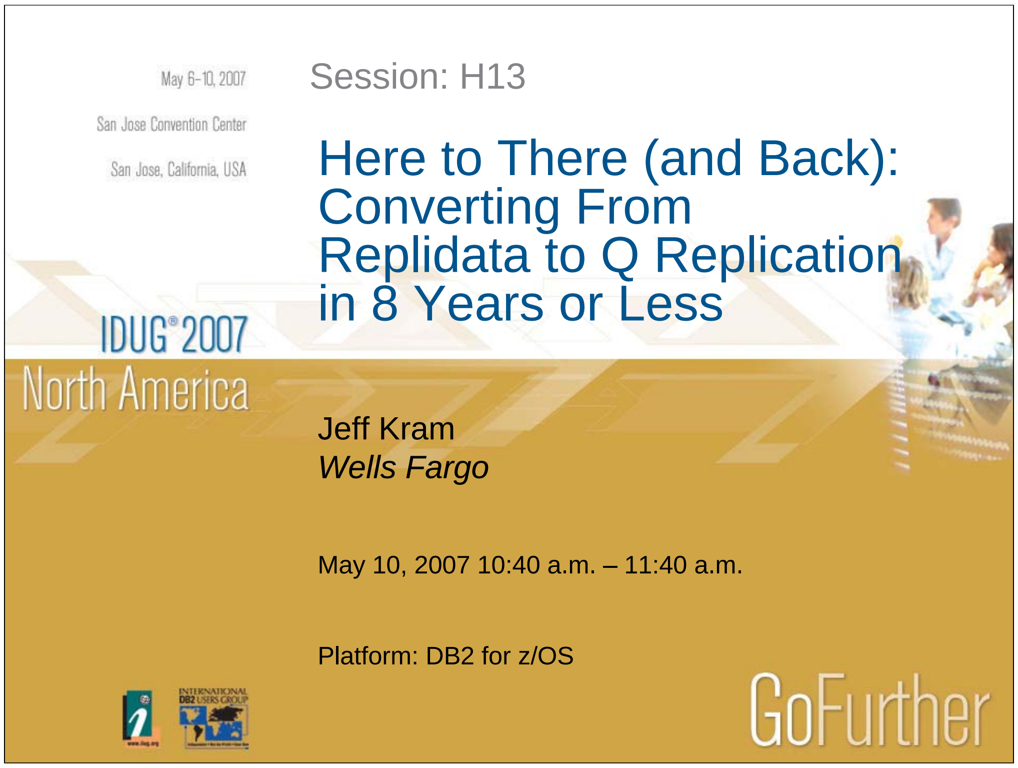


May 6–10, 2007

San Jose Convention Center

San Jose, California, USA

Session: H13

# Here to There (and Back): Converting From Replidata to Q Replication in 8 Years or Less

IDUG® 2007 North America

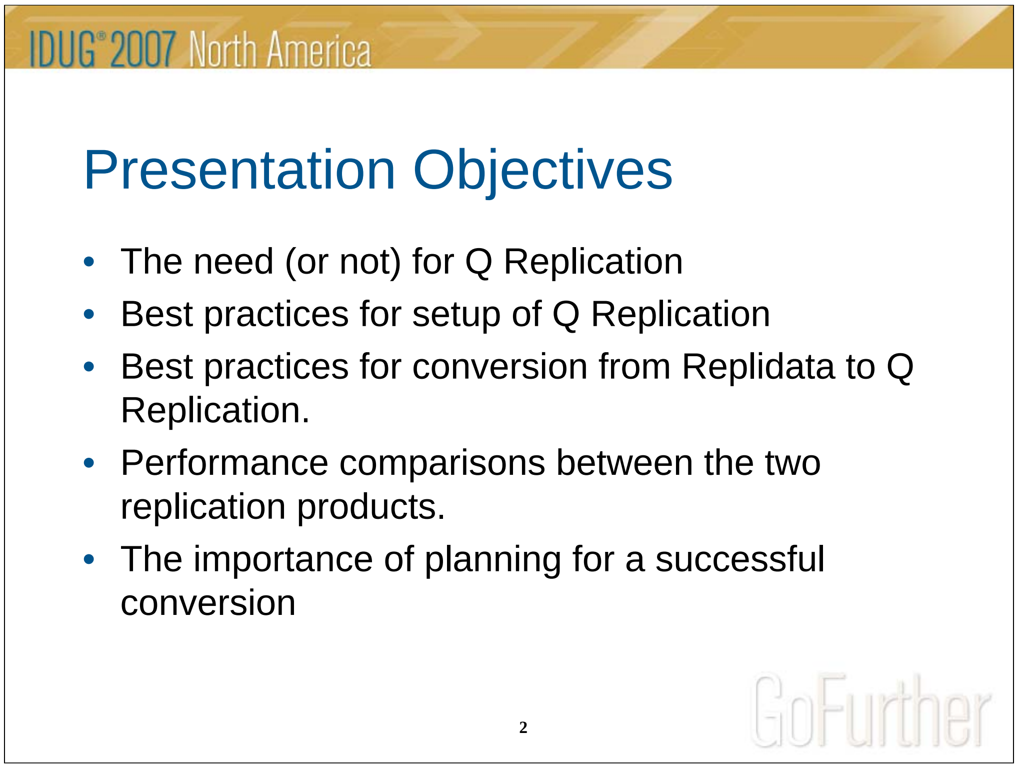
Jeff Kram
*Wells Fargo*

May 10, 2007 10:40 a.m. – 11:40 a.m.

Platform: DB2 for z/OS

GoFurther

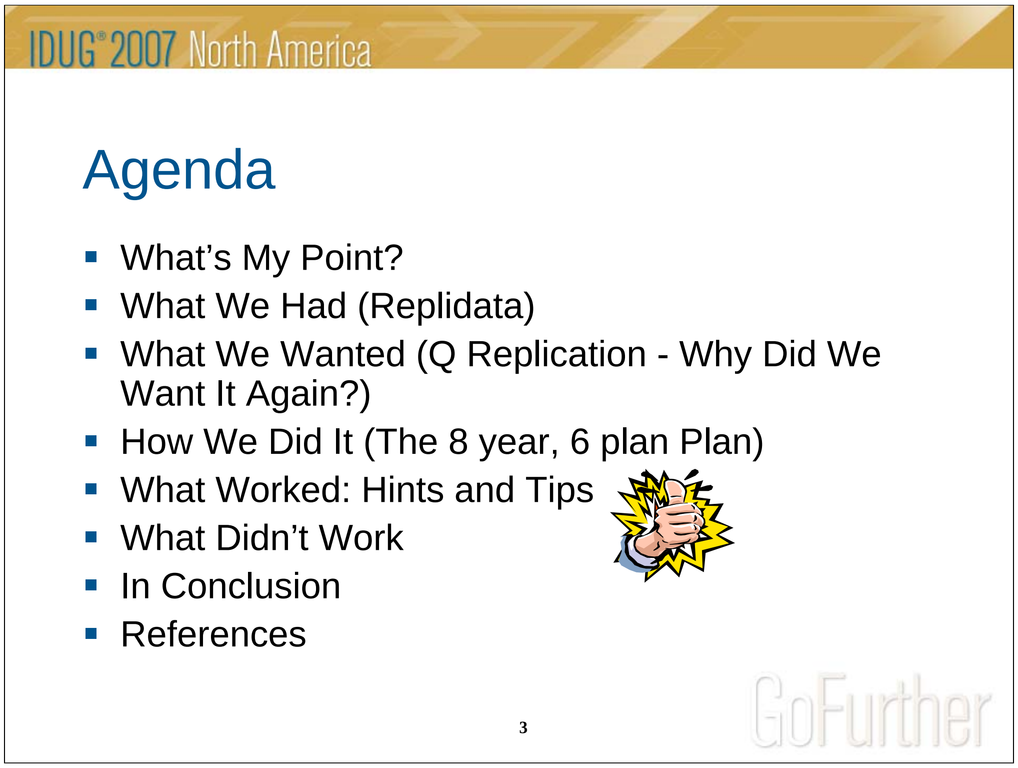
This presentation will cover our experience (Wells Fargo) from converting from Replidata to Q Replication as our replication product solution. Topics will include the pitfalls of setup and conversion: what challenges we faced in implementing Q Replication, performance issues, and hints and tips for a successful install and conversion.



## Presentation Objectives

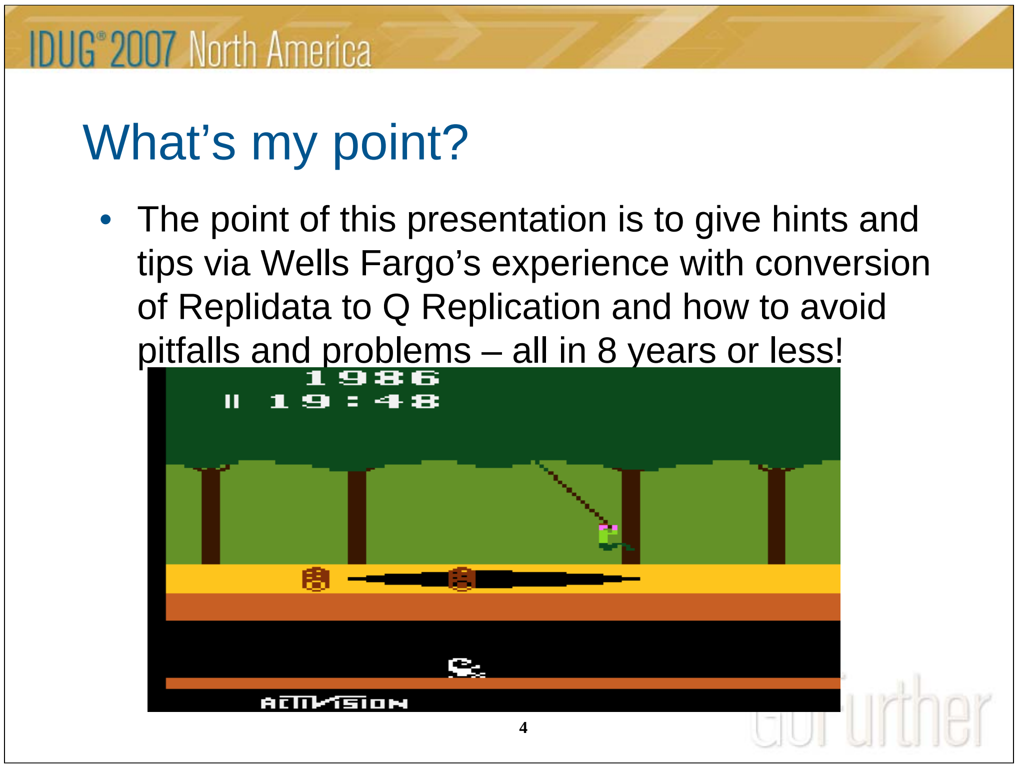
- The need (or not) for Q Replication
- Best practices for setup of Q Replication
- Best practices for conversion from Replidata to Q Replication.
- Performance comparisons between the two replication products.
- The importance of planning for a successful conversion

GoFurther



## Agenda

- What's My Point?
- What We Had (Replidata)
- What We Wanted (Q Replication - Why Did We Want It Again?)
- How We Did It (The 8 year, 6 plan Plan)
- What Worked: Hints and Tips
- What Didn't Work
- In Conclusion
- References



IDUG® 2007 North America

## What's my point?

- The point of this presentation is to give hints and tips via Wells Fargo's experience with conversion of Replidata to Q Replication and how to avoid pitfalls and problems – all in 8 years or less!

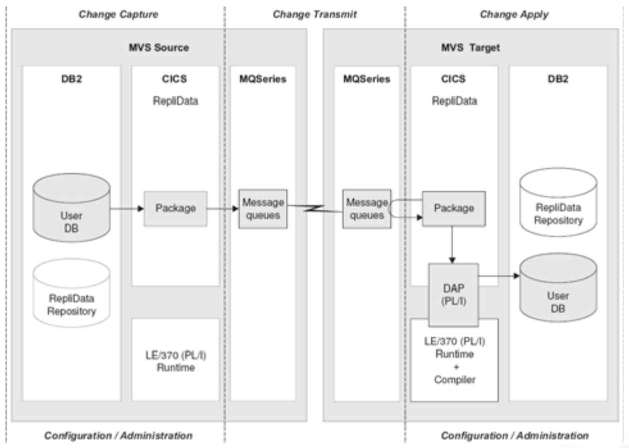
4

WebSphere Replication Server is the latest name – I'll be referring to it as Q Replication for this presentation.

Preface: I had no Replidata background going into this project – learned it all from my local Replidata guru. Attending Q Replication class, otherwise it's been all hands on. I'm also assuming some knowledge of Q Replication by the audience, but I will explain as many concepts as I can, where it applies.

IDUG® 2007 North America

## What We Had

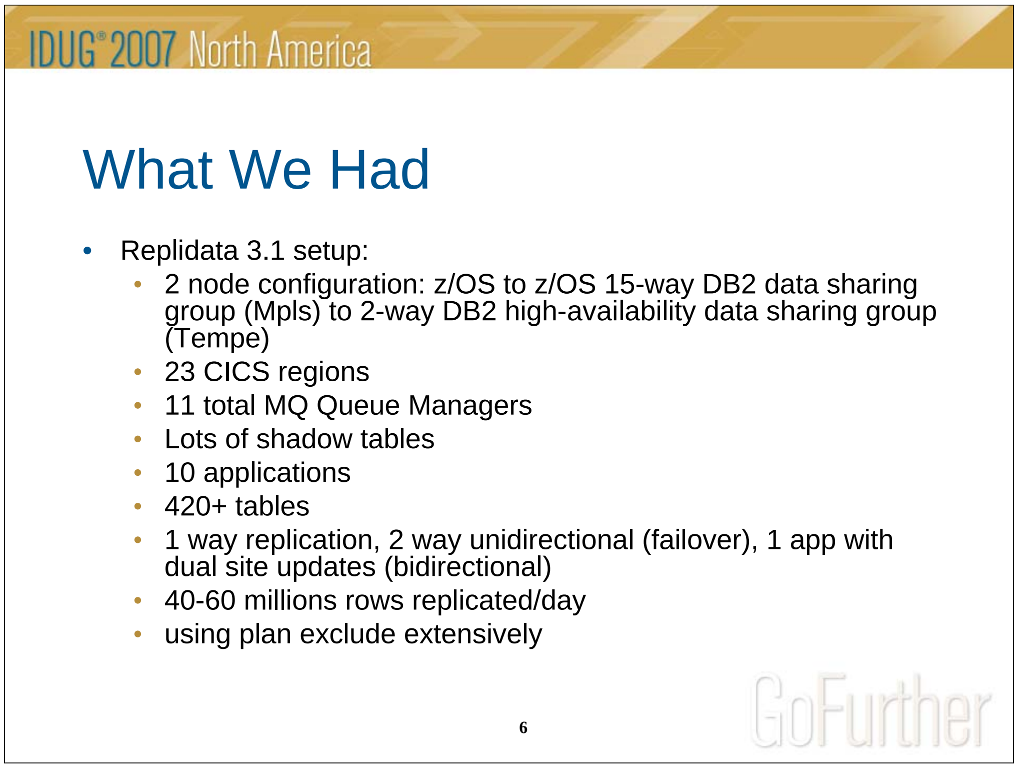


5

GoFurther

Graphic taken from Replidata manual – Concepts and Planning 3.1 (ISRD-COPL-03). Conceptually what Replidata looks like and how it operates.

For every table replicated, you need a shadow table (built in DB2). RepliData's started tasks read the DB2 logs, then write the changed data to MQ queues which are read by the source CICS region. The data is then written to an MQ queue, which transmits the packages to the target site MQ queue, which is then read by CICS and written to the application tables and the shadow tables. Each DB2 member has a log reader.


IDUG® 2007 North America

## What We Had

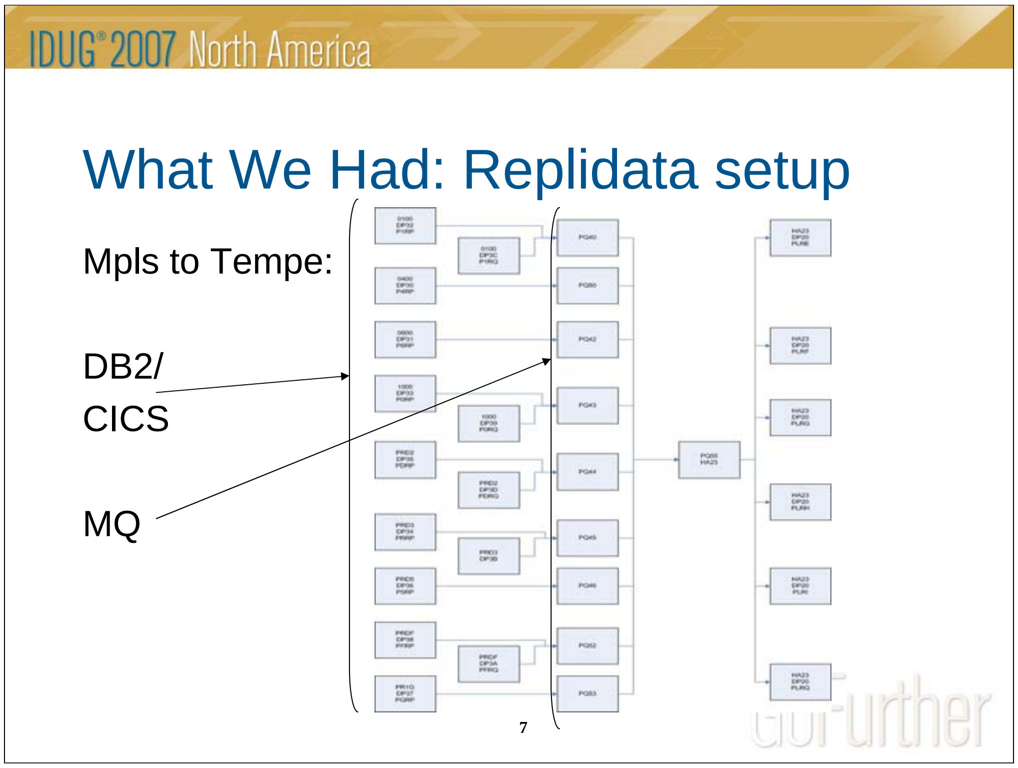
- Replidata 3.1 setup:
  - 2 node configuration: z/OS to z/OS 15-way DB2 data sharing group (Mpls) to 2-way DB2 high-availability data sharing group (Tempe)
  - 23 CICS regions
  - 11 total MQ Queue Managers
  - Lots of shadow tables
  - 10 applications
  - 420+ tables
  - 1 way replication, 2 way unidirectional (failover), 1 app with dual site updates (bidirectional)
  - 40-60 millions rows replicated/day
  - using plan exclude extensively

6

GoFurther

Lots of information here, but the point of this slide is to show we are large, have a fairly complex Replidata environment, and have certain requirements for replication.

The High-availability site is for critical apps only (5).



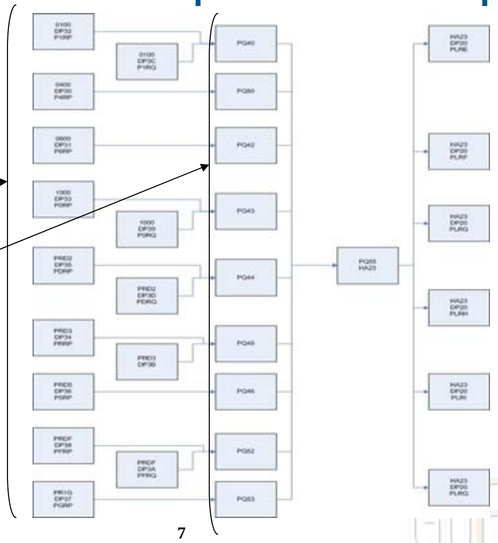
IDUG® 2007 North America

## What We Had: Replidata setup

Mpls to Tempe:

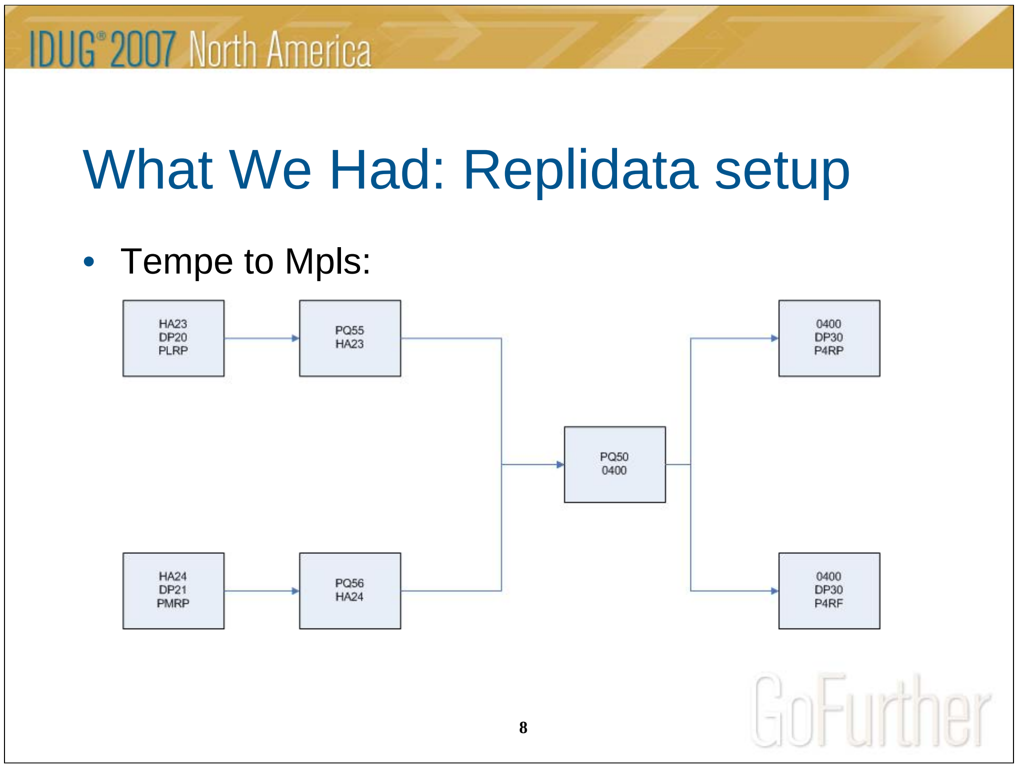
DB2/
CICS

MQ



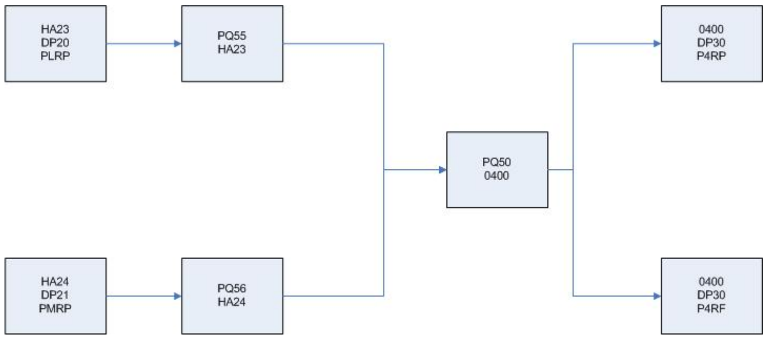
7

...And here is what our environment looks like with Replidata. This is the Mpls to Tempe piece of the puzzle. Note that some LPARS have more than 1 DB2 member on them – this would present somewhat of a problem for Q Replication when it came to automation and startup/shutdown/failover.


IDUG® 2007 North America

## What We Had: Replidata setup

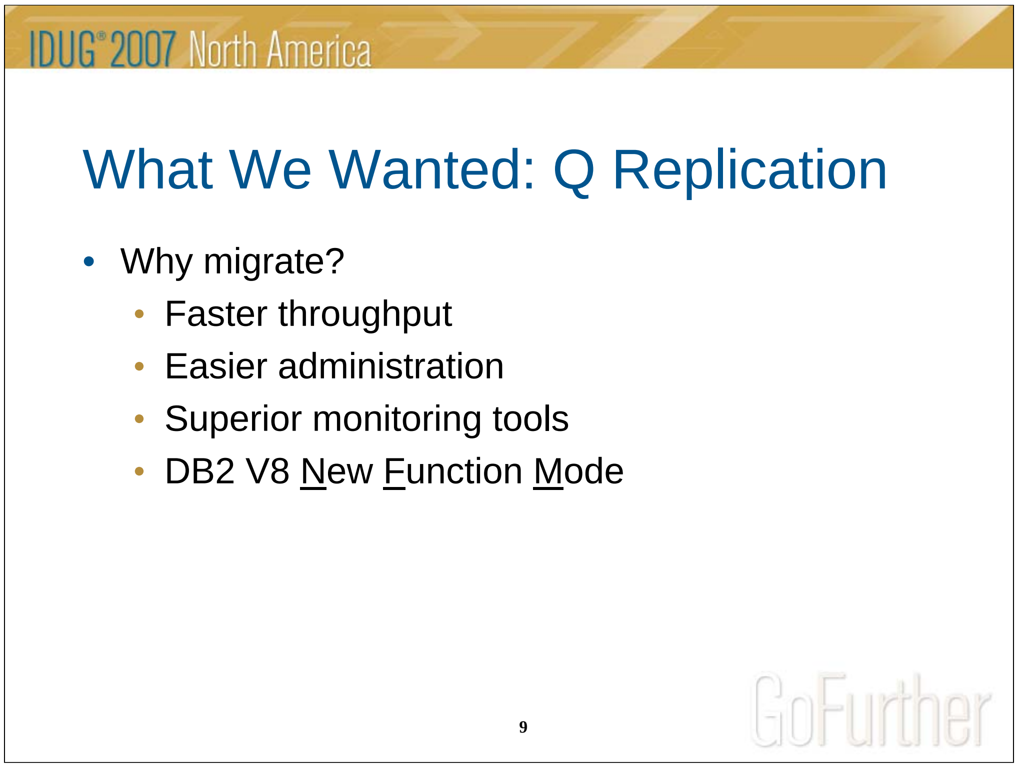
- Tempe to Mpls:



8

GoFurther

Here is the Tempe to Mpls view. Replication routes through one LPAR on the Mpls side.


IDUG® 2007 North America

## What We Wanted: Q Replication

- Why migrate?
  - Faster throughput
  - Easier administration
  - Superior monitoring tools
  - DB2 V8 <u>N</u>ew <u>F</u>unction <u>M</u>ode

9

GoFurther

Q Replication – formal name is Websphere Information Integrator v8.2 or WebSphere Replication Server (9.1):

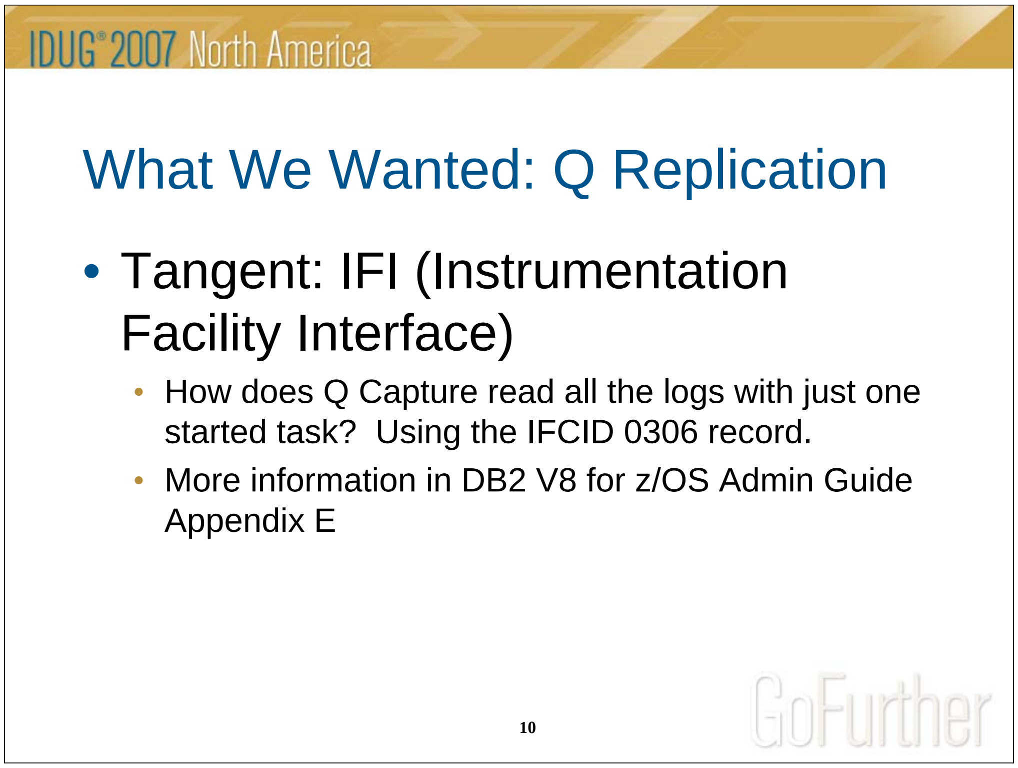
Faster throughput: Q Replication has half the number of messages being sent back and forth vs. Replidata. High throughput with low latency. Use of 1 log reader accomplished by reading the IFCID 306 record which contains the merged log record of all members of the data sharing group. Replidata needed 1 log reader per member of the data sharing group – in our case, 15. This is because Replidata did not have the 'flag' turned on to read the merged log records – Q Replication does. Performance is also increased with Q Replication due to the lack of a CICS component.

Easier administration: Adding tables to replication is very easy with the Replication Center GUI. Adding tables in Replidata was not so simple. There were new shadow tables to be built, binds to be done, in addition to the Replidata table replication entries. Replication Center came with Control Center which was downloaded as a part of DB2 Connect Enterprise Edition.

Superior monitoring tools: Real time monitoring and Performance tracking: nothing like it exists in Replidata. Monitoring tables keep track of latency, rows applied, trans applied, exceptions, etc. Dashboard lets you see real time replication latency in its various pieces – Q Capture, MQ, and Q Apply.

DB2 V8: This was the main reason for moving to Q Replication. Technically, as long as you don't use any new functions in DB2 V8 NFM you can use Replidata. This was not a practical solution. Neither Replidata 3.1 nor 4.1 supports NFM.

As you can see there are many advantages to migrating to Q Replication. In our case the reasons turned out to be more practical than anything – upgrading to V8 of DB2 for Z/OS meant we had to consider other replication alternatives.



IDUG® 2007 North America

## What We Wanted: Q Replication

- Tangent: IFI (Instrumentation Facility Interface)
  - How does Q Capture read all the logs with just one started task? Using the IFCID 0306 record.
  - More information in DB2 V8 for z/OS Admin Guide Appendix E

10

GoFurther

Both Replidata and Q Replication read the IFCID 306 record, but Q Replication has the merged log record flag/byte set.

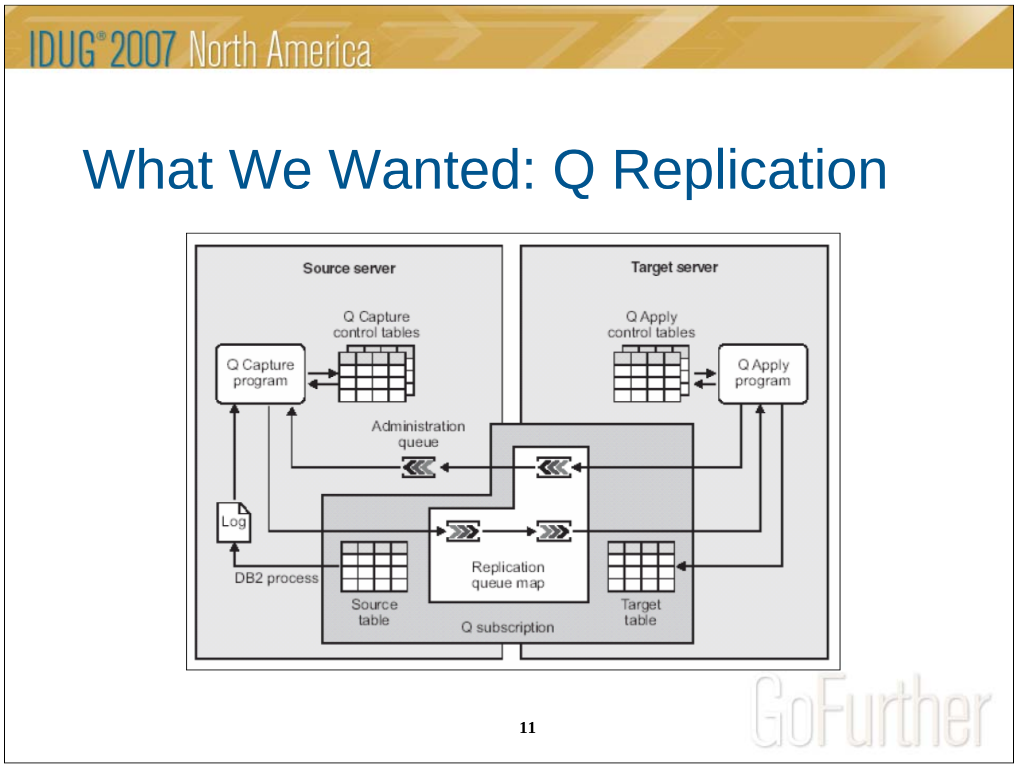
Q Capture retrieves log records via a READS request (synchronous) to the IFI for IFCID 0306.

Additional info (Text taken from an ETR opened with IBM):

The IFI is accessed through a call to an API provided by DB2 (CALL DSNWLI). The DB2 manuals have many pages dedicated to IFI. Log records are stored in ECSA memory buffers and then committed to DB2 log files (disk, tape). Both SQL and Q Capture would get the log records from buffers. But if Capture has been stopped for long enough then the log records would be fetched from disk or if the down time was long enough archived log files. Again, all of the log retrieval is performed by DB2's IFI.

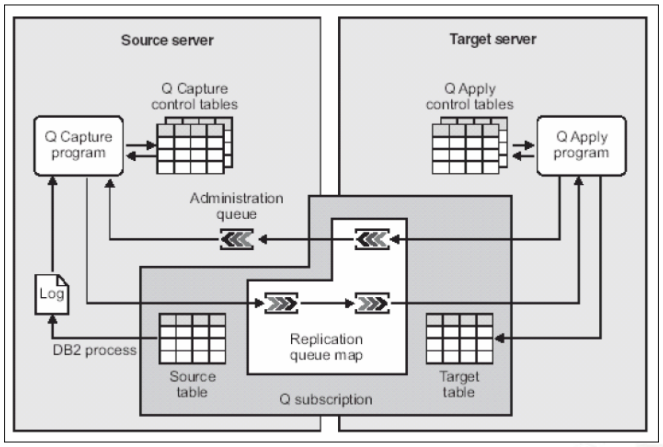
Description of the 306 record:

> IFCID 0306 IS DEFINED FOR THE IFI READS FUNCTION TO
> RETURN DB2 LOG RECORDS. THE APPLICATION CAN RETRIEVE
> LOG RECORDS BASED ON USER-SPECIFIED CRITERIA. FOR
> EXAMPLE, THE USER CAN RETRIEVE COMPRESSED OR DECOMPRESSED
> RECORDS.
> IN A DATA SHARING ENVIRONMENT, LOG RECORDS ARE RETURNED
> IN MERGED TIMESTAMP SEQUENCE FROM ALL ACTIVE MEMBERS OF
> THE DATA SHARING GROUP, IF WQALFLTR=X'00'.



IDUG® 2007 North America

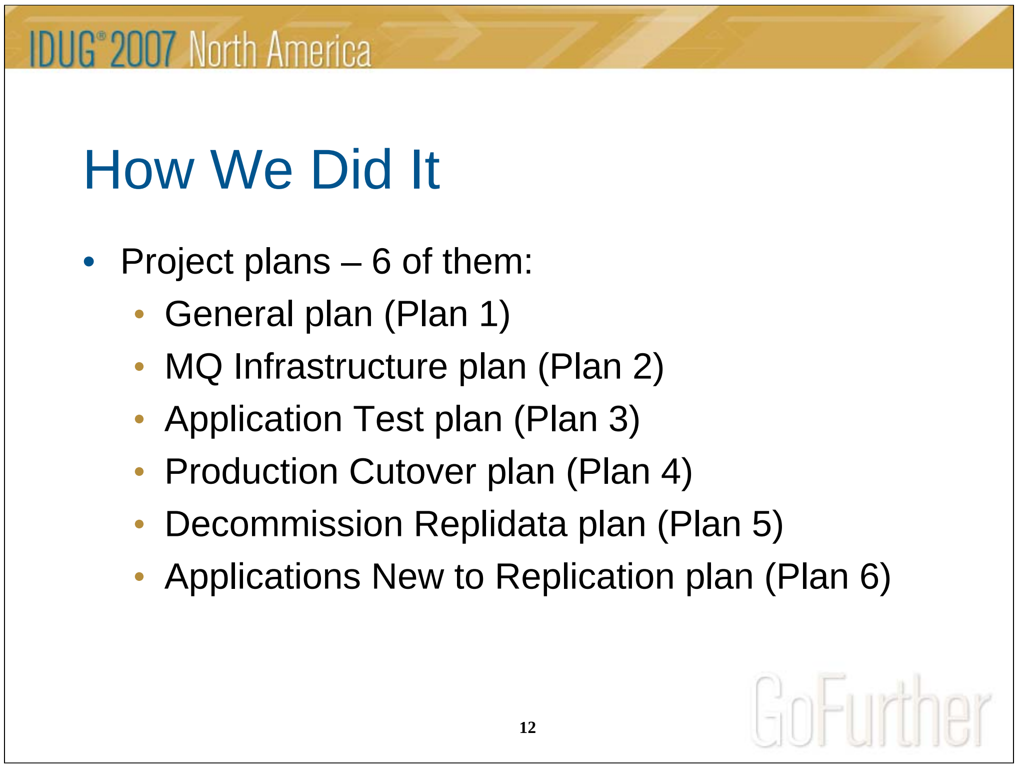
## What We Wanted: Q Replication



11

GoFurther

Taken from Redbook: WebSphere Information Integrator Q Replication: Fast Track Implementation Scenarios.  This is conceptually what Q Replication looks like: control tables for the Q Capture and Q Apply, with MQ as the delivery mechanism, and no CICS piece.  Changes made to a user (source) table on the Source side are recorded in the DB2 log, where they are read by the Q Capture started task and sent to the Target side via MQ queues.  The Q Apply program receives these changes and applies them to the (target) table using ‘plain old’ SQL.  Q Capture and Q Apply control tables keep track of which tables are replicated by way of subscriptions (a subscription defines the source table and target table and what type of replication and any conflict rules that apply), and talk to each other via MQ and the Admin queue.  An optional monitoring task can be set up to monitor and send out alerts for various warnings, errors, queue issues, and statuses.  There is also a USS (Unix System Services) component that allows you to run the Replication programs from a USS shell.  Since we run our programs via JCL, the only piece we use is the messages (MSGS) portion, which contains all the error codes and descriptions.


IDUG® 2007 North America

## How We Did It

- Project plans – 6 of them:
  - General plan (Plan 1)
  - MQ Infrastructure plan (Plan 2)
  - Application Test plan (Plan 3)
  - Production Cutover plan (Plan 4)
  - Decommission Replidata plan (Plan 5)
  - Applications New to Replication plan (Plan 6)

12

GoFurther

After an initial period of one plan, no PM, and delays with getting the product installed, we got a project manager and came up with a plan. 6 of them, actually:

General plan (Plan 1)

MQ Infrastructure plan (Plan 2)

Application Test plan (Plan 3)

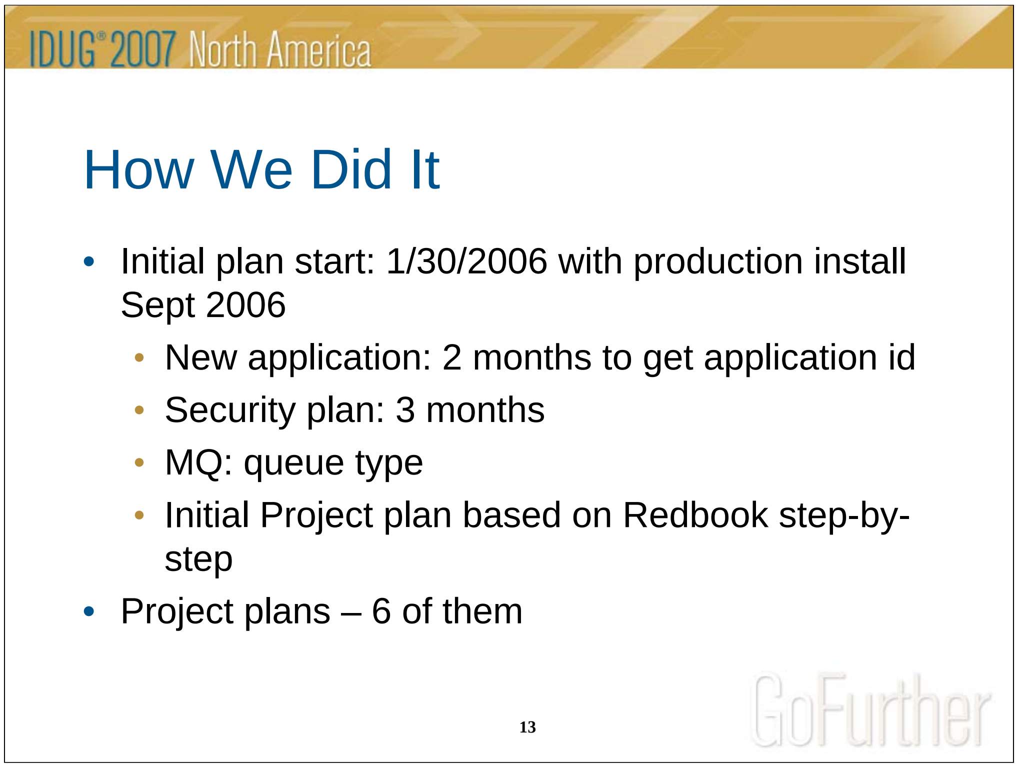
Production Cutover plan (Plan 4)

Decommission Replidata plan (Plan 5)

Applications New to Replication plan (Plan 6)

And lots and lots of meetings!

How did we arrive at this? A quick background...


IDUG® 2007 North America

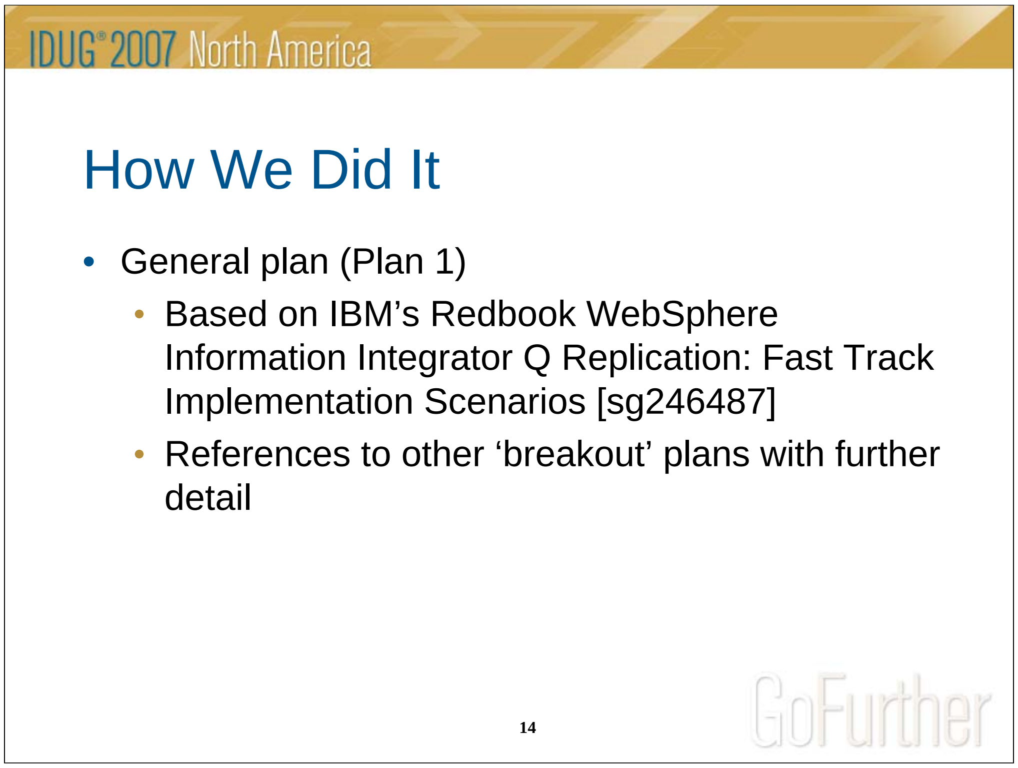
## How We Did It

- Initial plan start: 1/30/2006 with production install Sept 2006
  - New application: 2 months to get application id
  - Security plan: 3 months
  - MQ: queue type
  - Initial Project plan based on Redbook step-by-step
- Project plans – 6 of them

13

GoFurther

September 2006 was targeted because DB2 V8 was going to NFM in October. This being a new application, we had some paperwork to do, including a security plan and setting up the high level qualifiers (HLQ) in order to download the product. Replidata 4.1 was brought in to be installed in the event that Q Replication was delayed – which it was, but so was version 8.


IDUG® 2007 North America

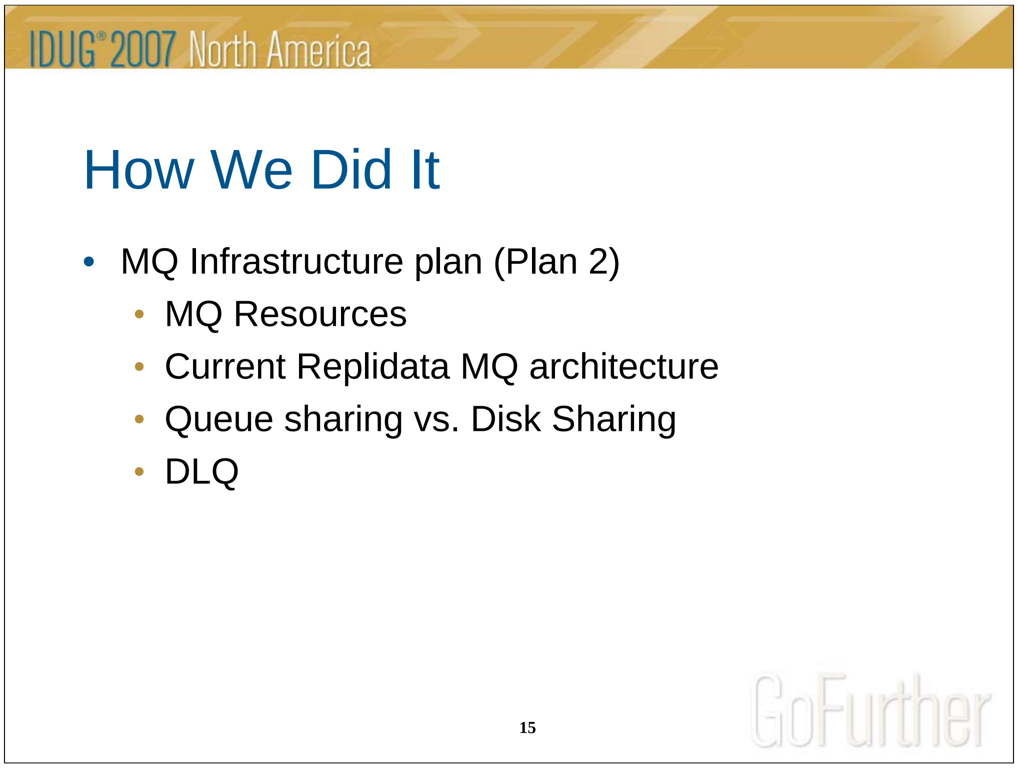
## How We Did It

- General plan (Plan 1)
  - Based on IBM’s Redbook WebSphere Information Integrator Q Replication: Fast Track Implementation Scenarios [sg246487]
  - References to other ‘breakout’ plans with further detail

14

GoFurther

This was the first plan before the others came about – copied almost directly from the Redbook step by step guide (Chapter 3.5). Eventually the other plans came about, so references were inserted into this general plan to refer to the others as needed.


IDUG® 2007 North America

## How We Did It

- MQ Infrastructure plan (Plan 2)
  - MQ Resources
  - Current Replidata MQ architecture
  - Queue sharing vs. Disk Sharing
  - DLQ

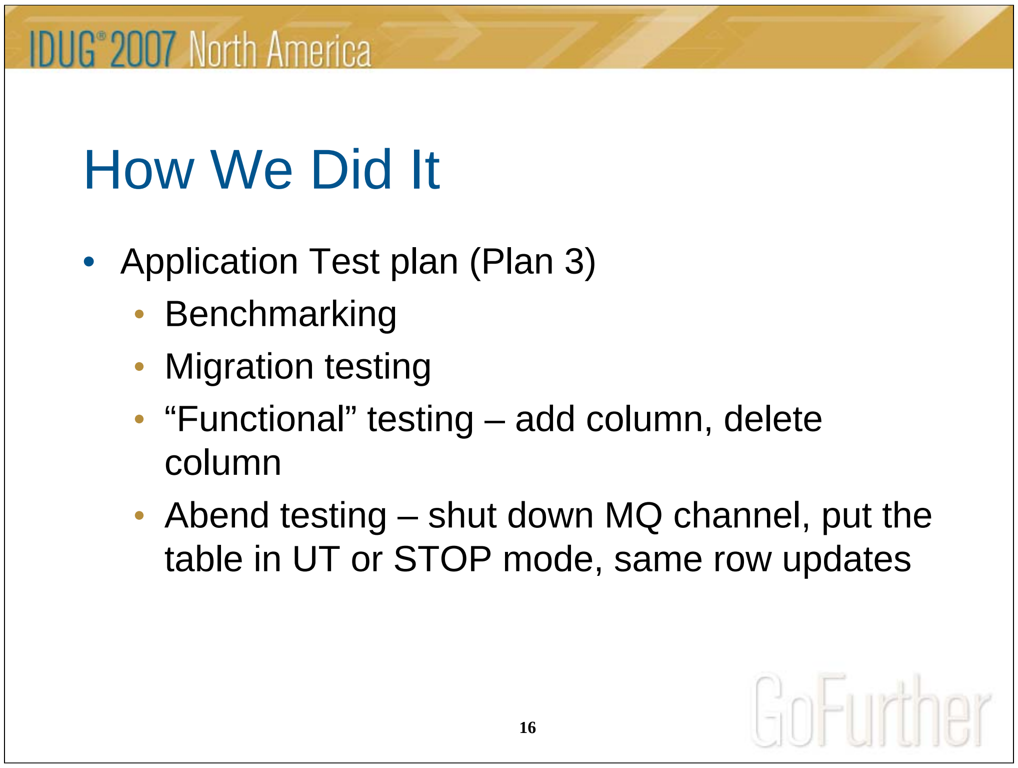
15

GoFurther

This plan dealt with all things MQ, specifically with the MQ infrastructure, and also capacity as it related to Coupling Facility (CF) storage for the queues. Initially a decision was made (conceptually) to go with queue sharing, where the queues were defined on all members of the LPARs, allowing Q Capture and Q Apply to run on any lpar. This would provide for the greatest flexibility in the event of a problem on one lpar, or during IPLs. However we later discovered that the CF needs for our plan would be greater than anticipated. Eventually disk sharing was decided as the route to take for now, with queue sharing to be reevaluated down the road.

What is Queue sharing vs. Disk Sharing? In a nutshell, queue sharing is like DB2 data sharing, where the queues are defined on all LPARs, and MQ – and therefore Q Replication – could be run on any available machine. This is accomplished by using the Coupling Facility. Disk sharing utilizes shared disk – accessible by any LPAR, but the queue managers need to be brought up on the new LPAR. In short, there is a longer recovery time for disk sharing than queue sharing, but queue sharing requires more resources.

DLQ – dead letter queues. Queue managers are normally set up with them. However, for Q Replication, it was decided that dead letter queues would complicate the process and potentially cause problems. Dead letter queues were not built for the new queue managers for Q Replication.



IDUG® 2007 North America

## How We Did It

- Application Test plan (Plan 3)
  - Benchmarking
  - Migration testing
  - "Functional" testing – add column, delete column
  - Abend testing – shut down MQ channel, put the table in UT or STOP mode, same row updates

16

GoFurther

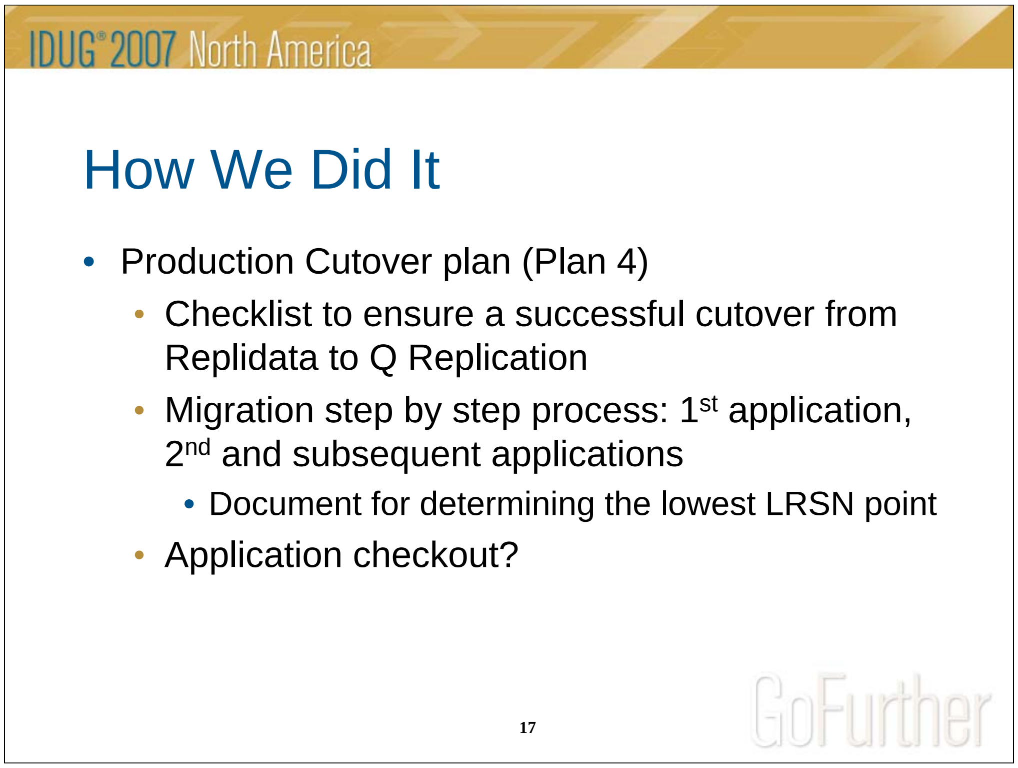
This plan was initially created to test Q Replication with our first application, and "kick the tires", and do some benchmark testing against Replidata, as well as do some "breakage" tests to see how Q Replication works (and to see how we fix it).

Benchmarking: Since Replidata doesn't keep track of any performance statistics (at least in our shop), the only really valid criteria we could come up with for comparing the two was wallclock time. A testing tool (TPNS) was used to simulate thousands of transactions, in order to get a reasonable wallclock estimate (by reasonable I mean more than a few seconds).

Migration testing: This plan also included the first steps of migrating an application to Q Replication. More detail on this is covered in Plan 4 – the production cutover plan.

Functional testing: This involved testing out everyday scenarios like adding and removing columns on tables, setting up subscriptions, starting and stopping them via Replication Center.

Abend testing: This involved breaking certain parts of the replication process and seeing what would happen – stopping a table, stopping MQ or a channel, updating the same column in a row to test the conflict rules. (With Bidirectional or Peer to Peer replication you decide what happens if the same column/row is updated at the same time – you can choose one site/server to always be the winner (bidirectional), or you can choose the update with the most recent timestamp as the winner (peer to peer).


IDUG® 2007 North America

## How We Did It

- Production Cutover plan (Plan 4)
  - Checklist to ensure a successful cutover from Replidata to Q Replication
  - Migration step by step process: 1st application, 2nd and subsequent applications
    - Document for determining the lowest LRSN point
  - Application checkout?

17

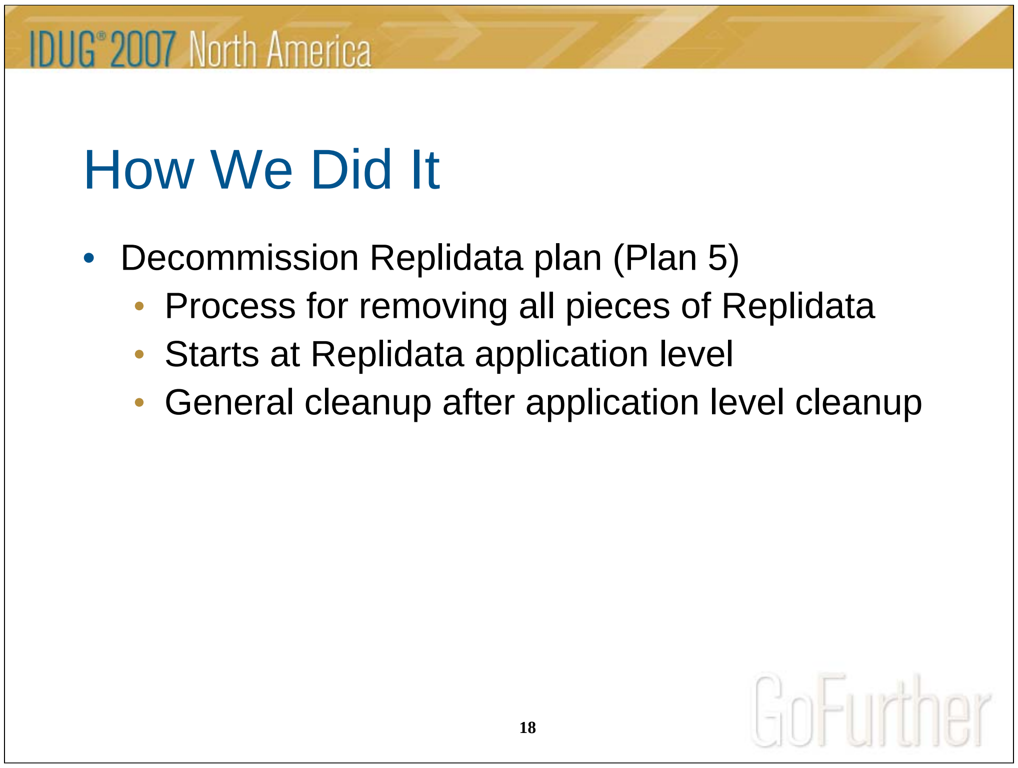
GoFurther

This plan ensured we had a successful cutover from Replidata to Q Replication. Our plan was to convert an application at a time, so we needed a step by step plan as to how to deactivate the application from Replidata and start them up on Q Replication without interruption. And we needed to do this for the subsequent applications without affecting the applications that had already migrated to Q Replication.

Document for determining the lowest LRSN point – can email on demand [getting OK from Donna first]. This piece is important because it provides seamless cutover; you need to determine what the last log record was that was read by Replidata to plug into Q Replication Q Capture job so there is no interruption in replication.

Thanks to Donna Kelsey from IBM who gave us this process.

How is the application involved in this process? In our case there was no application changes, except for the ones that opted for peer to peer replication (at the time of this, all were either unidirectional or bidirectional). All the application was responsible for was testing that their application still worked after the migration, and the replication was working. Some applications chose not to do any checkout.


IDUG® 2007 North America

## How We Did It

- Decommission Replidata plan (Plan 5)
  - Process for removing all pieces of Replidata
  - Starts at Replidata application level
  - General cleanup after application level cleanup

18

GoFurther

This plan involved the gradual retirement and cleanup of Replidata from the production environment. Process first starts after an application is converted to Q Replication – the shadow tables used by Replidata are put into a utility (UT) status before being dropped. Eventually all application related pieces are stopped, before the general cleanup begins. The general cleanup after application level cleanup includes:

•Free DB2 replidata plans

•Stop DB2 replidata database

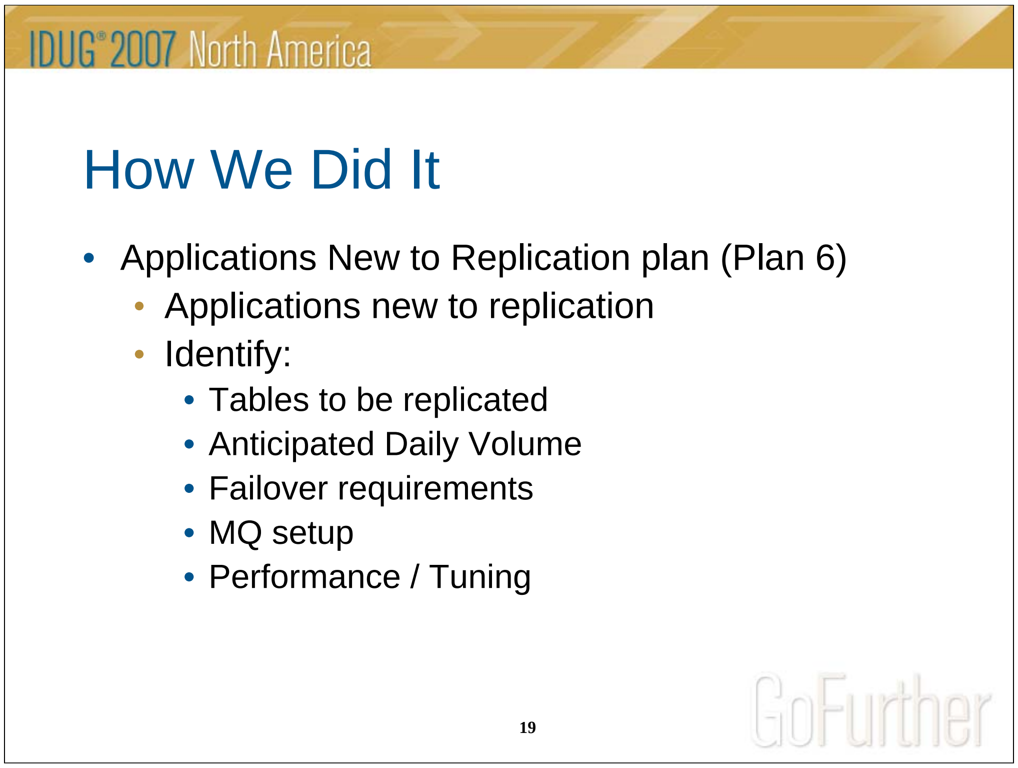
•Stop CICS regions

•Delete all VSAM files

•Stop MQ queue managers

•Security cleanup

•Drop DB2 replidata database



IDUG® 2007 North America

## How We Did It

- Applications New to Replication plan (Plan 6)
  - Applications new to replication
  - Identify:
    - Tables to be replicated
    - Anticipated Daily Volume
    - Failover requirements
    - MQ setup
    - Performance / Tuning

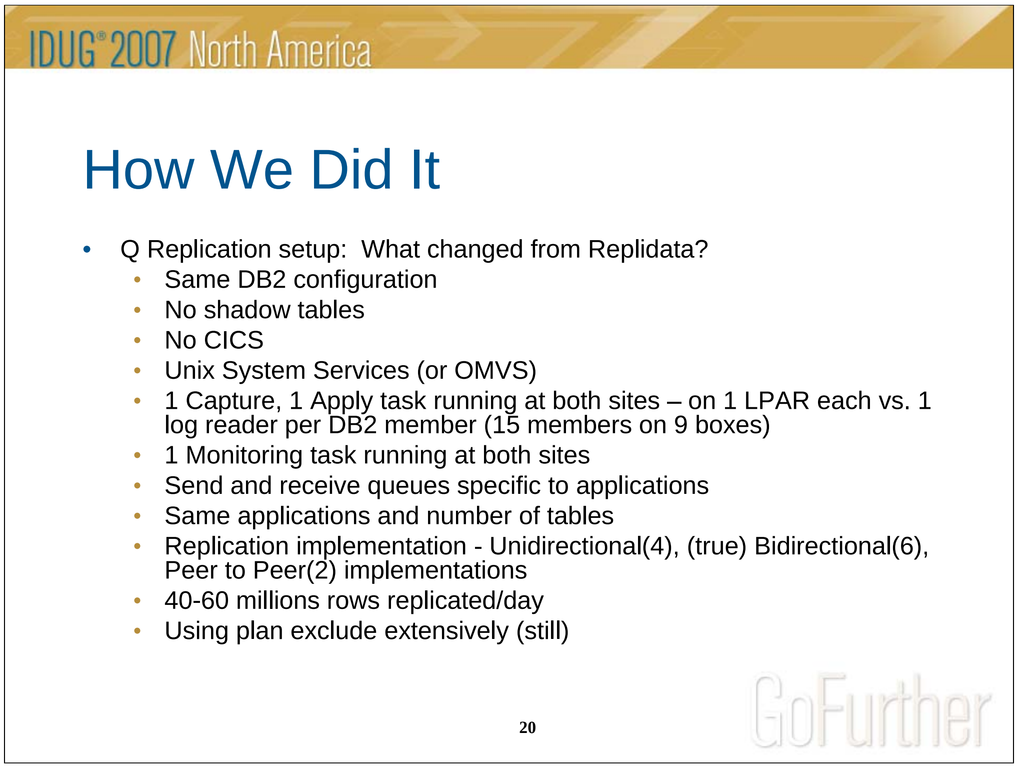
19

GoFurther

The purpose of this plan was to lay out the requirements for applications new to replication. Some requirements included the basics, like what tables to replicate, how many rows would be replicated, and any failover requirements. What do I mean by failover requirements? Requirements by the business need for continuing operation in the event that the source site becomes unavailable. Should processing stop, wait, or continue on the other site? This ultimately determines what type of replication (Unidirectional, Bidirectional, Peer to Peer) should be used.

MQ setup: should be same as other applications, but still needs to be verified.

Performance / Tuning: This would also include any capacity issues.


IDUG® 2007 North America

## How We Did It

- Q Replication setup: What changed from Replidata?
  - Same DB2 configuration
  - No shadow tables
  - No CICS
  - Unix System Services (or OMVS)
  - 1 Capture, 1 Apply task running at both sites – on 1 LPAR each vs. 1 log reader per DB2 member (15 members on 9 boxes)
  - 1 Monitoring task running at both sites
  - Send and receive queues specific to applications
  - Same applications and number of tables
  - Replication implementation - Unidirectional(4), (true) Bidirectional(6), Peer to Peer(2) implementations
  - 40-60 millions rows replicated/day
  - Using plan exclude extensively (still)

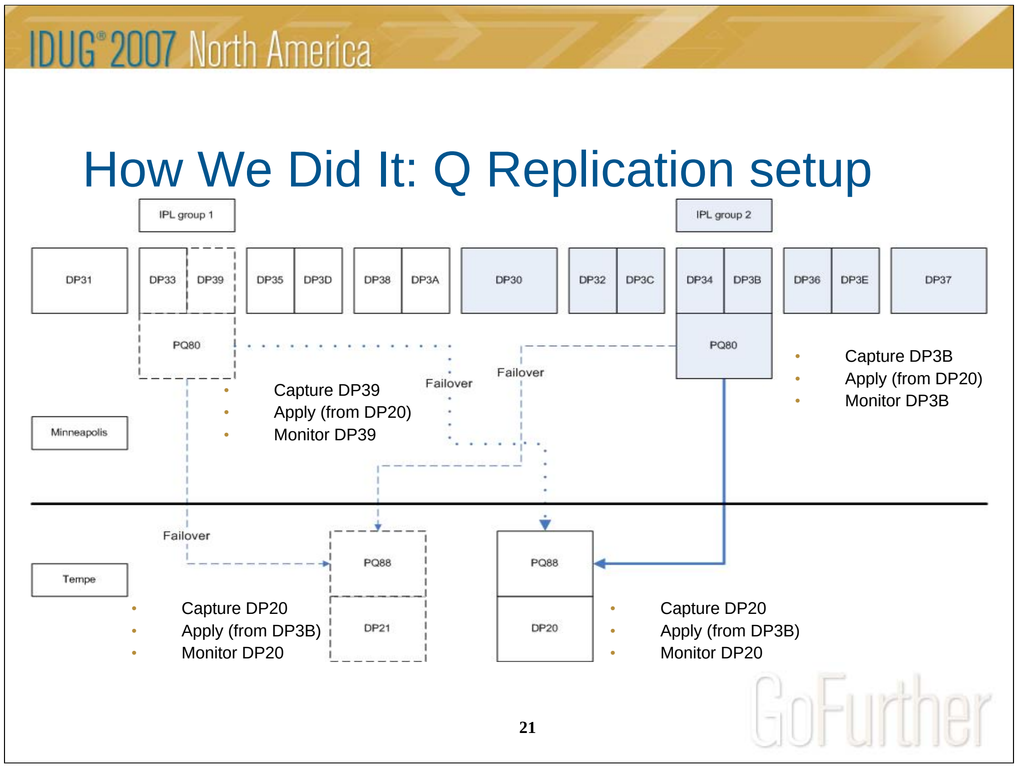
20

GoFurther

What changed? We are still replicating the same amount, but CICS is removed from the picture, and there are fewer log readers and queue managers involved.

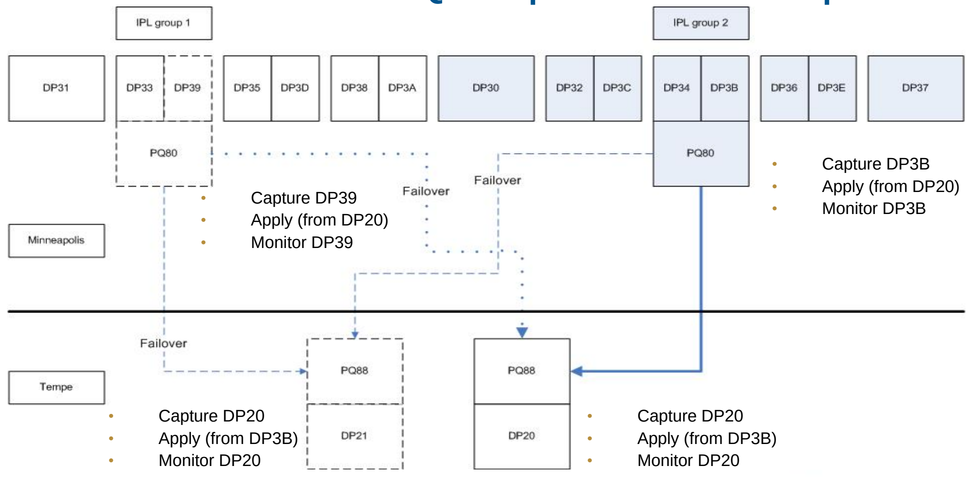
Some applications implemented more than one kind of replication for certain sets of tables.

No application table changes (Peer to peer replicated tables require application table changes, but as of this writing no applications had chosen this method).



IDUG® 2007 North America

## How We Did It: Q Replication setup



- Capture DP39
- Apply (from DP20)
- Monitor DP39

- Capture DP3B
- Apply (from DP20)
- Monitor DP3B

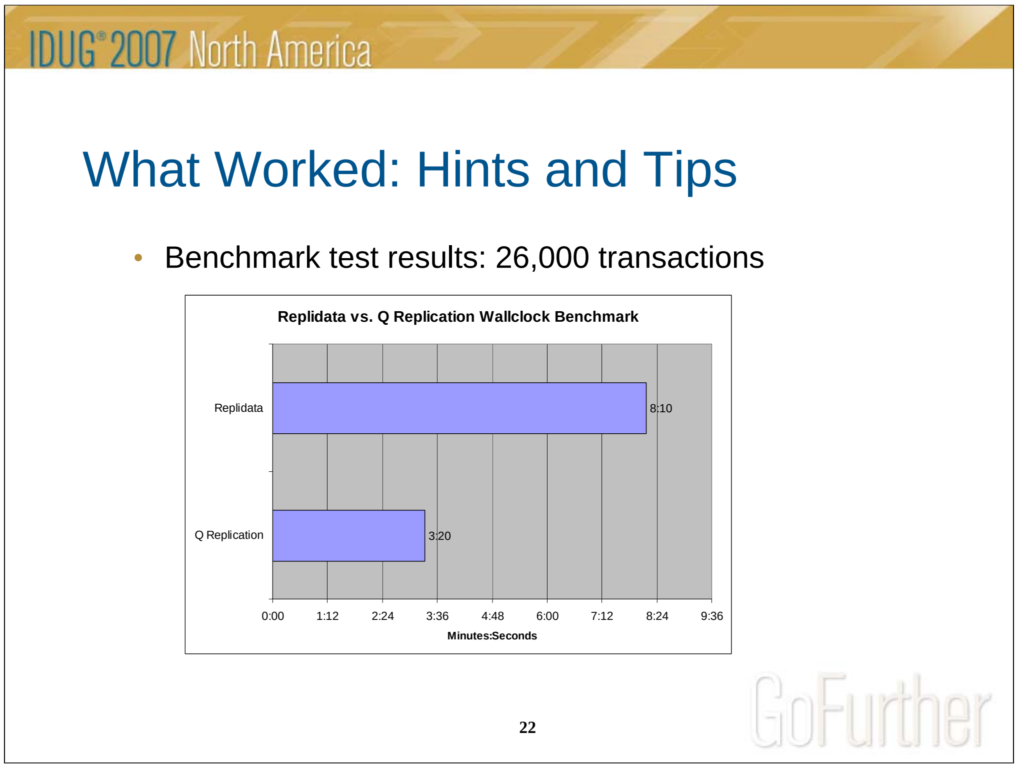
- Capture DP20
- Apply (from DP3B)
- Monitor DP20

- Capture DP20
- Apply (from DP3B)
- Monitor DP20

21

GoFurther

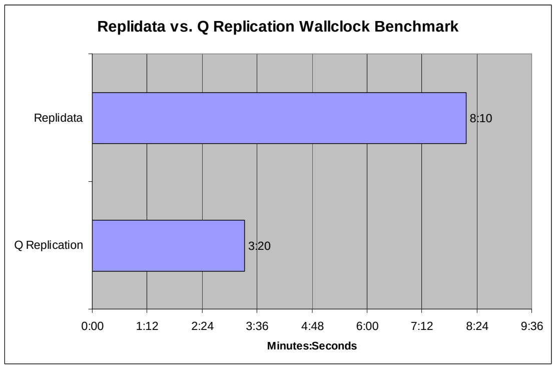
This shows our setup of Q Replication. As you can see, there is a Q Capture and a Q Apply started task running in each site, as well as a monitoring task. In the event of a failover, those jobs get moved to the failover LPAR, designated in the diagram by the dotted lines and saying "failover". During a scheduled outage (like an IPL) in Minneapolis, Q Replication would be stopped on DP3B and brought back up on DP39, after the MQ queue manager was restarted on that LPAR. After the primary LPAR was back up, the process would be reversed. The process is the same for Tempe.



IDUG® 2007 North America

## What Worked: Hints and Tips

- Benchmark test results: 26,000 transactions



22

GoFurther

Our benchmark test used one way replication.  We stopped the log readers for Replidata, and stopped the receive queues for the Q Replication test.

Our criteria was measured wall clock time for the apply process.

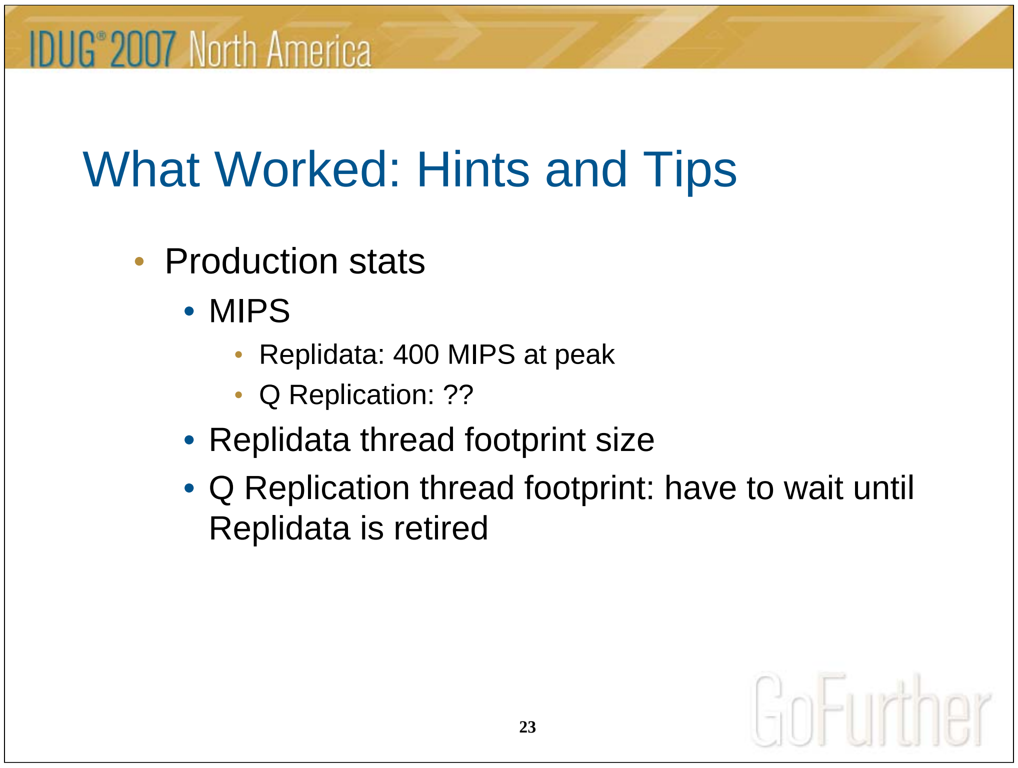
Note that the number of TPNS (testing tool) transactions (26,000) does not equal number of sql statements – over 314,000 for Q Replication, over 820,000 for Replidata.

CICS – 822,042 total calls

Q Rep – 593,362 calls

- ASNQAPAG
- ASNQP2PI
- ASNQBRWZ
- ASNQAHKT
- AVG trans applied (5 sec intervals): 661
- AVG rows applied (5 sec intervals): 4476 ~ 895 rows/sec

As this slide shows, with our basic criteria of wall clock time, Q Replication was noticeably faster than Replidata.



IDUG® 2007 North America

## What Worked: Hints and Tips

- Production stats
  - MIPS
    - Replidata: 400 MIPS at peak
    - Q Replication: ??
  - Replidata thread footprint size
  - Q Replication thread footprint: have to wait until Replidata is retired

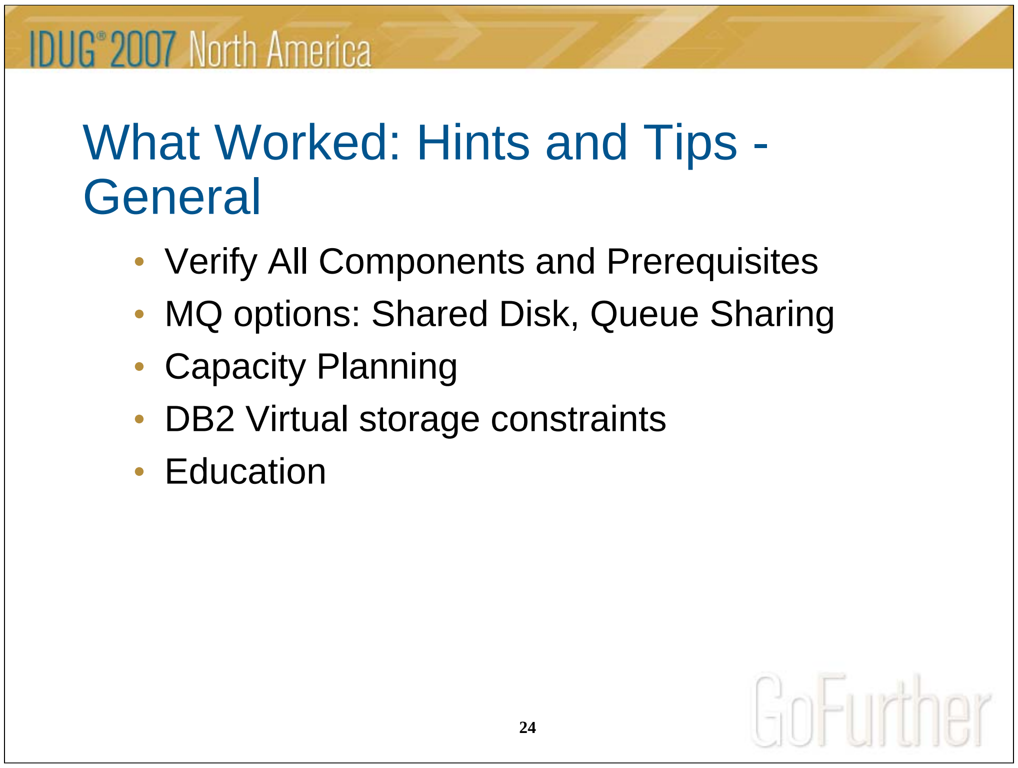
23

GoFurther

Again the Q Replication MIP usage won't be available until after this presentation has been finalized. Based on some initial research done in development, the numbers are a lot lower than Replidata, but we won't know for sure until it is up and running in production.

For the virtual storage conscious, the Replidata thread footprint size for our shop came in at just over 1 MB per thread.

Q Replication thread footprint size won't be able to be determined until Replidata has been removed.


IDUG® 2007 North America

## What Worked: Hints and Tips - General

- Verify All Components and Prerequisites
- MQ options: Shared Disk, Queue Sharing
- Capacity Planning
- DB2 Virtual storage constraints
- Education

24

GoFurther

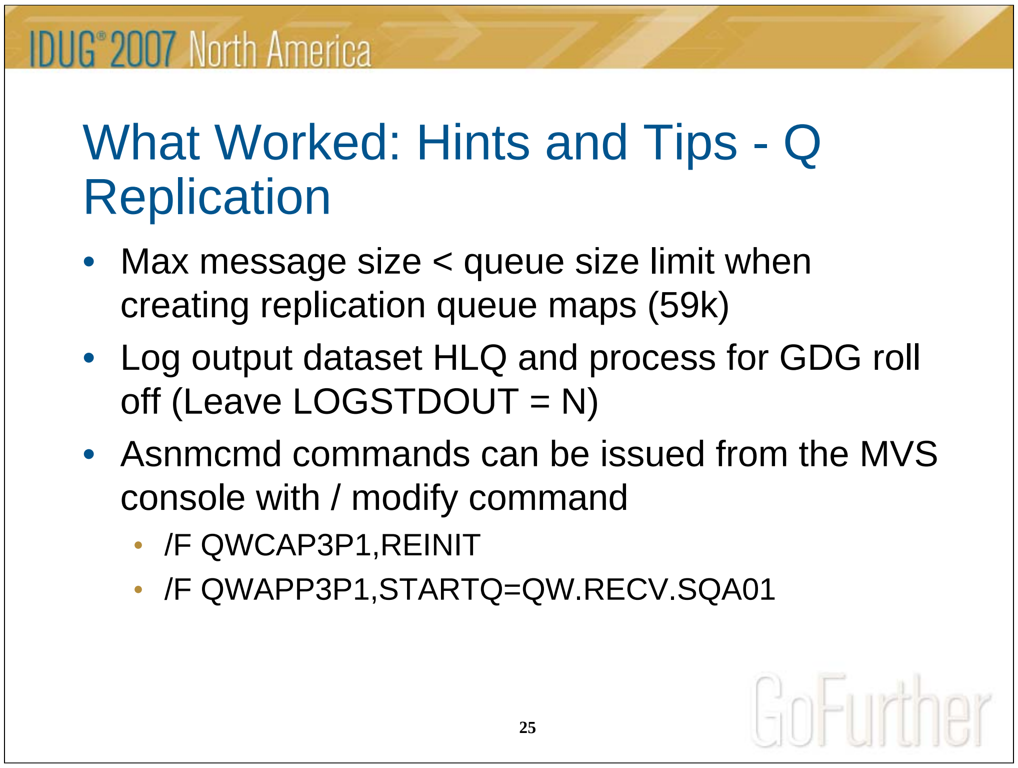
Verify all hardware and software components: may seem like a waste of time, but it’s easy to check and harder to fix once you’ve started down the path.

MQ options: This one is important. The option to use queue sharing vs. disk sharing should be carefully considered and discussed before continuing with the project. This is where having the necessary groups involved – capacity planning, storage, MQ, DB2, etc. – is important in the decision making process.

Capacity planning: Network, Coupling Facility (CF), disk storage, CPU are all factors to consider and research regardless of the MQ option chosen.

DB2 Virtual storage constraints: This was a big concern at our shop, given our issues with it for Version 8 of DB2. This was a factor in determining what DB2 member to use for the Q Capture and Q Apply (and Monitoring) started tasks.

Education: Attending the Q Replication class was very beneficial and highly recommended. Nothing beats hands on experience, but the class is useful for teaching the basics and especially the concepts of how it all works and fits together.


IDUG® 2007 North America

## What Worked: Hints and Tips - Q Replication

- Max message size < queue size limit when creating replication queue maps (59k)
- Log output dataset HLQ and process for GDG roll off (Leave LOGSTDOUT = N)
- Asnmcmd commands can be issued from the MVS console with / modify command
  - /F QWCAP3P1,REINIT
  - /F QWAPP3P1,STARTQ=QW.RECV.SQA01

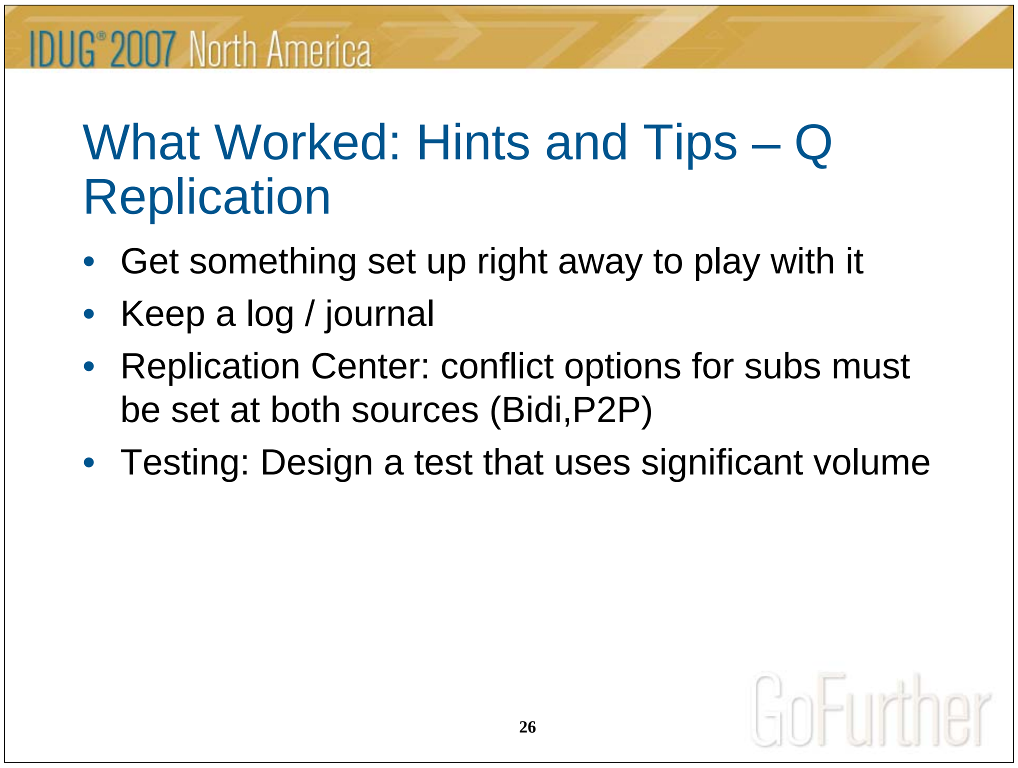
25

GoFurther

When creating replication queue maps, make sure the max message size is set to something less than the queue size limit.  Defaults to 63k, but didn't work for me until I used 59k.

Log output dataset HLQ and process for GDG roll off – in the started task proc, dummy out the log dataset, started task program, then backup to GDG when it comes down.  Good for keeping an archive of the log messages for historical research purposes.  (Thanks Donna and Jayanti.)

Asn* commands can be issued from the MVS console with /modify command – it's a good thing. The REINIT command is very useful when reinitializing tasks for new subs or changing monitoring alerts.


IDUG® 2007 North America

## What Worked: Hints and Tips – Q Replication

- Get something set up right away to play with it
- Keep a log / journal
- Replication Center: conflict options for subs must be set at both sources (Bidi,P2P)
- Testing: Design a test that uses significant volume

26

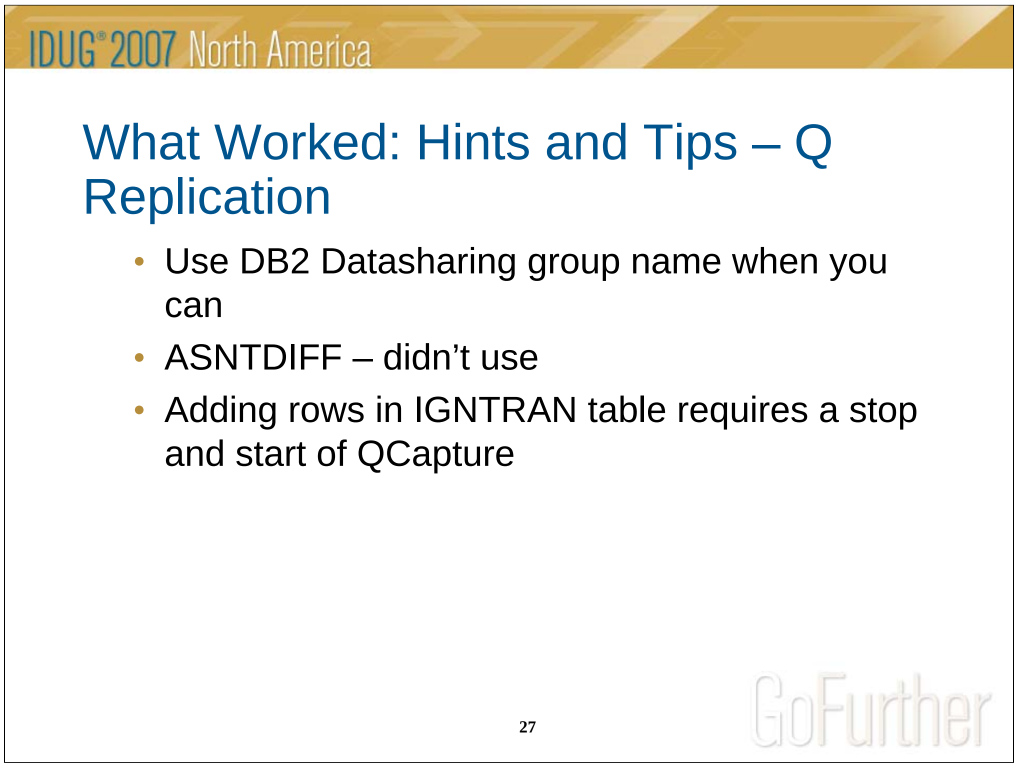
GoFurther

Get something set up right away to play with it: very helpful to ‘kick the tires’, especially during testing. Testing out the basics gave us a better understanding of the product, and in some cases, uncovered bugs (like in Replication Center).

I kept a Word doc with daily entries of hints and tips (for times like this), and tasks that I needed to do. It came in very handy when determining when something was done, and what hadn’t been done. The project plans provide good detail and a high level understanding of the project, but this document is more for the little details that don’t always get included in project plans (like questions to ask, setting up meetings, etc.).

Replication Center: first, use it. Second, the conflict option for subscriptions have to be updated for both subscriptions (when a subscription is created for 2 tables that have bidirectional replication, 2 subscriptions are actually created, one from the target to the source, and another for the source to the target.) Forgetting to update both could result in unpredictable results.

Testing: Design a test that uses significant volume to get accurate (and measurable) results. Manual testing could not create enough transactions in a timely manner to measure them before Replidata applied them.


IDUG® 2007 North America

## What Worked: Hints and Tips – Q Replication

- Use DB2 Datasharing group name when you can
- ASNTDIFF – didn't use
- Adding rows in IGNTRAN table requires a stop and start of QCapture

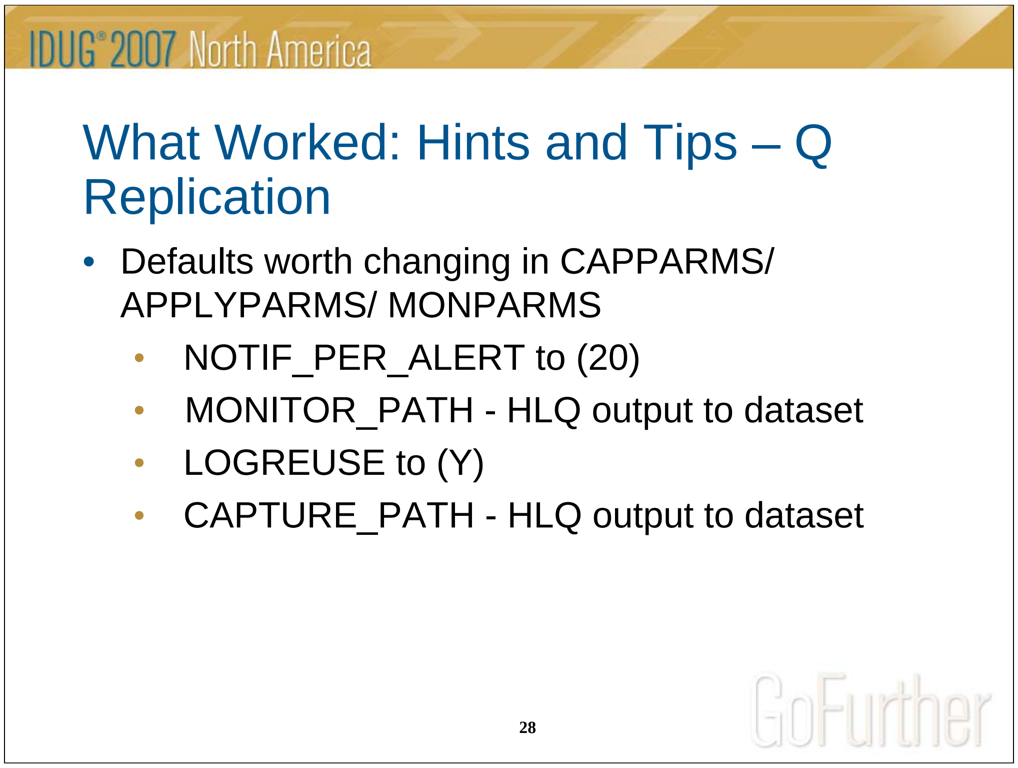
27

GoFurther

Use DB2 group name: Makes life much easier when you can use the group name, even though the manual says to use the member name.

ASNTDIFF: Tried to use, but needed create authority, big hassle. In our case we had existing applications converting, so what they had is what they continued to have with Q Replication. With Replidata we knew we had some inconsistency. This would have helped with our primary key issue (see later slide).

IGNTRAN table – try add your plans to ignore all at once as much as you can – requires stop and start of QCapture which may not be so easy to do in a production environment (except maybe during an IPL).


IDUG® 2007 North America

## What Worked: Hints and Tips – Q Replication

- Defaults worth changing in CAPPARMS/ APPLYPARMS/ MONPARMS
  - NOTIF_PER_ALERT to (20)
  - MONITOR_PATH - HLQ output to dataset
  - LOGREUSE to (Y)
  - CAPTURE_PATH - HLQ output to dataset

28

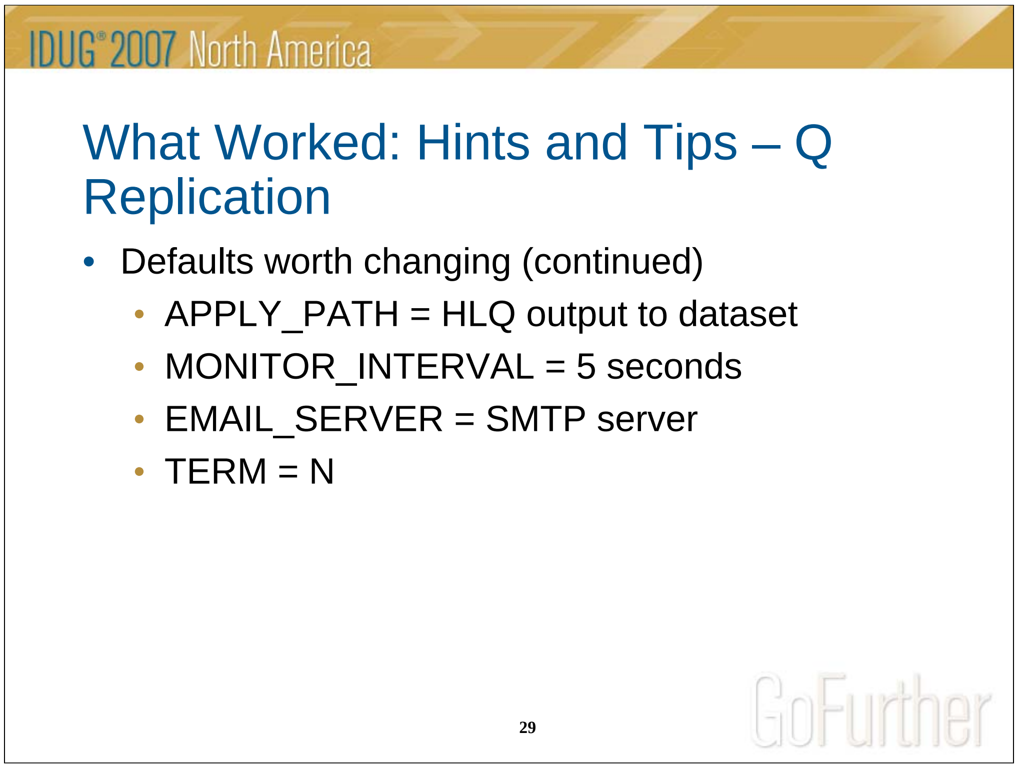
GoFurther

NOTIF_PER_ALERT - Default is 3. Inactive sub message email alerts wasn't showing all the subscriptions – solution was to change the NOTIF_PER_ALERT to 20.

MONITOR_PATH – Default is blank. Change to a high level qualifier for a dataset and the log messages will get written to it. Each time the task is brought down, you can use the log GDG rolloff process mentioned earlier to keep multiple versions of it and start fresh.

LOGREUSE – Default is N. Setting this to Y appends log messages to the output dataset you created using the CAPTURE_PATH parm. Leave LOGSTDOUT as 'N', especially if you're using the log rolloff to GDG mentioned earlier.

CAPTURE_PATH – Default is blank. Change to a high level qualifier for a dataset and the log messages will get written to it. Each time the task is brought down, you can use the log GDG rolloff process mentioned earlier to keep multiple versions of it and start fresh.


IDUG® 2007 North America

## What Worked: Hints and Tips – Q Replication

- Defaults worth changing (continued)
  - APPLY_PATH = HLQ output to dataset
  - MONITOR_INTERVAL = 5 seconds
  - EMAIL_SERVER = SMTP server
  - TERM = N

29

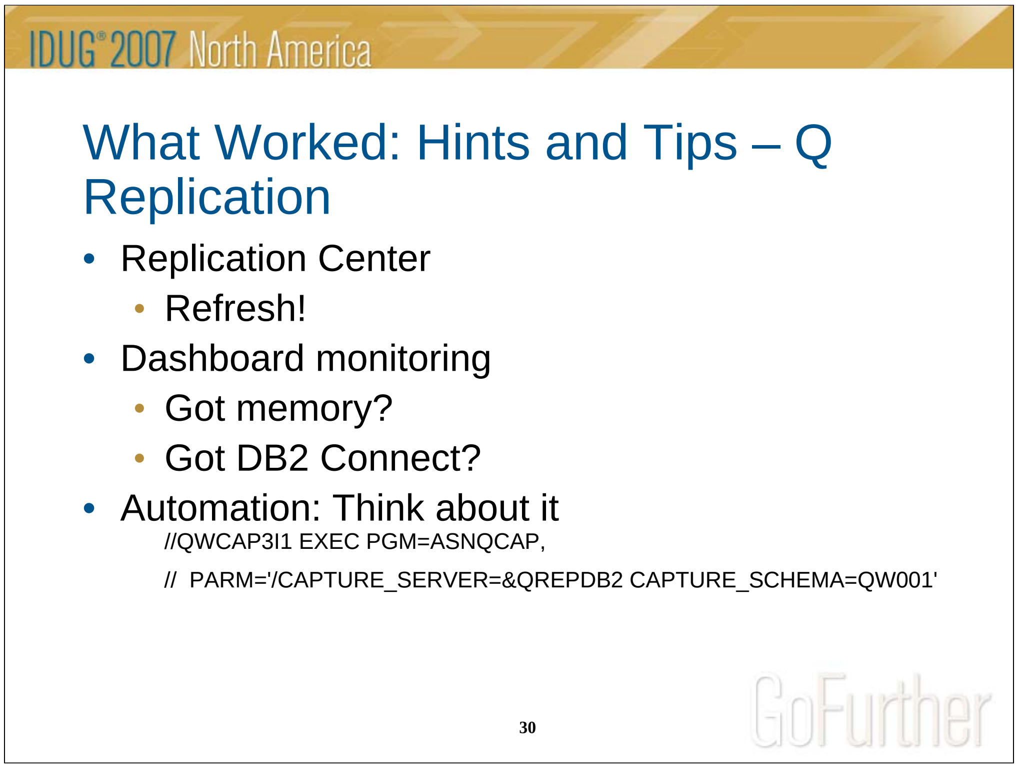
GoFurther

APPLY_PATH - Default is blank. Change to a high level qualifier for a dataset and the log messages will get written to it. Each time the task is brought down, you can use the log GDG rolloff process mentioned earlier to keep multiple versions of it and start fresh.

MONITOR_INTERVAL – Default is 300 seconds. Change value to 5 seconds to get more detailed stats. Dashboard recommendation is also 5 second intervals.

EMAIL_SERVER – Default is blank. Change to the SMTP server used in your shop to send out email alerts when certain alert conditions are met. Very helpful during testing too.

TERM – Default is Y. Change this to N so that Q Replication will remain up even if DB2 comes down.



## What Worked: Hints and Tips – Q Replication

- Replication Center
  - Refresh!
- Dashboard monitoring
  - Got memory?
  - Got DB2 Connect?
- Automation: Think about it

```
//QWCAP3I1 EXEC PGM=ASNQCAP,
//  PARM='/CAPTURE_SERVER=&QREPDB2 CAPTURE_SCHEMA=QW001'
```

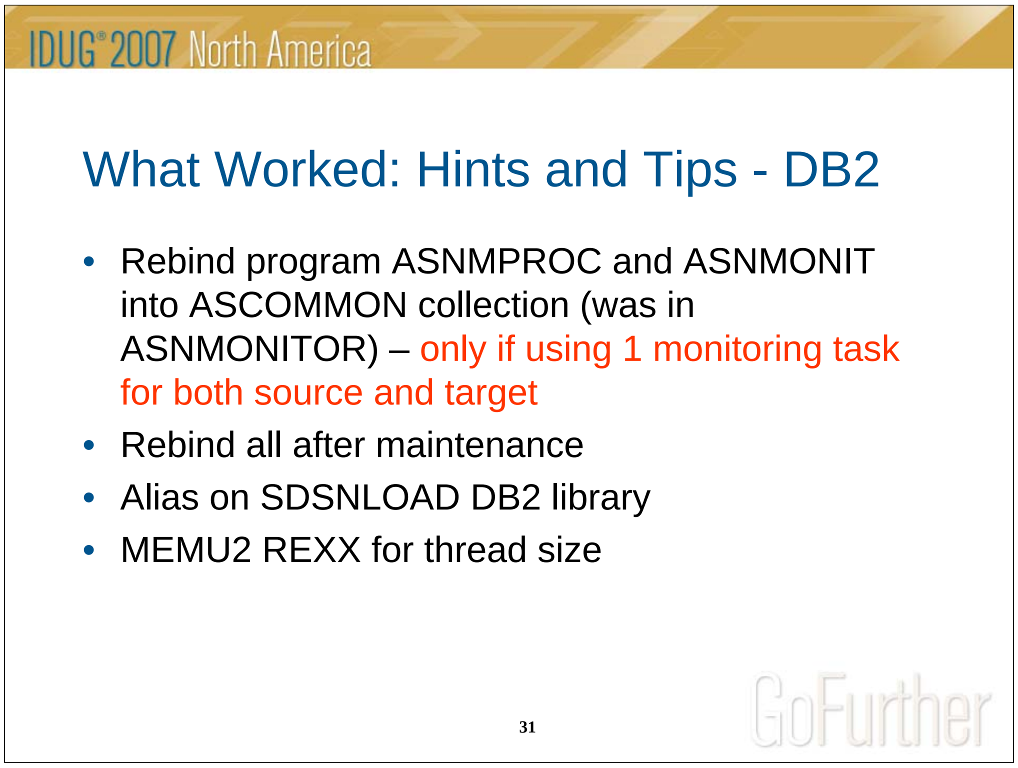
Replication Center: Makes administration much easier. Didn't really use ASNCLP (command line processor based subscription generator) to generate scripts. Don't forget to refresh your screen often!

Dashboard monitoring: Good visual representation of performance, activity, and status. Can be a memory hog on a PC. Have at least 1GB of memory installed. 2GB is better. 4GB and you can run anything you want!

This probably should have been mentioned earlier, but in order to connect to the mainframe with Replication Center and the Dashboard monitor, you'll need something like DB2 Connect to do this. I use the DB2 Connect Enterprise Edition that connects via our DB2 Connect gateway zLinux server.

Automation: We have multiple DB2 members on the same LPAR in production, but because of virtual storage concerns, we only wanted Q Rep to run on certain DB2 LPARs. So we can't specify group name because it might start on the wrong member of that LPAR. The solution was to make that parameter in the JCL a symbolic and have automation start the task with a certain value if it starts on that lpar, and another member when the started tasks get brought up on the failover LPAR during IPLs or abends/crashes.

And last but not least – Don't forget those rows in CAPPARMS and APPLYPARMS! Q Replication won't start without them.


IDUG® 2007 North America

## What Worked: Hints and Tips - DB2

- Rebind program ASNMPROC and ASNMONIT into ASCOMMON collection (was in ASNMONITOR) – only if using 1 monitoring task for both source and target
- Rebind all after maintenance
- Alias on SDSNLOAD DB2 library
- MEMU2 REXX for thread size

31

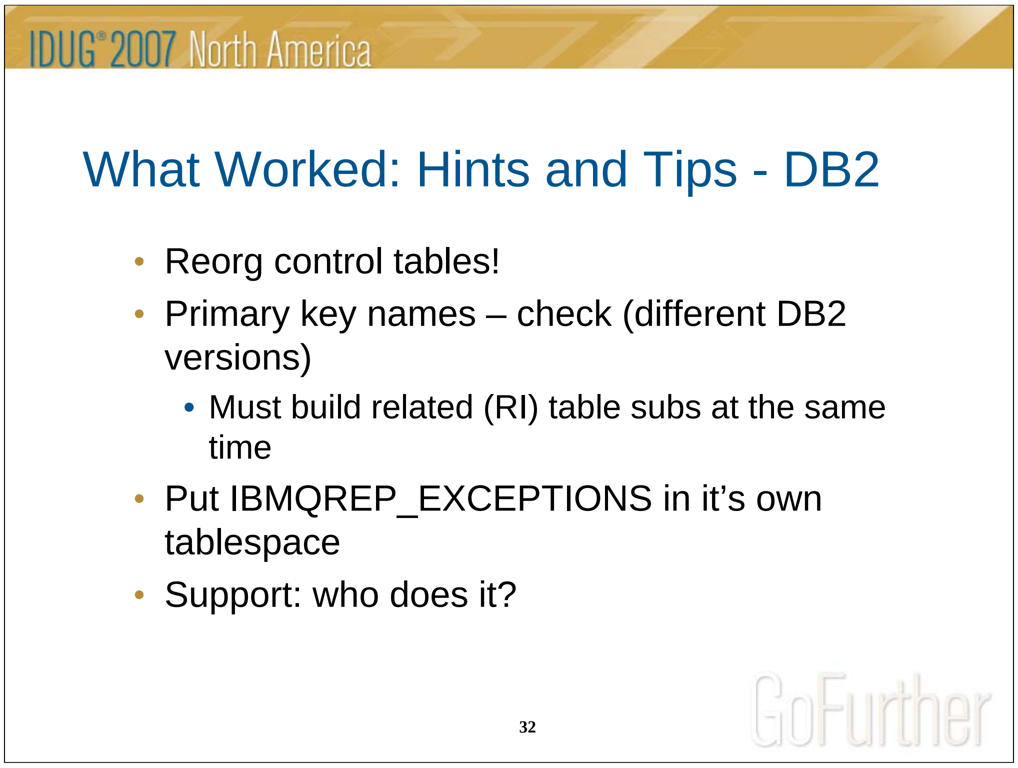
GoFurther

If you're only going to use one monitoring task for both the source and the target, you'll need to rebind the program ASNMPROC and ASNMONIT into the ASNCOMMON collection – didn't work otherwise when trying to monitor the other site.

Rebind all after maintenance: Bring down capture/apply/monitoring started tasks before doing so. Binding picks up the new maintenance and any new access paths with updated statistics. It's also a good idea to image copy the control tables, and also update the statistics (tables and indexes) on a regular basis.

Alias on SDSNLOAD DB2 library: Allows JCL to run on any LPAR to access DB2. We ran across this while preparing for version 8.

If you're concerned about DB2 virtual storage and the thread size for Q Replication and the number of them, consider running MEMU2 REXX JCL. Link to the REXX and JCL and instructions in the References section. It tells you the size of each thread in MB, and theoretically how many threads you could have in the system based on it.


IDUG® 2007 North America

## What Worked: Hints and Tips - DB2

- Reorg control tables!
- Primary key names – check (different DB2 versions)
  - Must build related (RI) table subs at the same time
- Put IBMQREP_EXCEPTIONS in it’s own tablespace
- Support: who does it?

32

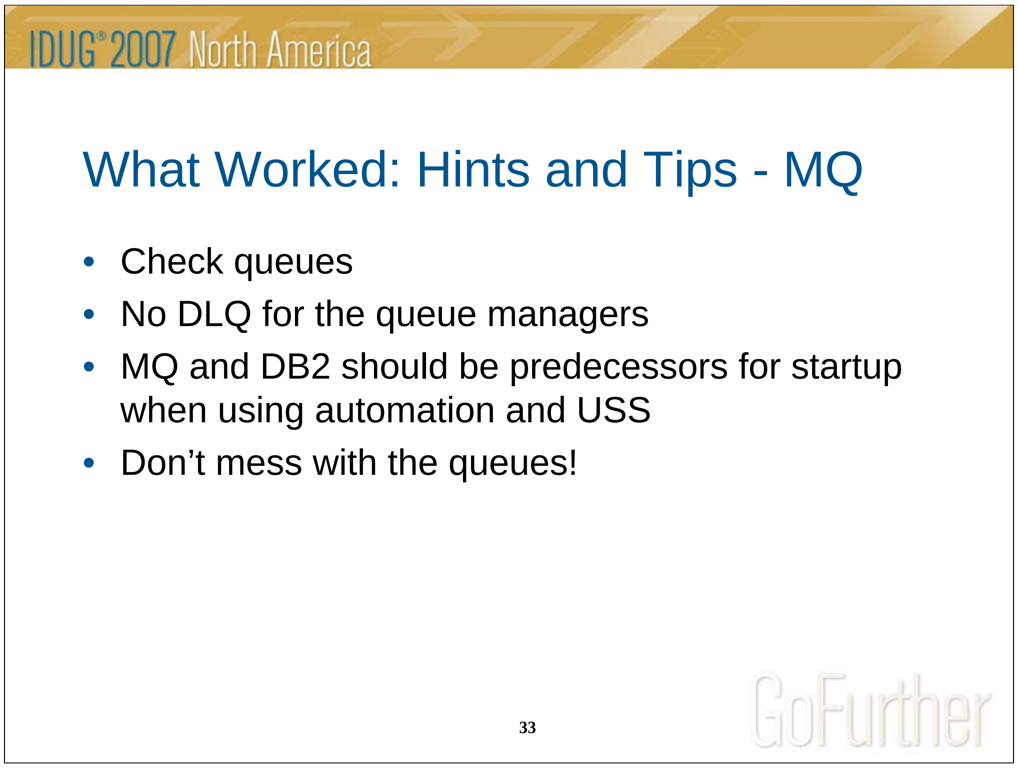
GoFurther

Reorg control tables:  This is very important – access paths can be terrible without it.  Scheduled reorgs are an option, but certainly do them as needed, especially after Q Replication has been running for a few days.

Primary key names – check the primary keys of the tables, otherwise you will have trouble creating subscriptions.  This happened to us because tables in the other subsystem had been created on a different version of DB2 than the source subsystem.  Same with foreign keys and RI between tables; check out your environments beforehand so it’s not a surprise.  ASNTDIFF would have helped in this regard.

IBMQREP_EXCEPTIONS in it’s own tablespace: recommended by IBM

Support – Determine who will support Q Replication in your shop – will it be the DBA group, systems group, or someone else?  Having clearly defined roles and responsibilities is essential.


IDUG® 2007 North America

## What Worked: Hints and Tips - MQ

- Check queues
- No DLQ for the queue managers
- MQ and DB2 should be predecessors for startup when using automation and USS
- Don’t mess with the queues!

33

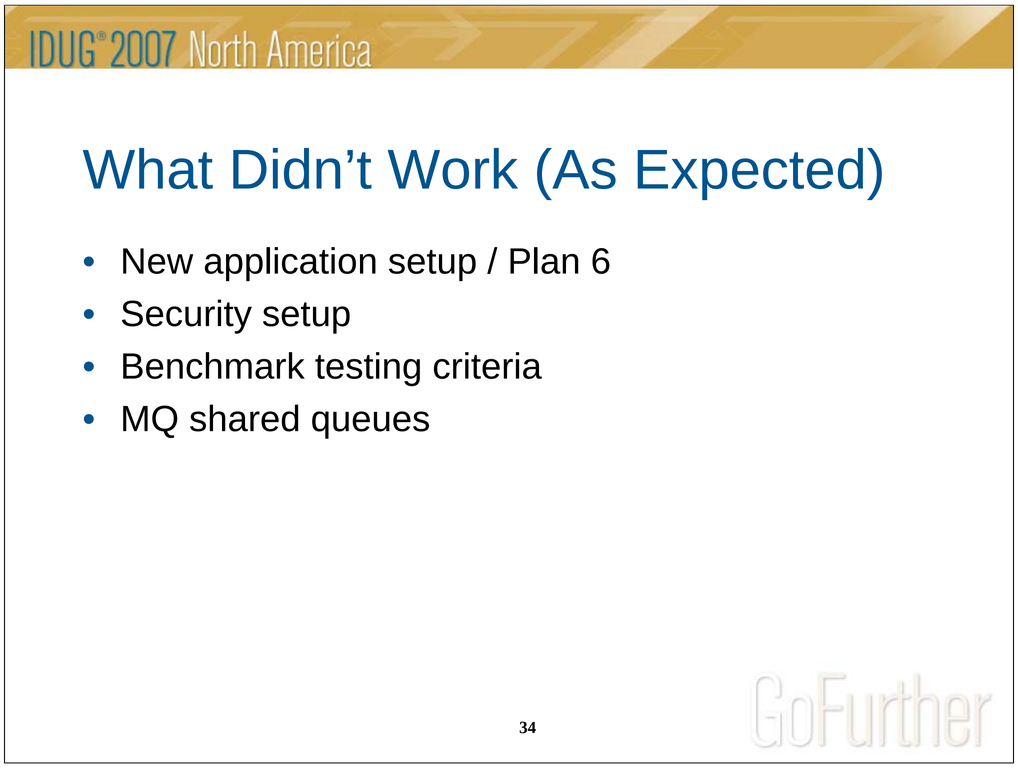
GoFurther

Check queues: Get MQ authority to look at your queues

DLQ (dead letter queue): Don’t have one for the Q Replication queue managers – keeps things cleaner.

MQ and DB2 as predecessors: This refers to automation. When setting it up, ensure MQ and DB2 tasks are up before Q Replication comes up. Q Replication doesn’t need DB2 to be up, but it’s not a bad idea. Definitely MQ though.

Unless you want some serious issues to deal with, don’t try to edit/delete manipulate the messages in the send and receive queues. Finding out the hard way was not my idea of a good time. Even if you have to restart the Q Capture job, do NOT empty the queues! Then you run into issues with gaps in the message sequence and the DONEMSG table.


IDUG® 2007 North America

## What Didn't Work (As Expected)

- New application setup / Plan 6
- Security setup
- Benchmark testing criteria
- MQ shared queues

34

GoFurther

New application setup/Plan 6: Not that it didn't work, we just didn't use it. Initially had an application that was interested, but fell through. Still a good idea though.

Security plan setup: Security plan was approved, just took longer than expected. What didn't work was thinking that it would be completed sooner!

Comparing Replidata to Q Replication was difficult because:

-Replidata has CICS component

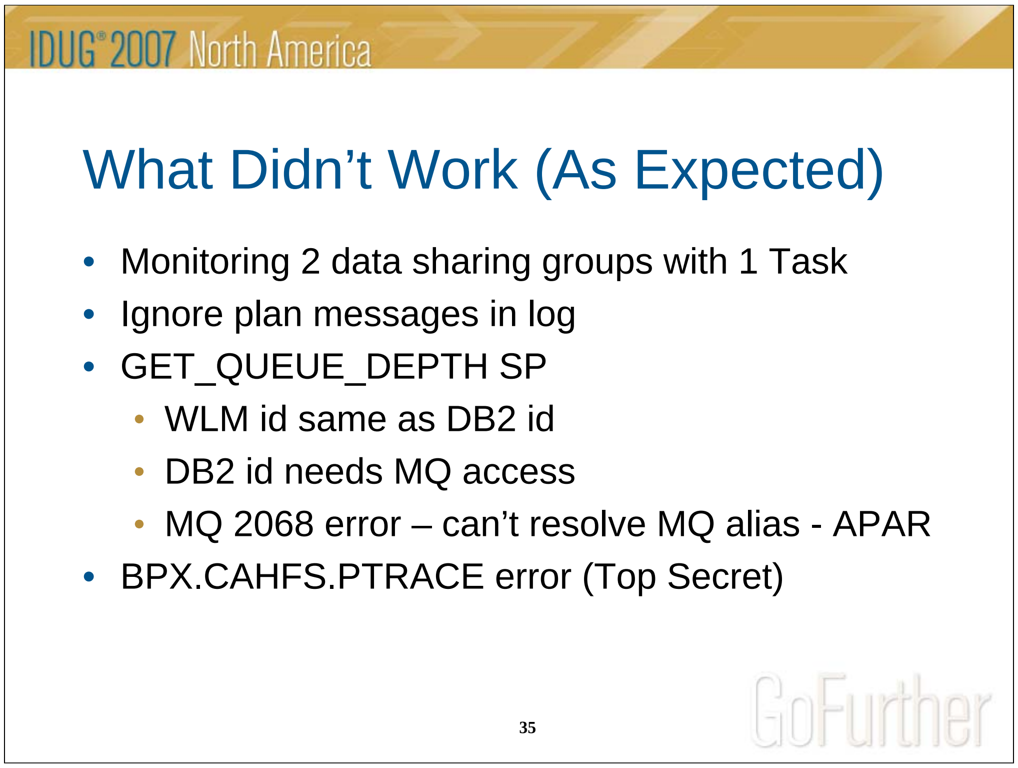
-Replidata keeps no performance statistics

-Volume in test environment was too low

-The solution? Wallclock timing and DB2 program CPU usage

-TPNS used to simulate lots of transactions to get significant volume to process (TPNS is a testing tool simulator – for my purposes it was a black box that someone else set up. TPNS = teleprocessing network simulator.)

MQ shared queues: Shared queues didn't work for us at this point because of the resource demands to the coupling facility. It will be revisited at some later point.


IDUG® 2007 North America

## What Didn't Work (As Expected)

- Monitoring 2 data sharing groups with 1 Task
- Ignore plan messages in log
- GET_QUEUE_DEPTH SP
  - WLM id same as DB2 id
  - DB2 id needs MQ access
  - MQ 2068 error – can't resolve MQ alias - APAR
- BPX.CAHFS.PTRACE error (Top Secret)

35

GoFurther

Monitoring 2 data sharing groups with 1 Task is possible, but the Capture/Apply task status cannot be monitored on the target site. The manuals state that this is limitation in z/OS for Q Replication. What we did was start a monitoring task for each data sharing group, which fixed the issue. Being notified whether or not the started task is up or down is important if you care about availability!

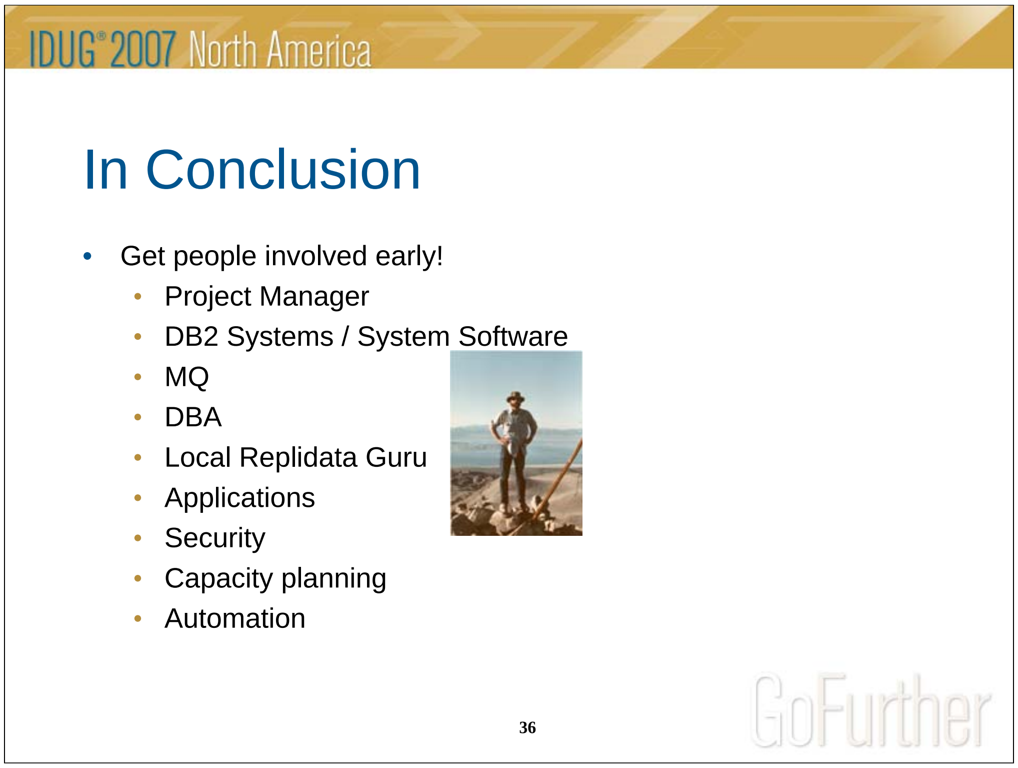
The ignore plan feature (something implemented with APAR PK27949) allows you to NOT replicate certain data based on plan id, authid, or auth token. When the criteria is met, Q Replication notes it in the IGNTRANTRC table. However it was also logging it in the started task, causing the log to fill up with unnecessary information. An additional APAR fixed this problem so that the messages went only to the IGNTRANTRC table.

GET QUEUE DEPTH SP: This stored procedure is used in the monitoring and alerts started task as an optional condition to check the queue depth. However it was abending with the error above. The problem turned out to be with queues that have aliases defined to them. IBM has identified this as a problem and will be putting out an APAR to fix it.

We received a READ ACCESS DENIED error on this dataset - BPX.CAHFS.PTRACE - during shutdown of started tasks. Solution was to grant this access -

```
TSS ADD(QWT#DVP) IBMFAC(BPX.CAHFS.PTRACE)       TSS
PER(QWT#DVP) IBMFAC(BPX.CAHFS.PTRACE) ACC(READ)
```

If you don't use Top Secret you may not have this error.



IDUG® 2007 North America

## In Conclusion

- Get people involved early!
  - Project Manager
  - DB2 Systems / System Software
  - MQ
  - DBA
  - Local Replidata Guru
  - Applications
  - Security
  - Capacity planning
  - Automation

36

GoFurther

Project Manager – invaluable.  Kept project moving, document updates and repository website.

DB2 Systems / System Software: DB2 systems did our install, but whoever does the software installation

MQ: POQ (plain old queues), disk shared, queue sharing are all reasons to have someone from MQ join your project immediately.

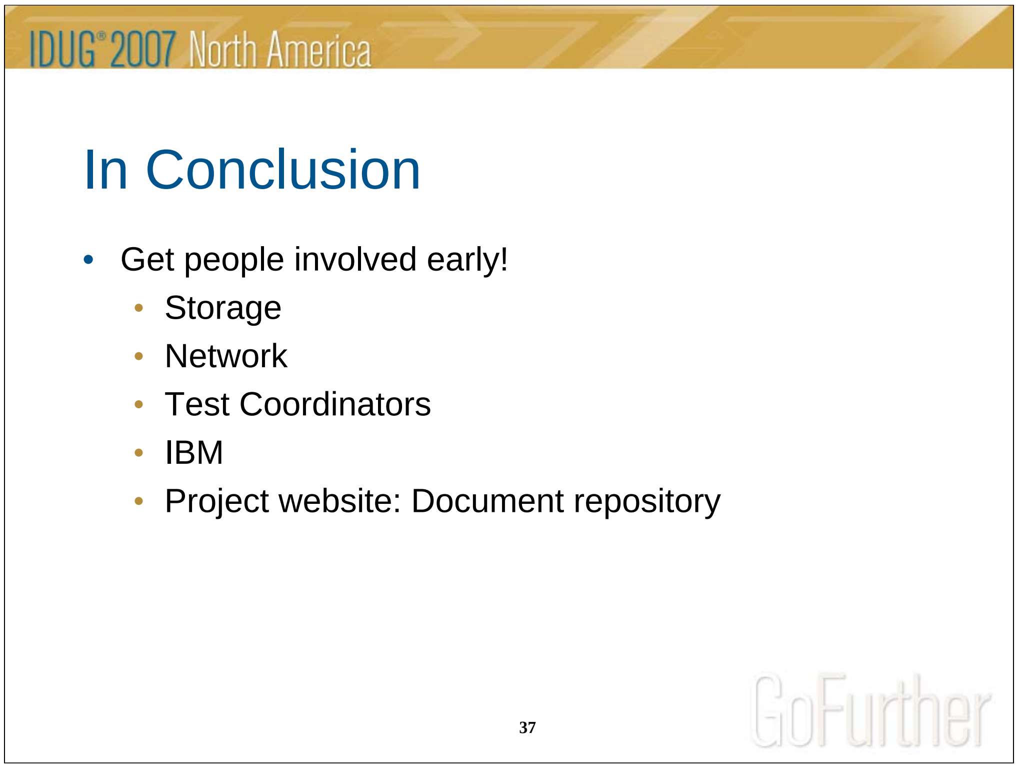
Local Replidata guru: He climbs mountains too.

Applications – get managers involved, checkouts will be needed, coordination with testing group also to schedule migrations.

Security – the dreaded security plan.  Contact security early on to see what is required for your project, especially if it's considered a new application.

Capacity planning – very essential in determining existing usage and what to expect for future usage with Q Replication.

Automation – planning for failovers and abend situations cannot be done without the help of automation.  They helped to solve our multiple DB2 member problem.


IDUG® 2007 North America

## In Conclusion

- Get people involved early!
  - Storage
  - Network
  - Test Coordinators
  - IBM
  - Project website: Document repository

37

GoFurther

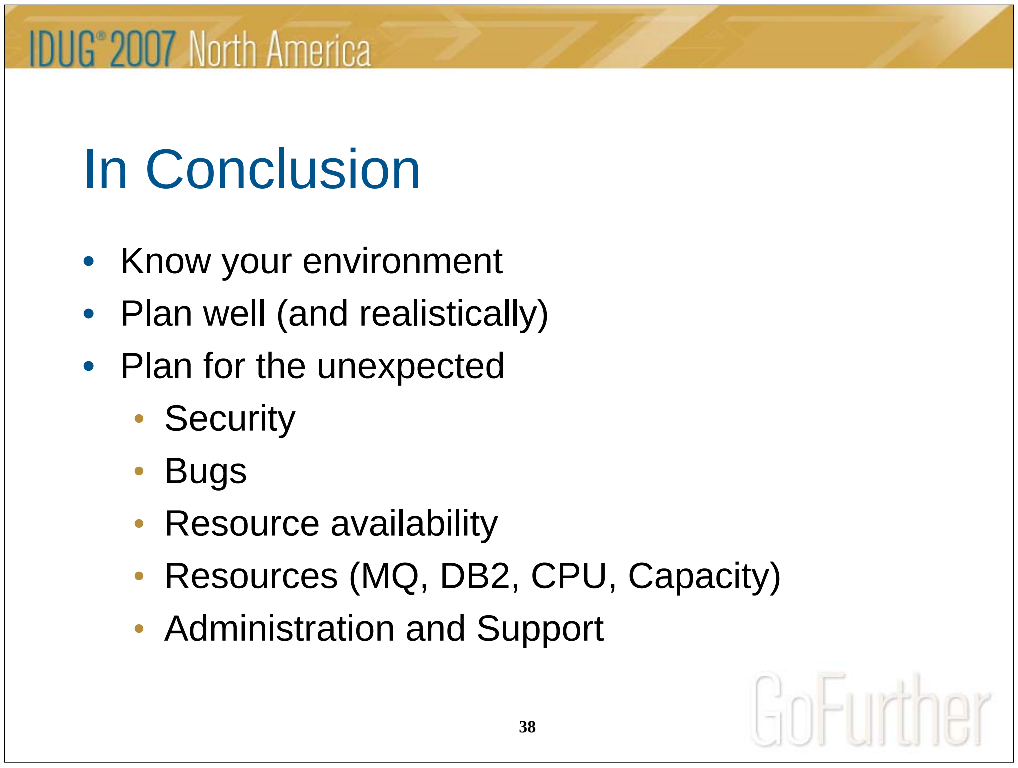
Storage – you're gonna need some space for all this stuff, right? Better get them involved too.

Network – TCP/IP stacks, failovers, this is the stuff that the Network people live for (or at least are tasked with).

Testing Coordinators – important to let everyone know when migrations will occur and if your timeline works with the enterprise testing area.

IBM – Having IBM assistance, whether it was via our advocate or from the development area or from answers by way of ETRs and PMRs, was critical to our success.

Project website: Having a document repository where everyone had access to the documents prevented a 'single point of failure' where only 1 person had control of the documents.



IDUG® 2007 North America

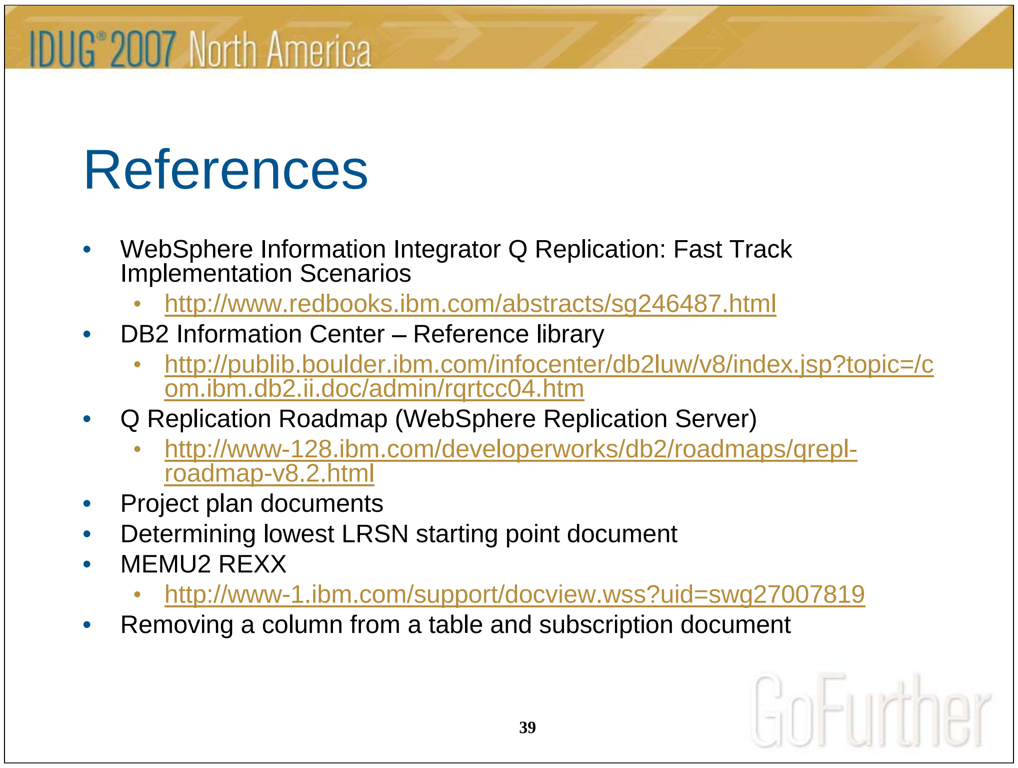
## In Conclusion

- Know your environment
- Plan well (and realistically)
- Plan for the unexpected
  - Security
  - Bugs
  - Resource availability
  - Resources (MQ, DB2, CPU, Capacity)
  - Administration and Support

38

GoFurther

This pretty much sums it up – get to know your environment, the people, and plan accordingly. Also plan for the unexpected, and be realistic about goals and timelines. You too can migrate to Q Replication in 8 years or less!


IDUG® 2007 North America

## References

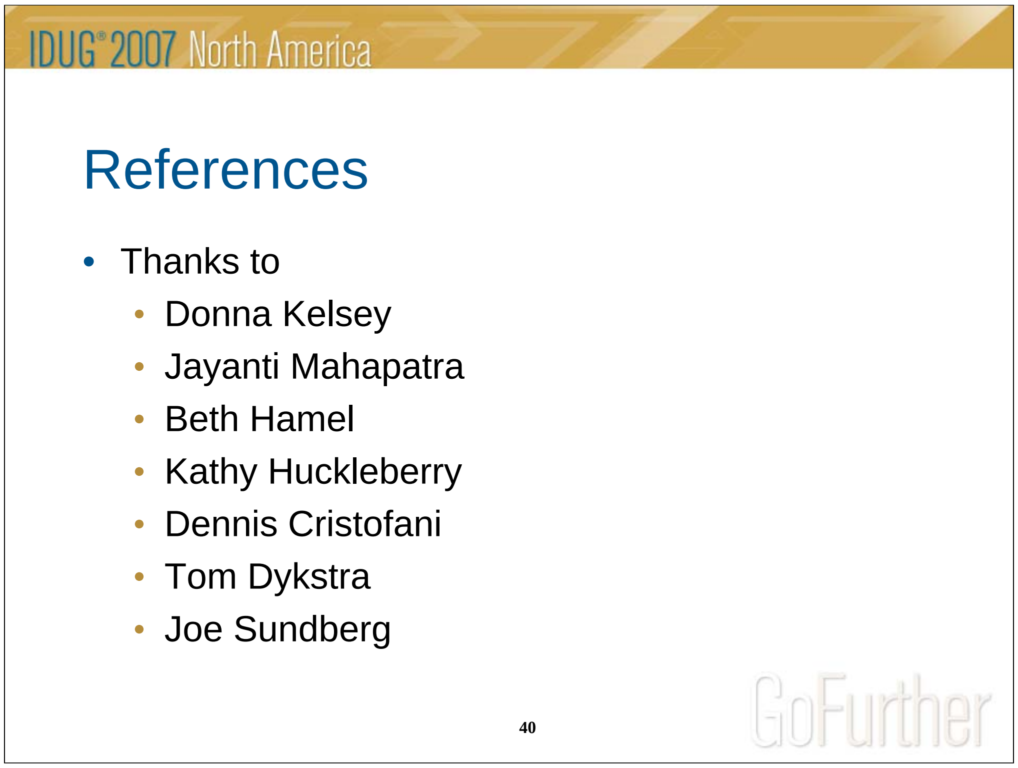
- WebSphere Information Integrator Q Replication: Fast Track Implementation Scenarios
  - http://www.redbooks.ibm.com/abstracts/sg246487.html
- DB2 Information Center – Reference library
  - http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.ii.doc/admin/rqrtcc04.htm
- Q Replication Roadmap (WebSphere Replication Server)
  - http://www-128.ibm.com/developerworks/db2/roadmaps/qrepl-roadmap-v8.2.html
- Project plan documents
- Determining lowest LRSN starting point document
- MEMU2 REXX
  - http://www-1.ibm.com/support/docview.wss?uid=swg27007819
- Removing a column from a table and subscription document

39

GoFurther

Project plan documents, determining lowest LRSN starting point, and removing a column from a table and subscription document available by request.

Anyone interested in seeing production throughput statistics can contact me directly.



## References

- Thanks to
  - Donna Kelsey
  - Jayanti Mahapatra
  - Beth Hamel
  - Kathy Huckleberry
  - Dennis Cristofani
  - Tom Dykstra
  - Joe Sundberg

Session: H13

## Here to There (and Back): Converting From Replidata to Q Replication in 8 Years or Less

Jeff Kram

Wells Fargo

jeff.kram@wellsfargo.com

GoFurther