

Sybase® Replication Server®

Real Time Loading into Sybase IQ

TABLE OF CONTENTS

1	Executive Summary	9	Replication Connection
1	Historical Overview	9	Enable RTL
1	Real Time Loading- Staging with High Speed Data Load	10	dsi_bulk_threshold
1	Staging vs. Real Time Loading	10	dsi_compile_max_cmds
1	Staging	11	Debug options
2	Real Time Loading	11	Trace Flags
3	Real Time Loading Considerations	11	Retry
4	RTL vs. Pre-RTL Replication	11	rs_ticket
4	RTL Timings	12	Customized Function Strings
4	Replication Server, Compilation, and Bulk Apply	12	RTL with Oracle
4	Compilation	12	Data Type Mapping
6	Bulk Apply	12	Enabling Data Mapping Between Oracle and Sybase IQ
6	Expanding the Technology Brings Improvements	13	Oracle Primary Database Restrictions
6	Important Configuration and Debugging Features	13	Putting it All Together
7	Configure Sybase IQ to Receive RTL Connections	13	Check the Logs
7	User Setup	13	Replication Agent Logs
7	T-SQL Compatibility Mode	13	Replication Server Logs
8	TEMP dbspace	14	Sybase IQ Logs
9	Server Connectivity	14	Conclusions and Recommendations
9	Configure Sybase Replication Server for RTL		
9	Server Connectivity		

EXECUTIVE SUMMARY

Historical Overview

Sybase has developed a new, innovative and dynamic staging technology, allowing full use of the native, high-speed data load mechanism Sybase IQ offers . . . Real Time Loading (RTL).

RTL addresses the need to have another system prepare the data for load by taking real time data movement from the source database, and applying staging technology to the data transfer in the background.

Up until now, the only option for loading data from an Online Transaction Processing (OLTP) database . . . airline reservation systems, financial transactions, etc., into a Sybase IQ data warehouse was a so-called staging process. Staging means that the OLTP is isolated from the data warehouse loading process to ensure data integrity, due to the rapidly changing data content in the OLTP database systems.

Direct replication into Sybase IQ was limited to low data movement environments. The reason lies in the fundamental differences on how Sybase IQ processes data, and data movement strategies Sybase Replication Server applies.

Sybase IQ is a column vector database. Retrieving a single row requires assembling the row from each column component. This process makes row by row processing ineffective, and the so- called “trickle feed” loading of Sybase IQ nearly impossible. The advantage of a column vector database is in the unmatched speed for complex data retrieval operations and data aggregation on huge data sets.

Real Time Loading — Staging with High Speed Data Load

Staging vs. Real Time Loading

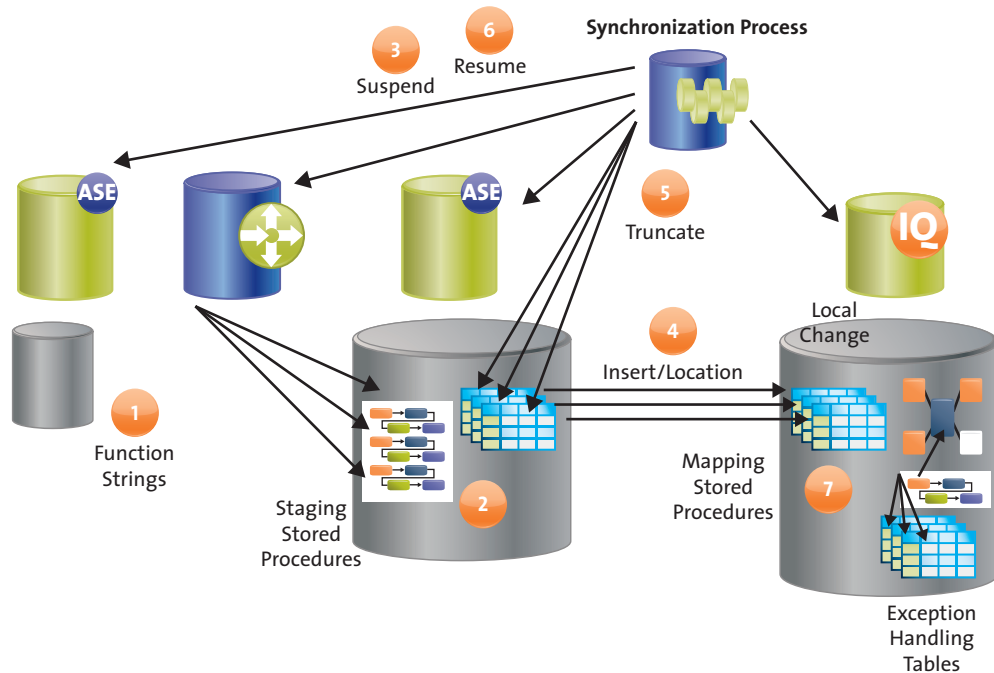
Both staging and RTL have advantages and disadvantages for loading data. Understanding how each option functions helps you determine what path is best for your organization. The following comparison explains the differences between staging and real time loading.

Staging

The basic building blocks of a staging system are illustrated in the diagram below, based on a typical staging setup:

1. Replicate OLTP data into a holding area using function strings
2. Transform the raw OLTP data into staging tables using stored procedures
3. Suspend replication from the OLTP system and ensure the queues are empty.
4. Transfer the data from the OLTP system to a landing area in the data warehouse with the INSERT ... LOCATION command.
5. Truncate the staging tables before resuming operations.
6. Resume the suspended replication from the OLTP system. (See step 3 above).
7. Map the stored procedure data from the landing area in the data warehouse to the final data storage.

Staging Illustration



Staging Considerations

Staging can cause huge latency between the OLTP system and the data warehouse that may be unacceptable for certain environments. In addition to the latency issue, there are many custom build components involved to make the staging process work.

Staging environments also require a lot of maintenance. Recovering from an error is a tedious operation. Although custom data transfer programs provide a lot of flexibility, they also tend to be the bottleneck in rapid development of new data transfer options.

Sybase IQ provides native loading mechanisms like the load table command, and the SQL command, INSERT with a LOCATION clause. There are also a few variations of these commands. For example: client side data loading, server side data loading, and parallel processing. However, all these commands require another system to prepare data for loading.

Real Time Loading

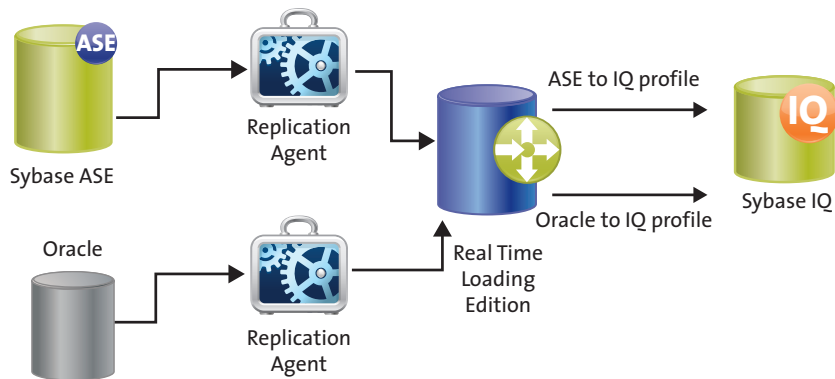
Basic Concept of Real Time Loading

From a high level point of view, Real Time Loading is essentially the clever combination of reducing the traffic volume and using the bulk load features of Sybase IQ.

These are the steps involved in the RTL process:

1. The Replication Agent collects the data on all transactions that contain tables referenced with replication definitions. Reference relationship information is specified in rep def and used by DSI module when applying the data in reference order.
2. The DSI process stores all net row changes into an in-memory database.
3. Based on the `dsi_compile_max_cmds` connection parameter, the net changed rows are being transferred to the Sybase IQ server with the INSERT ... LOCATION command.
 - a. Inserts are directly applied to the target table as defined by the replication definition.
 - b. Updates and deletes are transferred to a generated temporary table in the TEMP dbspace of the Sybase IQ server. Then a join update or delete will be issued to populate data from temp table to user table.
4. The DSI process cleans up any temporary tables from the Sybase IQ TEMP dbspace that have been created, and it frees memory used by them.

Real Time Loading Replication Server 15.6 Illustration



With RTL for Sybase IQ, the process of loading data from a Sybase ASE OLTP server or an Oracle 10g/11g database into a Sybase IQ data warehouse has been dramatically simplified. When comparing to the staging diagram, all the complex operations are handled by the Replication Server. That's why the RTL diagram is reduced to the data extract from source databases. The data movement from the databases is handled by the Replication Agents. The RTL operations are managed by Replication Server, as well as the high-speed data loading into the Sybase IQ target database.

Real Time Loading Considerations

Real time loading eliminates most of the staging steps as these steps were added into the inner workings of Replication Server 15.6. Replication Server 15.5 provides one of the fastest methodologies for loading data into Sybase IQ in real time.

Compilation: Command reduction and grouping

In addition to high-speed bulk loading of data into Sybase IQ from an in-memory database hosted by the Replication Server, a new advanced compilation technique was introduced. It contains two elements. Command reduction, reduces the number of row changes that need to be applied to the target database. The records are stored in the in-memory database. The second is command record or grouping, which groups inserts, updates and deletes together for bulk loading into replicate side.

The level of efficiency of reducing SQL statement transfers by applying the compilation technology heavily depends on how the source application is designed. The more random data changes are across a narrow range of data, the more effective compilation can be. The wider the range of data changes, the less reduction in net row changes you experience. However, even if compilation does not reduce SQL statements to be applied, using the in-memory database and the high-speed bulk load features produces a lightning fast data loading machine.

RTL and Replication Server Benefits

Compared to the traditional staging methodology, RTL provides a technology that allows for streamlined set ups of data transfer operations from OLTP systems into Sybase IQ.

However, the staging concept is still alive, even in the Real Time Loading of Replication Server. Instead of developing a lot of custom code, Replication Server packages all the moving parts and shields the set up with a few simple configuration commands.

This comes in handy when attempting an application upgrade with schema changes, or upgrading Sybase ASE, Oracle or Sybase IQ.

RTL vs. Pre-RTL Replication

Before Real Time Loading, using Replication Server was only recommended for low volume environments. If you don't use compilation for traffic volume optimization, or bulk apply, the only option left is to apply each data change, row by row. This is no problem for traditional database architectures. However, a column vector database is not built for row-by-row processing. Due to its column based approach, so called "set processing" is most effective with Sybase IQ.

RTL Timings

The following table shows statistics for loading and applying data changes for Real Time Loading, and pre-RTL.

ACTION	RTL		PRE-RTL
	Loading	Apply	Loading and Apply
Insert	10 sec	0 sec*	132 sec
Update	1 sec	1 sec	267 sec
Delete	1 sec	1 sec	466 sec

*No apply necessary from temporary table

These timings illustrate how important the bulk apply is in the Real Time Loading concept. Without the bulk apply, there is no high speed loading into Sybase IQ.

Replication Server, Compilation, and Bulk Apply

Replication Server does not allow you to use either compilation or bulk apply individually to enable RTL. By enabling the compilation on the DSI stream, RTL automatically enables the bulk apply. (Bulk apply is the methodology used by Sybase IQ to load data into the database, utilizing its high speed set processing capabilities, and built in parallel processing).

Compilation

Replication Server first collects all DML records from the transaction log of the source Sybase ASE server, regardless of the transaction association. Then, it applies the compilation rules described below to eliminate DML redundancies.

Compilation Rules

PREVIOUS OPERATION	INCOMING OPERATION	COMPILED OPERATION
Delete	Delete	Error
	Insert	Delete, Insert
	Update	Error
Insert	Delete	No op
	Insert	Error
	Update	Insert after image
Update	Delete	Delete
	Insert	Error
	Update	Update to new after image

Compilation is performed on per row and per table

Rows are identified by primary keys

Primary key updates are converted to delete followed by insert

Some additional rules apply to maintain the apply order. This is especially important on tables with referential integrity.

Apply Order

A basic rule for apply order processing is that any deletes on a table must be applied before subsequent inserts.

Referential constraints need to be carefully analyzed when setting up RTL. In most cases, RTL is able to determine the apply order from the referential constraints. However, if there is a circular reference, RTL may not be able to determine the correct apply order.

For example:

If there is a table A depending on table B, the following apply orders apply:

- Inserts to table A must be applied **after** inserts to table B
- Deletes to table A must be applied **before** deletes to table B
- Updates use **before** and **after** images to determine the order of apply

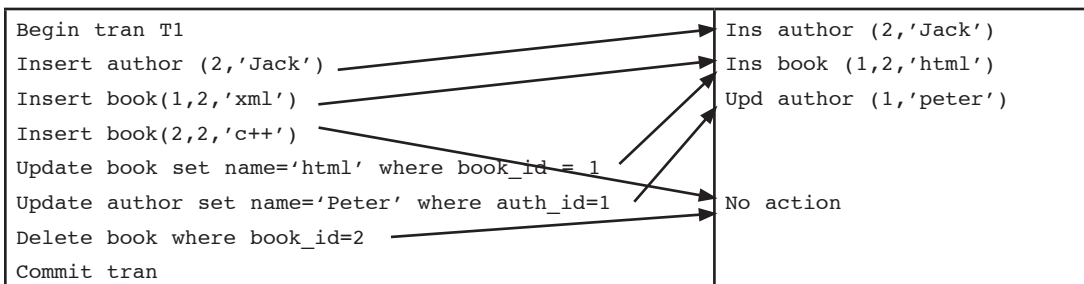
Compilation Examples

Assuming there are 2 tables used in 2 separate transactions, a situation like this is possible:

```
Begin tran T1
Insert author (2,'Jack')
Insert book(1,2,'xml')
Insert book(2,2,'c++')
Commit tran

Begin tran T2
Update book set name='html' where book_id = 1
Update author set name='Peter' where auth_id=1
Delete book where book_id=2
Commit tran
```

This would translate into the following scenario:



With this in mind, you can apply these rules to your environment and applications to determine how many SQL statements could be optimized and compiled.

Bulk Apply

As stated earlier, bulk apply utilizes its high speed set processing capabilities, and built in parallel processing. This is the true superstar of the Real Time Loading technology. The compilation process starts by creating the net new/changed records in the Replication Server in-memory database. Bulk apply uses the insert, with location technology, to transfer the data quickly from the Replication Server in-memory database into the TEMP dbspace tables in Sybase IQ. This only applies to update and delete statements. Inserts are added to the target table directly, as defined in the replication definition.

Updates and deletes are applied with so-called “set processing,” where the target table gets joined with the temporary table by the primary key. Updates and deletes are applied in one SQL statement.

Bulk apply is the fastest way to apply changes to a Sybase IQ table.

Expanding the Technology Brings Improvements

As you learned from the key components of RTL, there are two forces at work. One force streamlines the data flow from the source database, and the other applies high speed loading technology for Sybase IQ. This separation allows connecting the RTL technology in the Replication Server engine, with the available source data collectors.

Today, Sybase Replication Server supports a wide array of database systems, besides the Sybase products. Introducing the RTL technology into all the supported database systems is a matter of connecting the components. This technology allows Sybase IQ to become a central part of any data warehouse architecture and strategy. RTL effectively eliminates the data movement worries many IT managers are facing. Replication Server 15.6 currently supports Sybase ASE and Oracle 10g/11g as source databases.

IMPORTANT CONFIGURATION AND DEBUGGING FEATURES

Although RTL is a very robust technology, there are certain limitations that need to be addressed while configuring a RTL environment. Few of the Replication Server options need to be turned off to enable RTL.

Triggers on the receiving Sybase IQ table are fired when receiving data via the RTL methodology. Users have the option to exclude certain tables from the RTL data transfer, and force the standard row-by-row sequential replication. By excluding a table from RTL, the following rules apply:

- Custom function strings are not supported with RTL. If there are custom function strings present on a table, this table is excluded from the RTL.
- RTL does not support minimal column replication with Replication Server 15.5 and ESD #1. Replication Server 15.6 fully supports minimal column replication for both Sybase ASE and Oracle. This also applies to text/image columns with the `replicate_if_changed` option on.

Triggers on the receiving Sybase IQ table are fired when receiving data via the RTL methodology.

Users have the option to exclude certain tables from the RTL data transfer, and force the standard row-by-row sequential replication. By excluding a table from RTL, the following rules apply:

- You cannot compile commands on the mark-for-exclusion table in the disc compilation task.
- You cannot compile tables that depend on the marked table.
- You cannot compile tables that a marked table depends upon.
- Tables you cannot compile are replicated row-by-row (continuously) in commit order.

Configure Sybase IQ to Receive RTL Connections

There are 2 logins you need to create to enable the connection from the Replication Server to the Sybase IQ server.

User Setup

The Replication Server maintenance user `RS_maint` needs to be created.

```
grant connect to RS_maint identified by dbpwd ;
grant DBA to RS_maint ;
grant membership in group rs_systabgroup to RS_maint ;
```

A `sa` equivalent to enable Replication Server materialization

```
grant connect to sa identified by dbpwd;
grant DBA to sa ;
grant membership in group rs_systabgroup to sa ;
```

You also need to allow both users, `RS_maint` and `sa`, to be able to reference replication tables.

```
grant group to DBA ;
grant membership in group DBA to RS_maint ;
grant membership in group DBA to sa ;
```

T-SQL Compatibility Mode

In order to receive data natively from Sybase ASE, Sybase IQ needs to be set into T-SQL compatibility mode.

```
set option RS_maint.ALLOW_NULLS_BY_DEFAULT = 'OFF' ;
set option RS_maint.ANSINULL = 'OFF' ;
set option RS_maint.CHAINED = 'OFF' ;
set option RS_maint.CONTINUE_AFTER_RAISERROR = 'ON' ;
set option RS_maint.DATE_FORMAT = 'YYYY-MM-DD' ;
set option RS_maint.DATE_ORDER = 'MDY' ;
set option RS_maint.ESCAPE_CHARACTER = 'OFF' ;
set option RS_maint.ISOLATION_LEVEL = '1' ;
set option RS_maint.ON_TSQL_ERROR = 'CONDITIONAL' ;
set option RS_maint.QUOTED_IDENTIFIER = 'OFF' ;
set option RS_maint.TIME_FORMAT = 'HH:NN:SS.SSS' ;
set option RS_maint.TIMESTAMP_FORMAT = 'YYYY-MM-DD HH:NN:SS.SSS'
set option RS_maint.TSQL_VARIABLES = 'OFF' ;
```

For more information on these options, see: Sybase IQ -> Reference: Statements and Options -> Database Options

TEMP dbspace

For update and delete statements, RTL creates a temporary table in the Sybase IQ TEMP dbspace. This temporary table receives data via the INSERT ... LOCATION command from the in-memory database of the Replication Server. The data is applied through a SQL set command, joining the temporary table with the target table defined in the replication definition. The amount of additional space required for the replication depends on how much data is transferred.

Update Example

Updating rows on the source database results in set processing updates on Sybase IQ after all the data has been transferred into the Sybase IQ TEMP dbspace.

Create the temporary table `#rs_ubig_rtldemo_150_1`

Load the data with INSERT ... LOCATION into `#rs_ubig_rtldemo_150_1`

Execute:

```
update DBA.big_rtldemo
set c2 = w.c2 from DBA.big_rtldemo t,#rs_ubig_rtldemo_150_1 w
where t.id = w.id
```

Drop the temporary table `#rs_ubig_rtldemo_150_1`

Delete Example

Deleting rows from the source database also result in this set processing update on Sybase IQ after all the data has been transferred into the Sybase IQ TEMP dbspace.

Create the temporary table `#rs_ubig_rtldemo_150_1`

Load the data with INSERT ... LOCATION into `#rs_ubig_rtldemo_150_1`

Execute:

```
delete DBA.big_rtldemo
from DBA.big_rtldemo t,#rs_dbig_rtldemo_150_1 w
where t.id = w.id
```

Drop the temporary table `#rs_ubig_rtldemo_150_1`

Server Connectivity

In order for INSERT ... LOCATION to work, Sybase IQ must know about the Replication Server that contains the in-memory database with the net row changes.

You must add the information from the Replication Server into the interfaces file of the Sybase IQ server. Failure to add this entry causes the RTL process to fail. No data is transferred to Sybase IQ. This fills up the in-memory database, and ultimately backs up the log of the source Sybase ASE server.

Configure Sybase Real Time Loading Edition

A single command enables the RTL to replicate into Sybase IQ. There are a couple more connection parameters that you need to pay close attention to once RTL is activated.

Server Connectivity

The same rule applies to the Replication Server regarding server connectivity. The Sybase IQ server connection information must be declared in the interfaces file of the Replication Server.

Replication Connection

Creating a connection for the Sybase IQ server has to be done manually. There is no option in `rs_init` available to create the connection.

On the Replication Server, use the following command:

```
create connection to IQ.iqdb
using profile rs_ase_to_iq;standard
set username to RS_maint
set password to 'dbpwd'
```

The `rs_ase_to_iq` profile creates: function string classes, error classes, and the user defined data types. It also creates the necessary `rs_tables` in the target database.

Enable RTL

A single command enables RTL for the entire replication configuration from Sybase ASE to Sybase IQ. Even with all the other parameters in place, by disabling the `dsi_compile_enable` parameter on the connection, the Replication Server falls back into the row-by-row sequential replication. This comes in handy when debugging connectivity issues or compilation issues.

On the Replication Server, execute the following commands:

Connection Level

```
alter connection to IQ.iqdb set dsi_compile_enable to 'on'
```

This is the most commonly used approach to enable RTL.

Server Level

```
configure replication server
set dsi_compile_enable to 'on'
```

This enables RTL for all connections. This might not be exactly what you had in mind. So, be careful when enabling RTL at the server level.

Table Level

```
alter connection to IQ.iqdb
for replicate table named dbo.table_name
set dsi_compile_enable to 'on'
```

If you're already replicating data to a Sybase IQ server, you can enable RTL table by table.

To activate the configuration set by the above commands, issue these commands, no matter what method you use to enable RTL:

```
suspend connection to IQ.iqdb
resume connection to IQ.iqdb
```

dsi_bulk_threshold

RTL is not always the best method to transfer data into Sybase IQ. When used with low volumes of data, it actually is more expensive to use RTL. The expense is higher because of the overhead needed to create the temporary tables in the in-memory database, and the target Sybase IQ TEMP dbspace.

That's where the `dsi_bulk_threshold` parameter takes control. It essentially controls the number of simultaneous insert statements in the DSI thread. The default value is 20. When more than 20 simultaneous insert statements are being executed in the DSI thread, the replication server dynamically enables RTL. If there are less than 20, the standard row-by-row sequential replication is issued.

To optimize this value, it is important to carefully monitor the Replication Server, and analyze the timing of the replication.

To set the value of `dsi_bulk_threshold`, use:

- alter connection to change the bulk threshold connection parameters at the connection level:

```
alter connection to dataserver.database  
set dsi_bulk_threshold to value
```
- configure the replication server to change the server defaults:

```
configure replication server  
set dsi_bulk_threshold to value
```

To check the value of `dsi_bulk_threshold`, use:

```
admin config.
```

dsi_compile_max_cmds

There is another parameter that controls the volume that can be moved through the replication system. `dsi_compile_max_cmds` is part of the new High Volume Adaptive Replication (HVAR) (patent pending) and controls how many net changed rows are transferred to the target Sybase IQ before the temporary tables are flushed.

The default value is 100,000. This value seems high, but in a high volume environment, it is not uncommon to set this value well beyond the 1 million mark.

To set the value of `dsi_compile_max_cmds`, use:

- alter connection to IQ.iqdb

```
set dsi_compile_max_cmds to '500000'
```

Debug options

Debugging is essential to trace replication problems, or to pinpoint latency issues. There are 2 important trace flags that allow administrators to verify RTL is working correctly, or to triage a replication issue.

Trace Flags

DSI_BUF_DUMP

This prints the commands used during RTL into the Replication Server log. Note: This can use a lot of disk space quickly.

To enable this trace flag on the Replication Server, use:

- `trace "on", DSI, DSI_BUF_DUMP`

To disable this trace flag use on the Replication Server, replace the word "on" with "off".

DSI_TRACE_GROUP

This trace flag prints information about transaction grouping into the Replication Server log. If you need to know how individual transactions have been grouped together with the compile technology, this gives you the answer.

Note: This can use a lot of disk space quickly.

To enable this trace flag on the Replication Server, use:

- `trace "on", DSI, DSI_TRACE_GROUP`

To disable this trace flag use on the Replication Server, replace the word "on" with "off".

Retry

When bundling so many individual transactions into a single, huge transaction, retries can be difficult to resolve. The only possible way is to chunk the single transaction into smaller transactions with the `dsi_compile_max_cmds` parameter, and to suspend/resume the connection.

Keep reducing this parameter until you pinpoint the failing SQL. Once you pinpoint the failing SQL, use the skip transaction parameter when resuming the connection. In future releases of RTL, this should be addressed by Sybase engineering as an improvement opportunity.

rs_ticket

`rs_ticket` (Introduced in Sybase Replication Server 12.6) allows Replication Server administrators to insert flag records to measure the latency between the source database and the target database.

After you installed the necessary `rs_ticket` components, activating and testing the `rs_ticket` system is done with the following commands:

On the Replication Server

```
alter connection to IQ.iqdb set 'dsi_rs_ticket_report' to 'on'
suspend connection to IQ.iqdb
resume connection to IQ.iqdb
```

On the Primary Sybase ASE database

```
exec rs_ticket 'start'
<execute replication sql benchmarks>
exec rs_ticket 'stop'
```

On the Replicate Sybase IQ database

```
rs_ticket_report
```

Customized Function Strings

RTL does not support customized function strings. If you're in need of functionality that goes beyond the standard function strings, RTL introduces a new configuration parameter `dsi_command_convert`. This configuration parameter allows custom actions after the initial transaction activity.

PARAMETER VALUE	DESCRIPTION
iznone	Inserts are ignored
uznone	Updates are ignored
d2none	Deletes are ignored
t2none	An entire transaction is ignored
izdi	Inserts are converted into deletes and then inserted
uzdi	Updates are converted into deletes and then inserted

- E.g., if a user wants to ignore deletes, set the `dsi_command_convert` to 'd2none' instead of defining an empty function string. In this way, RTL can still be used.

RTL with Oracle

Sybase Replication Server 15.6 now supports the RTL technology from Oracle 10g or 11g into Sybase IQ. The Replication Agent on an Oracle server transfers the data straight into an inbound queue of the Replication Server, and allows for high speed data transfer from Oracle to the Replication Server. Once the data is in the inbound queue of the Replication Server, the same exact RTL methodologies as applied to a Sybase ASE source database take effect.

Data Type Mapping

Replication Server handles all the data type mapping between Oracle and Sybase IQ.

ORACLE DATATYPE	REPLICATION SERVER DATATYPE
RAW	rs_oracle_binary
DATE	rs_oracle_datetime
ROWID	rs_oracle_rowid
INTEGER	rs_oracle_int
INTERVAL	rs_oracle_interval
BINARY_FLOAT	rs_oracle_float
NUMBER	rs_oracle_decimal
TIMESTAMP(n)	rs_oracle_timestamp9
TIMESTAMP(n) (with local time zone)	rs_oracle_timestamptz

Enabling Data Mapping Between Oracle and Sybase IQ

To enable the data type mapping between Oracle and Sybase IQ, a separate connection to the Sybase IQ server needs to be created.

On the Replication Server, use the following command:

```
create connection to IQ.iqdb
using profile rs_oracle_to_iq;standard
set username to RS_maint
set password to 'dbpwd'
```

The `rs_oracle_to_iq` profile creates: function string classes, error classes, and the user defined data types. It also creates the necessary `rs_tables` in the target database.

rs_oracle_to_iq default settings:

- Set function string class to rs_iq_function_class
- Set error class to rs_iq_error_class
- Set connection parameter. E.g. set dsi_dataserver_make to iq
- Create some objects in replicate IQ. E.g. rs_lastcommit, rs_ticket_history, etc.
- Add some class level translation definition. E.g. rs_oracle_binary to varbinary

Oracle Primary Database Restrictions

Oracle performs minimal logging by default. If you are not using table replication definitions/subscriptions, perform one of the following steps:

- Create table replication definitions with send standby clause, three ways:
 - set pdb_auto_create_repdefs true in Replication Agent before marking table,
 - or use rs_create_repdef in Replication Agent,
 - or create the replication definition in RS directly

Or

- Enable full logging in Oracle to ensure the update command works correctly

Putting it All Together

To enable RTL with Oracle, there are just a few added steps necessary to connect an Oracle OLTP server with Sybase IQ.

In addition to the Replication Server 15.6 software the separate software package Replication Server Option for Oracle is required.

First, the Express Connect for Oracle needs to be installed on the Replication Server. This handles the replication connection from the Replication Server to the Oracle database.

Second, the Replication Agent needs to be installed on the Oracle sever. The Replication Agent reads the redo logs on an Oracle server and transfers the data to the Replication Server.

Once the Oracle server is connected to the Replication Server, the same Replication Server commands are used to define the data content that needs to be replicated into Sybase IQ. This simplifies the management of data movements in a heterogeneous environment. The same RTL techniques, as with a Sybase ASE source database, are applied to data arriving from an Oracle server.

Check the Logs

After installing RTL into your environment, you should examine the logs of the various components to ensure that everything is working correct.

Replication Agent Logs

The Replication Agent log on an Oracle server will reveal if the data flow is running smoothly or if there are any connectivity issues.

The status of the Sybase ASE Replication Agent can be retrieved directly from the Replication Server command line and the source database.

Replication Server Logs

Trace flags are very important in a replication environment. All output goes to a single file, the error log file, which is defined in the Replication Server start file.

Sybase IQ Logs

On Sybase IQ, the log is an important communication tool for the Sybase IQ server with the database administrator. When enabling RTL, on Sybase IQ, you should see an indication of set processing.

An indication that RTL is working can be found in the log and looks similar to the example below. Please ignore timings and use as an example only. The highlighted data below indicates set processing was used.

I. 10/16 17:45:47.

840 records were inserted into 'big_rtl-demo'.

I. 10/16 17:45:47. Insert for 'big_rtl-demo' completed in 0 seconds. 840 rows inserted.

I. 10/16 17:45:47. Update Pass 1 completed in 0 seconds.

I. 10/16 17:45:47. Update Pass 2 completed in 0 seconds.

I. 10/16 17:45:47. Update for 'rs_lastcommit' completed in 0 seconds. 1 rows updated.

I. 10/16 17:45:47. Insert Pass 1 completed in 0 seconds.

I. 10/16 17:45:47. Insert Pass 2 completed in 0 seconds.

If you see only 1 record count for inserts, updates and deletes on the replication target table, RTL is probably not in use. This can have multiple root-causes:

- The `dsi_bulk_threshold` may not be met
- The source table does not qualify for compilation, or
- RTL has been turned off for this specific table.
- In IQ release interface file under \$SYBASE, it does not contain RS server address entry.

When examining the logs of a RTL environment, you should always consider the entire system, including the data content. In the end, this could be perfectly normal.

CONCLUSIONS AND RECOMMENDATIONS

Sybase Replication Server Real Time Loading into Sybase IQ is an amazing technology. It opens up a whole new world of opportunities to connect data sources to Sybase IQ in real time. Real Time Loading into Sybase IQ eliminates the need to build your own staging environment that often brings maintenance headaches and latency issues with it. To increase the relevancy of Sybase IQ as a central data warehouse system in heterogeneous environments, the load process needed to be improved and streamlined. RTL delivers on this requirement.

As with every new technology, you should test, test, and then test some more before introducing into production environments. Although the technology is sound and the implementation is rock solid, Replication Server is built to move data out of the source database as fast as physically possible. If this data has nowhere to go, due to configuration or capacity issues on the target database, the data pipes fill up and eventually cause a backlog on the source database. There are numerous techniques within Replication Server to mitigate these situations.

With RTL, Sybase created a tool that allows clients to move Sybase IQ into the center of their data warehouse strategy. DBA's are now able to feed the data warehouse with "out-of-the-box" functionalities, and no custom coding from a variety of database flavors and with easy-to-use setup parameters. With the expansion of supported source databases from Sybase ASE to Oracle 10g/11g, Sybase IQ is now a very strong contender in assuming the central data warehouse single source of truth.