

#SHAREorg

**Free MQ! MQ Clients and what you can do with them**

**Mark Taylor**  
*marke\_taylor@uk.ibm.com*

WMQ Technical Strategy, IBM Hursley

Session 11510



SHARE  
Technology - Connections - Results

SHARE  
in Anaheim  
2012

CSS: F S

**Agenda**

- What are the MQ clients ?
- The MQ client and how it works
- How to connect a client to a server
  - Channel Table Configuration
- What facilities are available to clients
  - Transactions
  - Global Transactions
  - Security
  - Exits

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

SHARE  
Technology - Connections - Results

SHARE  
in Anaheim  
2012

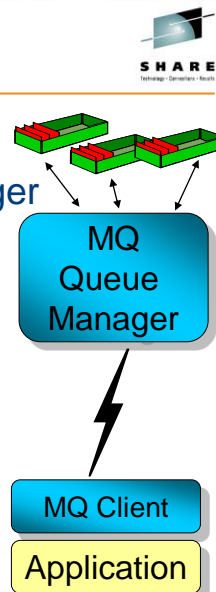
## What is a client ?

- Allows access to messaging API on a different machine than the queue manager

- Simpler administration
- Same programming capabilities (almost)
- Cheaper
  - Free in most cases

- However.....

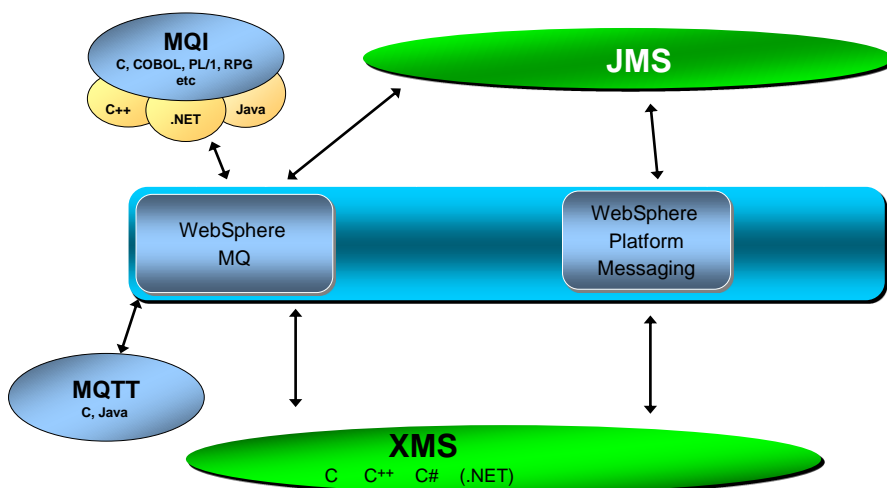
**No network – No messaging.**



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

**SHARE**  
in Anaheim  
2012

## Messaging Clients



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

**SHARE**  
in Anaheim  
2012

## Which client to use

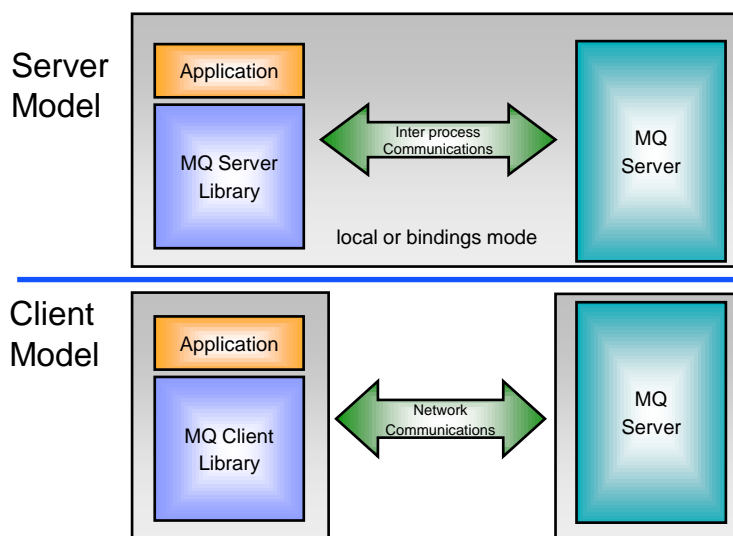


- Power of MQI vs Portability of JMS
  - JMS does not tie you to a provider (99% portable)
  - JMS available for non-JAVA languages in XMS
    - XMS is IBM specific though
- How important is speed ?
  - C tends to be faster than Java
  - MQI tends to be faster than JMS/XMS
- Mobile/small environments
  - Consider MQTT

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



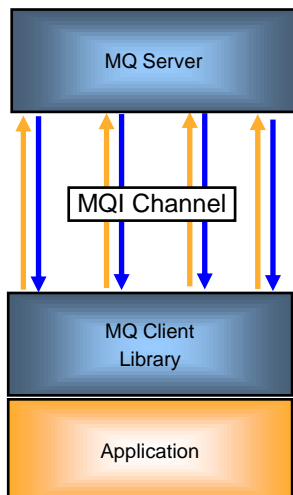
## What is an MQ Client ?



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## How does a client work ?



- Requires network access
- Each MQI Call shipped to server
- Response returned to application

### MQI Calls

MQCONN	MQCONNX	MQDISC
MQOPEN	MQCLOSE	MQSUB
MQPUT	MQPUT1	MQGET
MQCB	MQCTL	
MQINQ	MQSET	
MQCMIT	MQBACK	

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## How to install a client



1. **Install a MQ client and MQ server system**  
Install MQ server using the SERVER CD ROM  
Install the MQ client using the CLIENT CD ROM
2. **Install MQ client and server on the same machine**  
Install MQ server from SERVER CD ROM  
and select MQ clients you wish to install
3. **Install MQ client from SupportPacs site**  
Download SupportPac  
Extract and run installation program



See the platform Quick Beginnings for specific details

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## What about Licensing ?



- Installable clients can be downloaded for free
  - Available on many platforms
- Client attachment feature required for z/OS
- Extended Transactional (XA) Clients are **now** free



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Building a client application



- Compile your application as you would for local application
- Make sure you link your application with CLIENT libraries
  - libmqic\* for "C" applications on UNIX systems
  - mqic32.lib for "C" applications on Windows
  - imqb23\* imqc23\* for "C++" applications
- Take care when linking threaded programs
  - e.g. libmqic\_r.a for AIX
- Ensure that the correct runtime libraries are available
  - e.g. mqic32.dll for Windows
- From V7.1 onwards, libmqm can also act as client
  - Support for multi-version installations

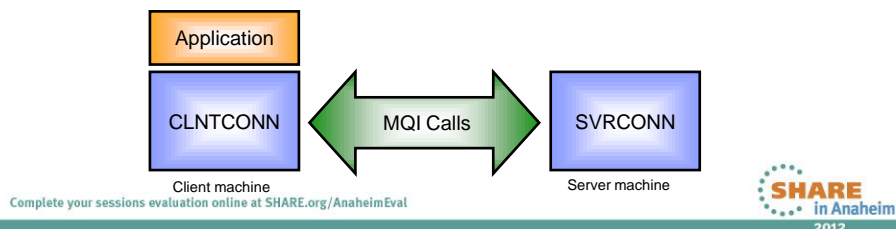
Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## How to connect a client to a server



- The client must be able to identify which channel it should use to communicate with the queue manager
- How to specify the client's connection to a queue manager:
  - Explicitly on the MQCONN verb
  - MQSERVER variable
  - Client channel tables
- Java client programs use either the MQEnvironment Java class or JNDI (using JMS) or explicit properties



## Using the MQSERVER variable



- The easiest way to define a client channel.
  - BUT has default CLNTCONN properties, ie.
    - No security, such as SSL
    - No exits
    - etc
- Takes precedence over channel tables
  - but is superseded by the use of the MQCNO structure.
- set MQSERVER=ChannelName/TransportType/ConnectionName
  - In Windows: use Control Panel -> System -> Advanced -> Environment Variables
  - In UNIX: export MQSERVER
- Examples:
  - MQSERVER=SYSTEM.DEF.SVRCONN/TCP/127.0.0.1
  - MQSERVER=SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(1415)
  - MQSERVER=SYSTEM.DEF.SVRCONN/TCP/JUPITER.SOLAR.SYSTEM.UNI
  - MQSERVER=SYSTEM.DEF.SVRCONN/LU62/BOX99

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Channel definition tables



- A channel definition table is:
  - A binary file (not editable by a user)
  - Created by RUNMQSC (or other MQ mechanism) as AMQCLCHL.TAB (by default) when client channels are defined
    - Use CSQUTIL MAKECLNT function on z/OS
  - Located in directory (by default):
    - <mq root>\qmgrs\QMGRNAME\@ipcc (Windows)
    - <mq root>/qmgrs/QMGRNAME/@ipcc (UNIX)
  - Read by the client if no MQSERVER variable defined and MQCONN options are not used

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

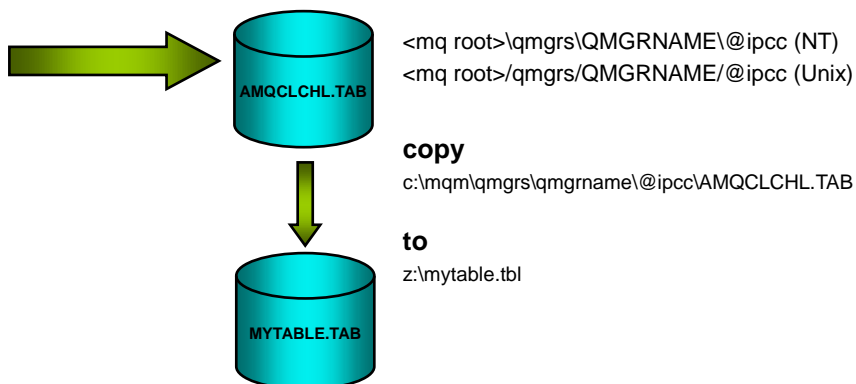


## How do I create and deploy a channel table ?



RUNMQSC

```
def chl(...) chltype(clntconn) ....
```



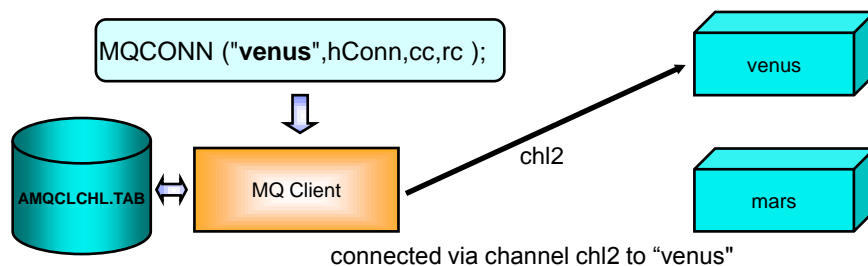
Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Using Channel Definition Tables: Example 1



- How is the QMNAME client channel attribute used?
  - def chl(chl1) chltype(clntconn) trptype(tcp) conname(host1) qmname(mars)
  - def chl(chl2) chltype(clntconn) trptype(tcp) conname(host2) qmname(venus)



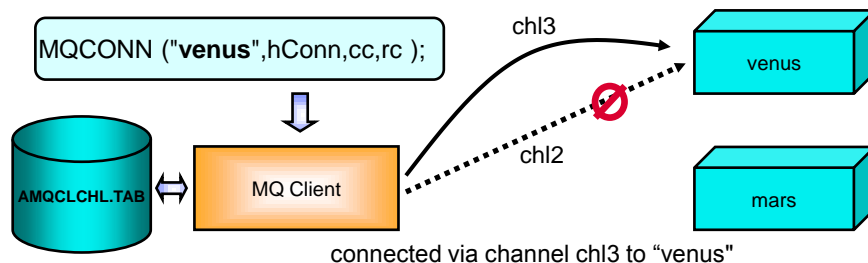
Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Using Channel Definition Tables: Example 2



- Multiple routes to the same Queue Manager
  - def chl(chl1) ....trptype(tcp) conname(host1) qmname(mars)
  - def chl(chl2) ....trptype(tcp) conname(tokenring) qmname(venus)
  - def chl(chl3) ....trptype(tcp) conname(ethernet) qmname(venus)
  - def chl(chl4) ....trptype(tcp) conname(dialup) qmname(venus)



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



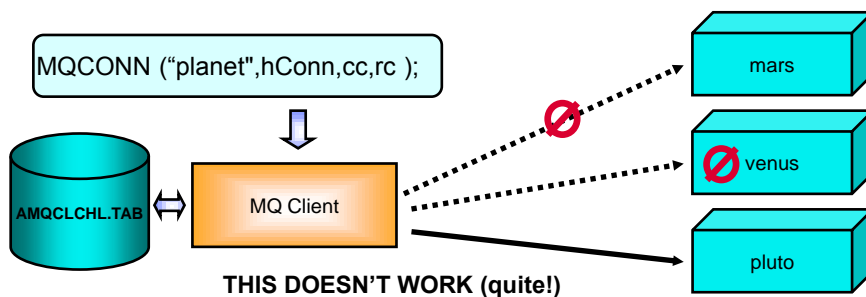


## Using Channel Definition Tables: Example 3



### How do we have back-up Queue Managers ?

- `def chl(chl1) ....trdtype(tcp) conname(ip.mars) qmname(planet)`
- `def chl(chl2) ....trdtype(tcp) conname(ip.venus) qmname(planet)`
- .....
- `def chl(chl5) ....trdtype(tcp) conname(ip.pluto) qmname(planet)`



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

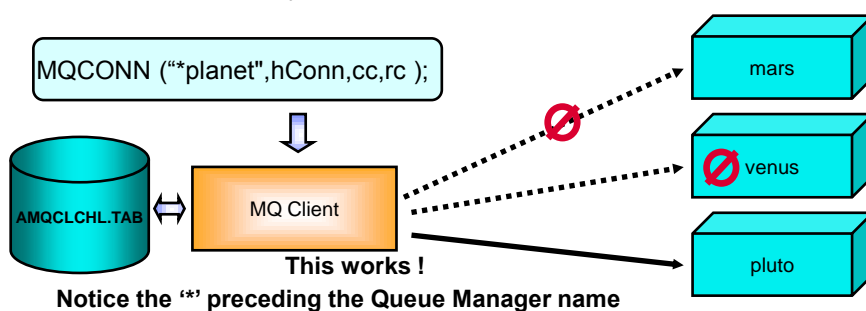


## Using Channel Definition Tables: Example 4



### How do we have back-up Queue Managers ?

- `def chl(chl1) ....trdtype(tcp) conname(ip.mars) qmname(planet)`
- `def chl(chl2) ....trdtype(tcp) conname(ip.venus) qmname(planet)`
- .....
- `def chl(chl5) ....trdtype(tcp) conname(ip.pluto) qmname(planet)`



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

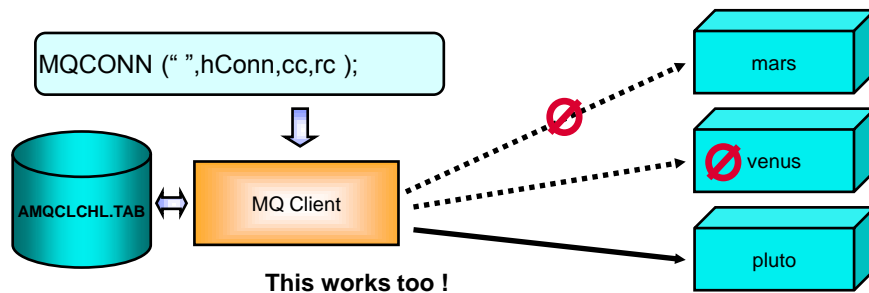


## Using Channel Definition Tables: Example 5



### How do we have back-up Queue Managers ?

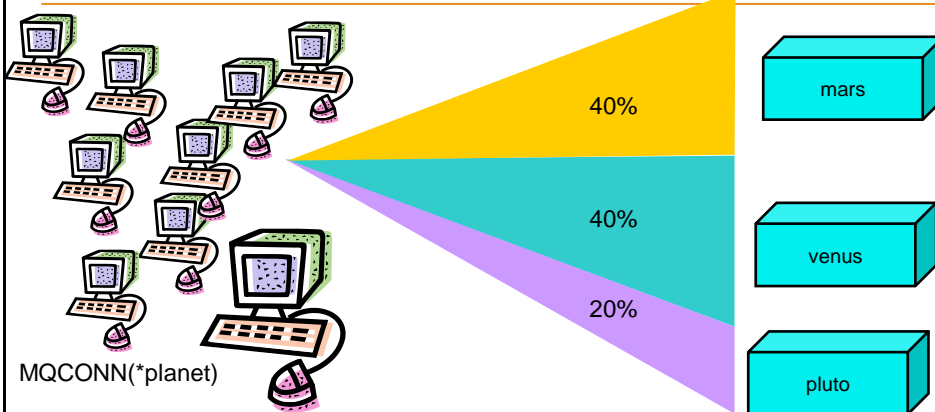
- def chl(chl1) ....trptype(tcp) conname(ip.mars) qmname()
- def chl(chl2) ....trptype(tcp) conname(ip.venus) qmname()
- .....
- def chl(chl5) ....trptype(tcp) conname(ip.pluto) qmname()



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Workload Balancing client connections



Name	CHLTYPE	TRPTYPE	CONNAME	QMNAME	CLNTWGHT	AFFINITY
chl1	CLNTCONN	TCP	ip.mars	planet	4	PREFERRED
chl2	CLNTCONN	TCP	ip.venus	planet	4	PREFERRED
chl3	CLNTCONN	TCP	ip.pluto	planet	2	PREFERRED

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



SHARE  
Technology · Services · Results

## Limiting client connections

Queue Manager

SVRCONN  
MAXINST(4)  
MAXINSTC(2)

Command Prompt - runmqsc TEST1

```
Starting MQSC for queue manager TEST1.  
DEFINE CHANNEL(SALES.CONNECT) CHLTYPE(SVRCONN)  
MAXINST(4) MAXINSTC(2)
```

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

2012

SHARE  
Technology · Services · Results

## Using MQCONN

MQCONN ( qmgr name, CNO, Hconn, cc, rc)

**MQCNO - Connection Options:**

```
MQCHAR4  StructId;           /* Structure identifier          */  
MQLONG   Version;            /* Structure version number     */  
MQLONG   Options;            /* Options that control the action of MQCONN */  
...  
MQLONG   ClientConnOffset;    /* Offset of MQCD structure for client connection */  
MQPTR     ClientConnPtr;      /* Address of MQCD structure for client connection */  
...
```

**MQCD - Channel Definition**

```
...  
MQCHAR   ChannelName[20];      /* Channel definition name      */  
...  
MQCHAR   ConnectionName[264]; /* Connection name              */  
...
```

If used, overrides MQSERVER and CHANNEL tables

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

2012

## Using MQCONN



```
MQCD cd = {MQCD_CLIENT_CONN_DEFAULT};

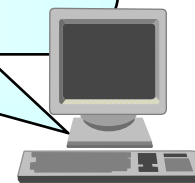
cno.Version = MQCNO_VERSION_2; // CD ignored if CNO not V2 or greater

cno.ClientConnPtr = &cd;

strcpy(cd.ChannelName,"SYSTEM.DEF.SVRCONN");

strcpy(cd.ConnectionName,"VENUS.SOLAR.SYSTEM.UNI");

MQCONN ( "", &cno, &hQm, &cc, &rc)
```



Complete your session evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

2012



## MQ Cloud Support: Pre-Connect Exit



- Supports movement by some to “Utility Compute”, Private Cloud configs, etc.
  - Rapid provision of applications allied with need to further decouple Client/Server connectivity
  - Server applications might move location – new addresses or queue managers
- MQ Client connects to a “service” rather than specific Queue Manager
- Can transparently change location of MQ server-side applications
  - No client code changes needed
  - No configuration files need to be updated at client machine
  - JMS/XMS applications already do this via JNDI lookup

Complete your session evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

CSD.F.5

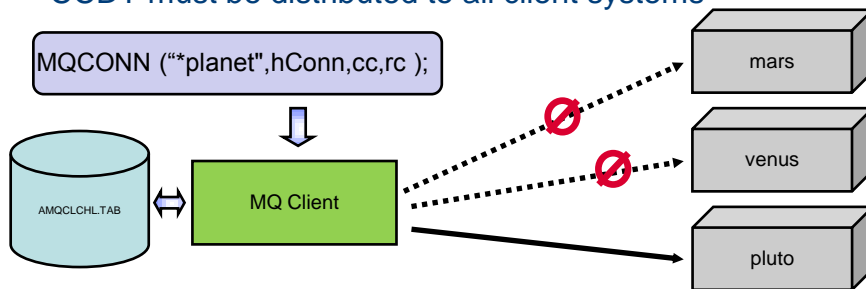


2012

## Dynamic Connections: CCDT



- How it used to be done ...
- The CCDT is used to select a queue manager from a list
  - Based on a pseudo-queue manager name prefixed with "\*"
  - CCDT is a locally-accessible file
- CCDT must be distributed to all client systems



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

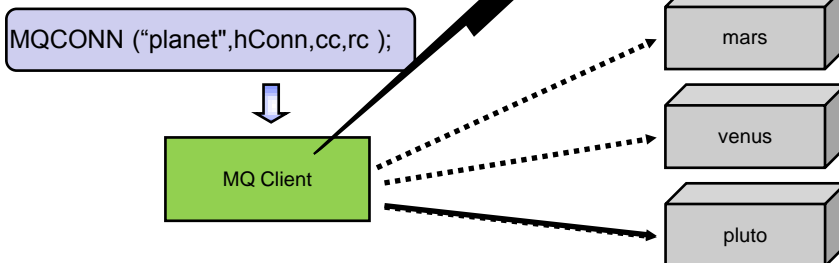
CBS F



## Dynamic Connections: Pre-Connect Exit



- Look up in a directory such as LDAP
- Make choice based on any accessible criteria
  - Is system running?
  - Workload?
  - Where is the application running now?
- No "\*" needed



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

CBS F



## Debugging Connection problems



- Check the error logs!
  - Server error log <root>\qmgrs\<QM>\errors\AMQERR01.LOG
  - Client error log <root>\errors\AMQERR01.LOG
- Double check the MQSERVER variable
- Does the amqsputc sample work?
- Is the network working ?
  - Can you "tcp ping" the host?
- Is there an MQ listener running?
- Is the channel table specified correctly
  - Do the environment variables point to the right place?

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

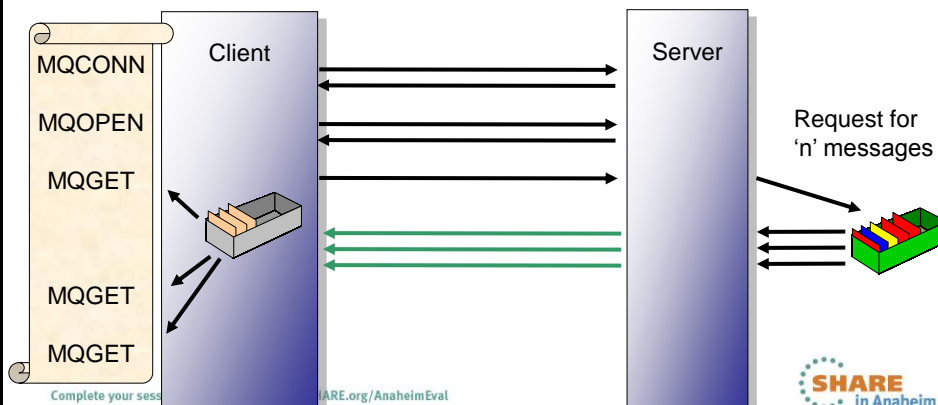


## WMQ V7 Enhancements for Client Performance Read Ahead



"Read Ahead" for Receiving Messages/Publications:

- Messages sent to a client in advance of MQGET, queued internally
- Administrative choice – no application changes needed
- Higher performance in client

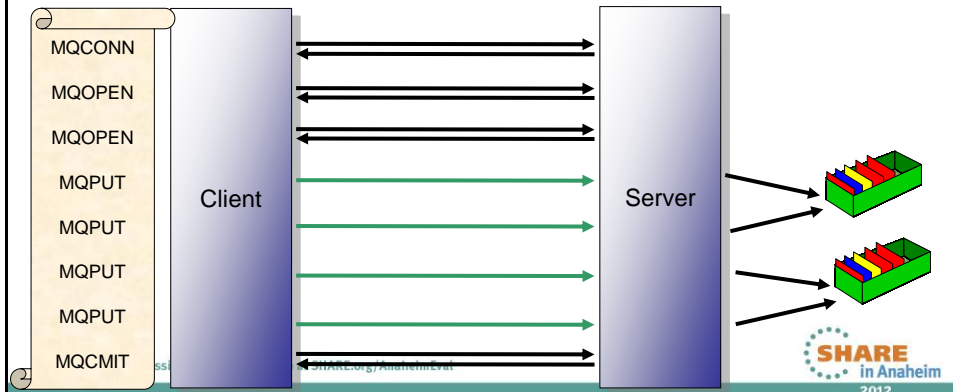


## WMQ V7 Enhancements for Client Performance Asynchronous Put



"Asynchronous Put" for Sending/Publishing Messages:

- Application can indicate it doesn't want to wait for the real return code
  - Maybe look for return code later – MQSTAT verb
- Maintains transactional semantics
- Higher performance in client

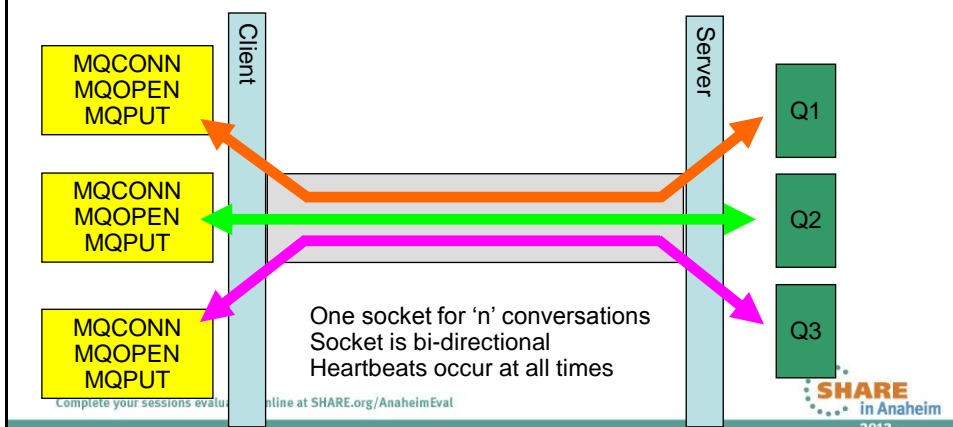


## WMQ V7 Enhancements Sharing Conversations



### Controlled by SHARECNV channel attribute

- 0 All sharing is off ; channel operates in MQ V6 mode
- 1 Sharing is off but channel still operates in MQ V7 mode
- Many Sharing up to negotiated value : Default 10



## Programming Considerations



- Take care when specifying the queue manager name on MQCONN if using client channel definition table...
- Most MQI calls are SYNCHRONOUS and tend to be slower than in a server environment.
- Always be prepared for MQRC\_CONNECTION\_BROKEN.
- Always code MQGMO\_FAIL\_IF QUIESCING.
- For optimum performance don't use really short lived connections (MQCONNs)
- Carefully code MQWI\_UNLIMITED on MQGET calls.
- Use Asynchronous MQPUT and Read Ahead if appropriate

**As always If you don't want to lose messages, code MQ\*\_SYNCPOINT on MQGET and MQPUT calls then issue MQCMIT**

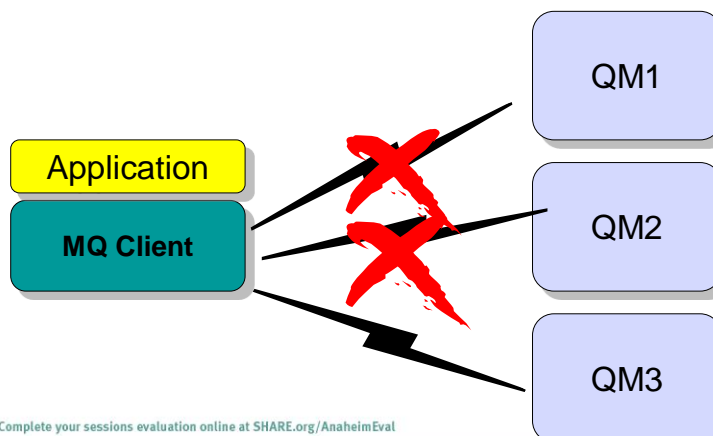
Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Automatic Client Reconnection



- Client library provides reconnection logic on detection of a failure
- Tries to hide queue manager failures by restoring state automatically
  - Can reconnect to the same or different QMgr
  - Re-opens queues, re-establishes subscriptions ...



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)





## Automatic Client Reconnection: What causes reconnection?



- Only explicit ends or failures
  - Communications failure
  - Queue Manager or Listener failure
  - STOP CONN
  - endmqm -s or endmqm -r
- The following will not cause reconnect
  - STOP CHANNEL
  - Any other endmqm

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Automatic Reconnection: MQI Considerations

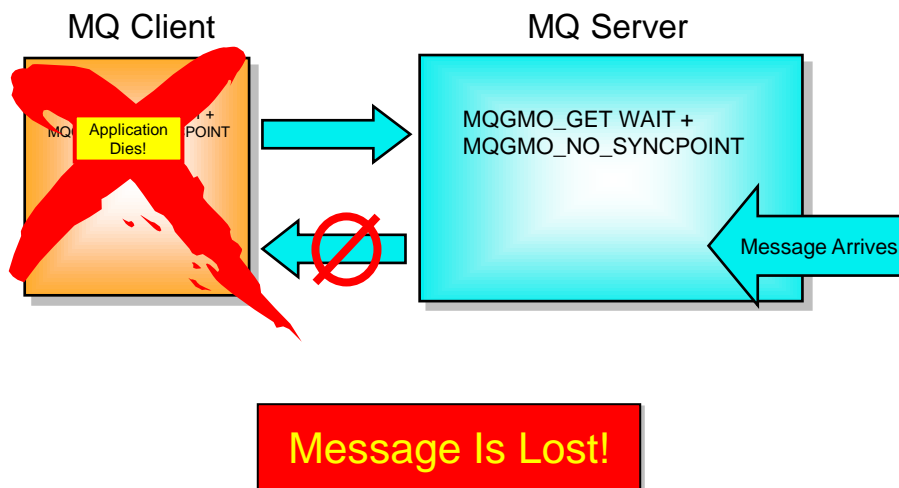


- Some MQI options will fail if you have reconnection enabled
  - Using MQPMO/MQGMO\_LOGICAL\_ORDER, MQGET gives MQRC\_RECONNECT\_INCOMPATIBLE
- New MQCONN options
  - MQCNO\_RECONNECT
  - MQCNO\_RECONNECT\_Q\_MGR
  - MQCNO\_RECONNECT\_DISABLED
- MQPUT of PERSISTENT message outside of syncpoint
  - May return MQRC\_CALL\_INTERRUPTED
- Event handler notified of reconnection 'events'
- MQSTAT may return only a 'subset' of information
  - Reconnection may have occurred during a message sequence
  - Can be used to query reconnection status

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



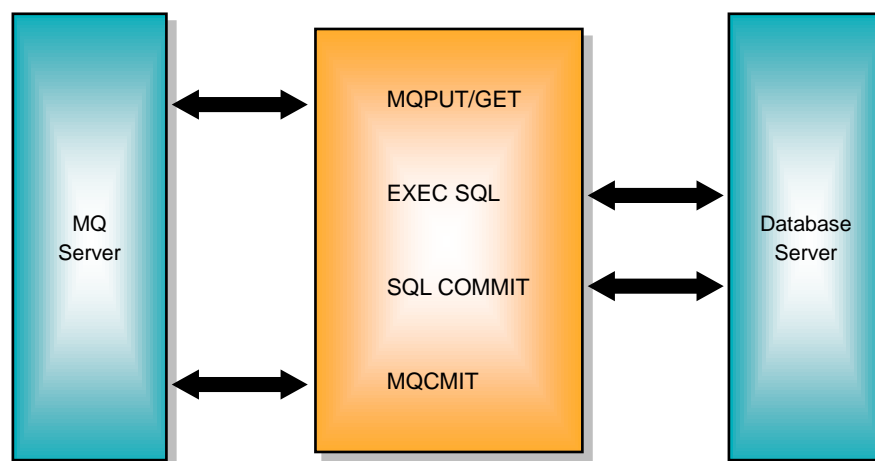
## Transactions



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Global Transactions



Multiple Resource Managers involved in the transaction

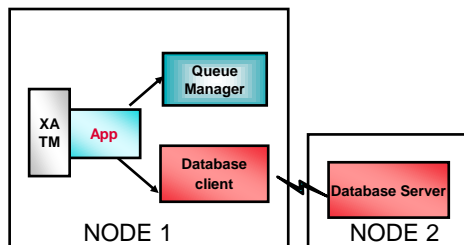
Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



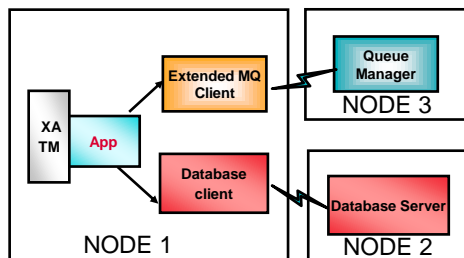
## Extended Transactional Client



### Local Application



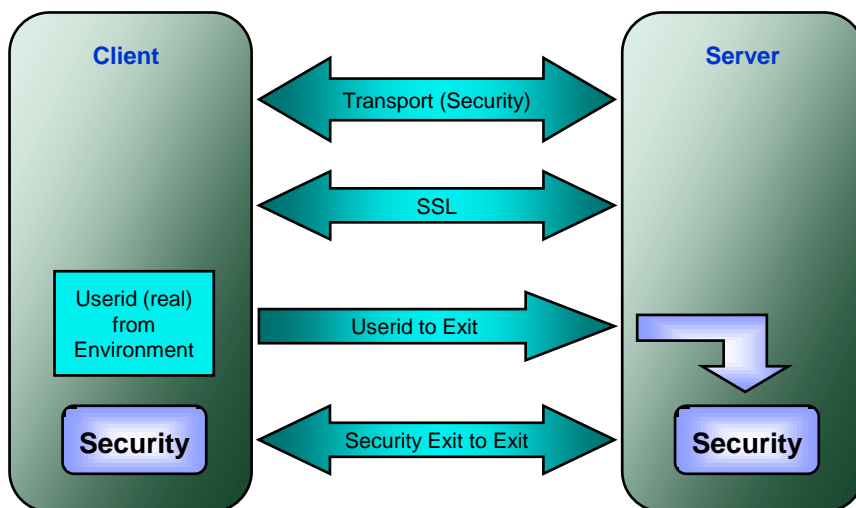
### Extended Transactional Client



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

2012

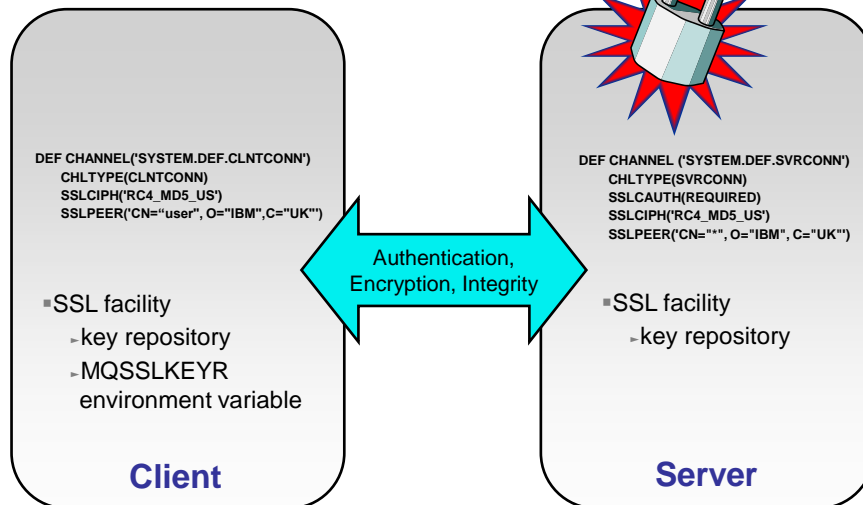
## Client Security



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

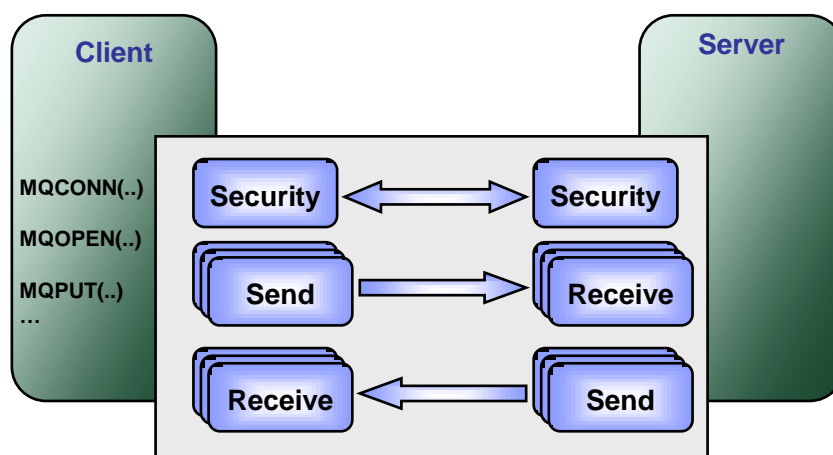
2012

## Client Security - SSL



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

## Exits



Message exits and Retry Exits are not applicable

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Summary



- Clients are a simple, low administration and cheap way of providing queuing throughout your network.
- Consider which client to use based on
  - Programming Language required (C,Java,C#, C++)
  - Programming model required (MQI vs JMS)
  - Performance
- Client applications can do the same as local applications
  - However, no network - no queuing

Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)



## Please submit evals – Session 11510



Complete your sessions evaluation online at [SHARE.org/AnaheimEval](http://SHARE.org/AnaheimEval)

