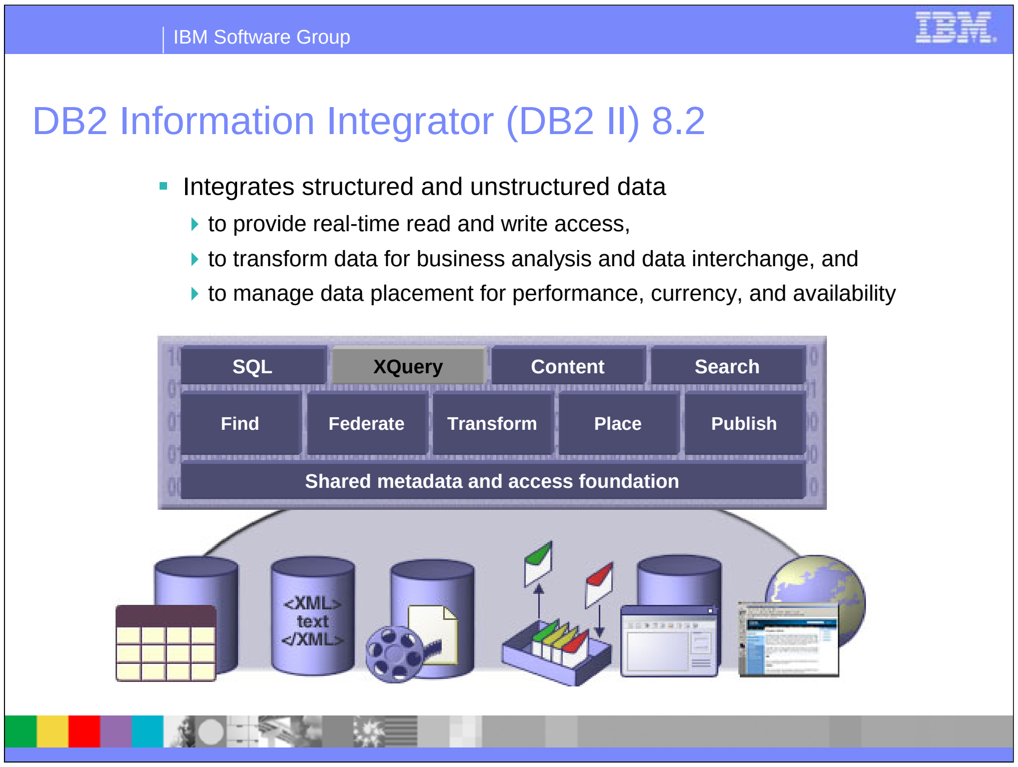
The DB2 Information Integrator Story - with a focus on Q Replication and Event Publishing

Beth Hamel
DB2 Replication Product Architect
hameleb@us.ibm.com

ON DEMAND BUSINESS™

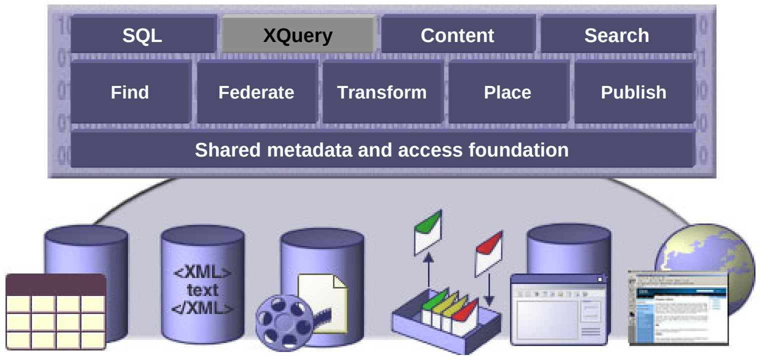
Agenda Topics

- DB2 Information Integrator in a z/OS world (and others!)
- Q Replication Features
- Publishing Data to MQ in an XML format
- Examples of Replication and Publishing Applications
- Production Results from a Q Replication Customer


IBM Software Group

DB2 Information Integrator (DB2 II) 8.2

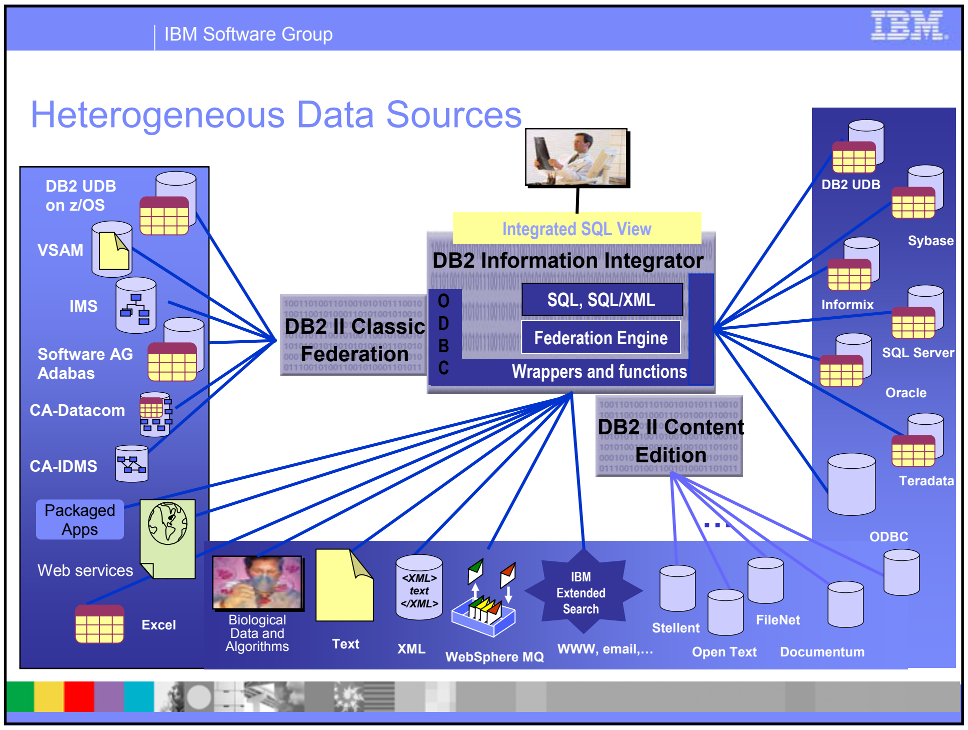
- Integrates structured and unstructured data
  - to provide real-time read and write access,
  - to transform data for business analysis and data interchange, and
  - to manage data placement for performance, currency, and availability



Businesses today must optimize every aspect of their business processes, to reduce costs, to better serve their customers, to react more quickly to changing business conditions. Accomplishing these goals requires horizontal integration of business processes. But business processes rely on IT systems, and under the covers, IT shops are dealing with increasing complexity. They have multiple hardware platforms, several different databases, other data in files or proprietary formats, and many applications to support. Streamlining and integrating business processes requires the integration of these various disparate systems. In addition, the company must be able to integrate information from its suppliers, and provide for the needs of its customers – all adding further diversity of systems and data formats. DB2 Information Integrator provides a means of bridging this diversity, providing an information infrastructure that makes integrating applications and business processes easier, while allowing the business to continue to leverage the existing diverse systems.

DB2 Information Integrator provides a platform allowing the integration of data of all types, from all types of information sources. In our vision, integration relies on a common foundation layer of metadata and data access capabilities. Multiple means of integrating data are supported, including the ability to find (indexed search), federate (real-time query of diverse data), and transform information into the needed formats. In addition, data may be placed (copied or moved) or changes to data may be captured and published to better support integration with applications and workflow processes. Access to the integrated data can, in our vision, be through any of a number of different programming interfaces.

In our very first release of DB2 Information Integrator we supported SQL access. In Version 8.2, we have made big steps towards our vision, adding two new access paradigms (content-style API and simple keyword search), two new integration disciplines (find and publish), and major enhancements in the power and performance of our federation and data placement capabilities. In a future release we will add XQuery support.

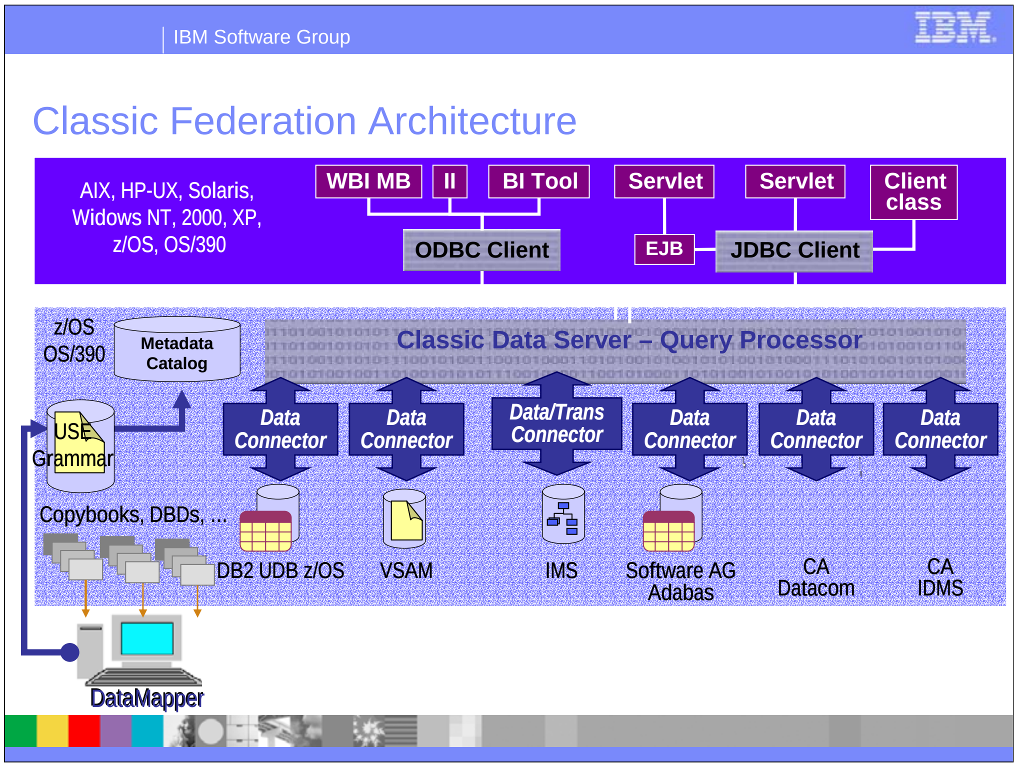


IBM Software Group

Heterogeneous Data Sources

This chart shows the sources that can be accessed by DB2 Information Integrator 8.2. They include all the common relational sources, and a broad range of other sources. All sources are available through a single integrated SQL view, and data from any source can be joined with data from any other in a single query. Note that many legacy sources are available through our Classic product, which runs on the mainframe, and interoperates with our LUW product.

- Transparency ->Off load the work from the developer since they work with a single virtual source with a common standard query language (SQL)
- Speed up the development
- Distributed queries over open platform systems and mainframe data



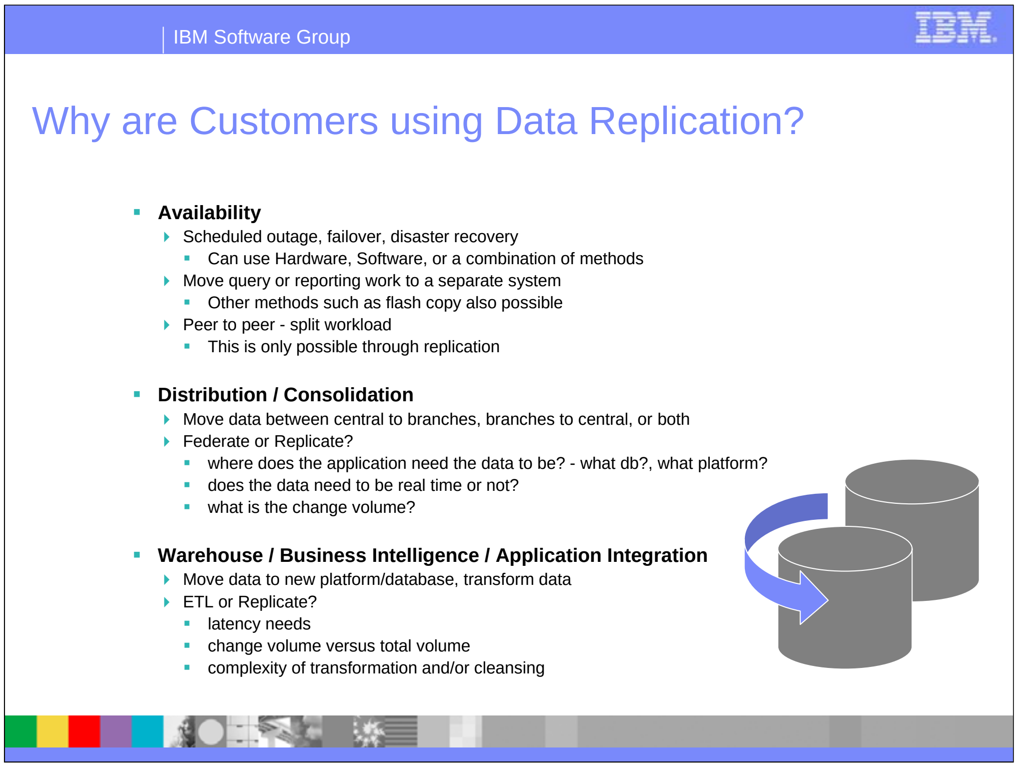
There are several major components in the Classic Architecture:

- Classic Data Server
- DataMapper and Metadata utilities
- Connectors to Datasources
- JDBC, ODBC, and CLI drivers

The Classic Data Server is a z/OS RDBMS whose data is stored in other database management products, including IMS, VSAM, CA's IDMS and Datacom, Software AG's Adabas, as well as in DB2 on z/OS and sequential files. The server runs on z/OS as a started task and provides communications, task, memory, and file management capabilities. It hosts services such as the query processor used to process SQL and the Correlation Server used manage changes from the databases.

The DataMapper and Metadata Utility produce relational table definitions in the Classic Catalog that map to elements in the native datasources. For example, the DataMapper GUI on Windows uses IMS Database Definitions (DBDs) and COBOL copybooks to produce a grammar called USE grammar, that is uploaded to the z/OS system and used by the batch Metadata utility to produce the table definitions in the Classic catalog.

Connectors use information in the catalog to map SQL requests from client applications into the native calls necessary to access a datasource. For example, there is an IMS ODBA connector that can make DL/I calls to IMS databases with full update capability, including RRS support for two-phase commit. There is also an IMS Transaction Connector which is used to execute IMS transactions. Shipped as part of II Classic Federation are a set of predefined stored procedures that map to the various forms of IMS transactions. In addition to basic single message non-conversational transactions, stored procedures are also provided for conversational transactions as well as transactions with multi-segment messages.



Why are Customers using Data Replication?

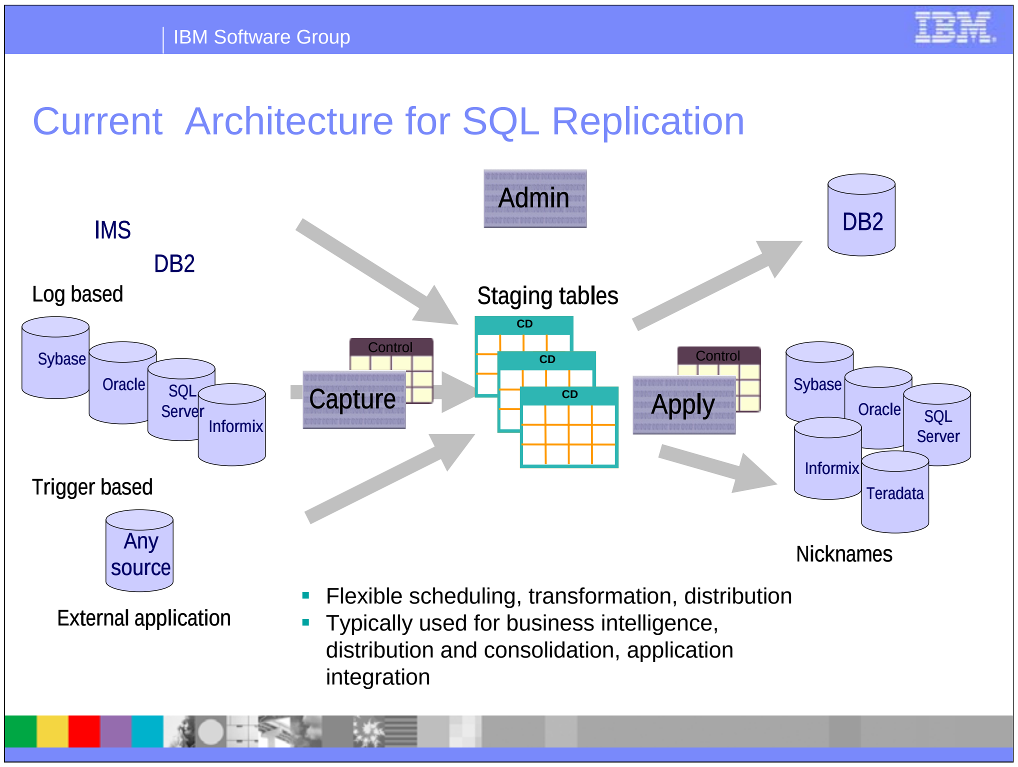
- **Availability**
  - Scheduled outage, failover, disaster recovery
    - Can use Hardware, Software, or a combination of methods
  - Move query or reporting work to a separate system
    - Other methods such as flash copy also possible
  - Peer to peer - split workload
    - This is only possible through replication
- **Distribution / Consolidation**
  - Move data between central to branches, branches to central, or both
  - Federate or Replicate?
    - where does the application need the data to be? - what db?, what platform?
    - does the data need to be real time or not?
    - what is the change volume?
- **Warehouse / Business Intelligence / Application Integration**
  - Move data to new platform/database, transform data
  - ETL or Replicate?
    - latency needs
    - change volume versus total volume
    - complexity of transformation and/or cleansing

Customers are using replication products to satisfy a wide variety of application needs.

Warehouses and ad-hoc query databases can be built using changed data, real time rather than through less frequent full extract/load processing. So in this case the customer must weigh the relative advantages and disadvantages of traditional ETL processing versus change capture replication.

Data can be accessed in place using federation, but when availability and/or performance of the application is critical, frequently the choice is made to replicate data to a local copy or cache.

The most frequently cited application of replication technology in recent years is availability. Copies of data may be used for scheduled outages, unscheduled outages, disaster recovery, or combinations of these. There are many choices to consider in a high availability scenario- hardware/software, logical/physical, synchronous/asynchronous. Some customers opt for a combination of methods for best coverage.



IBM Software Group

Current Architecture for SQL Replication

- Flexible scheduling, transformation, distribution
- Typically used for business intelligence, distribution and consolidation, application integration

•Sold under the name DB2 Datapropagator, as part of DB2 Information Integrator, or as part of DB2 for LUW, this is the architecture that has been available for the last 10 years. We are now calling this SQL Replication to make it distinct from Q Replication (the new queue based architecture)

•A Capture program or trigger captures changes 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.

•The staging table typically resides on the same system as the source table

•Staging table format is published to enable applications or ISV to provide capture function

•The Apply program fetches data from the staging tables using client/server db2 communications and applies it to the target tables using standard SQL statements

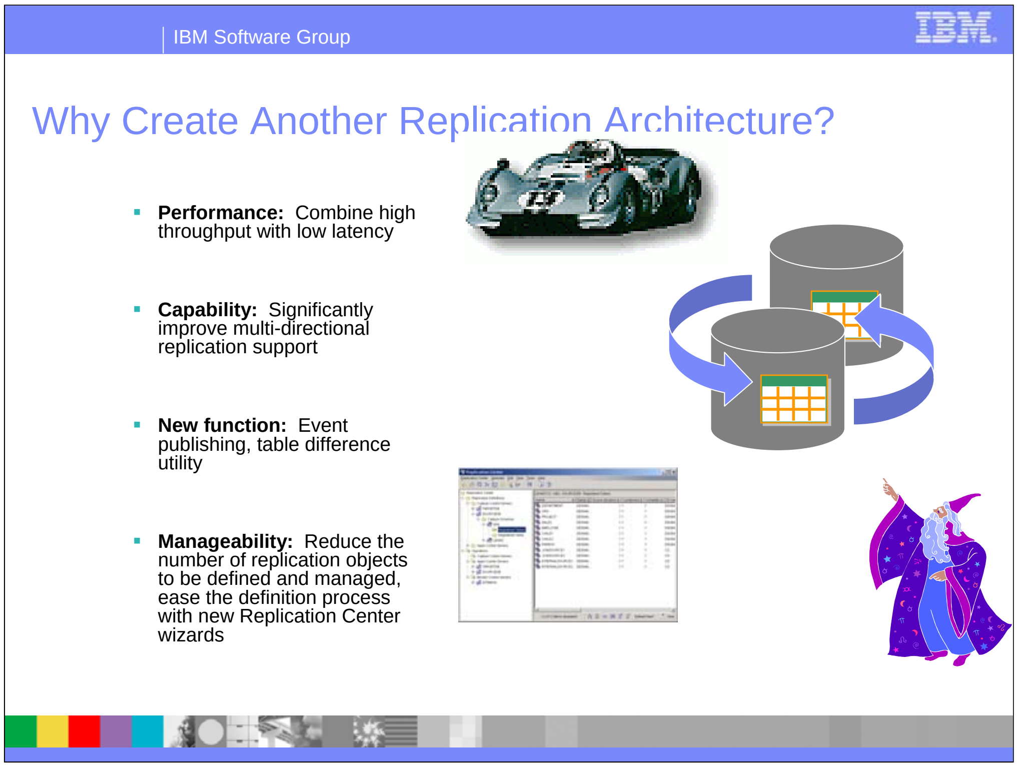
•One or more apply programs can subscribe to a CD table

•One apply program can replicate data to one or more target tables

•Target tables can be user copies, history tables, or staging tables

•Apply program handles column and row subsetting, performs SQL transformations, manages commit scope based on subscription sets and table vs transaction consistent delivery ➔ note that RI cannot be guaranteed for foreign sources as ordering across tables is unknown from trigger capture mechanisms

•Apply program references foreign source and target tables and control tables via nicknames



IBM Software Group

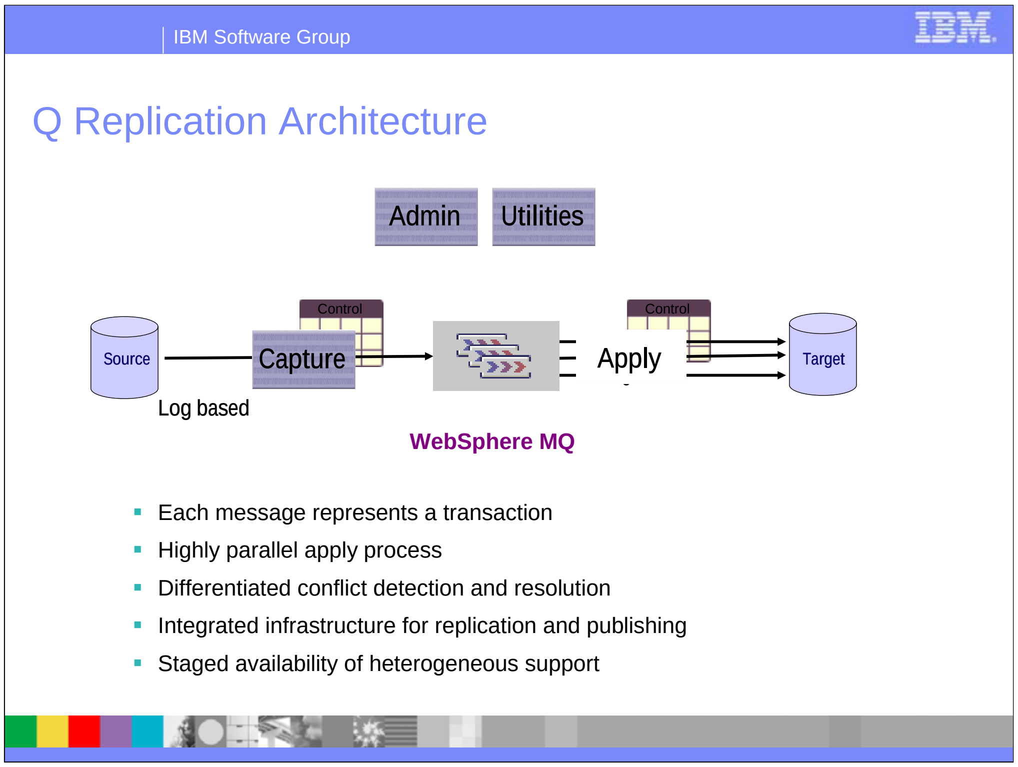
Why Create Another Replication Architecture?

- **Performance:** Combine high throughput with low latency
- **Capability:** Significantly improve multi-directional replication support
- **New function:** Event publishing, table difference utility
- **Manageability:** Reduce the number of replication objects to be defined and managed, ease the definition process with new Replication Center wizards

There is a growing demand for high speed low latency replication, primarily for the purposes of meeting high availability requirements. The implementations are varied, and include geographically distributed peer to peer applications, workload balancing, and primary/secondary failover configurations. In addition to speed, these implementations require robust methods for conflict detection and resolution.

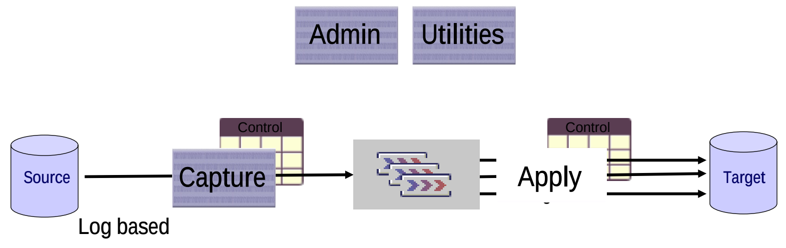
We also see the need for a solution that is easy to manage - requiring reduction in the numbers of objects to create and manage, and with easier methods to create and manage those objects.

In creating this new architecture, we also see an opportunity to create an infrastructure that can serve application messaging/publishing in addition to replication.



IBM Software Group

Q Replication Architecture



WebSphere MQ

- Each message represents a transaction
- Highly parallel apply process
- Differentiated conflict detection and resolution
- Integrated infrastructure for replication and publishing
- Staged availability of heterogeneous support

•Capture program stages data in queues

•Each message represents a transaction

•One or more data transport queues per source/target database pair

•Apply is significantly re-architected

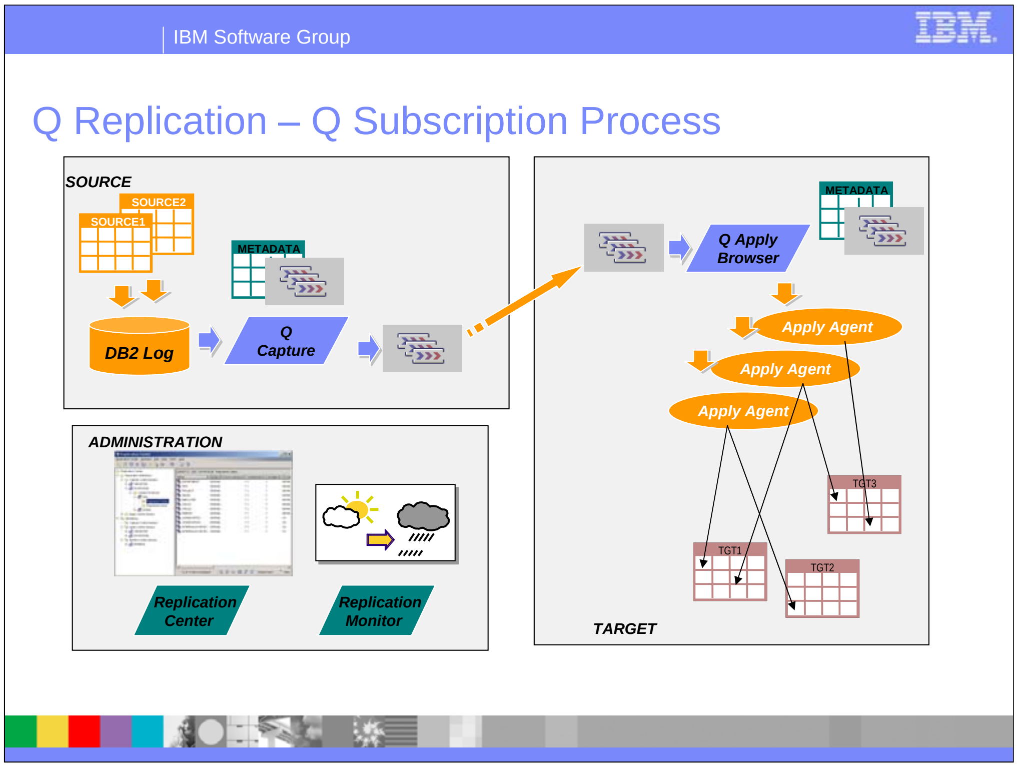
•Highly parallel in how it applies transactions to tables

•Data is always applied per source transaction units

•Data is applied such that source commit order is observed where necessary for data integrity

•Conflict detection very robust, including ability to handle deletes and key changes

•Data can also be published in XML format for external applications, using the same capture infrastructure



Q Replication – Q Subscription Process

This slide takes you through the implementation details of Q replication.

(1) First you install the programs and set up infrastructure for queues and control table metadata

(2) Subscribe to those tables of interest

(3) Changes to those tables will appear on the DB2 recovery log

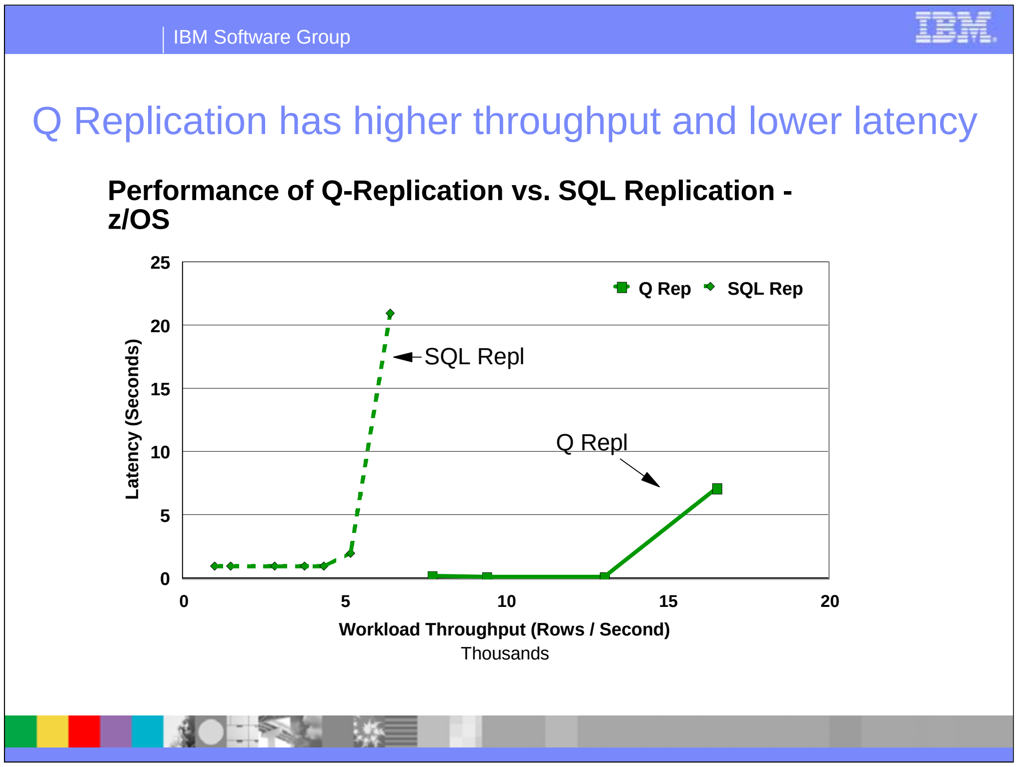
(4) The changes will be read by Q Capture and stored in memory

(5) Committed transactional data will be put to the data transport queue

(6) At a commit interval the data will be sent by MQ to the target receive queue

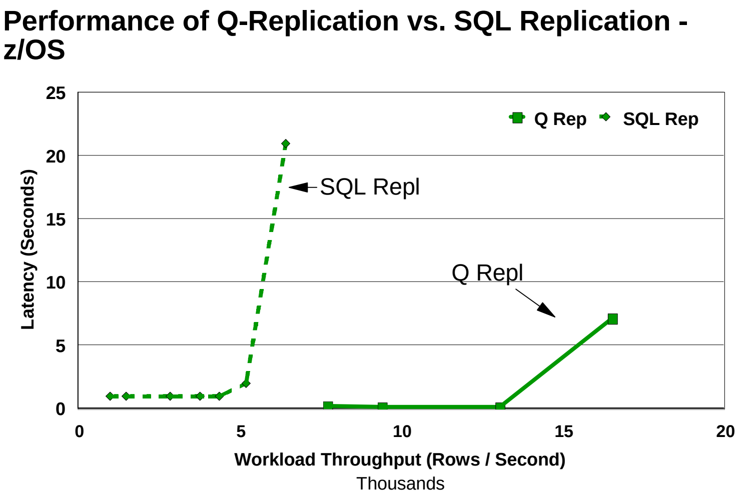
(7) Q Apply browser reads transactions from the queue, examines and tracks dependencies between transactions, and feeds transactions to Apply agents

(8) The alert monitor program keeps an eye on the both SQL and Q Replication server metadata and statistics.



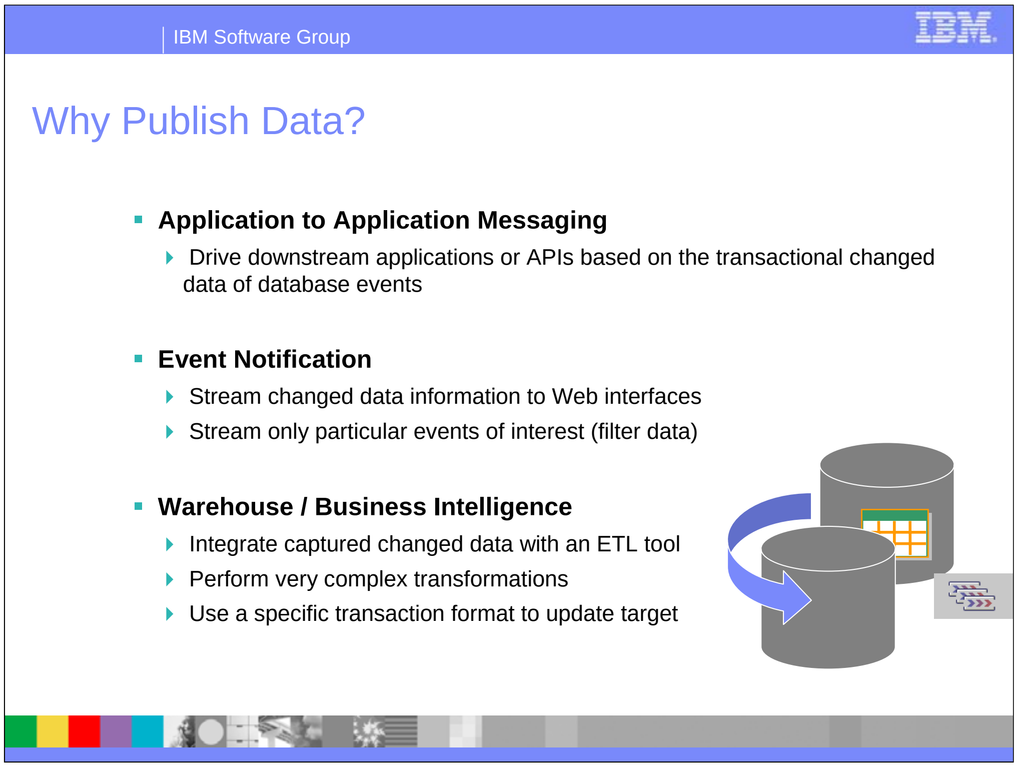
IBM Software Group

Q Replication has higher throughput and lower latency



This chart is one example of performance measurements taken at our own performance laboratory. Beta customers have also been impressed with the speed and latency of Q Replication.

Replication performance is subject to many variables. In this example we measured 10 row insert transactions, with avergae row length approximately 200 bytes. It is an intensive all insert workload, to stress the replication software.


IBM Software Group

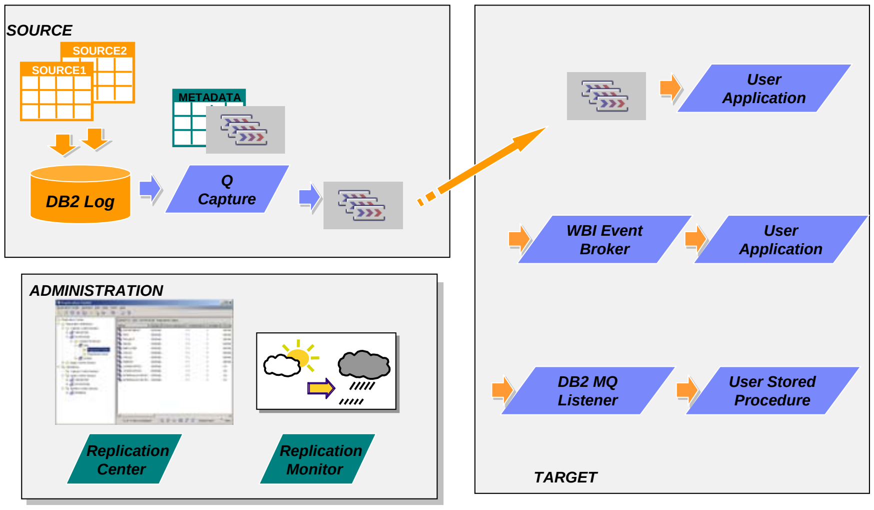
Why Publish Data?

- **Application to Application Messaging**
  - Drive downstream applications or APIs based on the transactional changed data of database events
- **Event Notification**
  - Stream changed data information to Web interfaces
  - Stream only particular events of interest (filter data)
- **Warehouse / Business Intelligence**
  - Integrate captured changed data with an ETL tool
  - Perform very complex transformations
  - Use a specific transaction format to update target

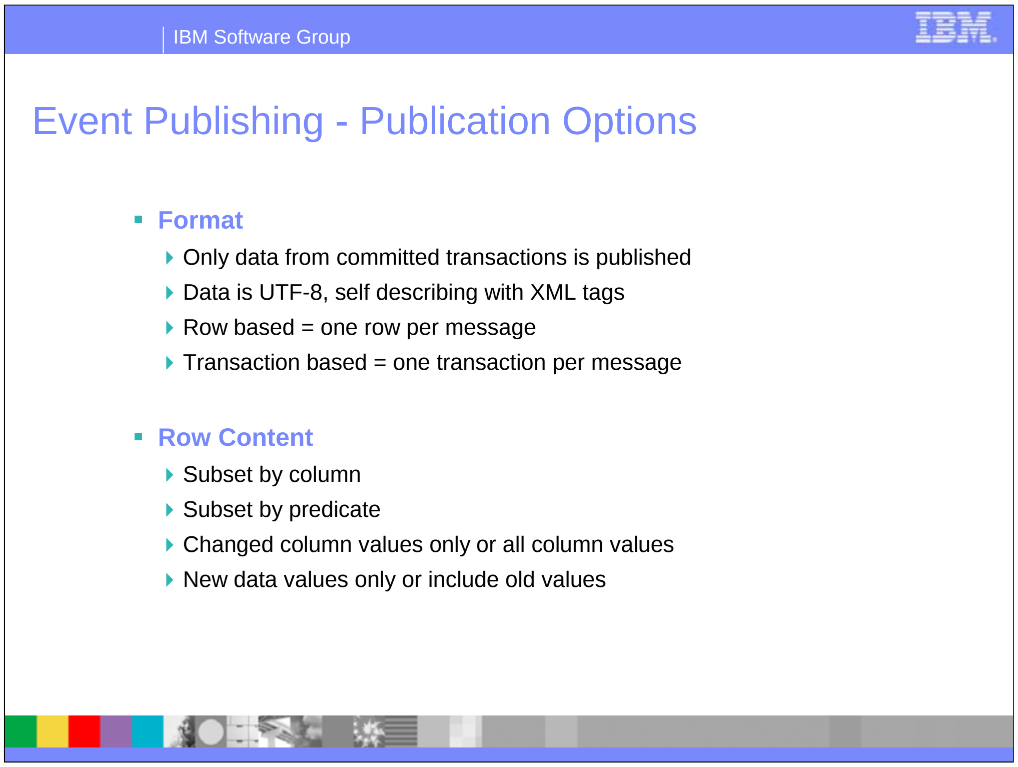
More and more customers are using message queuing to provide application to application communication. When the need exists to combine database activity with application messaging, then a strong advantage can be gained by using an asynchronous log based infrastructure to post messages that coordinate with database events. This eliminates the cost of 2 phase commit, or works in databases that cannot support a 2 phase commit. This also avoids availability concerns posed by an application that would otherwise require both the message queue server and the database to be available. In addition to the performance and availability gains, there is the simplicity of a central publishing mechanism that can be used without any special coding changes or modifications to old or new applications.

Database changes can be posted to a queue and then sent to downstream applications for further processing. Examples: streaming stock prices or wholesale item prices, moving data from an order database to shipping and/or billing databases, notifying all systems of customer information changes….

Q Replication – Event Publication Process



This slide depicts various configuration suggestions for event publishing. In addition to receiving published data directly from a user application, the data could first be brokered by the Websphere Business Integration Event Broker (formerly known as MQSI – MQ Series Integrator), and then passed on to other applications, or the data could be brokered by the MQ Listener function of DB2 on LUW or z/OS, and then passed on to your user written stored procedure.

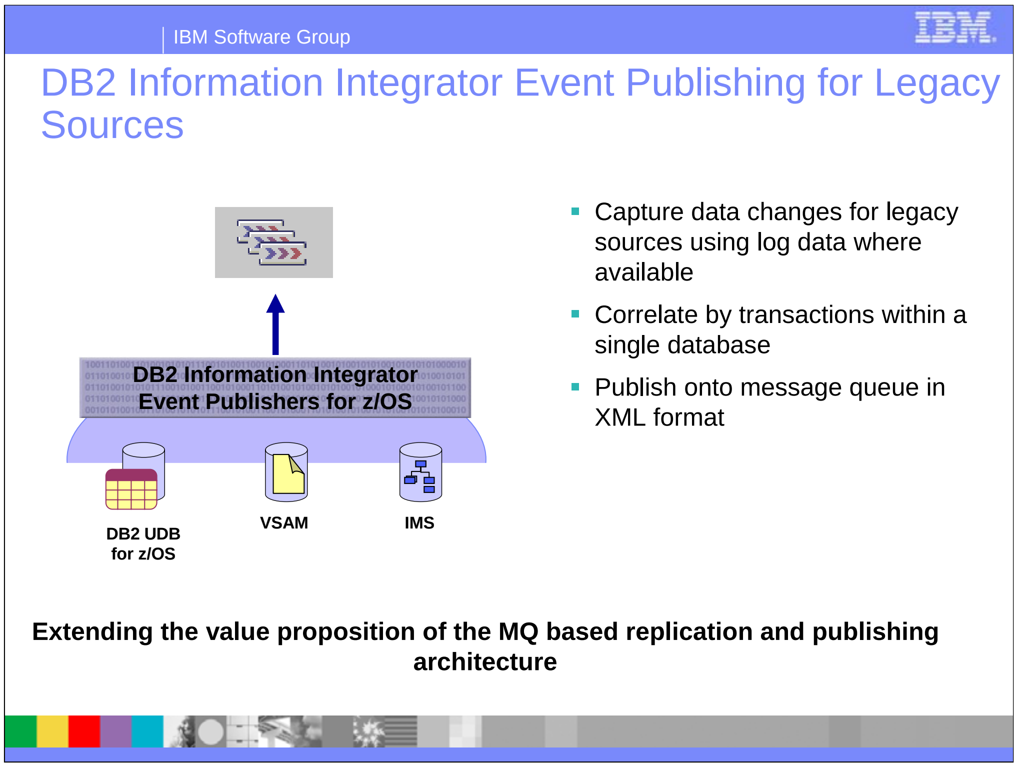


Event Publishing - Publication Options

- **Format**
  - Only data from committed transactions is published
  - Data is UTF-8, self describing with XML tags
  - Row based = one row per message
  - Transaction based = one transaction per message
- **Row Content**
  - Subset by column
  - Subset by predicate
  - Changed column values only or all column values
  - New data values only or include old values

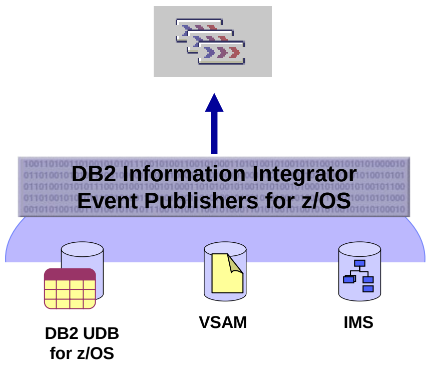
Data is captured in the same way that it is captured in Q Replication – transactional data is stored in memory until a commit record has been seen on the log. Then the committed data is translated into UTF-8, tagged with descriptive XML tags, and is put to a queue. You can choose for the messages to be made up of individual row changes, or of all associated row changes that were in a transaction.

Additionally, you have multiple options that dictate how much data is put into each row operation published: new/old values, all columns or only changed columns, etc.



IBM Software Group

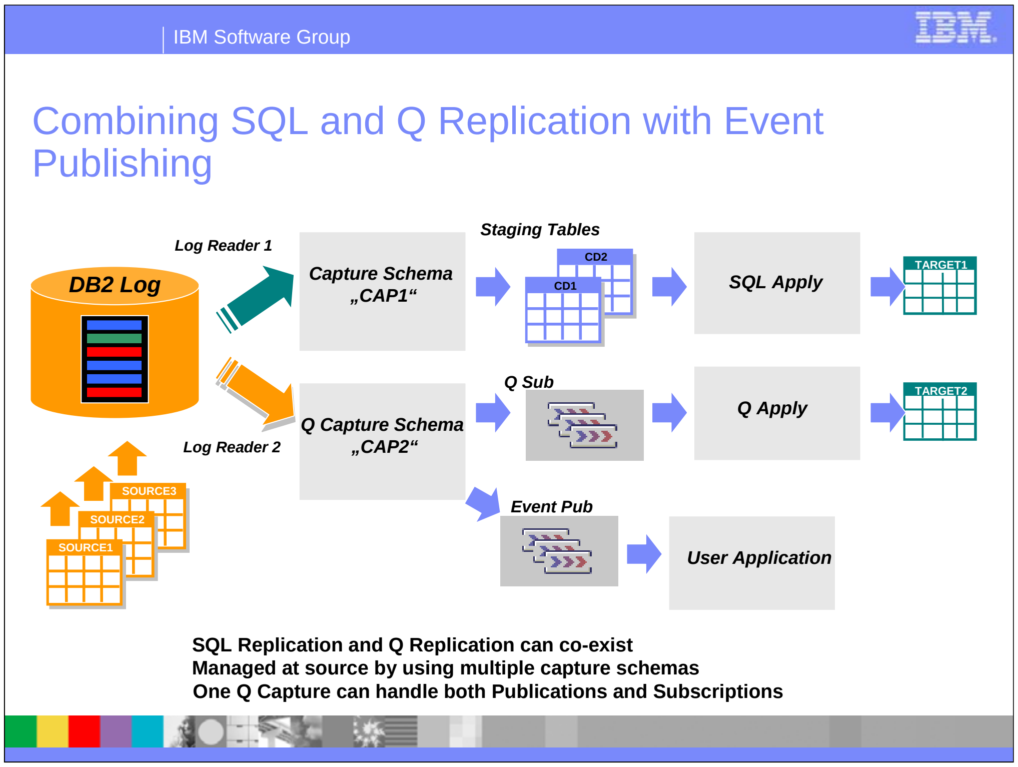
DB2 Information Integrator Event Publishing for Legacy Sources



- Capture data changes for legacy sources using log data where available
- Correlate by transactions within a single database
- Publish onto message queue in XML format

Extending the value proposition of the MQ based replication and publishing architecture

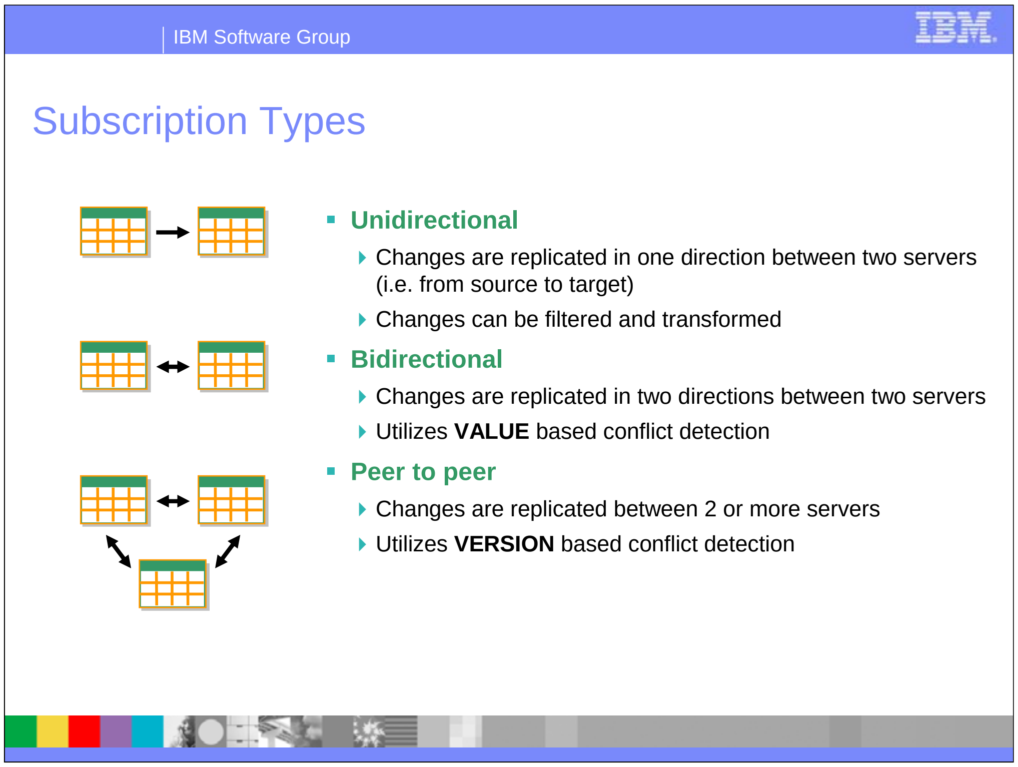
In addition to the event publisher for DB2, we are extending this technology by also offering event publishing from other legacy data sources. The first to be offered will be event publishers for IMS and CICS VSAM. All of our event publishers provide MQ messages in the same UTF 8 XML format. Most options that are offered for DB2 event publishing are also offered for the legacy sources. The major differences are (a) the non relational data must first be mapped to a relational format through a mapping tool, and (b) the legacy capture does not allow you to capture subsets of the data at the logical "row" level – subsetting must be performed at the application level.



Combining SQL and Q Replication with Event Publishing

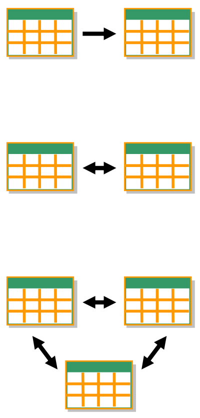
SQL Replication and Q Replication can co-exist
Managed at source by using multiple capture schemas
One Q Capture can handle both Publications and Subscriptions

This picture shows a coexistence configuration between SQL Replication, Q Replication, and Event Publishing. They can all co-exist on the same LUW database, DB2 z/OS subsystem , or datasharing group. Q Replication and Event Publishing tasks can be performed using the same Q Capture program (schema), but they do require separate queues. SQL Replication requires a separate Capture program (schema) – one Capture program cannot perform both SQL and Q Replication. One reason for this is that the Q Replication has been carefully designed to completely avoid 2 phase commit operations between DB2 and MQ.



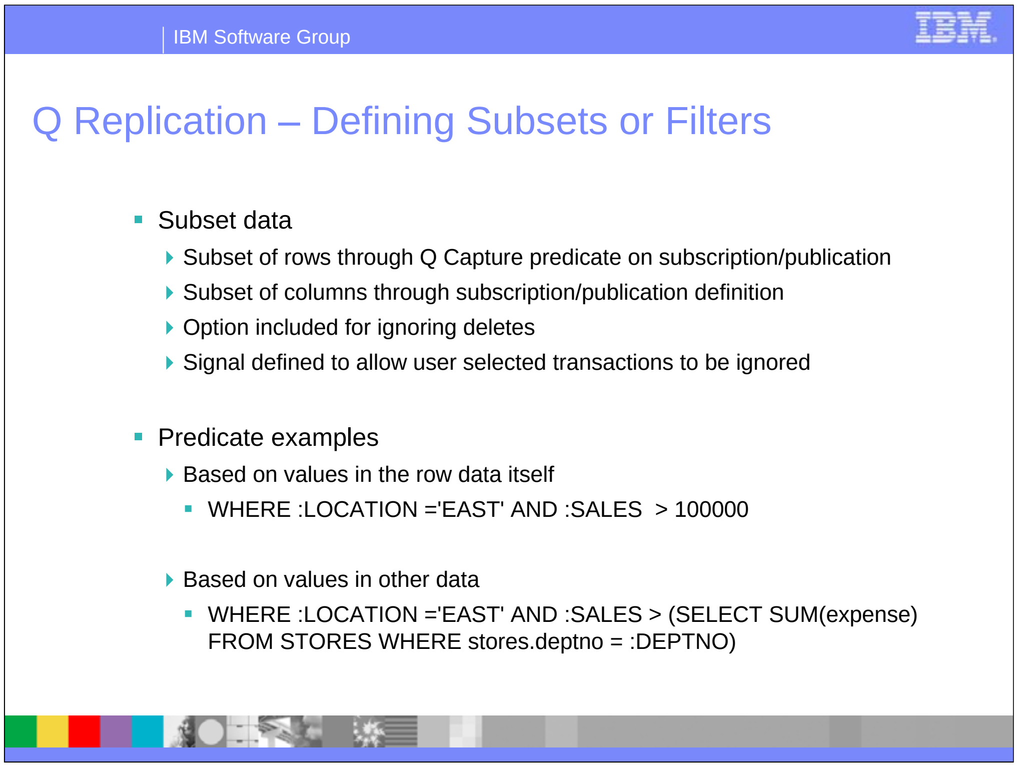
IBM Software Group

Subscription Types



- **Unidirectional**
  - Changes are replicated in one direction between two servers (i.e. from source to target)
  - Changes can be filtered and transformed
- **Bidirectional**
  - Changes are replicated in two directions between two servers
  - Utilizes **VALUE** based conflict detection
- **Peer to peer**
  - Changes are replicated between 2 or more servers
  - Utilizes **VERSION** based conflict detection

There will probably be a confusion point in terminology here between bidirectional and peer to peer. Bidirectional is a term that we are using to indicate that there are only 2 server databases involved, and that this is not a multi-master relationship. One server is designated as the winner of any conflicts, and column values are used to detect conflicts. Peer to peer is a true multi-master configuration that can allow greater than 2 servers, and provides a version (timestamp) based conflict detection and resolution method.

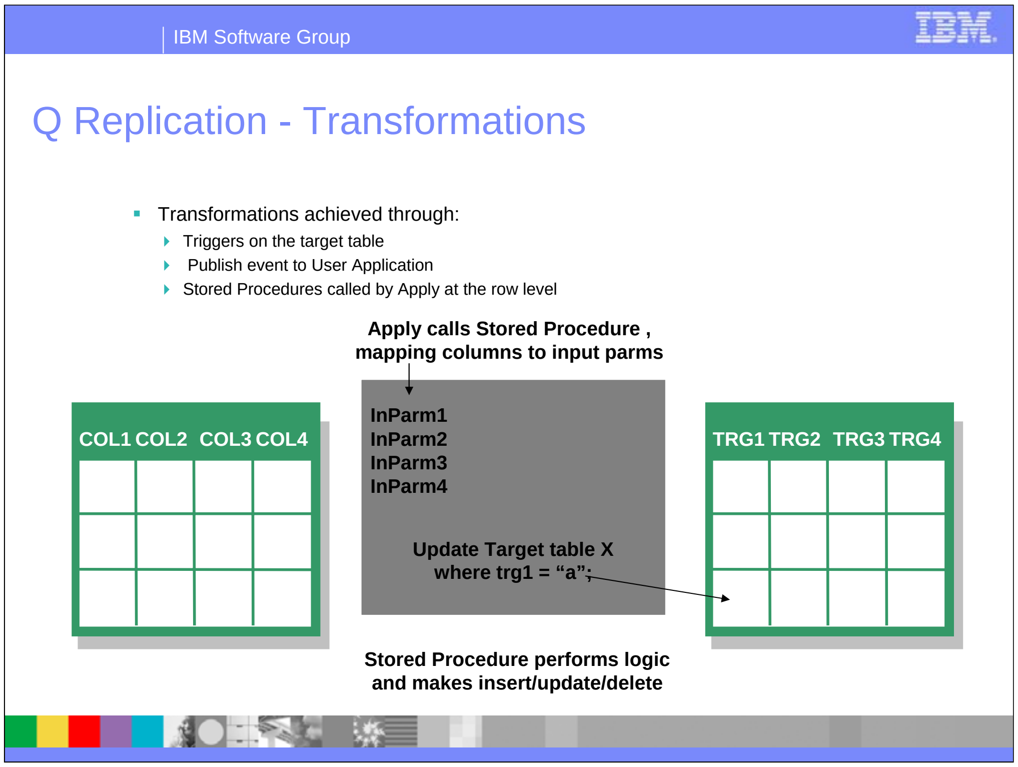


Q Replication – Defining Subsets or Filters

- Subset data
  - Subset of rows through Q Capture predicate on subscription/publication
  - Subset of columns through subscription/publication definition
  - Option included for ignoring deletes
  - Signal defined to 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 side versus evaluation during the Apply process in SQL Replication. 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.



IBM Software Group

Q Replication - Transformations

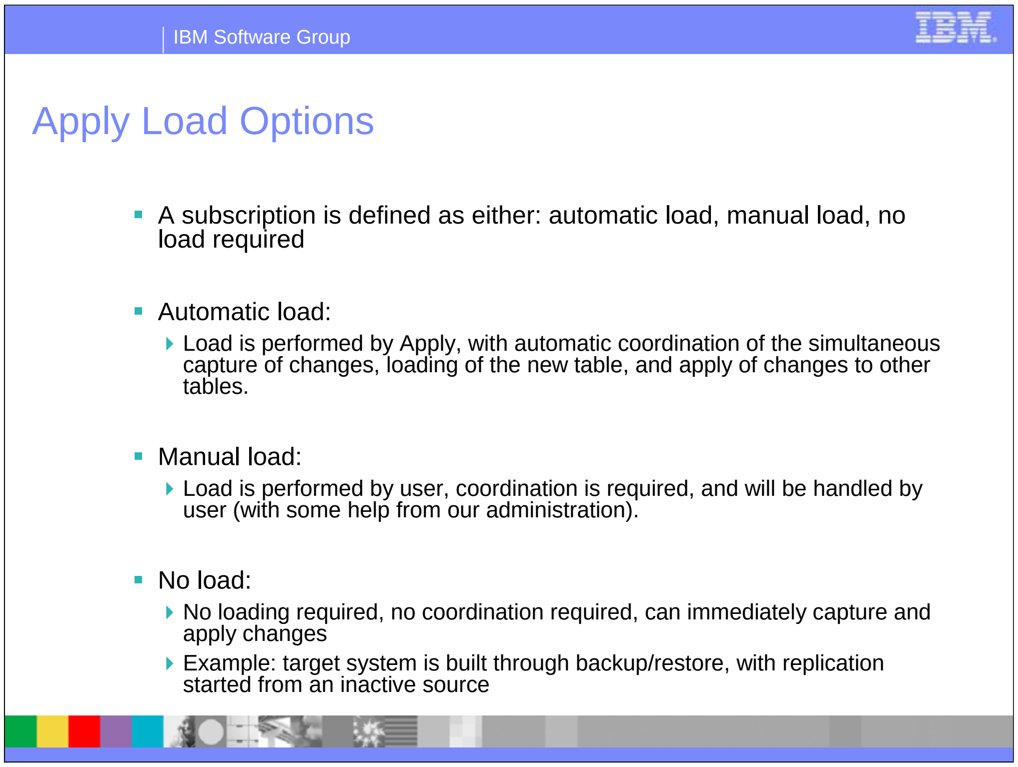
- Transformations achieved through:
  - Triggers on the target table
  - Publish event to User Application
  - Stored Procedures called by Apply at the row level

Apply calls Stored Procedure , mapping columns to input parms
COL1 COL2 COL3 COL4
InParm1
InParm2
InParm3
InParm4
Update Target table X where trg1 = "a";
TRG1 TRG2 TRG3 TRG4
Stored Procedure performs logic and makes insert/update/delete

One of the big benefits of SQL Replication is that transformations can be made very easily using SQL expressions. This capability is clearly different in the Q Replication case, where SQL is not being used. In Q Replication we provide a transformation exit capability in the way of a stored procedure call. The column values are passed in as the input parameters to the stored procedure, and the actual update of the table is made by the stored procedure, after any manipulations have been performed.

As previously described, the Event Publishing capability can be used to perform more extensive transformations.

Also, triggers and/or user defined functions could be defined on the target table to perform basic manipulations.

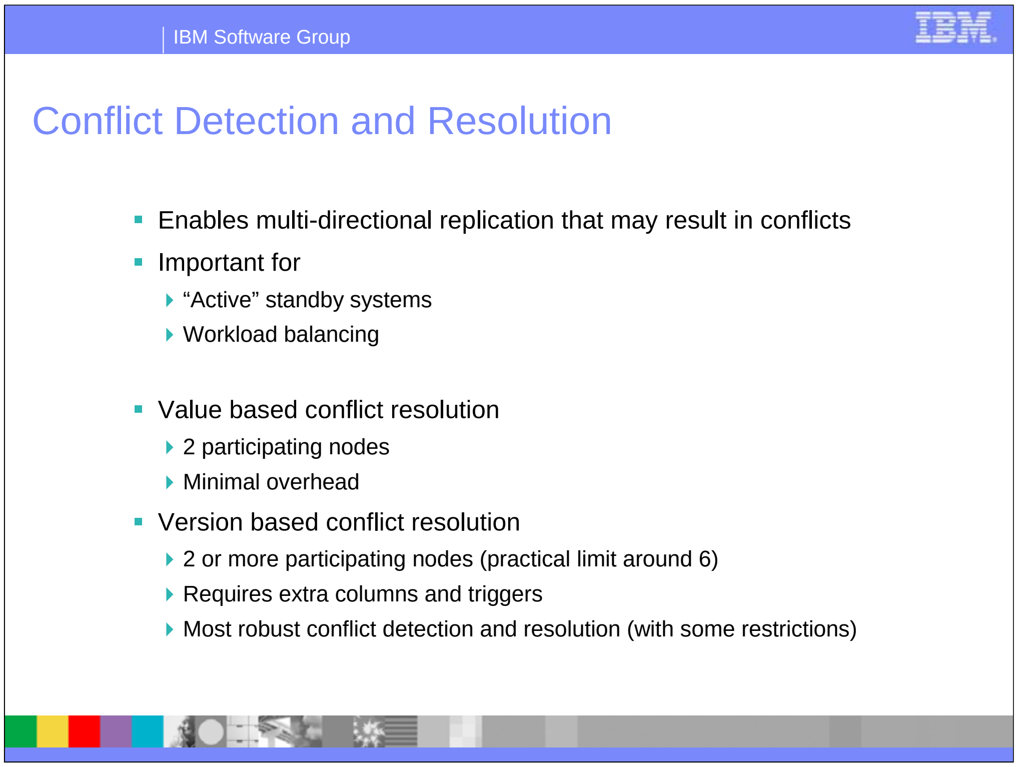

IBM Software Group

Apply Load Options

- A subscription is defined as either: automatic load, manual load, no load required
- Automatic load:
  - Load is performed by Apply, with automatic coordination of the simultaneous capture of changes, loading of the new table, and apply of changes to other tables.
- Manual load:
  - Load is performed by user, coordination is required, and will be handled by user (with some help from our administration).
- No load:
  - No loading required, no coordination required, can immediately capture and apply changes
  - Example: target system is built through backup/restore, with replication started from an inactive source

When source tables are being updated in parallel with the extraction of the source data to populate the target table (initially, before replication begins), then coordination is required between the Q Capture and Q Apply processes and the load itself. This coordination can be performed automatically by the product, or by the user if that is preferred.

When the source tables can be made temporarily inactive, still other methods can be employed that require no coordination. In this case the subscriptions can be defined as “no load required”.


IBM Software Group

Conflict Detection and Resolution

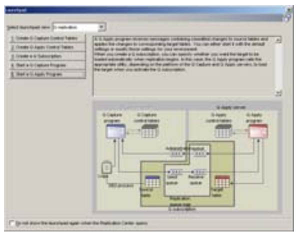
- Enables multi-directional replication that may result in conflicts
- Important for
  - “Active” standby systems
  - Workload balancing
- Value based conflict resolution
  - 2 participating nodes
  - Minimal overhead
- Version based conflict resolution
  - 2 or more participating nodes (practical limit around 6)
  - Requires extra columns and triggers
  - Most robust conflict detection and resolution (with some restrictions)

When multiple databases are allowed to update exactly the same records at exactly the same time, conflict detection and resolution must be incredibly robust. Here we recommend a version based method that can detect conflicts of all types. However, any robust method will require the addition of extra columns in the involved tables, and a mechanism (we offer triggers) to maintain these versioning columns.

When databases and applications are carefully set up such that only one copy is ever to be updated at a time, then it might be possible to live without conflict detection and resolution entirely. Or, it might be that one database will be a backup to the other, and backup scenarios may require conflict detection and resolution during switch and switchback timeframes. For this case, the peer to peer version based method can be used, but we also provide the value based alternate method, consuming less overhead, that may be entirely adequate for the application.

Replication Administration

- Replication Center GUI
  - Launchpads, Wizards, Online Help
  - Definitions, Operations, Monitoring



- Command Line Interface
  - Scripts or interactive mode
  - Example:

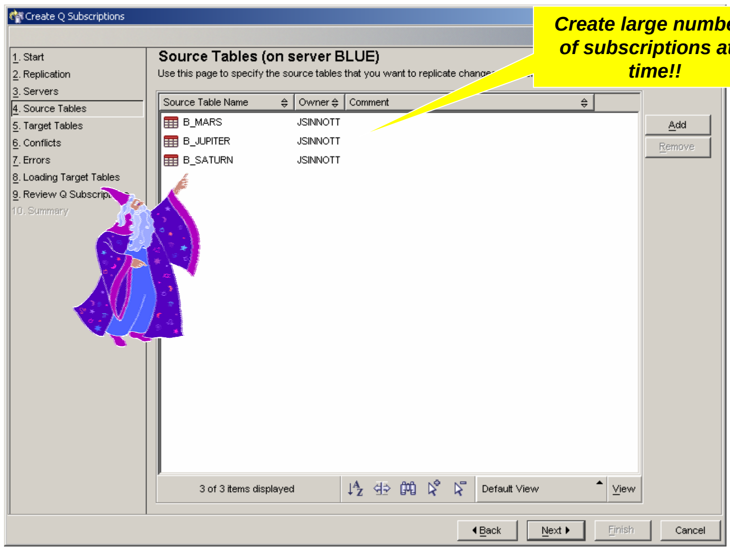
```
C:\asnclp
REPL > CREATE QSUB USING REPLQMAP ...
REPL > CREATE SUBSCRIPTION SET SETNAME ...
REPL > CREATE MEMBER IN SETNAME ...
```

- Java API's
  - Typically used when replication is embedded

As per V8.1, the administration is constructed on java APIs that can be called by a graphical user interface, a command, or a user program. It is strongly recommended that the novice user become more familiar with the product first by using the GUI, which has launchpads, wizards, and online help to get the user up and running very quickly. The more experienced user may prefer and find it faster to create command line scripts . Only vendors embedding the replication product would typically use a programmatic interface to the APIs.

Manageability of replication has been improved in Q Replication in several ways: one queue can be defined as the staging area for literally thousands of source objects. There is only one mapping to be defined – a subscription, rather than a registration pus a subscription. When many subscriptions will share common attributes, they can be built all at the same time using the mass subscription wizard.

Q Create Subscription Wizard



Source Tables (on server BLUE)

Use this page to specify the source tables that you want to replicate change

Source Table Name	Owner	Comment
B_MARS	JSINNOTT	
B_JUPITER	JSINNOTT	
B_SATURN	JSINNOTT	

Create large numbers of subscriptions at a time!!

Here is an example of the ease-of-use items added to the Replication Center : the Create Q Subscriptions wizard. Note that multiple subscription objects are being defined at once. The wizard walks the user through the various steps required to create subscriptions, and adjusts what steps are presented based on the user’s choices , eg unidirectional vs bidirectional or peer to peer.

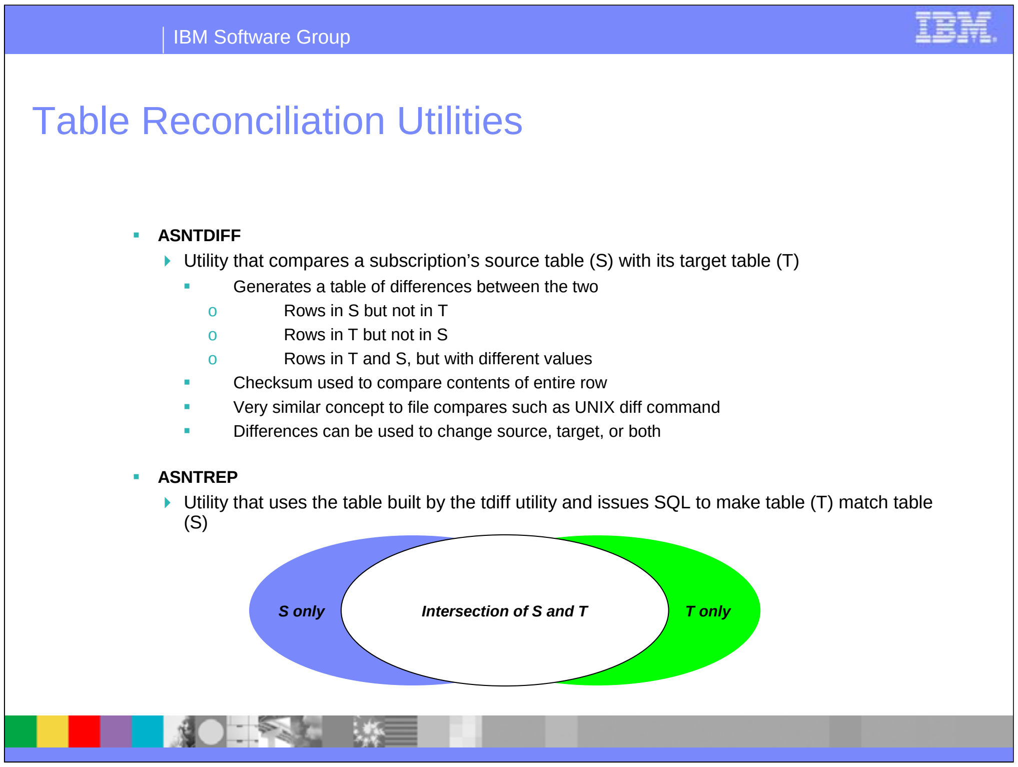
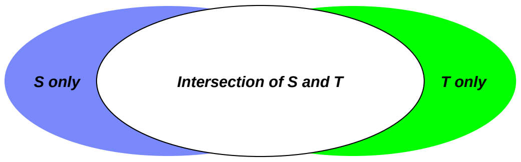

IBM Software Group

Table Reconciliation Utilities

- **ASNTDIFF**
  - Utility that compares a subscription's source table (S) with its target table (T)
    - Generates a table of differences between the two
      - Rows in S but not in T
      - Rows in T but not in S
      - Rows in T and S, but with different values
    - Checksum used to compare contents of entire row
    - Very similar concept to file compares such as UNIX diff command
    - Differences can be used to change source, target, or both
- **ASNTREP**
  - Utility that uses the table built by the tdiff utility and issues SQL to make table (T) match table (S)



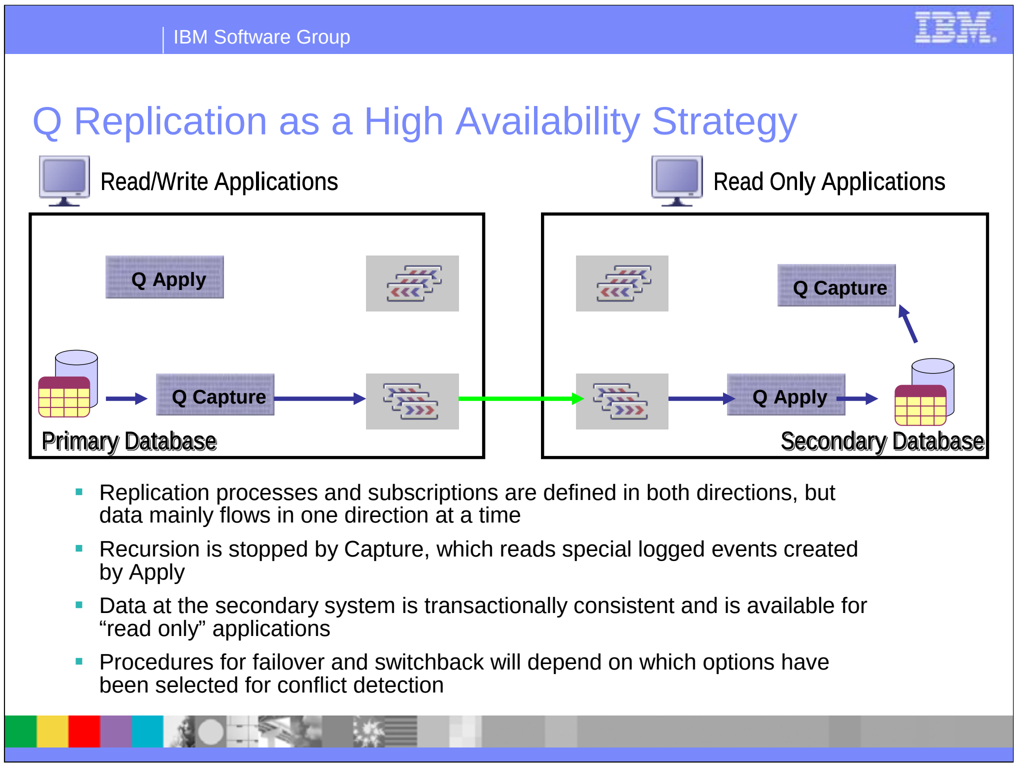
Tdiff has been used (in prototype with SQL Replication) with some internal and external customers:

www.ibm.com usage example (dBlue Application Delivery Team, eCare)

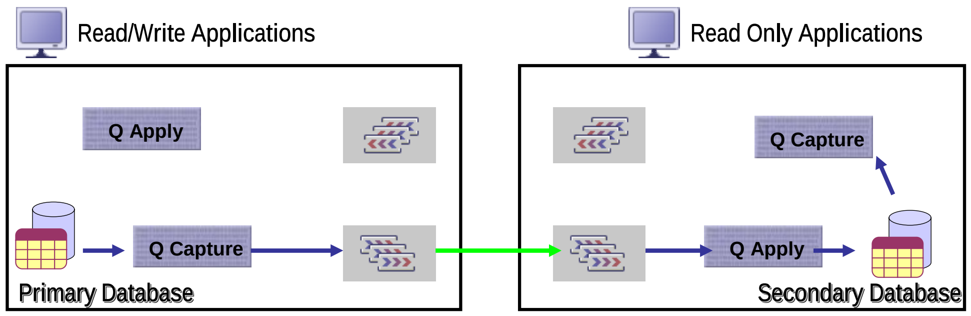
Ran 4 tdiff's in parallel (partitioned data via predicates) on one very large table to reconcile differences

Tables included LOBs (2 LOBs per row), 4.5 GB of LOB data in source/target tables, achieved 4.2 GB/sec diff

Note: Q Replication provides 3 utilities (1) monitor (mentioned on prior slide), (2) tdiff/trep (discussed on this slide), and (3) analyzer (not specifically discussed). Analyzer is a replication utility that is used as a serviceability aide. It looks at replication definitions/metadata and produces reports highlighting any detectable problems or inconsistencies in those definitions.

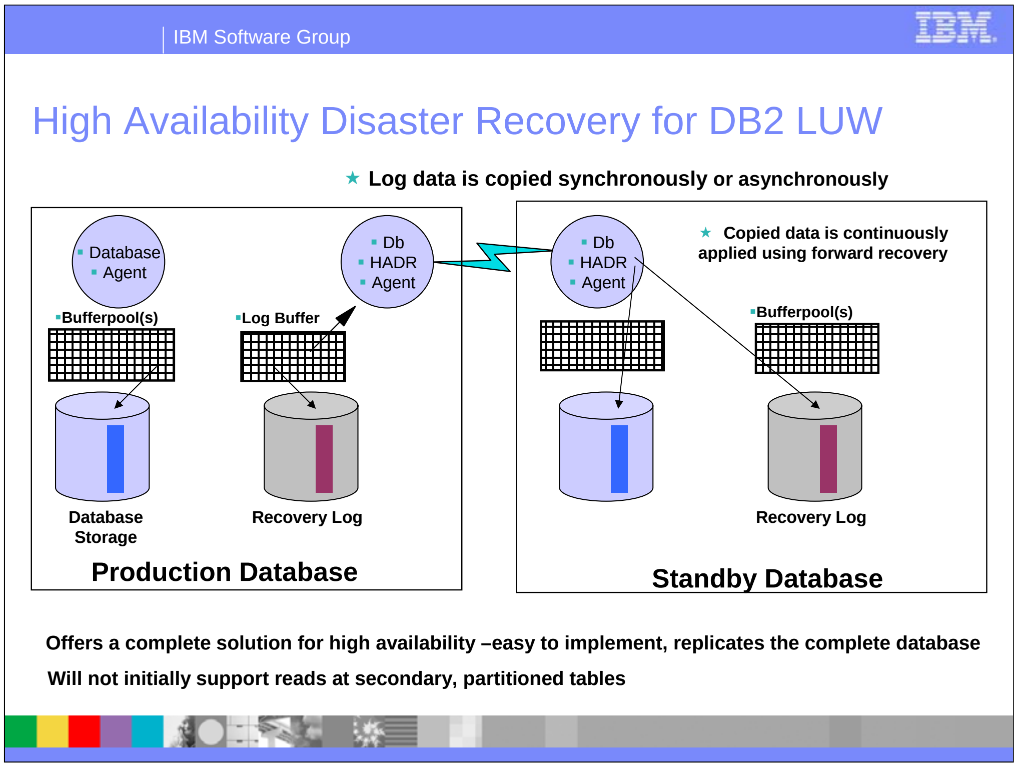

IBM Software Group

Q Replication as a High Availability Strategy



- Replication processes and subscriptions are defined in both directions, but data mainly flows in one direction at a time
- Recursion is stopped by Capture, which reads special logged events created by Apply
- Data at the secondary system is transactionally consistent and is available for “read only” applications
- Procedures for failover and switchback will depend on which options have been selected for conflict detection

In the next few charts, we will explore the use of replication as a high availability strategy. First we look at the basic setup. Replication is configured for either bidirectional or peer to peer, depending on the choice the user makes on how conflicts need to be handled. In either case , the replication is configured so that data changes can be captured in both directions so that the replication is ready immediately to handle a switch to the alternate server. No matter how the system is configured, recursion of updates is stopped at the Capture process, based on information provided by the Apply processes.


IBM Software Group

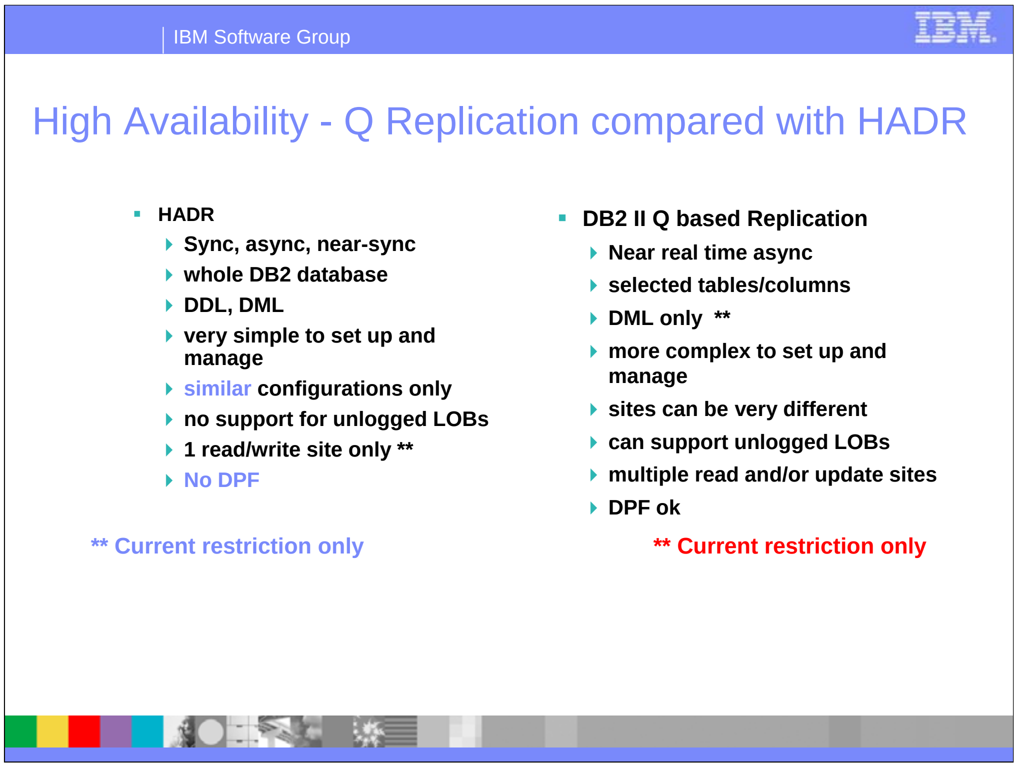
High Availability Disaster Recovery for DB2 LUW

Offers a complete solution for high availability –easy to implement, replicates the complete database

Will not initially support reads at secondary, partitioned tables

When considering a high availability solution for DB2 on LUW, consider also log shipping and the new DB2 LUW 8.2 HADR capability shown on this slide. In HADR, the data is replicated at a physical level, such that forward recovery is used to apply changes as seen in log records which have been sent from the primary to the standby.

There are advantages and disadvantages to the many high availability methods available (HADR, Q Replication, hardware), and you will want to carefully consider the benefits against your priorities. Some of these are examined in the following slide.


IBM Software Group

High Availability - Q Replication compared with HADR

- HADR
  - Sync, async, near-sync
  - whole DB2 database
  - DDL, DML
  - very simple to set up and manage
  - similar configurations only
  - no support for unlogged LOBs
  - 1 read/write site only **
  - No DPF

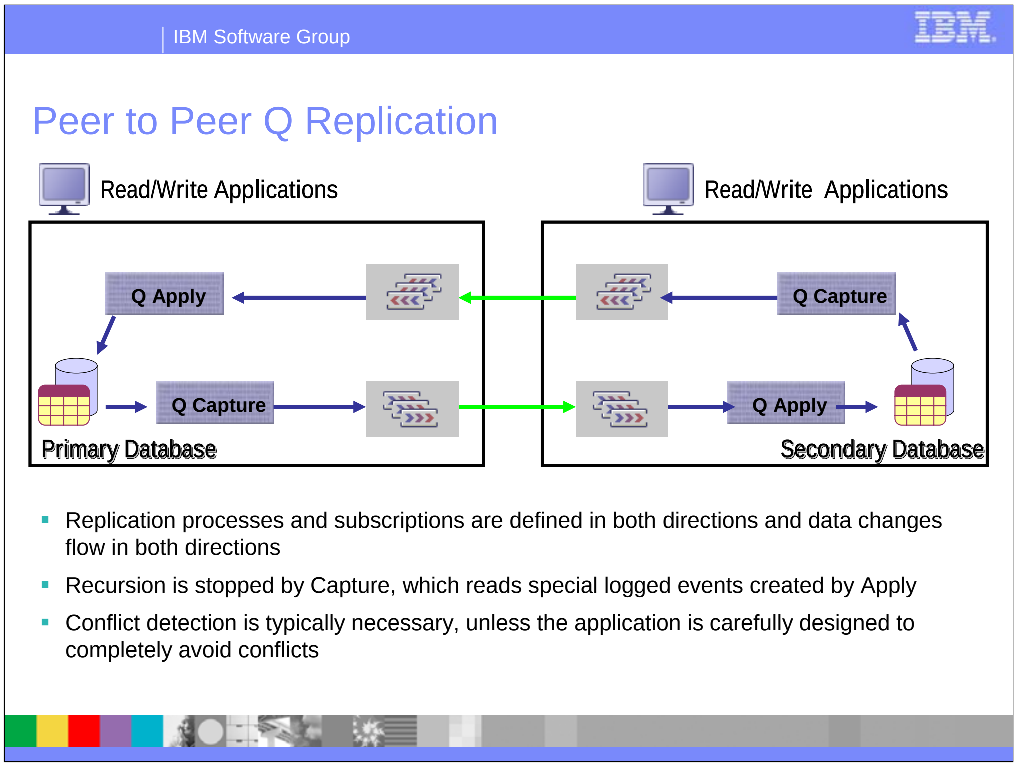
** Current restriction only

- DB2 II Q based Replication
  - Near real time async
  - selected tables/columns
  - DML only **
  - more complex to set up and manage
  - sites can be very different
  - can support unlogged LOBs
  - multiple read and/or update sites
  - DPF ok

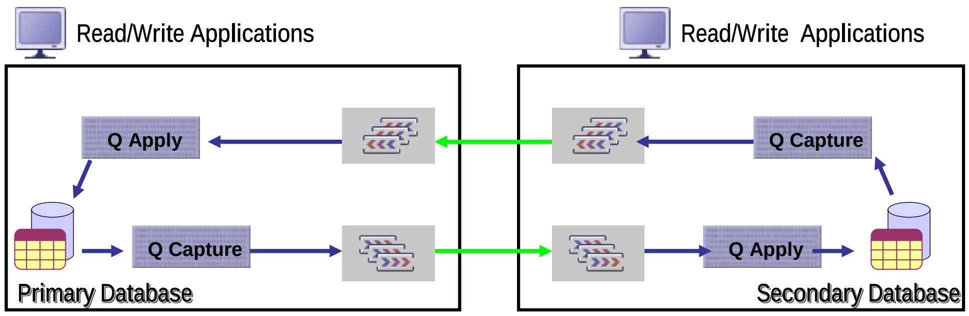
** Current restriction only

To net out this comparison, HADR is simpler to set up and manage because it covers the whole database. There are no filters or options to define. You do not have to consider issues such as identity columns and sequences, and you do not have to worry about adding new tables or altering existing tables – this is all covered.

So why do some users choose logical database replication? The most popular advantages are: (1)speed of takeover – the secondary in a replication configuration is live, (2) because the secondary database is live, it can be used for multiple purposes, and (3) because this is the only “logical” method available for high availability, it affords the most flexibility in hardware and software – primary and secondary systems can vary a great deal from one another. Therefore the advantages are (1) no down time = no loss of business, (2) greater ROI because the secondary can be multi-purposed, and (3) lower costs in building the secondary system.


IBM Software Group

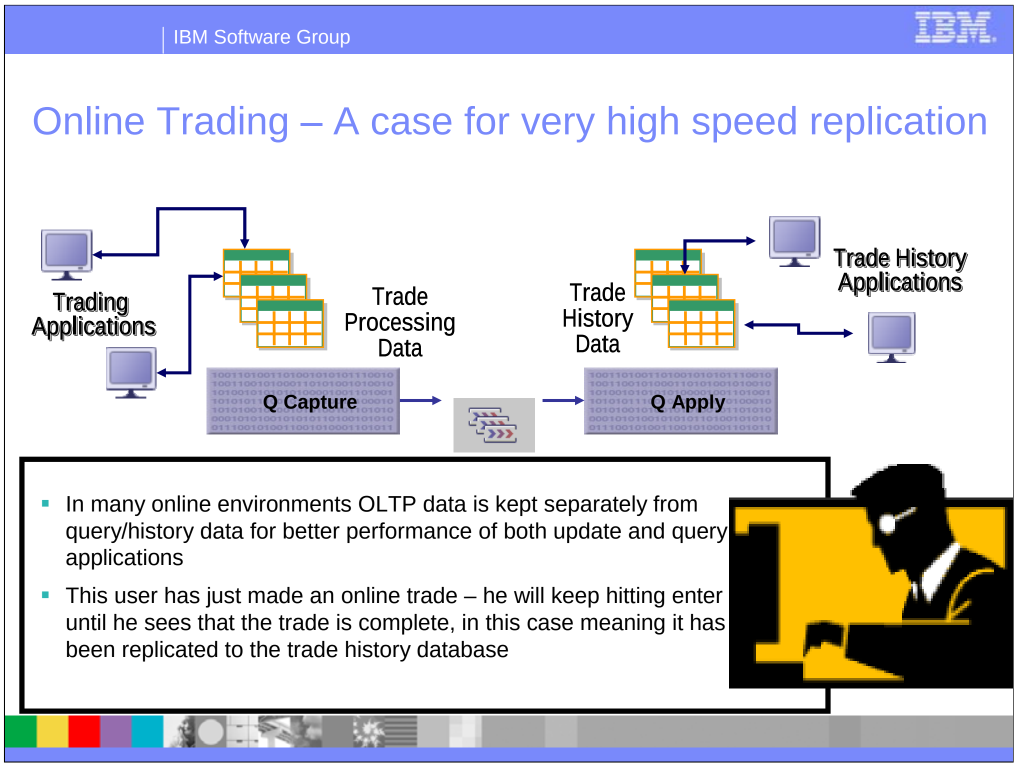
Peer to Peer Q Replication



- Replication processes and subscriptions are defined in both directions and data changes flow in both directions
- Recursion is stopped by Capture, which reads special logged events created by Apply
- Conflict detection is typically necessary, unless the application is carefully designed to completely avoid conflicts

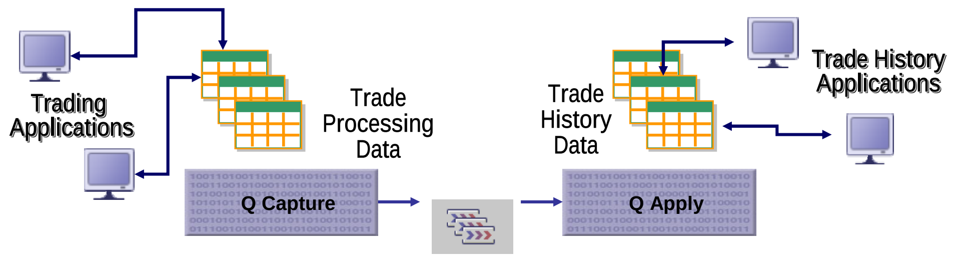
Peer to peer configurations are used to allow for workload balancing, often to bring a database closer to the user for better performance and availability. This is certainly the case for many online applications that are used all around the globe, and in particular for the online retail industry.

Peer to peer replication might be used in a configuration in which the same data rows may be updated simultaneously at multiple sites at the same time, or it may be that the same database is updated simultaneously at multiple sites at the same time, but in such a controlled fashion that it is not expected that more than one site would ever update any one individual row. However, it is also frequently the case that in a peer to peer configuration, it is desirable that if any one site becomes unavailable, then the applications that use this site failover to an alternate site. At this point the characteristics are identical to those previously discussed in the earlier slides on high availability. So even when applications are carefully planned and designed, it is typically necessary to have a strong plan to handle the failover and switchback cases, inclusive of conflict detection and resolution.



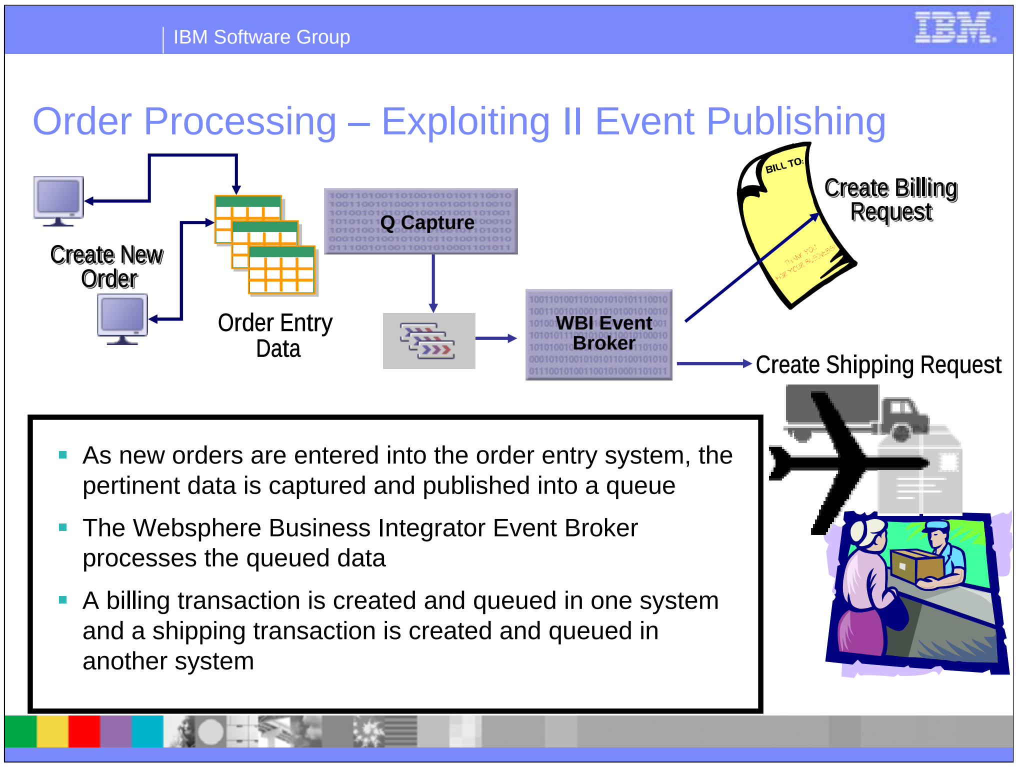
IBM Software Group

Online Trading – A case for very high speed replication



- In many online environments OLTP data is kept separately from query/history data for better performance of both update and query applications
- This user has just made an online trade – he will keep hitting enter until he sees that the trade is complete, in this case meaning it has been replicated to the trade history database

In many cases related business data is kept in two or more databases, possibly on different platforms, possibly from different vendors, and possibly in different formats. Different applications benefit from having the data in exactly the place where it is located and in that format, or it may just be that way because this is how the application was packaged. Replication has been bridging these gaps for years, but the new twist is that users have grown ever more demanding of their online environments. When a user makes an online trade, performs an online banking operation, uses frequent flyer points to buy or upgrade a flight, they want to verify that this action has occurred. Instant gratification is expected, and may be all the more desired because of insecurity of their internet actions. The user is not aware, and does not want to be aware, that multiple databases are used to run the business behind their actions. Many businesses are continuing to satisfy this database gap with replication, but they want it to be very fast, to provide the best seamless operation for their online customers.



IBM Software Group

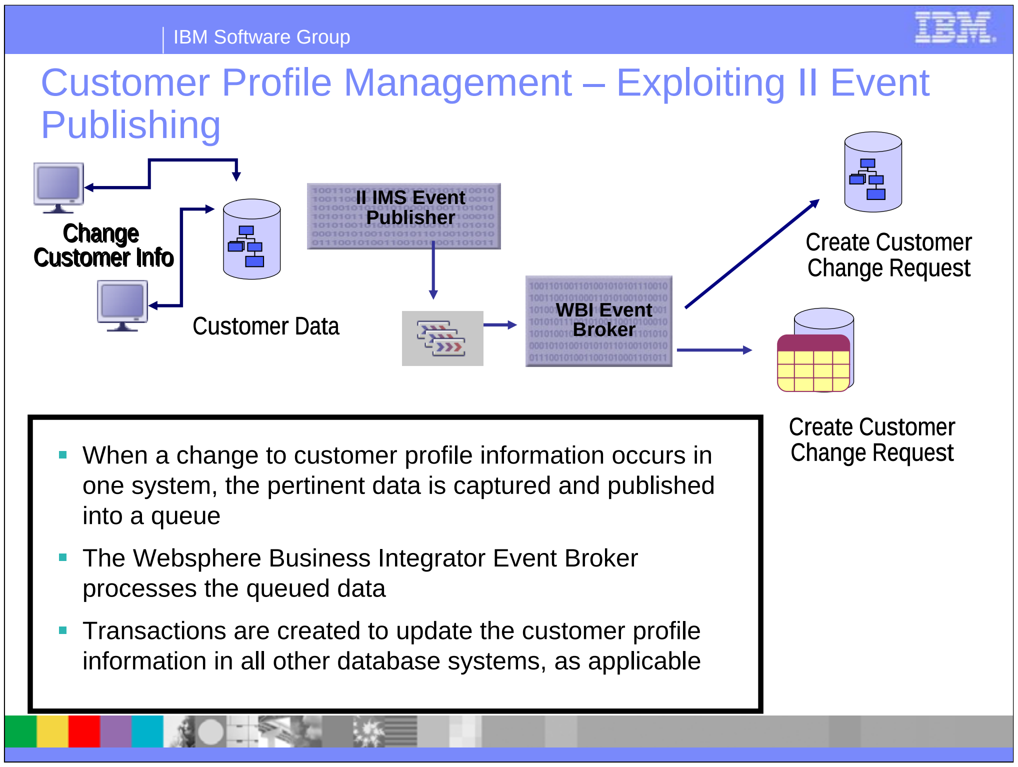
Order Processing – Exploiting II Event Publishing

- As new orders are entered into the order entry system, the pertinent data is captured and published into a queue
- The Websphere Business Integrator Event Broker processes the queued data
- A billing transaction is created and queued in one system and a shipping transaction is created and queued in another system

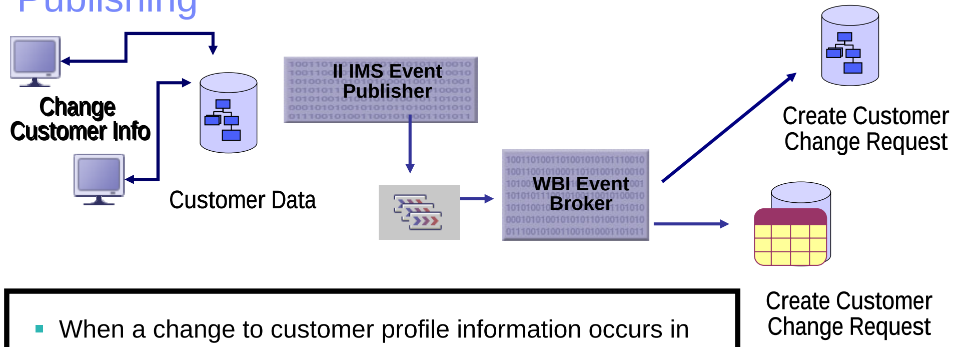
Business data needs to flow, within a company or between companies. There are many methods for creating this flow of data, and many businesses have turned to application messaging to perform this function. This is perhaps the very heart of service oriented architectures.

But here are several compelling reasons to use event publishing as the preferred method for application messaging – (1) the messages are created asynchronously from the originating application, reducing the performance impact on that application, (2) the application is similarly shielded from any loss of availability of the message queue or service, and (3) new or existing applications can be built without special coding for application messaging.

In this example, new orders can continue to be created while the connectivity to the in-house billing system or the external shipping system is down, even if queues are temporarily overfilled. When the connectivity is regained and message queues are again available, the orders can be processed and sent on.



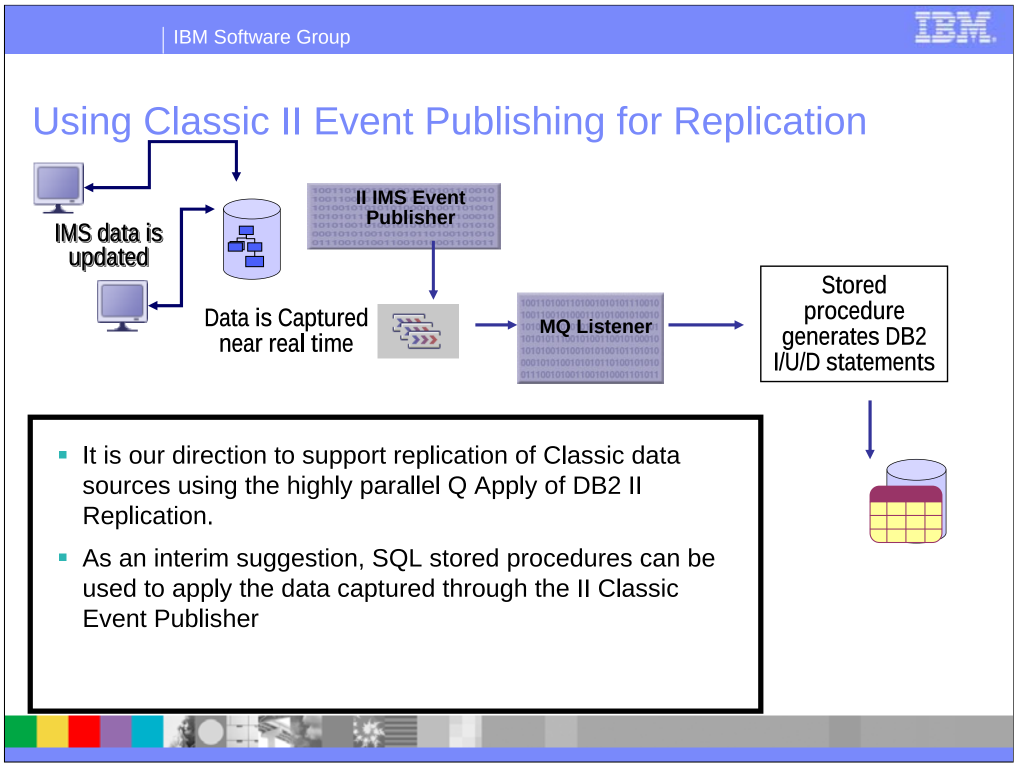
Customer Profile Management – Exploiting II Event Publishing



- When a change to customer profile information occurs in one system, the pertinent data is captured and published into a queue
- The Websphere Business Integrator Event Broker processes the queued data
- Transactions are created to update the customer profile information in all other database systems, as applicable

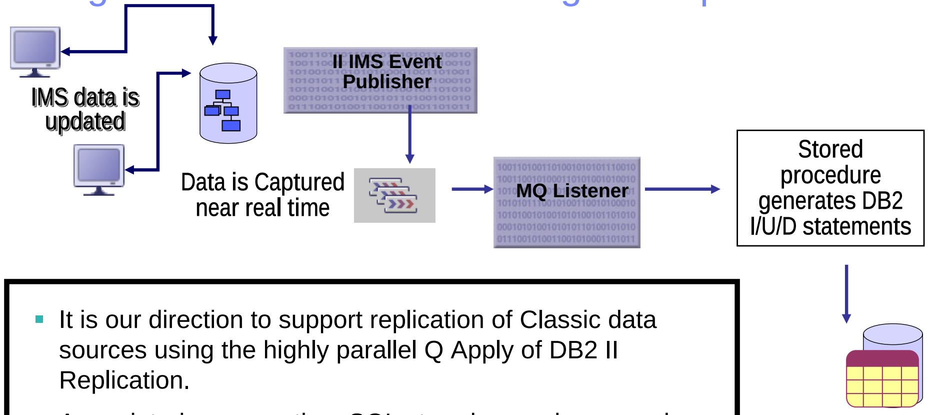
Here is another very common business issue – many heterogeneous systems exist within an enterprise, with a customer potentially defined in all or some of these systems. When a customer calls in with a change of information – change of name, address, status – it is highly desirable that this change be made in every system in which this customer exists within the business. This is a potent customer satisfaction issue.

Using the event publisher capability, very disparate systems can be linked together and benefit by sharing such important events. Depicted here is the ability to capture an event that occurs in an IMS database and publish that event to both a second IMS database as well as a DB2 database, where each of these databases represent various factions of the business.



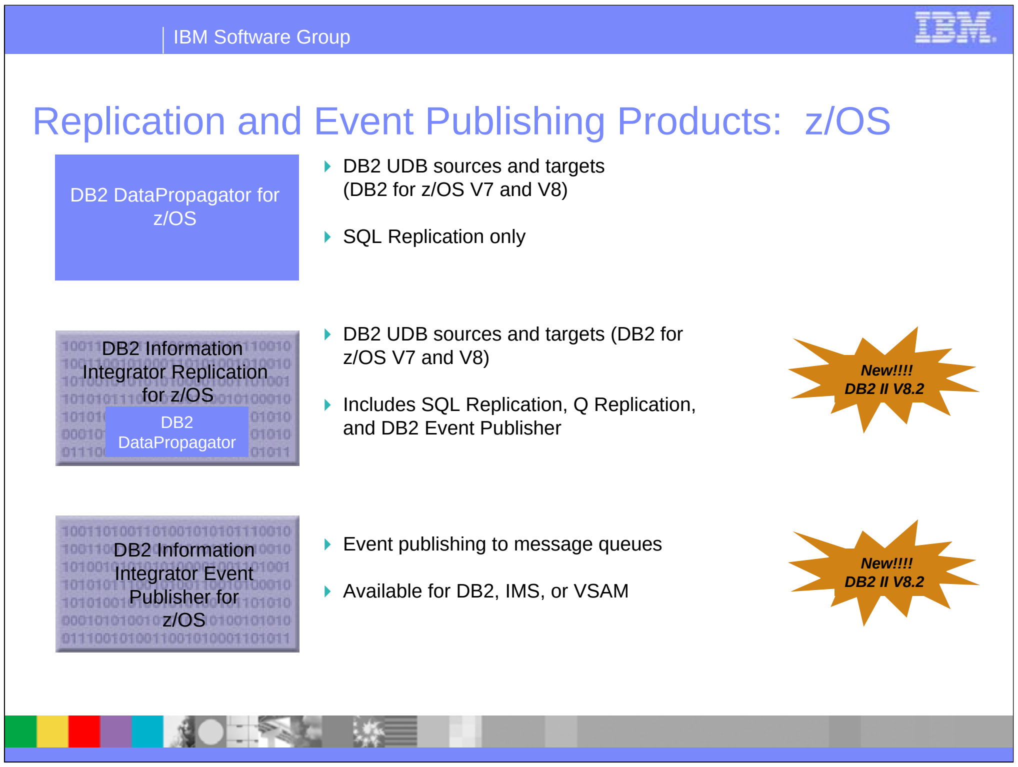
IBM Software Group

Using Classic II Event Publishing for Replication


IMS data is updated
II IMS Event Publisher
Data is Captured near real time
MQ Listener
Stored procedure generates DB2 I/U/D statements

- It is our direction to support replication of Classic data sources using the highly parallel Q Apply of DB2 II Replication.
- As an interim suggestion, SQL stored procedures can be used to apply the data captured through the II Classic Event Publisher

Until we offer a full blown Q Replication solution for classic sources, there can be temporary arrangements created using event publishing and user (or IGS) written programs. Look for our sample of this technology on developerworks. Note that this could be arranged in the reverse direction – capturing DB2 data and updating IMS or another legacy data source, even with SQL stored procs if using the Classic Federated feature.



IBM Software Group

Replication and Event Publishing Products: z/OS

DB2 DataPropagator for z/OS

- DB2 UDB sources and targets (DB2 for z/OS V7 and V8)
- SQL Replication only

DB2 Information Integrator Replication for z/OS
DB2 DataPropagator

- DB2 UDB sources and targets (DB2 for z/OS V7 and V8)
- Includes SQL Replication, Q Replication, and DB2 Event Publisher

New!!!! DB2 II V8.2

DB2 Information Integrator Event Publisher for z/OS

- Event publishing to message queues
- Available for DB2, IMS, or VSAM

New!!!! DB2 II V8.2

DB2 Data Propagator for z/OS 8.2 is the newest release of the product that has been available for many years under the same name. There are a small number of new features available in this release – most significant is the addition of table reconciliation utilities.

The new queue based replication architecture is in all cases marketed and sold as part of the DB2 Information Integrator brand.(the new version of which had the code name Masala) When purchased on z/OS, this package includes the SQL replication, the queue based replication, and the event publisher for DB2 on z/OS. Note that on z/OS this package does NOT include Websphere MQ, which must be purchased separately. The prerequisite version is 5.3.

It is also possible to purchase individually the Event Publishers for DB2, IMS, or VSAM. These are also part of the DB2 Information Integrator brand of products.

Highly recommend z/OS 1.4 or later.

Can run with OS/390 2.10 or later

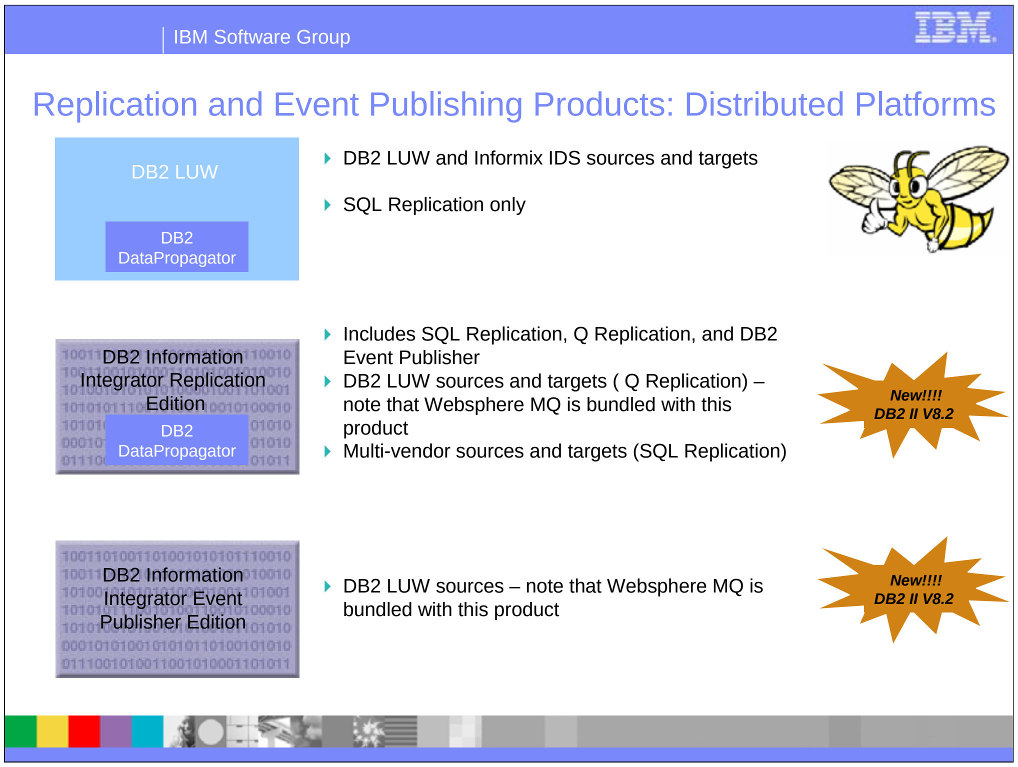
DB2 Universal Database for z/OS and OS/390 Version 7.1or later with PQ85495

WebSphere MQ for z/OS Version 5.3.1

XML Toolkit for z/OS and OS/390 Version 1.4.0 (for Q replication and Event publishing)

DB2 Administration Server (DAS) for z/OS (Replication Center)

390 Enablement (Replication Center)


IBM Software Group

Replication and Event Publishing Products: Distributed Platforms

DB2 LUW

DB2 DataPropagator

- DB2 LUW and Informix IDS sources and targets
- SQL Replication only

DB2 Information Integrator Replication Edition

DB2 DataPropagator

- Includes SQL Replication, Q Replication, and DB2 Event Publisher
- DB2 LUW sources and targets (Q Replication) – note that Websphere MQ is bundled with this product
- Multi-vendor sources and targets (SQL Replication)

New!!!! DB2 II V8.2

DB2 Information Integrator Event Publisher Edition

- DB2 LUW sources – note that Websphere MQ is bundled with this product

New!!!! DB2 II V8.2

There has been some confusion is this space and this slide attempts to help sort out this confusion. The newest release of DB2 is now announced as V 8.2, and had the code named “Stinger”. There has been a lot of well deserved excitement about this release, and included in the buzz has been the new queue based replication. This new replication will actually be marketed and sold as part of the DB2 Information Integrator brand (which has the code name Masala). These two products will be available at the same time, and Q Replication leverages DB2 V8.2. Q Replication only works with an 8.2 release of DB2 LUW.

There are a number of DB2 II packages that include replication – it is available in the standard, advanced, and replication editions. This package includes the SQL replication, the queue based replication, and the event publisher for DB2 LUW. Note that all of these multiplatform packages DO include Websphere MQ 5.3 in the bundle.

It is also possible to purchase individually the Event Publisher for DB2 LUW – this is the DB2 Information Integrator Event Publisher Edition.

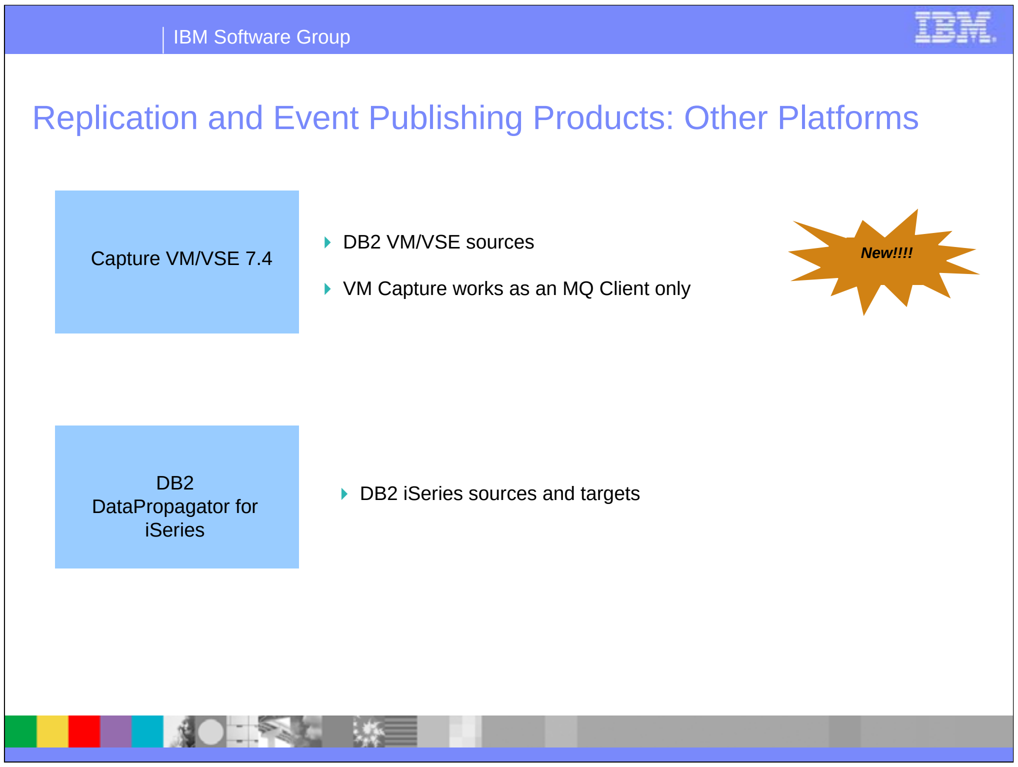
DB2 Universal Database for LUW V8.2

Q Replication inherits prerequisites of the database

WebSphere MQ Version 5.3.1 is included in the DB2 Information Integrator Package

Currently required that MQ Server be co-located with Capture and Apply components (will allow for client access later)

MQ Server does not run on all 64 bit platforms



Replication and Event Publishing Products: Other Platforms

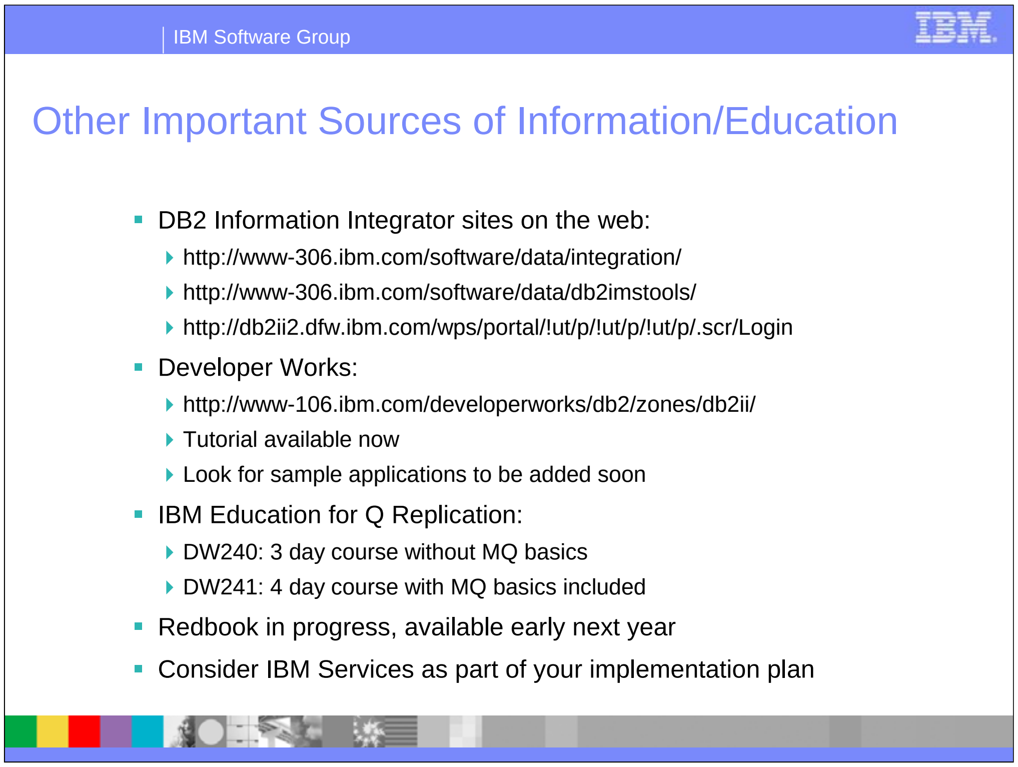
Capture VM/VSE 7.4

- DB2 VM/VSE sources
- VM Capture works as an MQ Client only

New!!!!

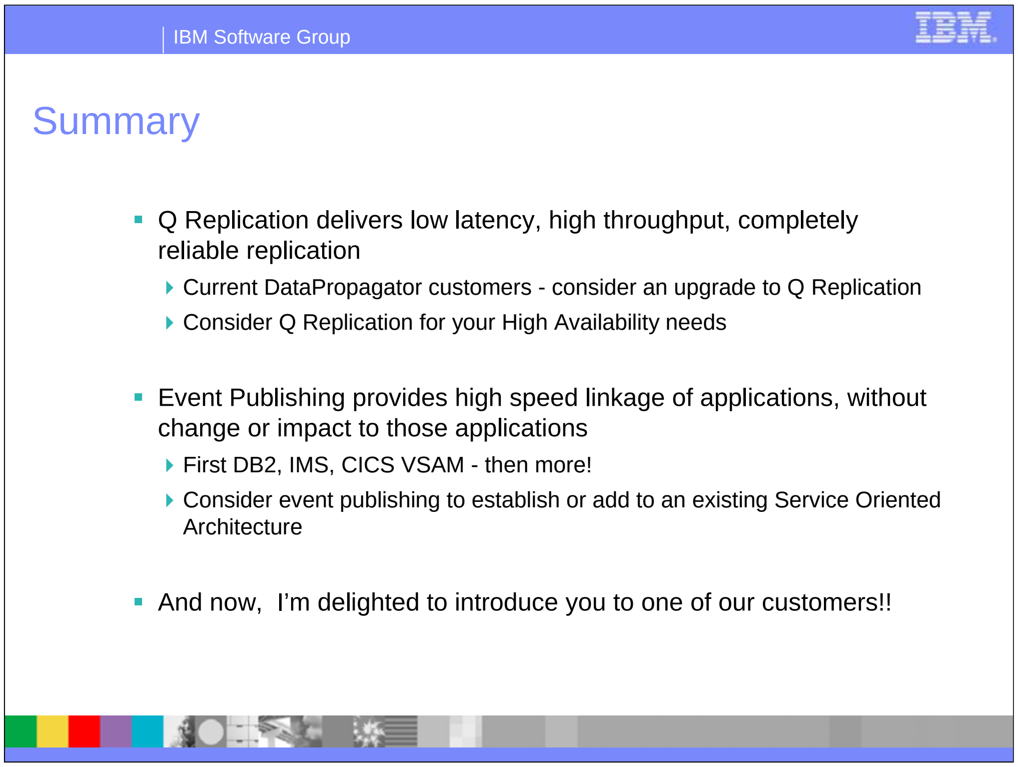
DB2 DataPropagator for iSeries

- DB2 iSeries sources and targets



Other Important Sources of Information/Education

- DB2 Information Integrator sites on the web:
  - http://www-306.ibm.com/software/data/integration/
  - http://www-306.ibm.com/software/data/db2imstools/
  - http://db2ii2.dfw.ibm.com/wps/portal/!ut/p/!ut/p/!ut/p/.scr/Login
- Developer Works:
  - http://www-106.ibm.com/developerworks/db2/zones/db2ii/
  - Tutorial available now
  - Look for sample applications to be added soon
- IBM Education for Q Replication:
  - DW240: 3 day course without MQ basics
  - DW241: 4 day course with MQ basics included
- Redbook in progress, available early next year
- Consider IBM Services as part of your implementation plan


IBM Software Group

Summary

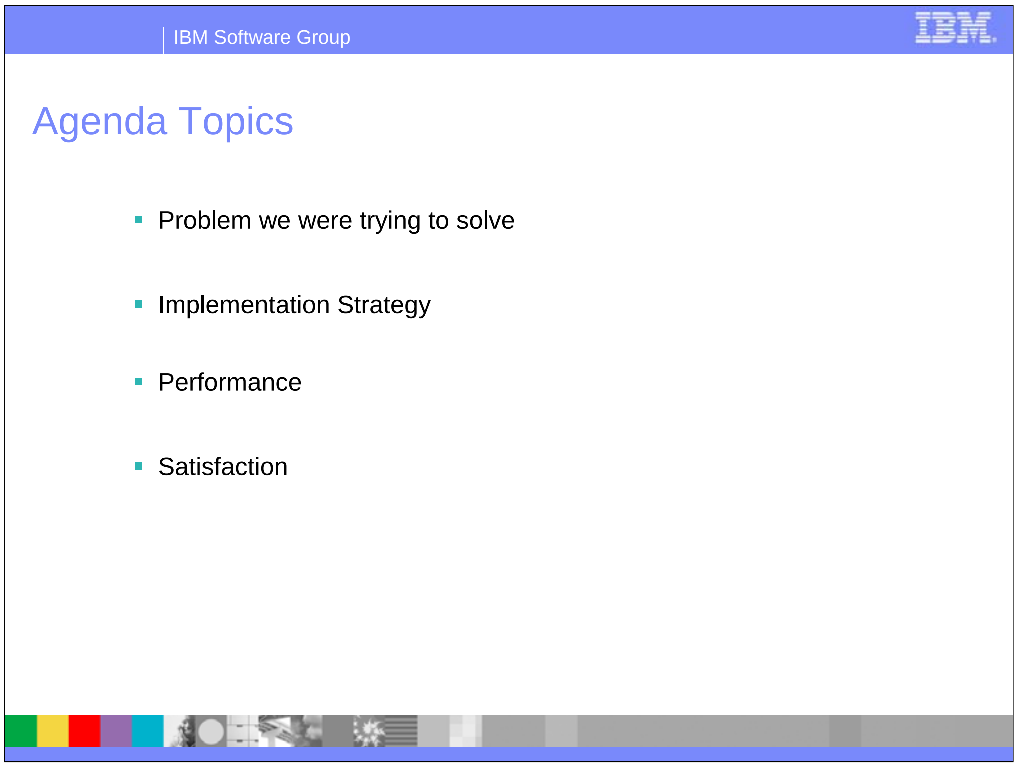
- Q Replication delivers low latency, high throughput, completely reliable replication
  - Current DataPropagator customers - consider an upgrade to Q Replication
  - Consider Q Replication for your High Availability needs
- Event Publishing provides high speed linkage of applications, without change or impact to those applications
  - First DB2, IMS, CICS VSAM - then more!
  - Consider event publishing to establish or add to an existing Service Oriented Architecture
- And now, I'm delighted to introduce you to one of our customers!!

Thank you for your time !!!

Calvin Petty

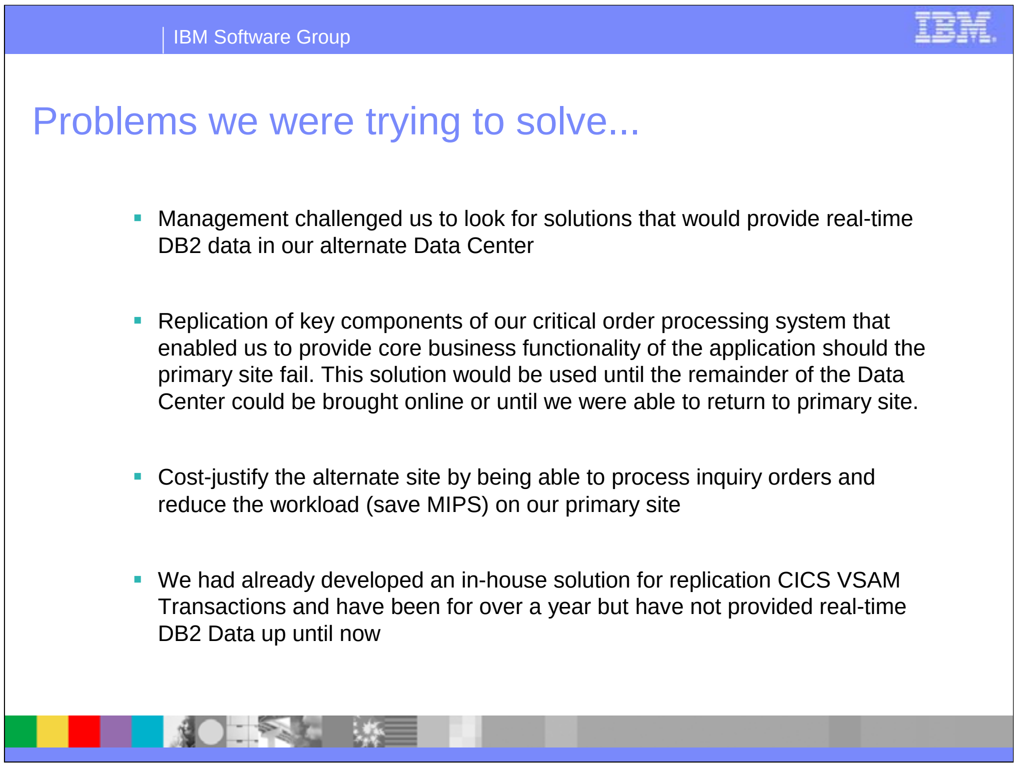
Email: Calvin.Petty@fmr.com

ON DEMAND BUSINESS™



Agenda Topics

- Problem we were trying to solve
- Implementation Strategy
- Performance
- Satisfaction

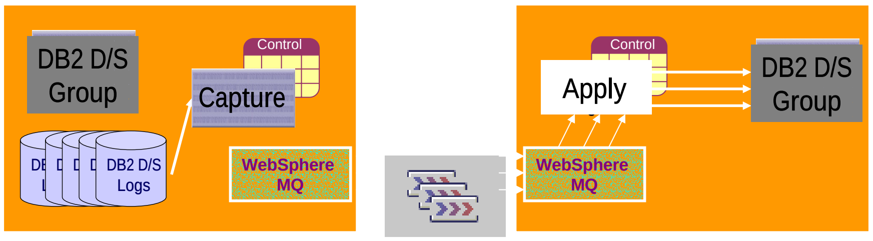


Problems we were trying to solve...

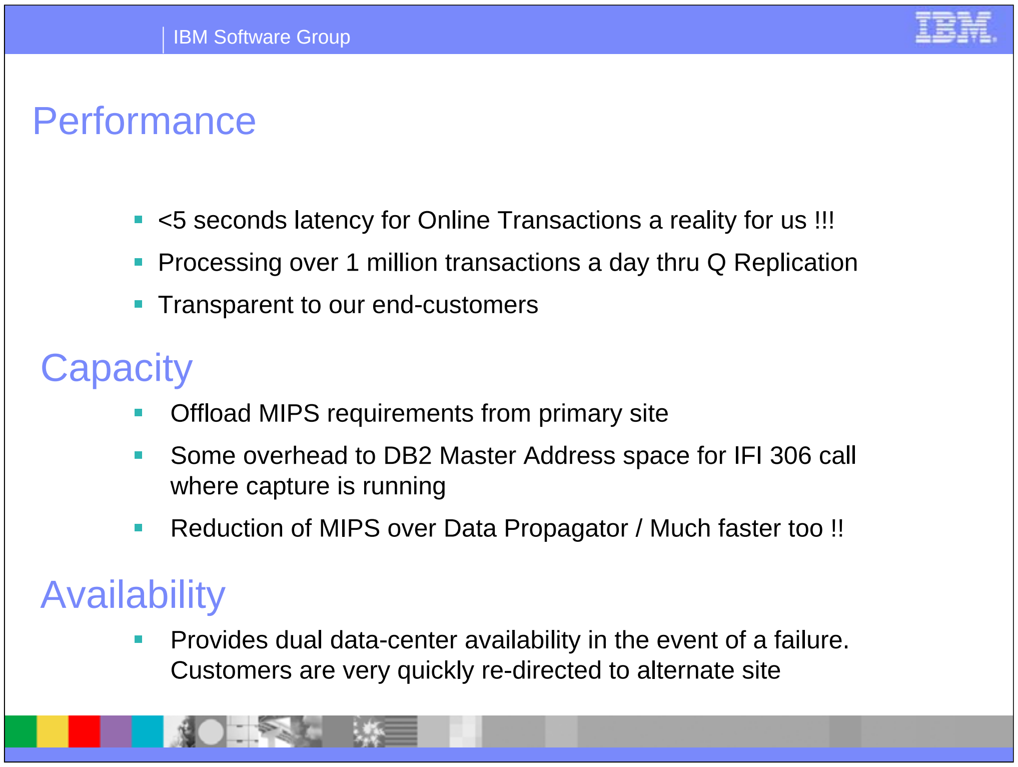
- Management challenged us to look for solutions that would provide real-time DB2 data in our alternate Data Center
- Replication of key components of our critical order processing system that enabled us to provide core business functionality of the application should the primary site fail. This solution would be used until the remainder of the Data Center could be brought online or until we were able to return to primary site.
- Cost-justify the alternate site by being able to process inquiry orders and reduce the workload (save MIPS) on our primary site
- We had already developed an in-house solution for replication CICS VSAM Transactions and have been for over a year but have not provided real-time DB2 Data up until now

Implementation Strategy

Unidirectional



- DB2 Datasharing group at our primary site processing update transactions
- Updates are transmitting to our alternate site upon commit
- Customers are performing inquires at our alternate site using real-time data from both DB2 and VSAM
- DB2 replication is 24/7



Performance

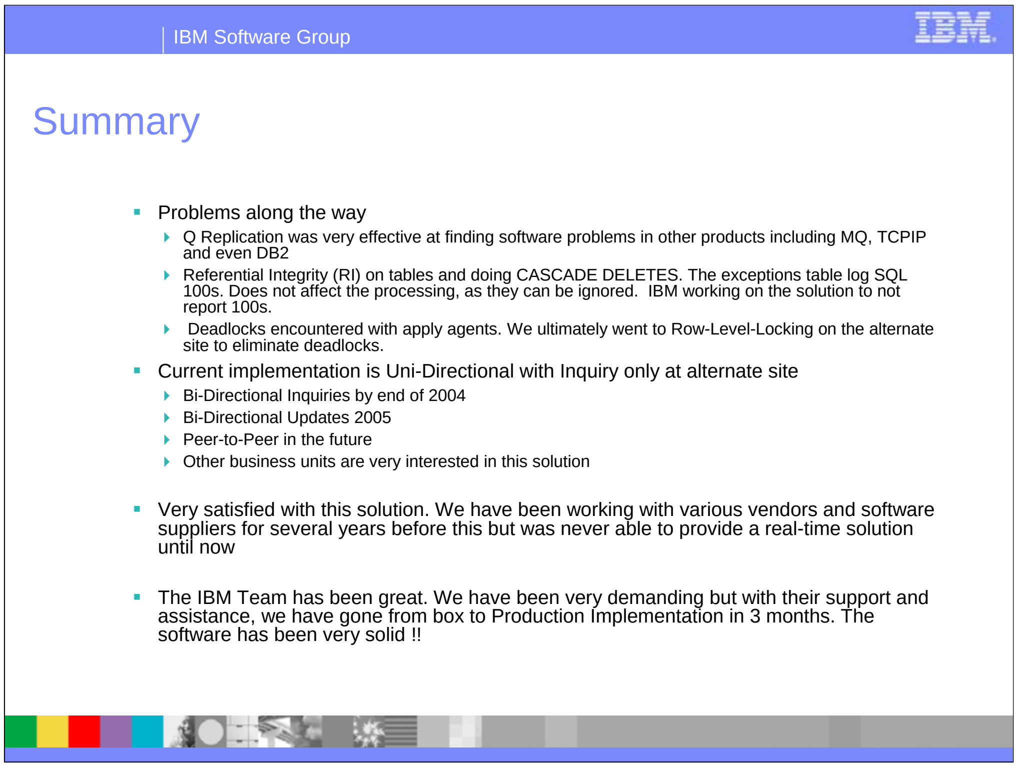
- <5 seconds latency for Online Transactions a reality for us !!!
- Processing over 1 million transactions a day thru Q Replication
- Transparent to our end-customers

Capacity

- Offload MIPS requirements from primary site
- Some overhead to DB2 Master Address space for IFI 306 call where capture is running
- Reduction of MIPS over Data Propagator / Much faster too !!

Availability

- Provides dual data-center availability in the event of a failure. Customers are very quickly re-directed to alternate site



IBM Software Group

Summary

- Problems along the way
  - Q Replication was very effective at finding software problems in other products including MQ, TCPIP and even DB2
  - Referential Integrity (RI) on tables and doing CASCADE DELETES. The exceptions table log SQL 100s. Does not affect the processing, as they can be ignored. IBM working on the solution to not report 100s.
  - Deadlocks encountered with apply agents. We ultimately went to Row-Level-Locking on the alternate site to eliminate deadlocks.
- Current implementation is Uni-Directional with Inquiry only at alternate site
  - Bi-Directional Inquiries by end of 2004
  - Bi-Directional Updates 2005
  - Peer-to-Peer in the future
  - Other business units are very interested in this solution
- Very satisfied with this solution. We have been working with various vendors and software suppliers for several years before this but was never able to provide a real-time solution until now
- The IBM Team has been great. We have been very demanding but with their support and assistance, we have gone from box to Production Implementation in 3 months. The software has been very solid !!