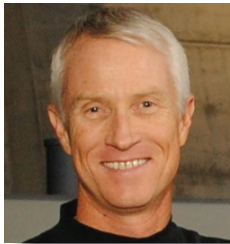




# Real Time Loading to SAP IQ

## How I learned to Stop Worrying and Love Real Time Loading

By Phil Adams, CSP, PMP



*Phil Adams has been an independent consultant since 1999. A highly resourceful and motivated professional, Phil has brought numerous high profile projects to completion in government, health care, pharmacy, and legal fields. Phil provides database, data warehouse, data movement, and business intelligence professional services specializing in the SAP Adaptive Server Enterprise, SAP Replication Server, SAP IQ, and Microsoft SQL Server platforms. Phil is also the founder and Principle Consultant of FireSteed Consulting. Phil currently works with Dobler Consulting, LLC.*

From the beginning, the dream was the ability to make decisions based on information in real time. An airline wants to close out cheaper fares based on reservation demand for a market segment using historical and real-time activity. A big box retail store wants to selectively lower prices based on a sale at several competitors. A hedge fund analyzes market fluctuations in trade pricing and volume data to determine risk to exit a position at just the right time.

For decades, the concept of gathering data and executing reports on aggregated data has whetted the business appetite for better and faster delivery of information on the business processes affecting companies. Much like the brain and nervous system of a living organism, the data needs to be delivered, aggregated, analyzed, and actionable information drawn from it.

The principle goal of analytics platforms is to deliver that actionable information in real-time. SAP Replication

Server has a 20+ year history of data replication to reporting platforms with minimal latency. SAP IQ also has a long history of delivering lightening-fast query response to aggregation of large data sets for reports.

As of SAP Rep Server version 15.5, a new feature was introduced called Real Time Loading (RTL) for SAP IQ and its SAP ASE companion, High Volume Adaptive Replication (HVAR). RTL and HVAR are similar technologies but with different end targets: RTL is optimized for and targets SAP IQ. HVAR is optimized for and targets SAP ASE.

For this article, we will focus on SAP Replication Server 15.7 with Real Time Loading to populate an SAP IQ 15.4 server. The intent is not to provide a step-by-step implementation since those steps exist in the Heterogeneous Replication Guide (<http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc36924.1571100/doc/pdf/rshetrep.pdf>), but to discuss specific configuration and implementation lessons learned. (Another source for a step-

by-step implementation guide/checklist is the presentation slides from ISUG-TECH in Atlanta. (<http://my.isug.com/p/do/sd/sid=3094&type=0>)

### The Performance Dilemma

For the last 30 years, database system performance has always proven challenging. Additionally, as the demand of OLTP performance has increased, so has the demand for reports for decision support. The two tuning approaches proved increasingly difficult to resolve on the same platform as loads increased. The solutions to the performance dilemma are:

1. Move reporting to another database server.
2. Smarter RDBMS's.
3. Move reporting to an optimized reporting platform like SAP IQ.
4. OLAP systems.

With the first option, the performance advantage of offloading workload and tuning for reports provided advantages. However, since reports usually aggregate from very large data sets, optimized platforms are required.

The second option was implemented in SAP ASE's engine optimization strategies *allrows\_oltp*, *allrows\_dss*, and *allrows\_mix*.

The SAP solution was the columnar analytics platform SAP IQ. While SAP IQ addressed the query performance issues very well, there was no way to replicate directly to SAP IQ efficiently so data was loaded to SAP IQ in bulk fashion. SAP Replication Server's traditional replication technique applies transactions to the target in a row-by-row fashion, but SAP IQ's optimal load technique is via bulk load.

SAP Replication Server 15.6 addressed this with the Real Time Loading feature. This feature applies transactions to SAP IQ in a bulk fashion instead of row-by-row. This approach is implemented in the HVAR and RTL functionality of SAP Replication Server. The implementation and operation of both appear similar. Both use a "Net Change database" called the CDB to collect, sort, and group transactions.

The resulting group of transactions are applied to SAP IQ or SAP ASE in an optimal bulk fashion. There are three ways to accomplish work faster in a computer:

- Work faster – CPU clock speed.
- Work smarter – Smarter caching and algorithms.
- Get Help – Multiple cores and parallel processing.

### SAP IQ Advantages

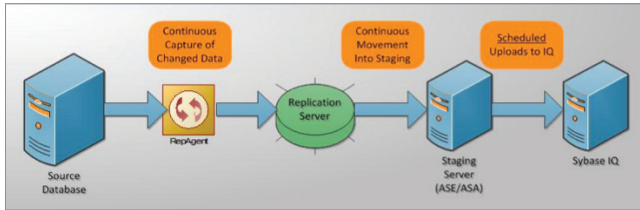
At heart, SAP IQ is a Relational Database Management System, so the known concepts of tables, columns, and SQL still apply as they do in SAP ASE or Oracle. However, SAP IQ's query speeds are 10x-100x greater than those of a traditional RDBMS. SAP IQ supports heavy ad-hoc query and was designed for analytics on terabyte and petabyte scale databases. SAP IQ is a disk-backed database, i.e. not an in-memory db. With its low TCO, SQL language interface, Multiplex feature, and ability to scale to 1000's of users, it is an extremely competent analytics platform.

SAP IQ uses a column-based storage strategy which means the data is not stored physically as rows but each column is stored as a separate structure: an index. The indexes use compression techniques (bit maps, dictionary lookup) and only the unique values are stored (cardinality is important for tuning, more on tuning for RTL later).

SAP IQ's optimal method of data loading is via bulk insert. Performance for single row insert, update, and delete is generally very slow. This is where SAP Rep Server's Real Time Loading features take advantage of loading and updating strategies that favor SAP IQ's preference for bulk data load.

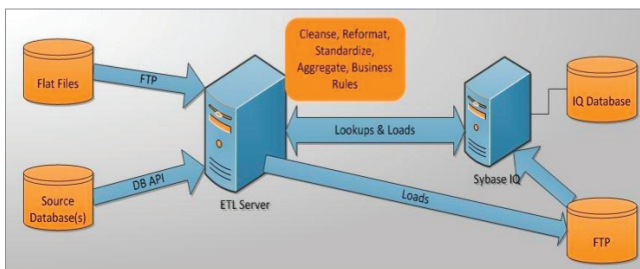
### SAP IQ Loading Before RTL

Before RTL, loads to SAP IQ were typically staged to an intermediary database server using replication. At scheduled times (off hours), the data was bulk loaded to SAP IQ (See **Figure 1**).



**Figure 1: Staging Database Replication to SAP IQ**

Another means of populating SAP IQ was using an ETL (Extract, Transform, Load) approach using a staging server. Data is moved to and prepared in the staging server. Other custom processes for table lookups and scrubbing execute in this intermediary server (**Figure 2**).

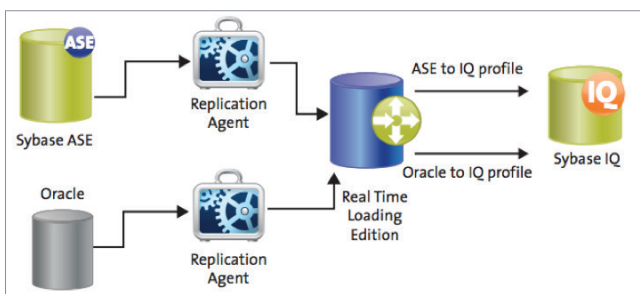


**Figure 2: ETL Technique for Replicating to SAP IQ**

However, neither the ETL nor the staging approaches are considered real-time replication methods.

### What is RTL?

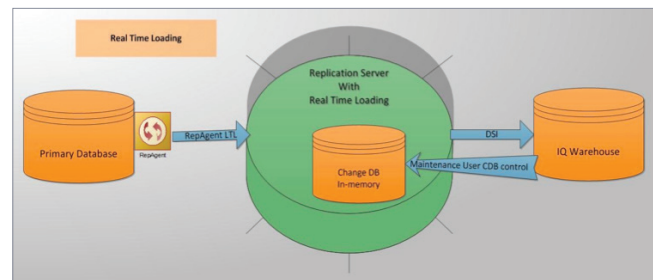
RTL enables the constant stream of replicated data to SAP IQ without the intermediary servers described above. RTL puts a minimum load on the SAP IQ server, minimizes locking, and so enables concurrency of the reporting applications. RTL removes many of the moving parts of the staging and ETL solutions and facilitates a more familiar Replication Server implementation (**Figure 3**).



**Figure 3: SAP Rep Server RTL to SAP IQ**

RTL is a licensed feature of SAP Rep Server 15.5+. It uses a 'Compilation' and bulk-apply strategy to populate the target SAP IQ server. Insert, update, and delete activities are grouped to fit SAP IQ's optimal bulk loading method. RTL sorts and groups transactions to build a 'net change' database. The 'net change DB' (CDB) is an in-memory database within SAP Rep Server, think of it as a transaction 'holding tank' that exists until a grouped set of transactions are committed.

Once the grouped transaction is applied, the CDB's memory is released and the next group of transactions are sorted and assembled into a grouped transaction in a new CDB (**Figure 4**).



**Figure 4: SAP Rep Server RTL to SAP IQ with CDB**

Inserts are applied directly to the target table from the CDB using the *insert..location* statement. Updates and deletes are optimized by inserting the changes to a temporary table in SAP IQ, and performing a join-delete or join-update to the target table.

Terminology for this transaction grouping and bulk application is "compilation". During compilation of a group of transactions, only the "net change", i.e. final outcome of a set of transactions is applied to SAP IQ. For example: if a row is repeatedly updated within the group of transactions, only the final result of the updates is applied to the row in SAP IQ.

RTL does not necessarily apply transactions in the same order they occurred in the source database. Certain transactions may require application in the same order. If replicated transactions fail, with a duplicate for instance, but no duplicate occurred in the source, those tables may need to be removed from RTL processing. If referential constraints are defined



in SAP IQ, the replication definitions should reflect this because RTL will follow replication definition defined constraints and order the transactions appropriately. The basic processing RTL executes is:

1. SAP Rep Server compiles (groups, sorts) transactions in the in-memory CDB.
2. SAP Rep Server creates # tables in SAP IQ to receive rows.
3. Inserts: SAP Rep Server has the maintenance user execute insert..location command directly to the SAP IQ table.
4. Updates & Deletes: SAP Rep Server has maintenance user execute the insert..location command to pull the CDB data into #temp tables.
5. Maintenance user executes update/delete with joins between the target and the #temp tables.
6. Commit, drop #temp table(s), and release memory of the CDB.

To enable RTL, set **dsi\_compile\_enable** to 'on' at the global level. The **dsi\_compile\_enable** can also be set at the connection and table level. Problematic tables can be excluded from RTL by turning **dsi\_compile\_enable** off just for the problem table.

### Special RTL Configuration

The following lists the parameters and areas that require special attention when setting up RTL to SAP IQ.

#### Prerequisites

- License SAP Rep Server and download RTL license.
- Suspend activity in primary before materialization.
- Suspend automated jobs like update stats and reorgs.
- Primary Keys must be defined in source DB and SAP IQ schemas.

### Building the SAP IQ Server

- The SAP Rep Server's interface file must have entry for the DSI to connect to the SAP IQ Server. The SAP IQ interfaces file must contain an entry for SAP Replication Server so that the maintenance user in SAP IQ can access the SAP Rep Server CDB.
- The maintenance user in SAP IQ logs into the SAP Rep Server to control the net-change database using an extern login.
- Map the local maintenance user to the SAP Rep Server login to control the CDB.
- Create the extern dbmaint login in SAP Rep Server before creating the extern login in SAP IQ.
- Create the server in SAP IQ first, 'aseodbc' is the only class.

```
create server RRS class 'aseodbc' using ''; <- empty string
create externlogin db_dbmaint to RRS
remote login db_dbmaint_rs identified by 'pa$$w0rd';
```

- **Minimize\_storage = 'on'** – be sure to turn this parameter on either globally or at the beginning of your DDL scripts. This will enable optimized FP[1-3] index creation. (See FP index rollover discussion below.)
- **Append\_Load = 'on'** – it is recommended to turn this for RTL performance. This option prevents SAP IQ from filling in the blank spaces in indexes. Instead, SAP IQ will append to the end of indexes. Some caution and monitoring (*sp\_iqindexfragmentation* for non-FP) is required since table fragmentation can occur.

```
set option public.Append_Load = 'On';
```

### Configure the SAP Replication Server

- SAP Rep Server **memory\_limit** defines the largest memory footprint of replication server. Allocate large enough to accommodate all in-memory allocations: 16,384 MB if you have the space.

- Memory usage for RTL is determined by:
  1. **dsi\_cdb\_max\_size** – the limit of the in-memory net change database.
  2. **dsi\_compile\_max\_cmds** – the number of commands that can be compiled in a grouped transaction.
- The Net Change Database (CDB) maximum size is configured with **dsi\_cdb\_max\_size**. The CDB is in memory, so allocate **memory\_limit** high enough to accommodate. If there are 2 DSIs with compilation enabled, and **dsi\_cdb\_max\_size** = 2048, then memory requirement for CDBs is 4096.
- **dist\_direct\_cache\_read** – default is 'on'. This is a part of Advanced Services Option/RTL option. It enables the DIST thread to read directly from the SQT cache. Do not turn it off.
- **dsi\_cmd\_batch\_size** – number of bytes sent in a batch. The default of 8192 is typically too small. A good start is 32,768.
- **dsi\_compile\_max\_cmds** – be sure this is large enough for large transactions, otherwise, RTL may not be used. Use 100,000 to start.
- **dsi\_compile\_retry\_threshold** – specifies the maximum number of commands in a group containing a failed transaction where SAP Replication Server will revert to continuous (non-RTL) mode replication.
- **block\_size** – consider increasing the SAP Replication Server block size. The default 16 kb block size defined for the queues requires allocation and deallocation that consumes processing resources. Increasing the block size can reduce the processing load.
- **Create connection** – new to RTL for SAP IQ is the profile **rs\_ase\_to\_iq**. Use this profile in the create connection command:

```
Create connection to IQ_DS.IQ_DB
Using profile rs_ase_to_iq:standard
Set username to iq_maint set password to 'pa$$w0rd'
```

### Create Database Objects

SAP PowerDesigner works nicely for reverse engineering the source database into a physical model. Then, change the database to SAP IQ and generate the DDL for the target SAP IQ database.

### Create Repdefs and Subs

MSA Replication simplifies the creation of and maintenance of replication. In SAP ASE 15.7, table-level replication definitions are needed for the primary key definitions.

Create subs without materialization since materialization is via Materialize Subscription.

### Materialize Subscription

The *insert..location* works nicely. Individual *insert..location* commands can be run in parallel (for 10 or 20 tables, depending on the hardware) to reduce materialization time.

### Verify and Monitor

- Monitors & Counters
- Sysadmin CDB (see below)
- SAP Control Center
- SAP IQ's server logs in \$IQDIR/logfiles/directory
- SAP Rep Server trace commands
- Bradmark

### What Prevents RTL Function?

There are a number of events that can stop SAP Replication Server from compiling transactions. Duplicates that violate primary key or unique index definitions, foreign key constraints, and data conversion issues. When these events occur, SAP Rep Server reverts to row-by-row application of transactions. The performance of non-RTL replication is detrimental to replication performance and typically causes transactions to back up in the queues, so errors must be resolved to continue RTL replication.

Certain replication errors are mapped to invoke retry logic in SAP Replication Server when RTL is active. This 'retry logic' is an enhancement to SAP Rep Server's normal function. When SAP Rep Server receives one of these errors, the defined logic causes SAP Rep Server to split the group transaction in half and

retry each half. If a failure occurs in one of the halves, that half will be split in half again, and retries of both halves execute. If both halves succeed, the grouped transaction has completed and RTL releases the current CDB and moves on to begin compiling a new group of transactions.

Other events that can cause SAP Replication Server to drop out of RTL mode and perform row-by-row updates are configuration parameters such as **dsi\_cmd\_batch\_size** and **dsi\_compile\_max\_cmds** being set too small resulting in lack of resources needed for RTL processing.

### SAP IQ Indexing Issues

SAP IQ index implementation is a key difference with other non-columnar RDBMS's physical structure. By default, flat Fast Projection (FP) indexes are created on columns when a table is created. The default index (if no other is specified) is a Flat Fast Projection.

Flat FPs do not have a look-up table, however, optimized FP<#> indexes have a tokenization (dictionary) look-up table. The three types of optimized Fast Project indexes are detailed and contrasted with the Flat FP in the chart shown in **Figure 5**.

FP1	FP2	FP3	Flat FP
IQ UNIQUE 256 or less	IQ UNIQUE 65536 or less	IQ UNIQUE 16777216 or less	IQ UNIQUE unspecified or greater than 16777216
Storage optimized for small number of unique values	Storage optimized for medium number of unique values	Storage optimized for 3-byte FP indexes	Storage optimized for large number of unique values
Faster query performance, less main IQ store space required	Faster query performance, less main IQ store space required	Faster query performance, less main IQ store space required	Queries may be slower
Need a small amount of extra cache for IQ temporary store for loads	Need extra cache for IQ temporary store for loads. The amount depends on the number of unique values and the data type.	Need significant extra cache for IQ temporary store for loads. The amount depends on the number of unique values and the data type.	No extra cache needed for loads
Loads may be slower if you have numerous columns with IQ UNIQUE <256	Loads may be slower	Loads may be slower	Loads are faster

**Figure 5: Optimized FP Index Details**

Take note of the numbers for FP3 indexes which are for cardinality  $\geq 65,537$  to  $\leq 16,777,216$  unique values. While FP1's and FP2's benefit performance and the look-up tables are small, FP3's tends to be a problematic for RTL performance due to the size of the lookup table. The entire lookup table must be read into memory. This can cause extraordinarily slow performance with RTL. Globally set **minimize\_storage = 'on'** when configur-

ing SAP IQ or when creating the table schema. This parameter will cause index 'rollover' when the number of unique values exceeds the capacity of a FP1, the index will roll over to a FP2. When the number of values of a FP2 is exceeded, the index will roll over to a FP3. This is when the performance problems begin. Reading in a 16 million row look-up table takes time, memory, and cycles.

To avoid rollover to the FP3 index, a special configuration parameter was created. Using **core\_options67= 65536** will cause FP2 rollover to skip the FP3 and go straight to a flat FP index.

To avoid index rollover entirely, implement SAP IQ 16 which changes to n-bit indexes.

### Indexing Recommendation

Each primary key index should have a unique HG index on the composite keys of the index. Each one of the columns making up the composite key should have an LF or HG indexes.

### Diagnostic Tools

All of the usual SAP Rep Server admin commands in SAP Replication Server for queue and transaction movement continue to display the same information with RTL implemented. SAP has added new commands and enhanced the existing tool set to provide information specific to RTL.

- **sysadmin cdb** – new command to view the currently processing net change database. Shows the number of commands being processed by a single CDB (commands\_in\_group). As CDBs are created and released, the naming convention of the CDB reflects the change from one CDB to the next.
- **sysadmin cdb, <queue\_number>, hold** – stops the currently processing CDB so that contents can be dumped/viewed.
- **sysadmin cdb, <queue\_number>, <queue\_type>, list** – shows details for a queue or for a specific table. This is also where you can see what command convert mappings have been created for a table.

- **admin statistics** – the Monitors and Counters feature of SAP Replication Server is a heavy subject in replication that requires more space than this article allows. See “*Sybase 15 Replication Server Administration*” by Saroj Kapoor Bagai and JaganBandi Reddy. The performance chapter (14) by Jeff Tallman gives an overview of using this package for gathering information about SAP Replication Server performance.
- **SAP IQ’s index advisor** – the Index Advisor in SAP IQ is a key performance monitoring tool since one of the primary performance issues in replication systems is frequently the target database. Proper indexing with primary key definition as well as avoiding over-indexing are key strategies to achieving target database server performance.

The Index Advisor provides run-time advice based on updates.

```
set option public.index_advisor = 'ON'
set option public.index_advisor_max_rows = 1000
```

Collects in-memory and use `sp_indexadvice` to view the heavy-hitters.

```
select * from sp_indexadvice()
order by Ninstdesc
```

The ‘Advice’ column will tell you what indexes the optimizer recommends. Index changes can be scripted straight from the text of the `sp_indexadvice()` ‘Advice’ column.

- **SAP IQ’s Server logs in \$IQDIR/log-files/directory** – indicates if multi-row transactions are occurring, bulk inserts, and #temp tables used with updates and deletes. Scripts using `sed` and `awk` to strip out the key data on row updates and length of time to perform those updates will help target problematic tables.
- **SAP Rep Server trace** - do not leave on for long periods of time, can produce large volume of SAP Rep Server error log messages.

```
trace 'on', dsi, dsi_trace_groupsize
trace 'on', dsi, dsi_trace_group
```

## Command Mapping

Command mapping is a new feature that maps commands to different sequences or to nothing. Set `dsi_command_convert` on the connection:

- `i2none` – insert to nothing.
- `d2none` – delete to nothing.
- `i2di` – insert to delete with insert, a form of autocorrection.
- `u2di` – update to delete with insert, a form of autocorrection.
- `t2none` – truncate table to nothing.

```
alter connection IQSRVR.iqdb
set dsi_command_convert to 'i2di, u2di'
```

To display current setting on a table:

```
adminconfig "table_name", <ds>, <db>
```

## Making the Case for RTL to SAP IQ

One question that frequently comes up is: How do I sell an SAP IQ and RTL solution to upper level management? First, start with a Proof of Concept (POC) on SAP IQ without replication. Chose a report based on its run-time. Choosing the most run-time intensive report will make the POC have impact. Only move the tables required for that report to SAP IQ and populate with insert...location. If at all possible, choose a report that requires little to no SQL changes. T-SQL compatibility in SAP IQ helps with this. Once the speed and ease of use are demonstrated, the need to real-time data will follow.

## Conclusion

This has been a brief introduction to some of the key configuration issues and methods of diagnosing performance issues with SAP Replication Server’s Real Time Loading to SAP IQ feature. ■

### Credits and Thanks

Special thanks to Jeff Tallman for his efforts on the Sybase Replication Server Performance and Tuning white paper, many Powerpoint presentations, and the extraordinary contribution he makes to this technical community. Thanks to Mark Mumy for his Powerpoint presentations and assistance over the years. Also a very special thanks to Rob Verschoor for the time and effort he continues to put in to produce the ASE, Rep Server, and IQ quick references.