (PMT). These must be created and managed by the DBA.
- Metadata partition mapping manages partitioning from the MicroStrategy 7 application. This method eliminates much of the maintenance at the database level.

See the Retek *Data Warehouse Operations Guide* chapter on Compression and Partitioning for additional details.

References

- Retek Data Warehouse (RDW) Operations Guide
- Retek Data Warehouse (RDW) User Guide
- Retek Software 11.0 Performance Tuning Guide
- Retek Extract Transform and Load (RETL) Performance Tuning Guide
- MicroStrategy 7: System Administrator’s Guide
- www.microstrategy.com