



**New England DB2 User Group**

**NEDB2UG**  
March  
2007

# Federated Q-Replication DB2-z/OS to ORACLE User Experience

Diwakar Rao  
Fidelity Investments.

This presentation provides an overview of IBM replication technologies, and explore tools and techniques to replicate from DB2 z/OS to oracle .

Federated ?

A federated database system is a type of meta-database management system (DBMS) which transparently integrates multiple autonomous database systems into a single federated database.

What's *transparent* about it? You can write SQL statements as if they were accessing DB2, and -- when the data resides in Oracle -- DB2 will transparently execute the SQL on the Oracle instance and return the result to DB2.

The federated database along with Oracle wrapper is like a "Oracle Transparent Gateway for DB2" used to access mainframe DB2 data from an Oracle server.

# Agenda

NEDB2UG  
March  
2007

- >>> Why Replicate ?**
- >>> Business requirement**
- >>> Federated Q-replication**
- >>> IBM Replication products**
- >>> Q-replication performance**
- >>> Setup and manage changes to the replication environment.**

Topics includes a brief overview of IBM replication product , and a lot of the technical details regarding how it works, what options will be available. Discuss basic Q-replication concepts, CCD target tables, roles played by MQSeries and Federated DB2 UDB database and provide tips and share user experiences in setting up and managing Federated Q-replication.

# Why Replicate?

NEDB2UG  
March  
2007

## >>> Disaster Recovery (High Availability)

- >>> Maintain standby copy for failover.
- >>> Minimize recovery time (HOT Stand-by)

## >>> Workload Isolation (Improve Performance)

- >>> Data Distribution/Consolidation
- >>> Regional Data Centers

## >>> Information Integration (Global Enterprise view)

- >>> Data warehouse/Reporting (ETL)
- >>> Event Notification, Analytics

### ➤ Disaster Recovery

Maintain a standby copy for failover. Minimize recovery time and prevent data loss. Transactional consistency.

### ➤ Workload Isolation

Maintain live copies for working in disconnected mode , read-only or read-write (Manage and resolve conflicts)

### ➤ Information Integration

Replication can copy changes from distributed sites to a central site for analysis, reporting, and for enterprise application processing.

Consolidation of data can be very useful for business intelligence applications such as OLAP or Data Mining.

Moving data to/from heterogeneous data stores.

- >>> Build Operational Reporting Database (ORD).
- >>> ORD needs to be in a 'current' state.
- >>> ORD should retain purged/deleted data for 7 years.
- >>> Need a means of identifying data that has been purged off of OLTP database (i.e. a 'deleted' flag).
- >>> System Requirements.
  - >>> Easily scalable with minimal operational cost.
  - >>> Conform to Enterprise Architecture.
  - >>> Solution should not impact OLTP Database performance.

Getting business requirements and SLA's clearly defined before we start helped us focus on the task in hand!

## >>> SQL Replication.

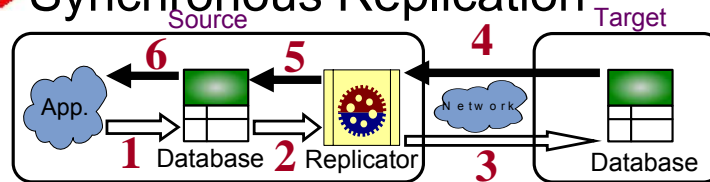
(aka IBM Data Propagator )

## >>> Q Replication.

## >>> Event Publishing.

- Data propagator has been available for over 10 years. IBM renamed Dataprop to SQL replication to make it distinct from the new Q Replication architecture.
- The new queue based replication architecture was released sometime in 2004 as part of DB2 Information Integrator and then part of WebSphere Information Integrator.
- This new Q based architecture can also help create an infrastructure that can serve application messaging/publishing in addition to replication.

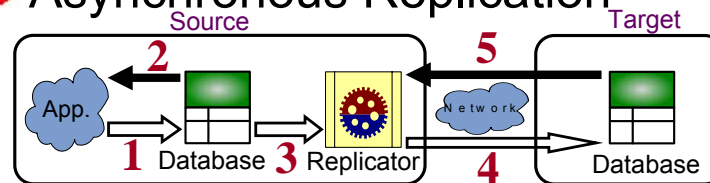
## >>> Synchronous Replication



⇒ : Data Write

⇐ : Write/Commit Acknowledgement

## >>> Asynchronous Replication



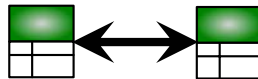
➤ **Synchronous** replication is a technique for replicating data between databases where the system being replicated does wait for the data to have been recorded on the target/replicated system before proceeding. Often, it can take a relatively long time to write data to the remote database and the client must wait for this to occur, leading to very long transaction response time. Could have a problem when the remote/replicated database becomes unavailable.

➤ **Asynchronous** replication is a technique for replicating data between databases where the system being replicated does not wait for the data to have been recorded on the target/replicated system before proceeding.

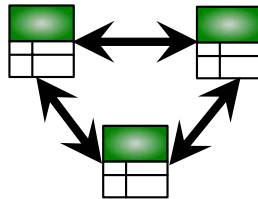
## >>> Unidirectional



## >>> Bidirectional



## >>> Multi-Master/Peer-to-Peer



➤ **Unidirectional** replication replicates changes in ONE direction between two servers (i.e. from source to target)

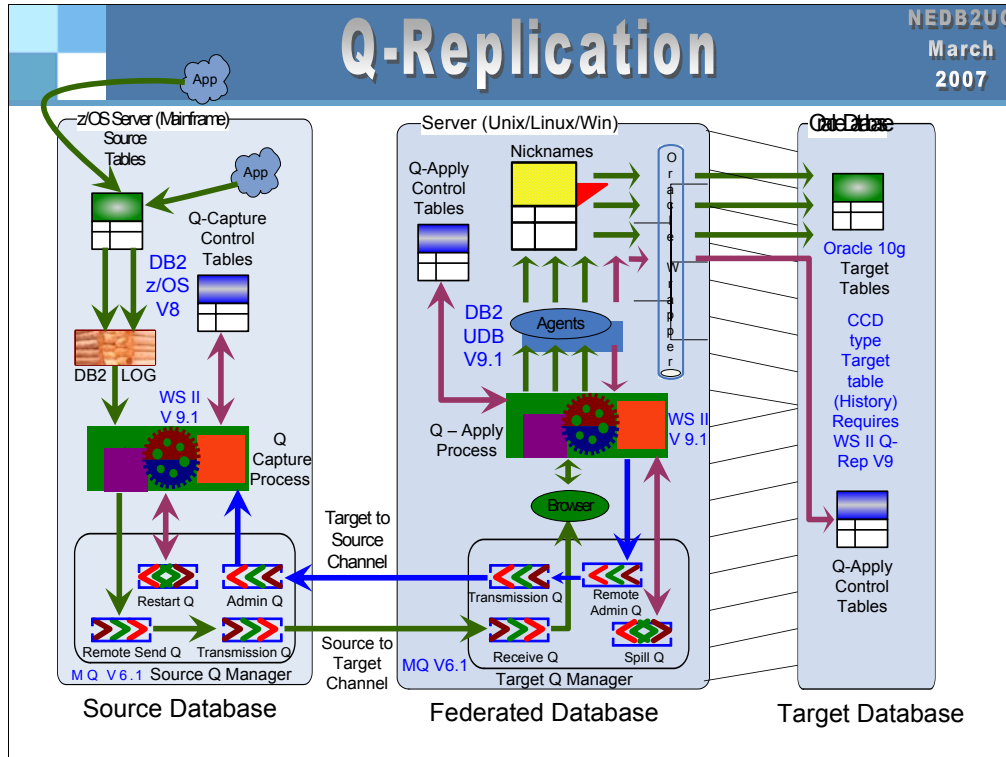
➤ The target table typically is read only.

➤ Unidirectional is the only option available for federated replication.

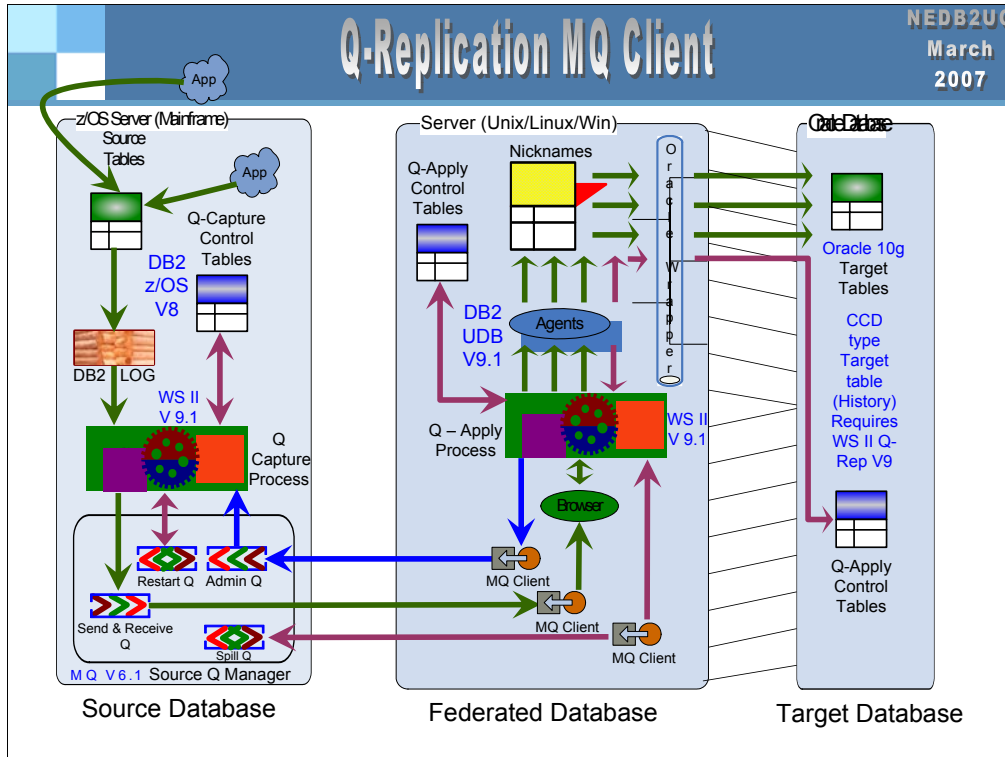
➤ **Bidirectional** replicates in two directions between two servers. One server is designated as the winner of any conflicts.

➤ **Peer to Peer** is a true **multi-master** configuration that can allow replication between tables on two or more servers. All servers are equal peers with equal ownership of the data; no server is the "master" or source owner of the data.

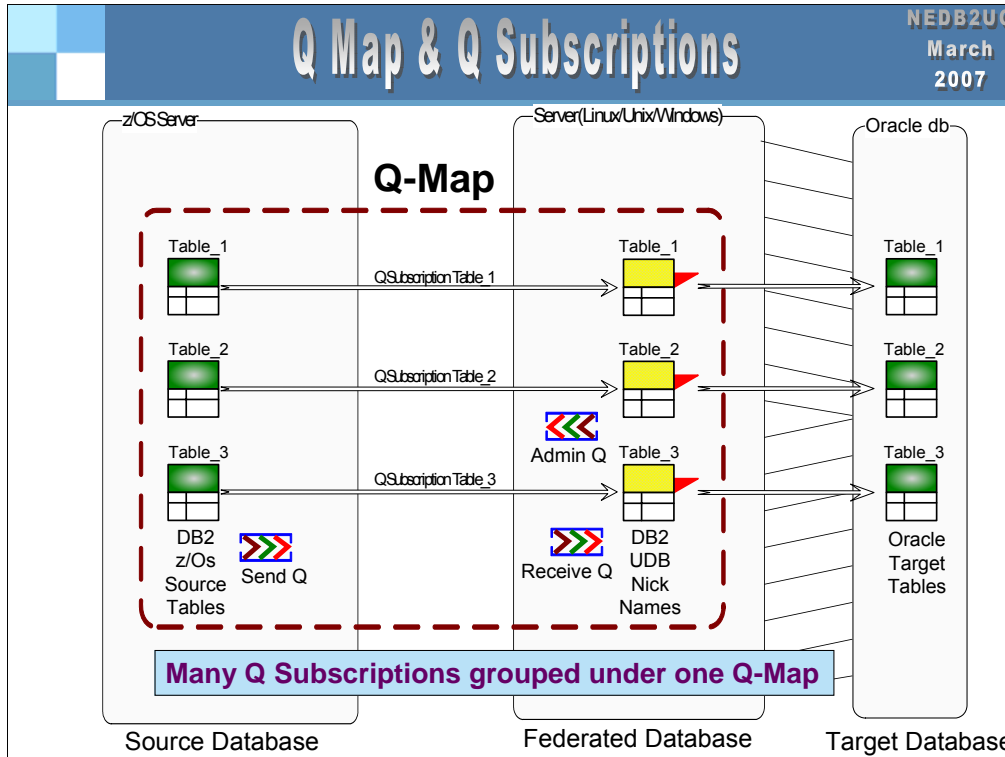
➤ With Q replication, you can replicate data back and forth between tables on two or more servers while applications update the identical copies of a table at all servers while keeping all copies of the table synchronized.



- Changes to subscribed tables will appear on the DB2 recovery log.
- The changes will be read by Q Capture and stored in memory.
- Committed transactional data will be put to the data transport queue (send queue).
- Messages on the SEND queue will be instantly sent by MQ to the target RECEIVE queue.
- For every incoming receive queue, Q Apply initiates one browser thread, the apply browser reads transactions from the receive queue, examines dependencies between transactions, and based on this dependency analysis applies data changes through parallel Q Apply agents, while also ensuring data integrity. Each browser can initiate many agents. Q Apply agents can apply transactions in parallel, emulating the parallelism of the originating source application and thereby dramatically increasing replication throughput.



- You can run the Q Capture or Q Apply on a system that uses a WebSphere MQ client to connect to the Queue Manager that the replication program works with.
- Allow you to isolate your messaging server from your database server.
- Distributed platforms only
- Allows separation of Database servers and MQ servers
- Allows replication support on platforms which currently lack MQ Server support
- IBM Recommendation: “For optimal performance, run the Q Capture and Q Apply programs on the same system as the queue manager that they work with.”
- Restrictions “A WebSphere MQ for z/OS subsystem cannot be a client.



➤ In Q replication, you create objects called Q subscriptions to define how data from a single source table is replicated to a single target table .

➤ Replication Q-Map identifies the MQ queues that a Q Capture program and a Q Apply program use to transport data and communicate. A single replication queue map transports data for one or more Q subscriptions.

➤ When you define a Q-Map, You define..

- The maximum size of a message you put on send Q.
- The amount of memory for the Q Apply program to process messages
- How often capture will send heart-beat message to apply task.

➤ When you define a Q-Subscription, you ..

- Map the source columns to the target columns.
- Can subset the columns and rows that you replicate from the source.
- The method of loading the target table.

➤ Q subscriptions that have dependencies must share the same replication queue map. Because the Q Apply browser at the receive queue detects dependencies between transactions.

Q Replication - Defining Subsets or Filters

NEDB2UG  
 March  
 2007

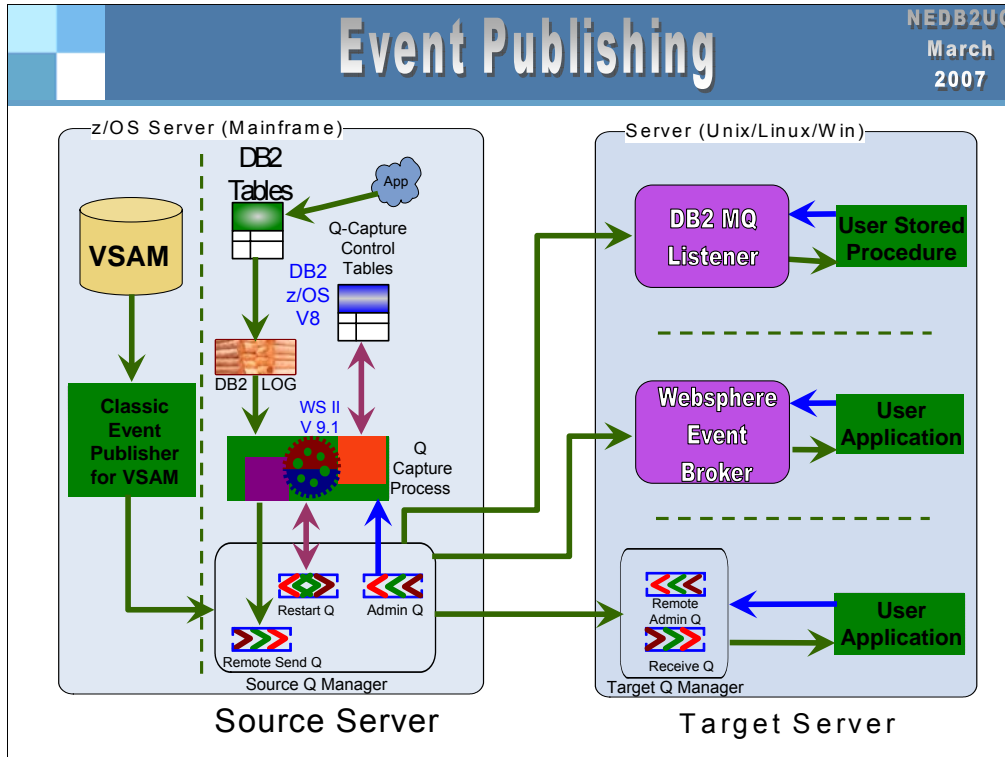
- **Subset data**
  - Subset of rows or columns at subscription level.
  - Option included for ignoring deletes.
  - Can allow user selected transactions to be ignored !
  
- **Predicate examples**
  - Based on values in the row data itself
    - WHERE :LOCATION ='EAST' AND :SALES > 100000
  
  - Based on values in other data
    - WHERE :LOCATION ='EAST' AND :SALES > (SELECT SUM(expense) FROM STORES WHERE stores.deptno = :DEPTNO)

➤ Column and row filtering is provided. The predicate is evaluated on the Q Capture.

➤ This allows the Q Capture to understand when a predicate column value has changed, and selectively convert certain updates to deletes or inserts as necessary, automatically.

➤ An option is available to suppress the replication of any deletes, and another option is available to mark transactions so that they will not be replicated.

➤ Evaluation can be made of the data in the row itself, and this is fast. It can also include lookups in other tables, but this can dramatically affect performance.



➤ In Event Publishing Changes to source tables are translated into XML messages and published to user applications.

➤ User Applications can directly process published data from message queues.

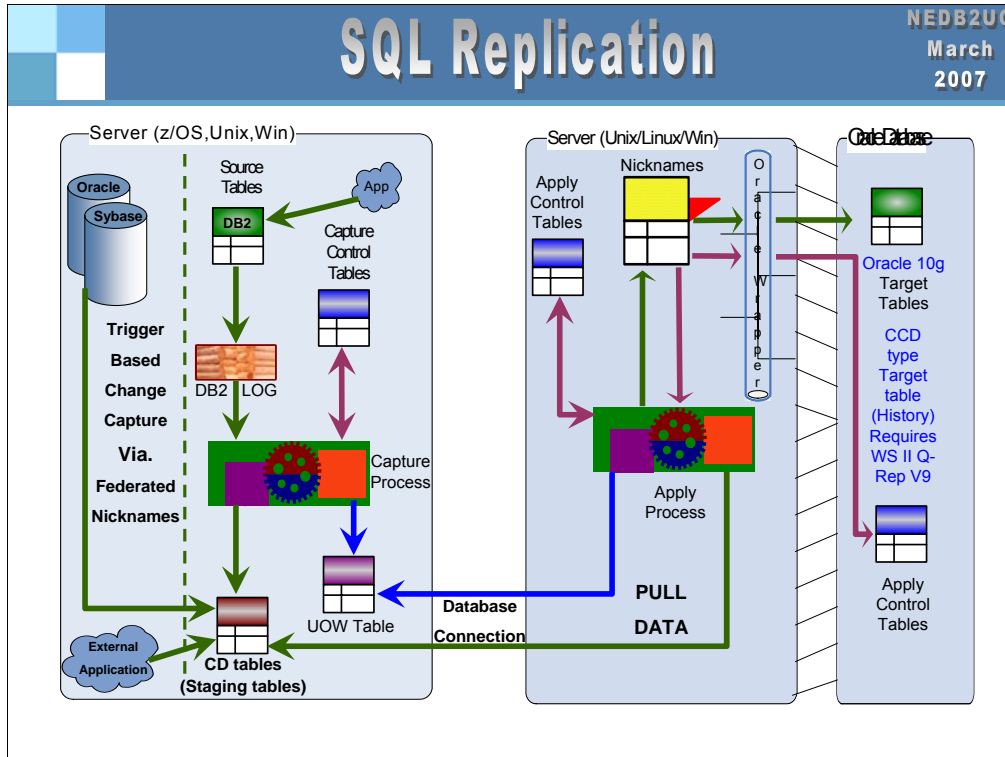
#### To update a Web site

Changes to a stock prices are captured from the DB2 log, and published as XML messages to a JSP application that runs on an application server . The XML messages are then used to update HTML pages that display the up-to-date stock information.

#### To feed a central integration broker

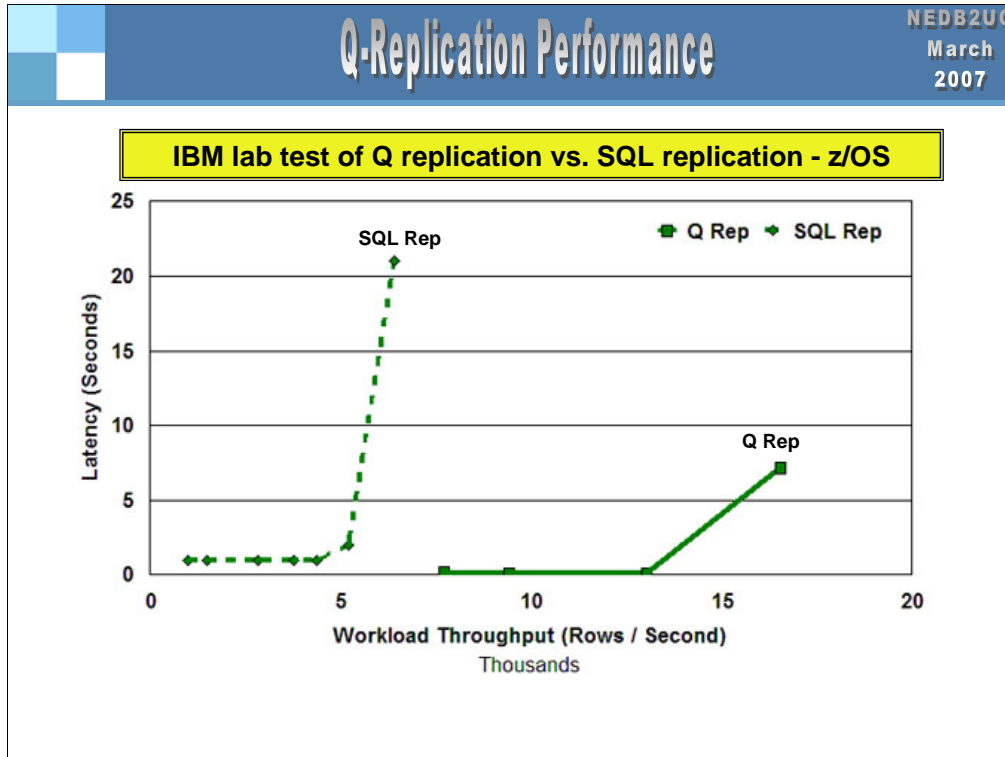
When a customer updates an address, the transaction is published as an XML message to a WebSphere Event Broker. The broker translates the XML message into formats that can be understood by different applications throughout the business, its partners, and suppliers.

➤ Classic Legacy Systems back into the game (VSAM, IMS)



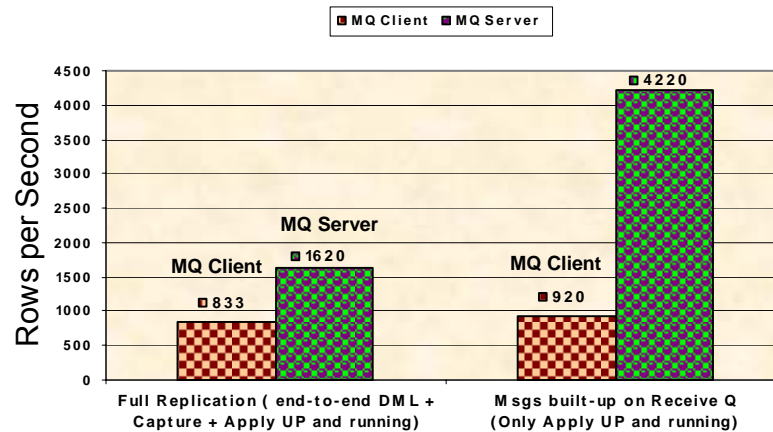
- Was DB2 Data propagator before being renamed to SQL Replication under “WebSphere Replication Server” product suite.
- A Capture program captures changes from the DB2 LOG (or) via triggers for Non-DB2 sources and moves them into a staging table, called a changed data (CD) table.
  - A single staging table can serve as source for multiple subscriptions or multiple staging tables can be created for a single source depending on the application requirements.
  - Publish once for N subscribers. A *pull* model.
  - You need to plan for additional log space based on tables involved in replication.
- The Apply program fetches data from the staging tables and applies it to the target tables using a database connection.
  - One or more apply programs can subscribe to a CD table.
  - One apply program can replicate data to one or more target tables.
  - Apply program handles column and row subsetting, performs SQL transformations, manages commit scope based on subscription sets and table vs. transaction consistent delivery.
  - Apply program references foreign source and target tables and control tables via nicknames.

- >>> What makes Q-Replication faster than SQL-Replication ?
  - >>> Elimination of DB2 staging table (CD) that result in reduced DB2 logging.
  - >>> Push changed data in “compact” format using MQ engine.
  - >>> Apply uses parallel agents, providing high-throughput.



- Chart from an article published by John Aschoff (aschoffj@us.ibm.com), Performance Analysis - WebSphere Information Integration, IBM Silicon Valley Lab
- Workload show almost three times the throughput for Q replication compared to SQL replication, while maintaining much lower latencies at the high throughput points. Q replication is able to replicate more than 12,000 rows per second, with an end-to-end latency of less than 2 seconds, and consistently less than 1 second at lower rates. The workload used in these measurements consisted of INSERTs only, simulating a moderately complex transaction with 10 INSERTs per transaction, and 212-byte rows.
- You can find the complete article “ Websphere II Q replication performance considerations” @ <http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0503aschoff/>

## Q Replication MQ Client vs. MQ Server



- Refer to Slide No 8 and 9 for MQ Client and MQ Server architecture diagram.
- Performance test involved running a batch job that inserted 600,000 rows into 13 tables, Insert 50 rows into each table and COMMIT. 13 tables involved had varying record length between 50 bytes to 1000 bytes. The Insert step ran in loop "n" number of times.
- Q Capture will cut 1 MQ message for each UOW if the message is < 64KB (default), if the UOW is > 64KB, Q capture will split it into multiple messages (maximum message size limit can be specified when you define the Q-MAP)
- For this test **2000** miles of network separated the Source DB2 and Target Oracle server.
- For the first category on the graph "Full Replication" workload show almost two times the throughput for MQ SERVER setup compared to MQ CLIENT architecture, While MQ server setup maintained a consistent end2end\_latency of 2 - 3 seconds, the MQ client setup started with a 2 second end2end latency and gradually increased to 18 seconds by the time all rows got replicated.
- The second category "Msgs. Built-up on the Receive Q" demonstrates the impact of "long distance network". When we eliminate the network from play, we see a significant improvement in MQ server setup and a very marginal improvement in MQ client architecture (coz. it still has to get msgs. from the remote MQ on z/OS server).
- As per "unofficial" IBM performance estimates, one can expect 12k to 13k rows replicated per second in a MQ Server setup and 4k to 5k rows per second in MQ Client architecture. (I'm guessing this is when you have the source and target database servers pretty close!)
- **IBM Recommendation:** "For optimal performance, run the Q Capture and Q Apply programs on the same system as the queue manager that they work with."

Tuning Replication

NEDB2UG  
 March  
 2007

- >>> IBMQREP\_CAPPARMS control table on Capture Server.
  - >>> MEMORY\_LIMIT
  - >>> COMMIT\_INTERVAL
  - >>> SLEEP\_INTERVAL
- >>> IBMQREP\_RECVQUEUES control table on Apply Server.
  - >>> NUM\_APPLY\_AGENTS
  - >>> MEMORY\_LIMIT
- >>> IBMQREP\_APPLYPARMS control table on Apply Server.
  - >>> AUTOSTOP
  - >>> MONITOR\_INTERVAL (not a tuning parameter!)

➤ The commit interval parameter specifies how often, a Q Capture program commits transactions to MQ. Shorten the MQ commit interval and DB2 committed transactions on the send Q will be pushed through to the receive Q with less delay. Lengthen the commit interval to reduce CPU overhead.

➤ The sleep interval parameter specifies the idle time that a Q Capture program waits after reaching the end of the active log and assembling any transactions that remain in memory. Increase the sleep interval to save CPU, Lower the sleep interval to reduce latency.

➤ You specify the value for number of apply agents and memory for the apply task when you create a replication queue map.

➤ If you set autostop=Y, the Q Apply program shuts down after all receive queues are emptied once. You have to manually restart the Apply program to process messages received after the apply program last shutdown.

➤ The monitor interval parameter tells a Q Apply program how often to insert performance statistics into the IBMQREP\_APPLYMON table. This is the one table you query all the time to find end-to-end performance statistics.

# Architecture Summary

NEDB2UG  
March  
2007

Architecture	SQL Replication	Q Replication	Event Publishing
<b>Data Distribution (1:Many)</b>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•CCD Staging to Fan-Out Topologies</li> </ul>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•CCD Staging to Fan-Out Topologies</li> </ul>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•Consumer Process</li> </ul>
<b>Data Consolidation (Many:1)</b>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•Through Target Views</li> </ul>	<ul style="list-style-type: none"> <li>•No</li> <li>•No target Views supported</li> </ul>	<ul style="list-style-type: none"> <li>•Consumer Process</li> </ul>
<b>Join sources</b>	<ul style="list-style-type: none"> <li>•Yes</li> </ul>	<ul style="list-style-type: none"> <li>•No</li> </ul>	<ul style="list-style-type: none"> <li>•Consumer Process</li> </ul>
<b>Computed Target Table Columns</b>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•SQL Expressions</li> <li>•Source Views</li> <li>•Stored Procedures</li> </ul>	<ul style="list-style-type: none"> <li>•No</li> <li>•Stored Procedures can be Target Objects instead of Tables.</li> </ul>	<ul style="list-style-type: none"> <li>•Consumer Process</li> </ul>
<b>Horizontal/Vertical subsetting (&amp;) Before/After images</b>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•Full SQL Power</li> <li>•Base and Change aggregate target tables</li> </ul>	<ul style="list-style-type: none"> <li>•Yes</li> <li>•No Aggregates</li> </ul>	<ul style="list-style-type: none"> <li>•Yes</li> </ul>

- Database objects that can be targets
- Q-Replication
  - DB2 tables and views.
  - Tables on non-DB2 relational databases.
- SQL- Replication
  - DB2 tables and stored procedures.
  - Tables on non-DB2 relational databases.
- Base aggregate — an aggregate of a source table based on SQL column functions and GROUP BY filters that you define Apply issues a select against the source table each time it processes a base aggregate target table and inserts new rows in the target table.
- Change aggregate — an aggregate of the changes to a source table based on SQL column functions and GROUP BY filters that you define Apply issues a select against the CD table each time it processes a change aggregate table and inserts new rows in the target table.

- Supports Unidirectional subscriptions only.
- One Apply program instance per federated target.
- One set of Apply control tables for every federated database.
- Requires some of the control tables to be in the target database.
- Q Apply program updates both the control tables and the target tables in the same commit scope.
- Uses nicknames for these control tables at the federated server.
- Supports target to be a DB2 stored procedure writing to nicknames.

- The Q Apply program writes changes to a non-DB2 target table by issuing DB2 SQL to a DB2 Information Integrator federated server nickname.
- A nickname is a pointer to the non-DB2 target rather than a physical table.
- The DB2 Information Integrator federated server uses the nickname and the Q Apply program's SQL to write changes to the non-DB2 target table on behalf of the Q Apply program.
- A subset of the Q Apply control tables are built in the non-DB2 database. The rest are built in a DB2 UDB database that is created on the DB2 II federated server.
- The method for automatic loading of a non-DB2 target table is limited to the DB2 UDB EXPORT and IMPORT utilities. The IMPORT utility uses a nickname to write to the non-DB2 target table.

## Consistent-Change-Data type target table

- >>> Maintain an AUDIT trail of database changes.
- >>> Retain history of source data for compliance or support reporting.
- >>> CCD type target table has 2 attributes.
  - >>> Condensed = Y : Contains only the latest value for the row.
  - >>> Condensed = N : One row for every Update, Insert, or Delete.
  - >>> Complete = N : Starts with no data.
  - >>> Complete = Y : Contains every row from the source table.
- >>> Optional auditing columns.
  - >>> IBMSNAP\_OPERATION : A flag, (I)NS, (U)PD or (D)ELETE.
  - >>> IBMSNAP\_AUTHID : User ID that updated the source table.

- By using a consistent-change-data (CCD) table as your target type, you can keep a history of source changes. For example, you can track before and after comparisons of the data, when changes occurred, and which user ID updated the source table.
- Consistent Change Data (CCD) tables have been a popular type of target table in SQL Replication . New to Q replication in V9.1
- Fan out scenario staging table : SQL replication can use a Q Replication CCD target table as a source table for a fan out architecture.
- Pay attention to CONFLICT ACTION (“I” Ignore (or) “F” Force ) on DELETE’s with CONDENCED=“Y” & CONFLICT ACTION=“F” targets (A reuse of primary key on source + subsequent delete will replace the target row with OPERATION “I”). i.e. you will LOSE history!

Managing changes		NEDB2UG March 2007
Replication Center GUI	ASNCLP Command line Interface	
<p><b>Replication Center Launchpad</b></p> <p><b>Getting Started with the Replication Center</b> Welcome to the launchpad for the Replication Center! Click on a button to set up replication. After you click a button, the Select launchpad view field appears at the launchpad views for the three different tools.</p> <p><b>Q replication</b> This high-volume, low-latency replication solution captures changes via WebSphere MQ message queues to a target replicate to update tables in decision support systems, table by geographically-distributed applications, and more.</p> <p><b>Event publishing</b> This publishing solution, which is a feature of Q replication, committed changes as XML messages on WebSphere MQ changes. You can use them to feed data to your web appli</p> <p><b>SQL replication</b> This replication solution captures changes that are made to changes are replicated from the staging tables to target tab captured only once. You can also use the staging tables to</p>	<pre>ASNCLP SESSION SET TO Q REPLICATION; ----- CREATE QSUB SUBTYPE U USING REPLQMAP DB2_TO_ORACLE_QMAP1 (SUBNAME STAFF STAFF0001 Q.STAFF OPTIONS HAS LOAD PHASE E TARGET NAME Q.STAFF ERROR ACTION Q); ----- DROP QSUB USING REPLQMAP DB2_TO_ORACLE_QMAP1 FOR SUBNAME LIKE "STAFF0001";</pre>	

What is ASNCLP ?

- Command line processor to define Replication Scenarios.
- Calls same Java API's as the Replication Center
- Interactive and Script Mode supported

Execute ASNCLP script file example.

**asnclp -f createsub.cmd**

NEDB2UG March 2007	<h1>Target Table Load</h1>
<p>A subscription is defined as: Automatic load, No load or Manual load.</p> <ul style="list-style-type: none"> <li>&gt;&gt;&gt; Automatic load: Load is performed by the Apply task.</li> <li>&gt;&gt;&gt; No load: No loading required, no coordination required.</li> <li>&gt;&gt;&gt; Manual load: Load is performed by user, Steps involved to load Federated Oracle target tables.           <ul style="list-style-type: none"> <li>&gt;&gt;&gt; Stop and Start subscription ("CAPSTOP" &amp; "CAPSTART").</li> <li>&gt;&gt;&gt; Unload source in comma delimited format, FTP unload file to target server, Load the target table using SQL*Loader utility.</li> <li>&gt;&gt;&gt; Signal "LOADDONE".</li> </ul> </li> </ul>	

Options for Loading target tables.

**Automatic load** : "This option is not available if the target table is in a non-DB2 database. "

**No load** : No loading required, can immediately capture and apply changes.

**Manual load:** Load is performed by user, coordination is required, and will be handled by user.

With Manual Load, Load the target table using a utility of your choice, and then notify Q capture program when the table is loaded.

1. Stop capture for a table by SIGNAL "CAPSTOP" and start by SIGNAL "CAPSTART"  
 "INSERT INTO ASN1.IBMQREP\_SIGNAL (SIGNAL\_TYPE, SIGNAL\_SUBTYPE, SIGNAL\_INPUT\_IN) VALUES ('CMD', '**CAPSTART**', 'EMPLOYEE0001');"  
 (The Q Capture program starts sending changed transactions to the temporary spill queues)
2. Unload from the source, FTP file to target server, Load the target table.  
 (The Q Apply program puts changes from the source table in a temporary spill queue while waiting for the loading to finish.)
3. Signal "LOADDONE" by inserting a 'LOADDONE' signal as shown in step 1  
 (or) run ASNCLP command  
**LOAD DONE** QSUB SUBNAME EMPLOYEE0001;  
 (The Q Apply program starts applying rows from the spill queue)

Refer to Appendix B for tips and techniques to unload from DB2 z/OS, FTP and load Oracle target tables.

## Add a new column to a replicated table.

1. Start source DB2 z/OS table in read only (RO) mode. (Outage)
2. Verify Apply has processed all messages on receive Q.
3. STOP SUB. for the table you want to alter.
4. DROP SUB. and Nickname for the table you want to alter.
5. ALTER source DB2 z/OS tables.
6. Generate "Create new SUB" for table you just altered.
7. ALTER target Oracle tables, column characteristics based on "create oracle table DDL" generated in step 6.
8. Run SQL script generated in step 6.
9. SIGNAL to CAPSTART and LOADDONE to activate the SUB.
10. Start source DB2 z/OS table in read write (RW) mode.

➤ **"Federated targets:** If the nickname for the target table has columns that are not part of the existing Q subscription, you can use the ADDCOL signal to add them to the Q subscription. You must drop the Q subscription and re-create it after you alter the target table (you cannot add columns to a nickname). "

## DBA on-call will get paged when..

- >>> Monitor **Capture log** and alert on any ASN\*E messages.
- >>> Monitor **Apply log** and alert on any ASN\*E messages.
- >>> Monitor MQ and alert if source **Send Q** (or) target **Receive Q** depth is > set threshold.
- >>> Monitor and alert if **sender** (or) **receiver MQ** channel status is down.
- >>> Install and configure Replication Alert Monitor to provide automated monitoring and alerts on errors.

- >>> Install and configure Q-Capture on z/OS.
- >>> Install and configure DB2 UDB and Websphere Replication Server on target server.
- >>> Configure DB2 UDB to support Federation.
- >>> Install queue managers and queues for the Q Capture and Q Apply programs.
- >>> Create and configure Q Capture and Q Apply Control tables.
- >>> Create Queue-Map and Q-Subscriptions.
- >>> Verify setup and start replicating!

- If datasharing on z/Os, install capture on CPU with most DML activity.
  - Configuring DB2 UDB to support federated access to Oracle ..
    - Enable UDB for Federation by changing a DBM configuration.
    - Create Oracle Wrapper, Server definition and User mapping
    - All of this can be easily done by Downloading “Federation Configuration Wizard “ @  
<http://www-1.ibm.com/support/docview.wss?uid=swg27007070>
  - After you install queue managers on z/OS and target server, turn to a easy to use tool to configure your queue definitions “Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing” @  
<http://www-1.ibm.com/support/docview.wss?uid=swg27007070>
  - Use Replication Center to create Queue-map and Q-subscriptions.
  - White paper! By Dell Burner from IBM (thank you Dell !)
- Quick start for Q replication to Oracle and Sybase @**  
<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0505burner/index.html#main>
- & a Invaluable IBM source “WebSphere Replication Server (Q replication) Information roadmap @ <http://www-128.ibm.com/developerworks/db2/roadmaps/qrepl-roadmap-v8.2.html>

- >>> For Q replication, WebSphere MQ server or client installation is required.
- >>> A restricted use license of WebSphere MQ V06.0 is included with WebSphere Replication Server. (z/OS and LUW)
- >>> A restricted use license of DB2 Enterprise Server Edition V9.1 is included with WebSphere Replication Server.

➤ To obtain the current information for supported software levels for WebSphere Replication Server, refer to

➤ **DB2 z/OS**

[http://www-306.ibm.com/software/data/integration/replication\\_server\\_z/requirements.html](http://www-306.ibm.com/software/data/integration/replication_server_z/requirements.html)

&

➤ **DB2 LUW**

[http://www-306.ibm.com/software/data/integration/replication\\_server/requirements.html](http://www-306.ibm.com/software/data/integration/replication_server/requirements.html)

➤ **All Information Integrator products**

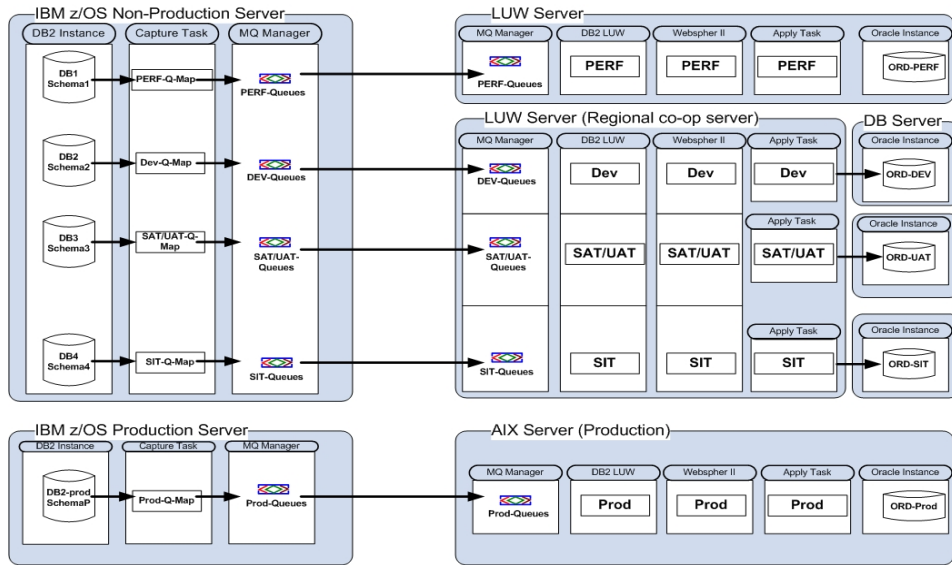
<http://www-306.ibm.com/software/sw-bycategory/subcategory/SWB50.html>

➤ Restrictions “A WebSphere MQ for z/OS subsystem cannot be a client.”

# Environment Setup

NEDB2UG  
March  
2007

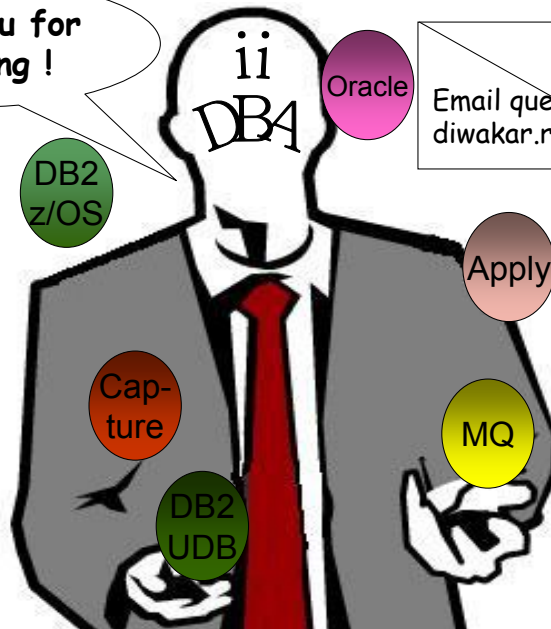
## Typical Environment Setup



# Information Integrator DBA !

FIS Technology  
Database  
Engineering

Thank you for  
attending !



Email questions to  
diwakar.rao@fmr.com

# Appendix

NEDB2UG  
March  
2007

- >>> Appendix A : Initial data sync.  
(Manual unload and Load process)
- >>> Appendix B : SPUFI DB2 LUW and Oracle tables.
- >>> Appendix C : Replication Utilities
- >>> Appendix D : Links to Additional Material

# Appendix A

NEDB2UG  
March  
2007

## Manual Unload and Load (1 of 2)

- Unload source DB2 tables using IBM unload utility in comma delimited format. (comma DELIMITED is new in DB2 v8).
- Convert IBM Load card to Oracle SQL Loader control card. (in the notes find sample code to convert SYSPUNCH output to SQLLDR control card).
- FTP Unload and SQLLDR control cards to target server.
- Use shell scripts to load all FTP'ed files.
- SIGNAL LOADDONE after loading target oracle tables.

```
/* REXX */ (SAMPLE CODE TO CONVERT DB2 UNLOAD CARD TO ORACLE SQL*LOADER CONTROL CARD)
"ISREDIT MACRO "
"ISREDIT CHANGE X'7F' X'40' ALL "
"ISREDIT C P'TEMPLATE =====' 1 UNRECOVERABLE ALL "
"ISREDIT EX ' DSN(' ALL "
"ISREDIT EX ' DISP(OLD' ALL "
"ISREDIT EX ' SORTKEYS' 1 ALL "
"ISREDIT EX ' FORMAT DELIMITED' ALL "
"ISREDIT DEL ALL EX "
"ISREDIT C POSITION(*) X'40' ALL "
"ISREDIT C P' CHAR(=====)'CHAR' ALL "
"ISREDIT C P' VARCHAR(=====)'VARCHAR' ALL "
"ISREDIT C ' DECIMAL ' 'DECIMAL EXTERNAL' ALL "
"ISREDIT C ' INTEGER ' 'INTEGER EXTERNAL' ALL "
"ISREDIT C ' SMALLINT ' 'SMALLINT EXTERNAL' ALL "
"ISREDIT C ' TIMESTAMP EXTERNAL ' 'TIMESTAMP' ALL "
"ISREDIT C ' TIMESTAMP ' 'TIMESTAMP' ALL "
"ISREDIT C ' TIMESTAMP ' 'TIMESTAMP 'YYYY-MM-DD-HH24.MI.SS.FF6" ALL "
"ISREDIT C ' DATE EXTERNAL ' 'DATE EXTERNAL' ALL "
"ISREDIT C ' DATE EXTERNAL ' ' DATE 'YYYY-MM-DD" ALL "
"ISREDIT C ' TIME EXTERNAL ' 'TIME EXTERNAL' ALL "
"ISREDIT C ' TIME EXTERNAL ' ' DATE 'HH24.MI.SS" ALL "
"ISREDIT C 'LOG NO RESUME YES' X'40' ALL "
"ISREDIT C P'LOAD DATA =====' 'LOAD DATA' ALL "
"ISREDIT C P' EBCDIC =====' TRUNCATE ALL "
CLINE1 = ""TRAILING NULLCOLS""
CLINE2 = ""FIELDS TERMINATED BY OPTIONALLY ENCLOSED BY""
"ISREDIT F 'INTO TABLE' FIRST "
"ISREDIT LINE_AFTER .ZCSR = &CLINE1"
"ISREDIT LINE_AFTER .ZCSR = &CLINE2"
"ISREDIT EX 'INTO TABLE' ALL "
"ISREDIT C ' . ' : ' X ALL "
"ISREDIT EX ALL "
"ISREDIT F 'FIELDS TERMINATED""
"ISREDIT C 'BY OP' X'C2E8407F6B7F40D6D7' NX ALL"
"ISREDIT C 'SED BY' X'E2C5C440C2E8407D7F7D' NX ALL"
"ISREDIT RES "
```

## Manual Unload and Load (2 of 2)

>>> FTP Unload and converted SQL\*LDR control cards to target server.

>>> Sample FTP JCL.

```
//ftpjob01 JOB ...  
//FTP EXEC PGM=IBMFFTP,PARM=('EXIT')  
//SYSPRINT DD SYSOUT=*  
//OUTPUT DD SYSOUT=*  
//INPUT DD *  
servename  
userid  
password  
cd /ftpfiles/db2/data  
put 'xyz.abc.db1.table1.unload' table1.dat  
put 'xyz.abc.db1.cards(table1)' table1.ctl  
QUIT
```

>>> Sample shell script to load all FTP'ed files (in notes section).

```
# sample shell script to load Oracle tables for every *.ctl files in a directory  
CTL_DIR=/ftpfiles/db2/data  
DATA_DIR=/ftpfiles/db2/data  
SQLLDR_FILE=/ftpfiles/db2/data/sqlldr_script.sh  
UNAME=oracle-database-user  
PASS=oracle-user-password  
DB=oracle-database-name  
export CTL_DIR DATA_DIR SQLLDR_FILE UNAME PASS DB  
cd $CTL_DIR  
for i in `ls *.ctl`  
do  
  LENGTH=`expr "$i" : ".*"`  
  LENGTH=`expr $LENGTH - 4`  
  file_name=`expr substr $i 1 $LENGTH`  
  echo $file_name  
  echo "sqlldr $UNAME/$PASS@$DB direct=true errors=0 \\\ " >> $SQLLDR_FILE  
  echo "control=$CTL_DIR/$file_name.ctl \\\ " >> $SQLLDR_FILE  
  echo "data=$DATA_DIR/"$file_name".dat \\\ " >> $SQLLDR_FILE  
  echo "log=$DATA_DIR/$file_name.log" >> $SQLLDR_FILE  
  echo " " >> $SQLLDR_FILE  
done  
sh $SQLLDR_FILE > sqlldr_script_dtpy.log  
rm $SQLLDR_FILE
```

## SPUFI DB2 LUW & Oracle Tables

- >>> Very handy when you want to query/change DB2 LUW (or) Oracle tables (via. nicknames) from SPUFI on mainframe.
- >>> To setup SPUFI access to remote tables, Insert rows into sysibm.ipnames, locations and usernames (as shown in the notes).
- >>> Bind SPUFI package DSNESM68 to remote DB2 UDB database. (you can either run the bind in DB2 client configuration assistant GUI – right click the DB and bind (or) Bind from mainframe by specifying the remote LINKNAME -- BIND PACKAGE(DB2RMT1.DSNESPCS) MEMBER(DSNESM68) with SQLERROR(CONTINUE) )
  - >>> SPUFI example  
SELECT \* FROM ASN1.IBMQREP\_SUBS;  
SELECT \* FROM DB2RMT1.ASN1.IBMQREP\_TARGETS;
- >>> Need to create the Oracle wrapper, server definition & user mapping before you can create nicknames and access Oracle tables.

```
INSERT INTO SYSIBM.IPNAMES (LINKNAME, SECURITY_OUT, USERNAMES,  
IBMREQD, IPADDR)
```

```
VALUES ('DB2RMT1', 'P', 'O', 'N', '00.00.000.00');
```

-- LINKNAME = makeup any name to identify a remote location, must use the --  
same name in LOCATIONS and USERNAMES table.

-- TSO PING *target logical hostname* to get IPADDR.

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME, IBMREQD, PORT)  
VALUES ('DB2DB1', 'DB2RMT1', 'N', '60000');
```

-- LOCATION = Federated DB2 database name.

```
INSERT INTO SYSIBM.USERNAMES (TYPE, LINKNAME, NEWAUTHID,  
PASSWORD, IBMREQD)
```

```
VALUES ('O', 'DB2RMT1', 'db2id1', 'db2idpw', 'N');
```

-- NEWAUTHID and PASSWORD can be case sensitive.

```
COMMIT;
```

## Replication Utilities

**>> asntdiff**

Write differences between the source and the target table, into a table as a set of SQL operations to perform to resynchronize the tables (e.g., insert, update, delete)

**>> asntrep**

Repairs target based on differences found by asntdiff

**>> asnqfmt**

Format the contents of a send queue

**>> Replication Center Alert Monitor**

Provides automated monitoring of your replication environment

Alerts you to error and other conditions

**>> Q Replication Analyzer**

Generates report about the state of your replication environment

## Links to Additional Material

**>>> WebSphere Replication Server (Q replication) Information roadmap**

<http://www-128.ibm.com/developerworks/db2/roadmaps/qrepl-roadmap-v8.2.html>

**>>> Quick start for Q replication to Oracle and Sybase**

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0505burner/index.html#main>

**>>> Q Replication Tools**

<http://www-1.ibm.com/support/docview.wss?uid=swg27007070>

**>>> Information Integrator Q replication Performance considerations**

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0503aschoff/>

**>>> All Information Integrator products**

<http://www-306.ibm.com/software/sw-bycategory/subcategory/SWB50.html>