Telcordia Technologies

Performance from Experience

IMS TECHNICAL CONFERENCE

Customer Experience: Using the Latest and Greatest in OTMA

Session E91

Steve Nathan

stephen.nathan@telcordia.com

An SAIC Company

RETURN TO INDEX

Disclaimer

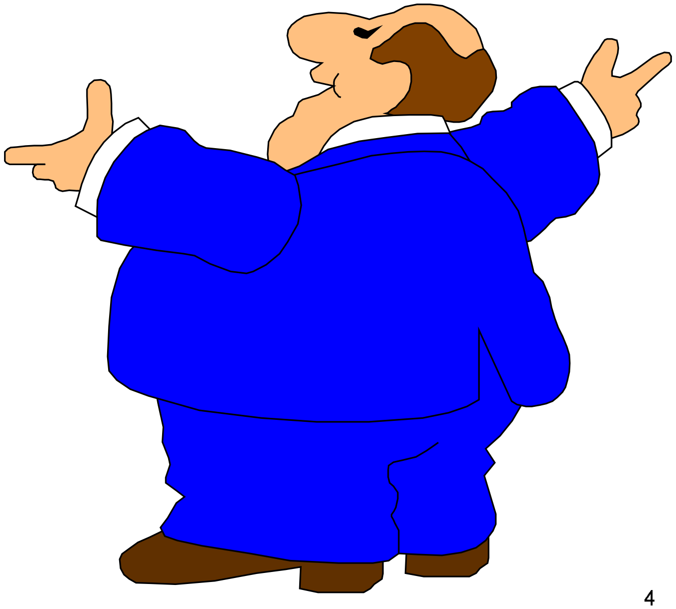
- The purpose of this presentation is to provide a technical perspective of Telcordia's experience using IMS.
- Although this document addresses certain IBM products, no endorsement of IBM or its products is expressed, and none should be inferred.
- Telcordia also makes no recommendation regarding the use or purchase of IMS products, any other IBM products, or any similar or comparable products.
- Telcordia does not endorse any products or suppliers. Telcordia does not recommend the purchase or use of any products by the participants.
- Each participant should evaluate this material and the products himself/herself.

Acknowledgements

- Special thanks to Jack Yuan and his team from IBM IMS OTMA development for all the enhancements in PQ32402 and all of the other enhancements in OTMA

Presentation Outline

- Introduction
- IMS OTMA Interface
- IMS OTMA Exits
- IMS OTMA Security
- Sources of Documentation



Telcordia Technologies

Performance from Experience

Introduction

- Telcordia has given several presentations about its experiences using IMS OTMA
- There have been many changes and enhancements in the last year
- This presentation will concentrate on what is new and exciting in OTMA

Introduction

- IMS/ESA 5.1 introduced the OTMA (Open Transaction Manager Access) feature
  - Activated by PN64987
  - PN87811 provides IMS re-sync support
- This feature uses the MVS cross-coupling facility (XCF) to send data to IMS from other MVS applications (OTMA clients)
  - No VTAM or TCP/IP is involved
- IMS 6.1 includes activation and re-sync in base code
- There are many other APARs for IMS OTMA
  - **Be completely up to date**

Telcordia Technologies

Performance from Experience

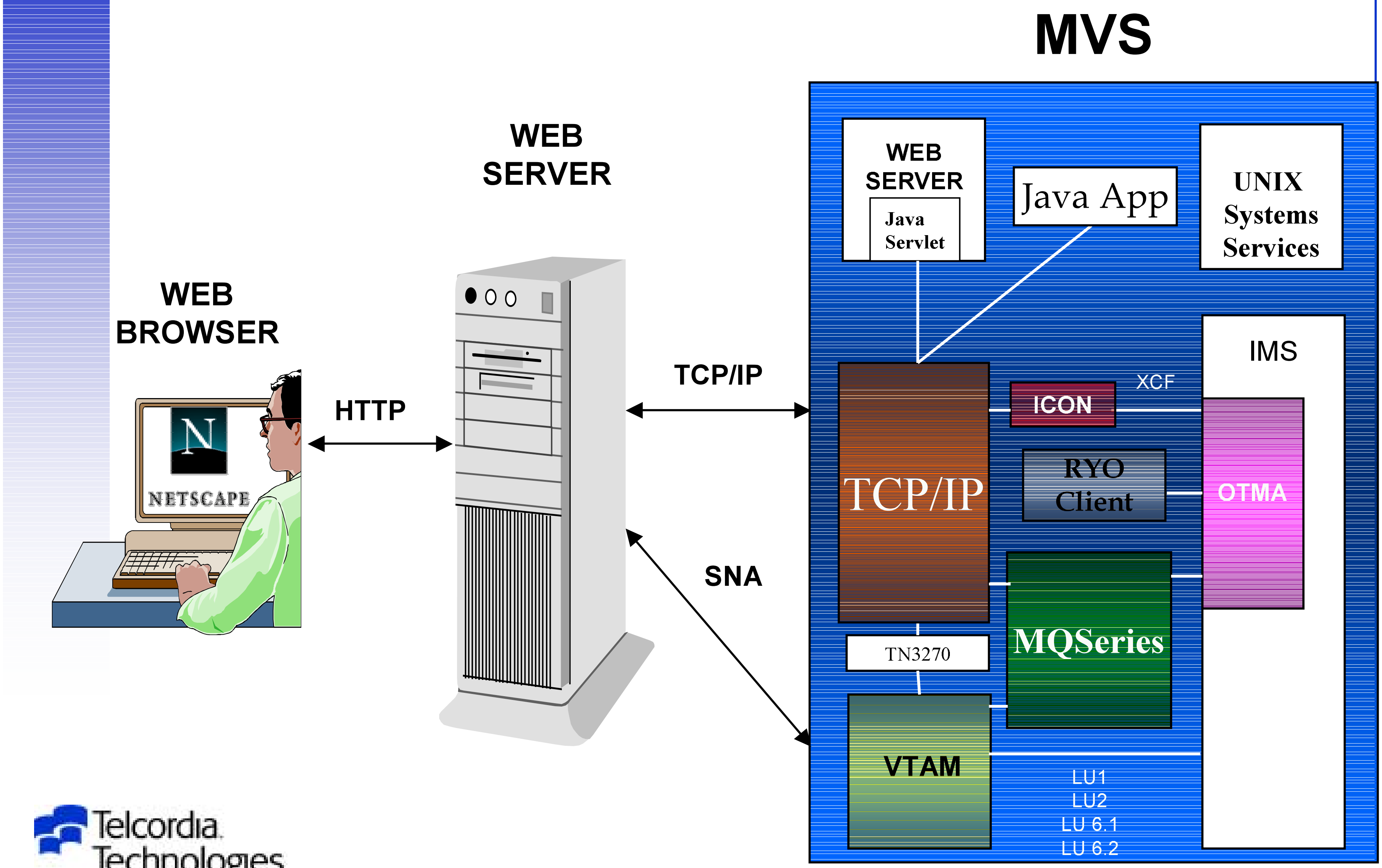
Introduction

- IMS Connect is an IBM provided OTMA client for TCP/IP
  - This was the IMS TCP/IP OTMA Connection (ITOC)
  - See Session E93
- MQSeries includes an IMS OTMA client
  - "MQSeries-IMS Bridge"
    - Allows existing IMS applications to access MQSeries messages without coding changes
  - See Session E92
- You can write your own OTMA client using the OTMA Callable Interface
  - See Session E93

Introduction

- There are other vendor supplied OTMA clients
- This is a partial list I found by searching the web
  - Tuxedo - BEA Systems
  - Component Broker - IBM
  - Orbix - Iona
  - Uniface - Compuware
  - TCP/IP - Interlink
  - SNA Server - Microsoft
  - AS/IMS (DCE) - IBM

Introduction



Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- Define IMS OTMA parameters in the IMS Control Region JCL or DFSPBxxx
  - OTMA=Y/N (autostart OTMA)
  - GRNAME=XCF group of IMS Control Region
  - APPLID1=XCF member name of IMS Control Region
  - OTMANM=XCF member name of IMS - APAR PQ12917
  - USERVAR=XCF member name of IMS - only if RSR or XRF
  - APPCASY=Y/N - APARS PQ17309 & PQ19330
    - This now has nothing to do with OTMA - it is APPC only
  - OTMAMD=Y/N - APAR PQ32402
    - Allow DFSYPRX0 to override OTMA output client if OTMA input
  - OTMASP=Y/N - APAR PQ32402
    - Set the default output TPIPE to SYNChronized if no DRU exit
  - OTMADB= (More Later)
  - OTMASE=C,F,N,P - APAR PQ30626 & PQ35615
    - Provide default OTMA security level

IMS OTMA Interface

- Define OTMA Destination Resolution Exit name for each client
  - Member DFSYDTx in IMS.PROCLIB
  - Can be overridden by OTMA client
- Set OTMA security via command
  - /SEC OTMA xxxx (NONE, PROFILE, CHECK, FULL)
- Start (and stop) OTMA via command
  - /STA OTMA (like /STA DC)
  - /STO OTMA (like /STO DC)
- There are no SYSGEN requirements for OTMA

IMS OTMA Interface

- The default OTMA security for IMS cold start is FULL
- If you bring IMS up cold with OTMA=Y and OTMA clients try to connect expecting NONE they will fail
  - See OTMA security considerations at the end of this presentation
  - This happens all the time in our test environment
- Recommendation
  - Use OTMA=N in the DFSPBxxx
  - During IMS cold start issue the following commands via automation
    - /SEC OTMA xxxx (NONE, PROFILE, CHECK, or FULL)
    - /STA OTMA
  - During IMS warm start issue the following command via automation
    - /STA OTMA
- This is not needed if you use the new OTMASE parameter

IMS OTMA Interface

- Tpipes (Transaction Pipes)
  - OTMA equivalent of LTERMs
  - Input Tpipe names are specified by the client
  - Asynchronous output Tpipe names
    - The destination in the CHNG call for modifiable ALTPCB's
    - The destination name in static ALTPCB's
    - Can be overridden by the DFSYDRU0 exit
  - Control blocks are dynamically created by IMS
  - Tpipe name must be unique only within a client (multiple clients can use the same Tpipe name)
    - IMS recognizes Tpipes as **XCFmember**.**Tpipename**
  - A Tpipe name cannot be the same name as an IMS transaction
  - Client manages the use of Tpipes (number, routing, etc.)

IMS OTMA Interface

- Tpipes (continued)
  - There are two kind of TPIPES: SYNChronized and Non-SYNChronized
    - SYNChronized Tpipes exchange sequence numbers and can be resynchronized across failures
    - SYNChronized Tpipes will survive an IMS warm start but will be deleted during an IMS cold start
    - Non-SYNChronized Tpipes will be deleted during an IMS warm start
  - Warning: /STOP TPIPE ALL stops all existing Tpipes and all new Tpipes as they are created
    - Fixed by APAR PQ32402

IMS OTMA Interface

- OTMA Commit Mode
  - Specified by OTMA client for each message
  - Similar to APPC-IMS Commit Mode
  - Commit Mode 0 - Commit-then-send
    - IMS sends output after syncpoint is complete
    - OTMA requires ACK to dequeue message
    - Queued on a control block called a QAB
  - Commit Mode 1 - Send-then-commit
    - IMS sends output and waits for ACK before syncpoint is complete
    - Increases region occupancy
    - Messages are not queued
    - “Queued” on a control block called a TIB

Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- OTMA Commit Mode Gotcha's
  - CM1 input message processed by response mode transaction which then message switches to nonresponse mode transaction generates CM0 output
  - CM1 input messages routed through workload router incorrectly produce CM0 output
    - Fixed by APAR PQ27158
  - CM1 input message which does ISRT - PURG - ISRT - PURG - ISRT - PURG to IOPCB will generate one multi-segment message instead of 3 single segment messages
  - ALTPCB output is always treated as CM0 even if input was a CM1 OTMA message

Telcordia Technologies

Performance from Experience

IMS OTMA Interface

- Commit mode has an effect on IMS Response Mode
  - If the message is sent with Commit Mode 1 (send-then-commit) then IMS will treat the transaction as RESPONSE mode even if it is defined as NONRESPONSE
  - If the application does not respond to the IOPCB IMS will respond with a DFS2082 RESPONSE MODE TRANSACTION TERMINATED WITHOUT REPLY message
  - This applies even if there are message switches before ending
  - You must apply PQ23213 or client can hang

IMS OTMA Interface

```
/DIS OTMA
GROUP/MEMBER  XCF-STATUS  USER-STATUS          SECURITY
   OTMAZZS
   -ZZS          ACTIVE      SERVER               NONE
   -CSQ2M        ACTIVE      ACCEPT TRAFFIC
   -ITOCA        ACTIVE      ACCEPT TRAFFIC
```

```
/DIS TMEMBER CSQ2M TPIPE ALL
MEMBER/TPIPE   ENQCT  DEQCT    QCT           STATUS
    CSQ2M
    CSQ00035       1      1      0           SYN
    CSQSTUFF       3      3      0           SYN
    CM1TPIPE       0      0      0
```

- CM1 TPIPE counts are always 0

IMS OTMA Interface

- Use IMS TPIPE trace for debugging
  - /TRA SET ON TMEMBER xxxx TPIPE yyyy
  - Produces x’6701’ IMS log records
  - Format with DFSERA10 and DFSERA30
- Diagnosis and Reference Guide needs updating
- OTMA headers start at offset x’50’ in the MSG PREF section
  - They are truncated - IBM will fix - sometime
- ACK’s are not traced - in or out

IMS OTMA Interface

- Message text is in the I/O BUFF section
  - Only the first segment of a multi-segment message is traced
  - APAR PQ25881 identifies input and output message by the ID= in the trace record
    - TIB0 - Input message before DFSYIOE0
    - TIB2 - Input message after DFSYIOE0
    - QAB0 - CM0 output message before DFSYIOE0
    - QAB2 - CM0 output message after DFSYIOE0
    - SLM0 - CM1 output message before DFSYIOE0
    - SLM2 - CM1 output message after DFSYIOE0

IMS OTMA Interface

- /TRA SET ON TABLE OTMT (not a typo) is an internal trace for IBM use only
  - It does not produce enough information for IBM to debug
- You can specify OTMADB=Y in your DFSPBxxx to get more trace data for IBM
  - **DO NOT!!!**
  - It will flood your MVS console with WTO’s
  - It can only be removed by recycling IMS

IMS OTMA Interface

- The OTMA client communicates information to IMS and gets information from IMS in a prefix passed in front of the message
  - The prefix has 4 sections mapped by macro DFSYMSG:
    - **Control**: TPIPE name and type, message type, chaining, etc.
    - **State Data**: Commit mode, IOPCB LTERM and MODNAME override, etc.
    - **Security**: Security scope, userid, RACF group, Utoken
    - **User**: Client specific
      - Saved on input (in the IMS Message Prefix)
      - Passed back on output
      - ICON shares with user
      - MQSeries passes MQMD (message descriptor)
      - OTMA/CI client can use for its own purposes - APAR PQ32398
      - Accessible by OTMA user exits

IMS OTMA Interface

- Warnings about the Security Prefix
  - TMAMSFLN - “Length of following fields” is not used
  - TMAMSULN - “Length of userid fields” **DOES NOT** include its own length like almost every length field in the OTMA headers
    - If you have an 8 byte userid this field should be set to x’09’
      - One byte for TMAMSUTY (“Field Type”)
      - 8 bytes for the userid
  - The same is also true for:
    - TMAMSRLN - “Length of Utoken Fields”
    - TMAMSGLN - “Length of Group Fields”

IMS OTMA Interface

- IMS Output Messages - WARNING!!!
  - All messages from OTMA originated transactions inserted to an alternate output PCB return down the incoming Tpipe by default (except for program-to-program message switches)
  - Alternate destinations can be reached by using the OTMA routing exits DFSYPRX0 and DFSYDRU0
  - Asynchronous output from non-OTMA originated transactions is also possible using the OTMA routing exits
    - Must be supported by the OTMA client
      - **ITOC does not support**
      - **IMS Connect will support**
      - **MQSeries does support**
      - **OTMA Callable Interface does support**

IMS OTMA Interface

- The maximum OTMA input segment size is 32767
  - LLZZ + 32763 bytes of user data
  - IMS will break up the segment into pieces that fit into the LGMSG and chain them together
- The maximum OTMA total message length input to ITOC is 10 million bytes
  - LLLL + OTMA prefixes + user data + EOM
- The maximum OTMA total message length input to MQSeries is the MQ Max Message Length
  - This is usually 4MB
  - MQSeries long message support increases this to 100MB
    - This has not been tested with OTMA
- Thanks to Peter Simpson of ABN Amro Bank and Bob Millar of IBM Hursley for this information

IMS OTMA Interface

- In an IMS 6.1 Shared Queue environment all OTMA messages must run on the IMS copy that received them
- In IMS 7.1 Commit Mode 0 messages can run on any IMS copy in the shared queue group
  - IOPCB output is routed back through the original IMS
  - ALTPCB output is sent from the processing IMS
- Commit mode 1 messages must still run on the IMS copy that received them
  - IBM would like to remove this restriction in a future release

IMS OTMA Interface

- The following list of restrictions is taken directly from the IMS/ESA V6 OTMA Guide and Reference
  - **The maximum total length of all prefixes for an OTMA message is 4096 bytes. This length does not include any application data.**
  - **Existing IMS application programs that use SETO calls might not run as expected. APPC/IMS application programs using SETO calls might require modification to use implicit OTMA support.**
  - **IMS conversational and Fast Path transactions must be defined as send-then-commit. Existing Fast Path applications can run with OTMA.**
  - **OTMA transactions cannot send input with an ACK message. For Fast Path transactions, OTMA always requests an acknowledgement for output.**
  - **A transaction from an IMS terminal (e.g. SLU2) cannot route output directly to a client, but must use an OTMA Prerouting exit routine (DFSYPRX0).**

Telcordia Technologies

Performance from Experience

IMS OTMA Interface

- OTMA Restrictions (continued)
  - OTMA messages cannot use IMS shared queues, but must remain local to the IMS subsystem to which the OTMA client is connected. Program switches for OTMA transactions also remain in the local IMS subsystem. (Enhanced in IMS 7.1 for Commit Mode 0 messages)
  - OTMA does not support the IMS Message Format Service (MFS). However, the MFS message output descriptor (MOD) name can be specified by the client in the prefix of an OTMA message.
  - OTMA does not support IMS Front-End-Switch.
  - OTMA messages cannot be encrypted.
  - All userids must be verified by RACF, unless the client specifies no security checking in the security-data section of the message prefix.
  - IMS modules that contain XCF macros must be reassembled for new releases of IMS.

IMS OTMA Interface

- OTMA Restrictions (continued)
  - OTMA has read only access to MSDB. No update access is available to MSDB from OTMA.
  - OTMA does not operate in the IMS DBCTL environment.
  - OTMA does not allow IMS terminal control commands like but not limited to /EXIT, /FORMAT, /HOLD, /RCL, and /SIGN commands.
    - (The equivalent of /EXIT can be accomplished by setting flags in the OTMA header)

IMS OTMA Exits

- There are two IMS OTMA output routing exits
  - DFSYPRX0: Pre-routing exit
  - DFSYDRU0: Destination resolution exit
- Invoked for CHNG call to modifiable ALTPCB
- Invoked for ISRT call to static ALTPCB
  - Invoked in XM mode so no SVCs or IMS services
- Invoked even if input message was not from OTMA
  - Allows asynchronous output to OTMA
- Not invoked for ISRT to IOPCB
- Not invoked for ISRT to transaction
- Name of OTMA client in parameter list for both exits is x’00’ if message did not come from OTMA
  - Documentation says it is blanks - **Doc will be updated**

IMS OTMA Exits

- DFSYPRX0: Pre-routing exit
  - Actions:
    - RC=0: Input message came from OTMA, destination is OTMA or input message did not come from OTMA, output is not OTMA
    - RC=4: message originally did not come from OTMA, but destination is OTMA
      - Need to set XCF member name of OTMA client
    - RC=8: message came from OTMA, but destination is not OTMA
  - IMS sample exit is not useful
    - MQSeries sample exit in Appendix B of the *MQSeries for OS/390 System Management Guide* is very useful for MQSeries client
  - Needed because there can be a different DFSYDRU0 exit for each client
    - This exit determines the client and thus the DFSYDRU0 exit to invoke
  - To be passed the SCD address to help in debugging - APAR PQ35615 - more later

IMS OTMA Exits

- OTMA Exit Architecture Problem
  - An client sends a message via ICON to OTMA
  - The message is scheduled and the transaction does a CHNG-ISRT-PURG to an ALTPCB to an MQSeries OTMA client
  - DFSYPRX0 gets invoked and its only option is RC=0 (Input is OTMA, Output is OTMA)
  - The DRU0 exit that next gets control is the ICON DRU0, not the MQSeries DRU0
  - It is up to the ICON DRU0 to set RC=8 (destination is a new OTMA client) and provide the MQSeries client name
  - The MQSeries DRU0 exit will then be invoked
  - This means the every client DRU0 exit has to know about all the other clients
  - IBM has a requirement to enhance DFSYPRX0 to let it pass the message directly to the proper DRU0 exit

Telcordia Technologies

Performance from Experience

IMS OTMA Exits

- OTMA Exit Architecture Problem - **SOLVED!!**
  - APAR PQ32402 now allows the DFSYPRX0 exit to override the OTMA client name even if the input was from an OTMA client
    - Many thanks to IBM for this
    - OTMAMD=Y must be specified in DFSPBxxx
  - This means that all decisions about OTMA clients can be made in one place and the DFSYDRU0 exits do not have to know about other OTMA clients
  - There are then 5 cases mapping to 3 return codes:
    - Input is not OTMA; Output is not OTMA - RC=0
    - Input is not OTMA; Output is OTMA client - RC = 4
    - Input is OTMA; Output is same OTMA client - RC=0
    - Input is OTMA; Output is new OTMA client - RC=4
    - Input is OTMA; Output is not OTMA - RC=8
      - **For RC=4 the new OTMA client name must be specified**

IMS OTMA Exits

- OTMA Exit Architecture Problem - **SOLVED!!**
  - A table can be built which maps all IMS destinations that should be routed to an OTMA client
  - A generic DFSYPRX0 can then be coded which only requires changes to the table for new destinations/clients
  - Pseudocode for this exit is as follows:
    - If the exit is invoked for a destination in the table the output will be directed to OTMA
      - **If the input was OTMA and the table client is the original OTMA client - set RC=0**
      - **If the input was OTMA and the table OTMA client is not the original OTMA client - set RC=4 and specify the new OTMA client as the table OTMA client**
      - **If the input was not OTMA - set RC=4 and specify the OTMA client as the table OTMA client**
    - If the exit is invoked for a destination not in the table the output will be directed as follows
      - **If the input was not OTMA the destination is not OTMA - set RC=0**
      - **If the input was OTMA the destination is not OTMA - set RC=8**
        - **This last item may vary by installation**

IMS OTMA Exits

- DFSYDRU0: Destination Resolution exit (default name)
  - Name can be overridden
    - In DFSYDTx
    - In CSQFSYSP in CSQZPARM for MQSeries
      - Recommendation is DRU0xxxx (xxxx=QMGR name), then exit can know XCF member name for asynchronous output
    - On DATASTORE card for ITOC
    - Via otma_openx function for OTMA Callable Interface (PQ32398)

IMS OTMA Exits

- DFSYDRU0: Destination Resolution exit
  - Actions:
    - RC=0: destination is the original OTMA TPIPE
    - RC=4: destination is non-OTMA LTERM
    - RC=8: destination is new OTMA Client (need to specify)
      - The new client DRU0 exit will then be invoked
      - No need for this after PQ32402
    - RC=12: destination is invalid
      - A1 status on CHNG call
  - Can override the output TPIPE name - PQ27207
  - Can create a SYNChronized TPIPE
    - Needed for MQSeries – See Session E92

IMS OTMA Exits

- DFSYDRU0 (continued)
  - Passed address of OTMA user data
    - Early IMS documentation said maximum was 256 bytes
    - Real maximum is 1024 bytes
    - For asynchronous output you must value this as expected by the OTMA client
  - NOT passed address of first segment of output message
    - There is none on a CHNG call
    - IBM is looking at alternatives

IMS OTMA Exits

- DFSYDRU0 (continued)
  - Debugging IMS exits has always been difficult - including DFSYDRU0
  - IBM is trying to help us
    - Now passed address of SCD at offset x’5C’ in the parameter list - PQ32402
    - Now the flag at offset x’1A’ will have the x’04’ bit turned on if the TMEMBER/TPIPE trace is active for this TPIPE - PQ32402
  - If the trace is on the for the TPIPE the code can choose to issue IWTO macros (using the SCD address) for debugging

IMS OTMA Exits

- DFSYDRU0 (continued)
  - IMS sample exit is not useful
    - MQSeries sample exit in Appendix B of the *MQSeries for OS/390 System Management Guide* is very useful
      - Follow this code VERY carefully
  - Fifth input parameter is DFSINTX0 table address
    - **Not documented**

IMS OTMA Exits

- DFSYIOE0: Input/output edit exit
  - Used to modify length or data of segment
  - Can cancel segment or message
  - Enhancements in APAR PQ12783
    - Get address of OTMA prefix control section
    - Get address of OTMA prefix state data section
    - Get address of OTMA prefix user data
      - Can be updated
      - Cannot change length
    - Provide IOPCB LTERM override name on input
    - Provide IOPCB MODNAME override name on input
  - Can provide MODNAME override name on output - APAR PQ32402
    - May be needed for MQSeries

IMS OTMA Exits

- DFSYIOE0: Input/output edit exit (continued)
  - Can be used to invoke MFS Segment Edit Routine
    - You have to establish the environment
  - Third parameter, static work area, was only halfword aligned
    - It is doubleword for DFSYPRX0 and DFSYDRU0 - IBM has fixed
  - If invoked from output generated by a non-Message Driven BMP the transaction code is blanks
    - This makes it difficult to do editing based on the source of the message
    - IBM is working to provide the PSB name to the exit
    - Until then you can edit based on the output TPIPE name

IMS OTMA Exits

- DFSCTRN0: Transaction Authorization exit
  - Not an OTMA exit
  - Is called when transaction input is from OTMA
  - The userid was not passed to the exit when the input was from OTMA
    - IBM fixed - APAR PQ28799

IMS OTMA Exits

- DFSCCMD0: Command Authorization exit
  - Not an OTMA exit
  - Is called when command input is from OTMA
  - The fourth parameter in the function-specific parameter list is documented as containing the userid associated with the command
    - It contained binary zeroes if the command was from OTMA
    - IBM fixed - APAR PQ21864
      - This APAR also addresses the inconsistent use of the OTMA security section by MQSeries and ITOC

IMS OTMA Exits

- DFSNPRT0: Input Message Routing Exit
  - Not an OTMA exit
  - Flag at offset +8 in the parameter list is set to x'00000008' to indicate OTMA input
  - Address of first message segment at offset +12 was zero
    - IBM fixed - APAR PQ32402
  - Address at offset +28 points to the OTMA member name of the client that sent the message

IMS OTMA Exits

- DFSQSPC0: Queue Space Notification Exit
  - Not an OTMA exit
  - Invoked with new “caller” vector code for OTMA
  - Invoked under the OTMA TCB
    - There is one per OTMA client
    - Watch reentrancy

IMS OTMA Exits

- If error in exits, user can get AX status code on IMS call
  - Not documented as an OTMA error: Fixed in IMS 6.1 doc
- X’67D0’ records are written to the IMS log
  - Return codes (OTMA Guide: SC26-8026-01)
    - X’1C’ - internal interface error
    - X’20’ - DFSYDRU0 overrides exceed maximum limit
    - X’24’ - invalid destination from DFSYDRU0
    - X’28’ - unknown return code from DFSYDRU0
    - X’2C’ - DFSYPRX0 returned invalid XCF member name
    - X’30’ - DFSYPRX0 required XCF member name not returned
    - X’34’ - unknown return code from DFSYPRX0
    - X’38’ - destination is in different client and XCF member name from DFSYDRU0 is invalid
    - X’3C’ - DFSYDRU0 returned an invalid user data length

IMS OTMA Security

- Requires RACF 1.9.2 (or equivalent)
- Two levels of security

  1. Connections from client to IMS (during client bid)

     - Uses the RACF Facility class
     - IMS checks ***IMSXCF.XCFgroupname.client-member-name***
       - Client must have read access to this class

IMS OTMA Security

- Two levels of security (continued)
  2. OTMA Security (set by /SECURE OTMA)
     - CHECK
       - Existing RACF calls are made
       - IMS commands are checked against the CIMS class
       - IMS transactions are checked against the TIMS class
     - FULL
       - Same as CHECK, but the ACEEs are built in the dependent regions as well as the control region
     - NONE
       - No calls to RACF are made
     - PROFILE
       - Each message defines the level of security checking to be done

Information on the Web

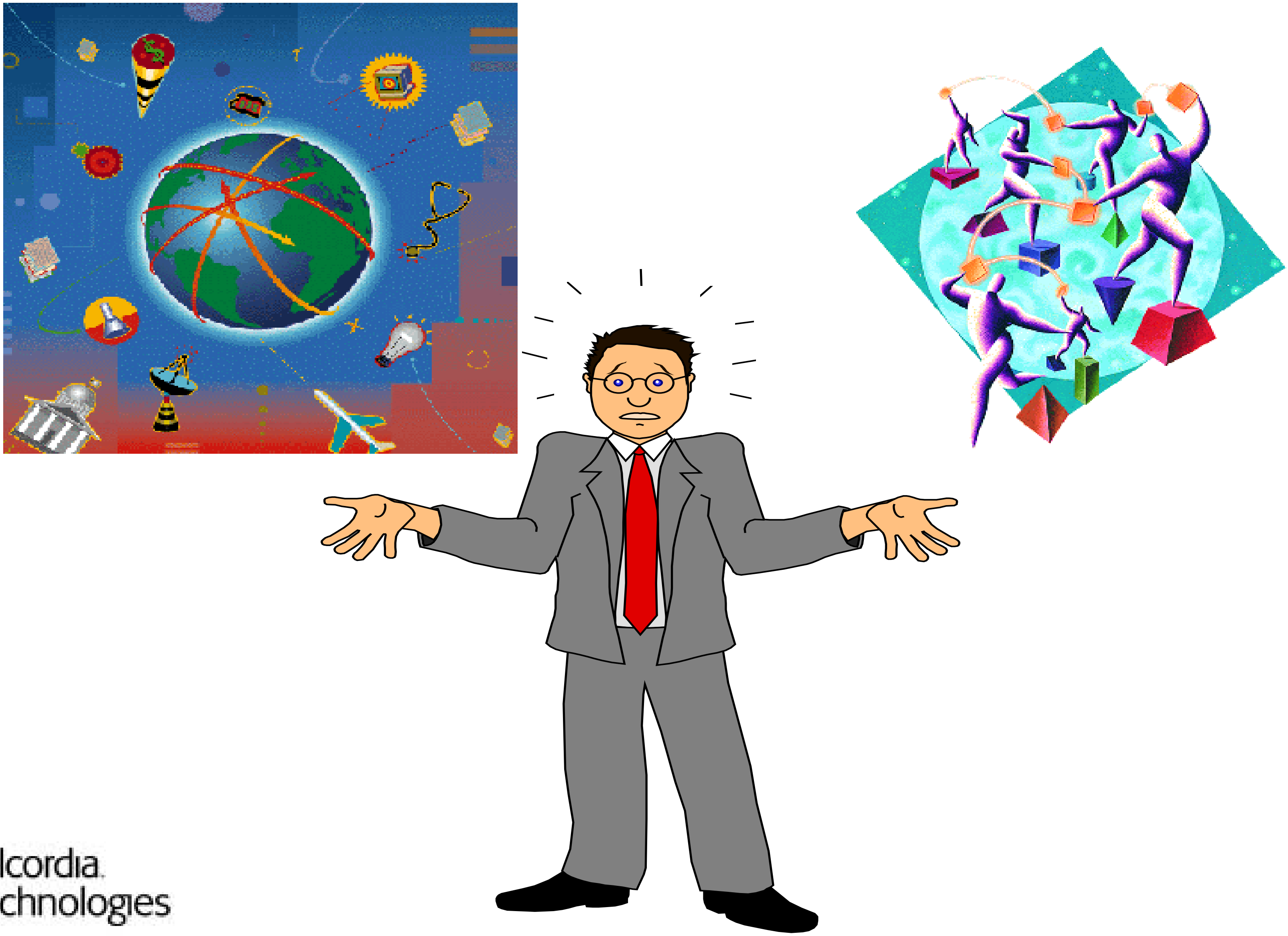
- http://www.software.ibm.com/ts/mqseries/
- http://www.software.ibm.com/data/ims
- http://www.redbooks.ibm.com/

IMS OTMA Documentation

- IMS/ESA V6 OTMA Guide and Reference - SC26-8743
- IMS/ESA Application Programming : Transaction Manager - SC26-8729
- IMS/ESA Customization Guide - SC26-8732

RETURN TO INDEX

Questions?



Telcordia Technologies

Performance from Experience