APPC and OTMA

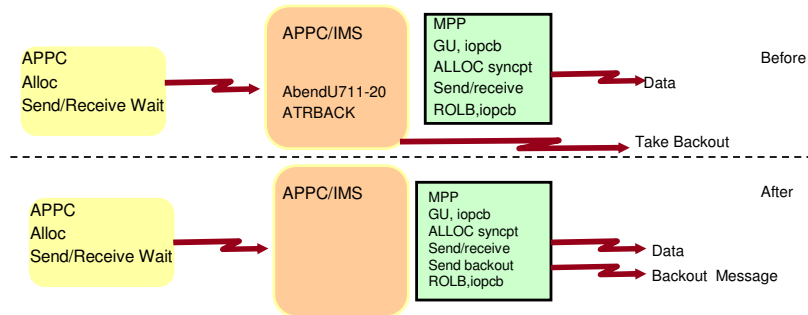**Information Management** software

IBM

Topics

- APPC and OTMA RRS protected conversation processing with ROLB

- APPC and OTMA Shared Queues Enhancement

- APPC LU 6.2 Input/Output Edit Exit (DFSLUEE0) Enhancement

- OTMA ACEE

- OTMA DFS2082I for CM0

- Performance Enhancement

- MQ Message Expiration

307

IBM

APPC/OTMA RRS protected conversation processing with ROLB call

- **Remove ABENDU711-20 for ROLB call restriction**

  IMS 11 APAR PK84674
  PTF UK59654

  – The ROLB call will not affect the entire RRS protected Unit of Work.
    - The RRS protected input remains in process until a commit point
    - The outbound protected conversation will not be notified to backout

- **Migration Consideration**

  – IMS Application must now notify outbound protected application to do backout

APPC	APPC/IMS	MPP	Before
Alloc		GU, iopcb	
Send/Receive Wait	AbendU711-20	ALLOC syncpt	Data
	ATRBACK	Send/receive	
		ROLB,iopcb	Take Backout

APPC	APPC/IMS	MPP	After
Alloc		GU, iopcb	
Send/Receive Wait		ALLOC syncpt	Data
		Send/receive	
		Send backout	Backout Message
		ROLB,iopcb	

308

APPC/OTMA RRS protected conversation processing with ROLB call:

APAR PK84674 removes the current ABENDU0711 RC20 restriction that is enforced when an application ROLB call is processed. The ROLB call will now be isolated to the issuing IMS application only and will not affect the entire RRS protected Unit of Work. The protected input message remains in process for the IMS application until a commit is reached. It also changes the expected application behavior for modified message driven IMS applications that can receive
protected input messages from OTMA and/or APPC/MVS. If the IMS modified application issues an outbound protected conversation, the outbound protected conversation is no longer included in the ROLB processing meaning the outbound protected conversation will not be backed out as part of the ROLB call since ATRBACK is not issued.

IBM

APPC and OTMA SQ Enhancement

- New capability that removes the dependency on RRS in a Shared Queues environment for
  - APPC synchronous conversations and OTMA CM1 (send-then-commit) interactions
    - Applies only to synclevel=None | Confirm
      - Synclevel=Syncpoint still requires RRS

- Communications use XCF services

309

IMS 12 supports the execution of APPC synchronous conversations and OTMA Commit Mode 1 (CM1) or Send-then-Commit transactions with sync level of NONE or CONFIRM in a Shared Queues (SQ) environment using XCF communications thereby eliminating the need for Resource Recovery Services (RRS). Note that the use of Synclevel=syncpoint still requires RRS.

IBM

Implementation

▪ New options for the existing AOS= parameter in DFSDCxxx to request the use of XCF

– AOS=B: Synchronous transactions synclevel=NONE|CONFIRM can be processed in a back-end system using XCF communications

• Note: Processing synclevel of SYNCPT depends on the RRS option

– RRS=Y: transactions can be processed at either FE or BE, using RRS

– RRS=N: transactions are only processed at the FE

– AOS=S: allows synchronous transactions with synclevel of NONE|CONFIRM to be processed in a back-end system using XCF communications

– Note: Processing synclevel SYNCPT is equivalent to AOS=F.

– AOS=X: allows synchronous transactions with synclevel of NONE|CONFIRM to be processed in a back-end system using XCF communications

– Note: Processing synclevel of SYNCPT is equivalent to AOS=N

Note: (1) Choice of B|S|X is dependent on how syncpoint messages are to be processed

310

Existing AOS= options in the DFSDCxxx member of IMS PROCLIB include:

▪AOS=N which specifies that shared message queue support for synchronous APPC/OTMA is not active. The default is inactive (N).

▪AOS=Y which specifies that shared message queue support for synchronous APPC/OTMA is active (Y) and uses RRS

▪AOS=F which activates the function by force (F) even if another member in the IMSplex cannot activate the function..

The new options that support XCF include:

▪AOS=B which requests the use of XCF to communicate between the FE and the BE systems for synchronous transactions with sync level of NONE and CONFIRM. The processing of requests that are associated with synclevel of SYNCPT depends on the RRS specification. If RRS=Y then transactions the synclevel SYNCPT transaction can be processed at either a FE or BE IMS system using RRS. If RRS=N then transactions with synclevel SYNCPT are only processed at the FE IMS.

▪AOS=S which also requests the use of XCF to communicate between the FE and the BE systems for synchronous transactions with sync level of NONE and CONFIRM along with RRS Multisystem Cascaded Transaction support for synchronous transaction with sync level of SYNCPT. Under this option, SYNCPT request processing is equivalent to AOS=F meaning that if RRS is not active on one system, then the systems that have specified S (force) will still queue incoming transactions globally (without any affinity.) Therefore, if a system without RRS tries to process one of these transactions, IMS abends the application with a U711 code.

▪AOS=X which, like AOS=B and AOS=S, requests the use of XCF to communicate between the FE and the BE systems for synchronous transaction with sync level of NONE and CONFIRM.

RRS will not be used for synchronous transaction with sync level of SYNCPT. In this case, the functionality of AOS=X specification is equivalent to the AOS=N specification.

The B and X parameters are only applicable to the FE system. The BE system is not required to specify this parameter. The BE will invoke either the new XCF message processing or the existing RRS message processing depending upon the message having the XCF indicator or the RRS indicator, respectively. Note that the existing parameters of Y, N, and F are still available and their functionalities are unchanged.

IBM

Example of a flow using XCF – Synclevel=NONE

AOS = B | S | X

The FE IMS queues the synchronous synclevel NONE message into the SQ structure with a newly defined indicator and waits to be notified when the response is available
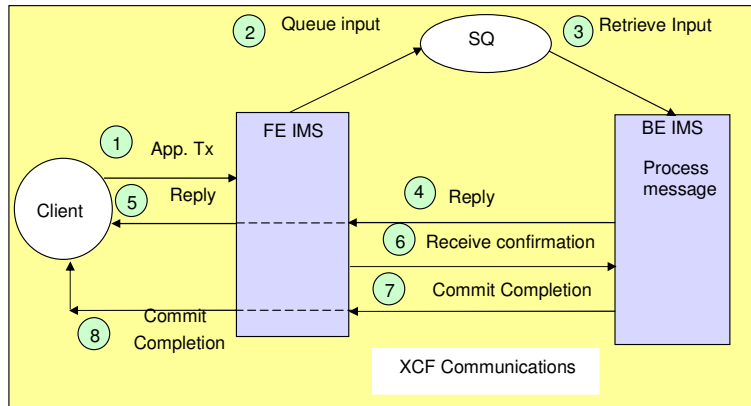
A BE IMS in the same SQ group retrieves the message and recognizes that the input message contains the XCF indicator

311

When a message is received, the FE IMS control region determines whether the input transaction specifies the sync level of NONE or CONFIRM and that XCF enablement is specified (AOS specification is B, S or X) and queues the APPC synchronous conversation or OTMA CM1 input transaction message into the SQ structure with a newly defined indicator. Any back-end (BE) IMS control region that processes this input will recognize the new indicator and start using XCF to send and receive messages to and from the originating FE IMS control region. The FE IMS system gets notified when a transaction response is available. The following describes the flow:

(1) A client application enters an OTMA CM1 or APPC synchronous conversation transaction with synclevel NONE to the FE IMS system.

(2) The input transaction message with the XCF indicator is put on the SQ using the existing transaction ready queue type.

(3) A dependent region on the BE IMS system retrieves and processes the input message.

(4) The BE IMS system sends the transaction response to the FE and waits for confirmation.

(5) The FE forwards the transaction response to the client application.

(6) The FE sends the receive-confirmation to the BE and waits for the commit completion.

(7) The BE receives the receive-confirmation, completes the commit process and sends back the commit completion message to the FE.

(8) The FE forwards the commit completion to the client application.

Example of a flow using XCF – Synclevel=Confirm

AOS = B | S | X

The FE IMS queues the synchronous synclevel CONFIRM message into the SQ structure with a newly defined indicator and waits to be notified when the response is available
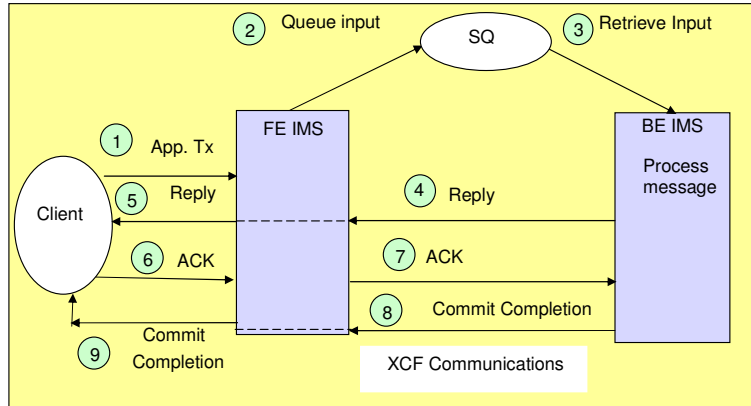
A BE IMS in the same SQ group retrieves the message and recognizes that the input message contains the XCF indicator

312

Synchronous requests with synclevel=CONFIRM differ slightly from the previous flow due to an extra flow from the client. The following describes this scenario:

(1) A client application enters OTMA CM1 or APPC synchronous conversation transaction with synclevel=CONFIRM from the FE IMS system.

(2) The input transaction message with the XCF indicator is put on the SQ.

(3) A dependent region on the BE IMS system retrieves and processes the input message.

(4) The BE IMS system sends the transaction response to the FE and waits for an ACK.

(5) The FE forwards the transaction response to the client application.

(6) The client application returns an ACK to acknowledge receipt of the output response

(7) The FE forwards the ACK to the BE and waits for a commit completion.

(8) The BE receives the ACK, completes the commit process and sends back the commit completion message to the FE.

(9) The FE forwards the commit completion message to the client application.

Front-end Logging

- New AOSLOG=Y|N keyword in the IMS DFSDCxxx proclib
  - Specifies whether or not the FE system is to write a 6701 log record for:
    - Response messages returned from the BE system via XCF
      - Applicable to all synclevels (NONE, CONFIRM and SYNCPT)
    - Error messages returned from the BE system via XCF
      - Applicable to all synclevels of (NONE, CONFIRM and SYNCPT)
  - ID=TIB3
    - For diagnostics

- /DIAGNOSE SET AOSLOG(ON|OFF)
  - Enhancement to the /DIAGNOSE command to control AOSLOG capture for events related to APPC and OTMA synchronous transactions in a shared queues environment

313

IMS 12 also provides an AOSLOG=Y|N parameter in the DFSDCxxx member of proclib to request that a 6701 log record be written for the following cases:

▪Response messages returned from the BE system via XCF for transactions with all synclevels of NONE, CONFIRM and SYNCPT.

▪Error messages returned from the BE system via XCF for transaction with all synclevels of NONE, CONFIRM and SYNCPT.

The AOSLOG parameter is only applicable to the FE system. The BE system is not required to specify this parameter.

The /DIAGNOSE command which retrieves diagnostic information for system resources such as IMS control blocks, user-defined nodes, or user-defined transactions at any time without taking a console dump. In IMS version 12, the command has been enhanced to allow or disallow the capture of events related to APPC and OTMA synchronous transactions in a shared-queues environment. When enabled, the AOSLOG events are written to the OLDS as type X'6701' records. If AOSLOG(ON) is specified in a non-shared queues environment or when AOS=N is specified in the DFSDCxxx PROCLIB member, the command is rejected with a DFS2859I message.

IMS Version 12

IBM

6701 Log Record Example

```
   INTERNAL TRACE RECORD          ID = TIB3  SEGNO=00  RECNO = 000002D5 TIME  22:49:45.249   DATE 2010.081
MSG PREF
 0C65E040 000000    027A0000 0040D580  85D50000 F9F9F9F9  40404040 D1C1E5E3  D9C1D5D1 C5B81AA3   *.:... N.EN..9999    JAVTRANJE..T*
 0C65E060 000020    68332B64 C0000000  0C732060 C8E6E2F1  40404040 40404040  40404040 C9D4E2F1   *............HWS1         IMS1*
 0C65E080 000040    40404040 C5B7CA11  CA942423 121000A8  C5B81AA3 68332B64  00000000 00000000   *   E....M.....YE..T............*
 0C65E0A0 000060    00000000 00000000  0C732060 00000000  00000000 C9D4E2F1  00000000 00000000   *...........-........IMS1........*
 0C65E0C0 000080    00000000 00000000  00000000 C5B81AA3  682A9C64 00000000  00000000 00000000   *............E..T................*
 0C65E0E0 0000A0    00000000 00000000                                                            *........                       *
MCI PREF
 0C65E0E8 000000                       01400000 0000F9F9  F9F94040 4040A0F0  00000002 00000000   *      . ....9999    .0........*
 0C65E100 000018    00000000 00010000                                                            *........                       *
STATE
 0C65E108 000000                       00481020 00000000  00000000 00000000  00000000 00000000   *        .......................*
 0C65E120 000018    00000000 00000000  00000000 0000C5B8  1AA36814 11E60000  00000000 00000000   *..............E..T...W..........*
 0C65E140 000038    00000000 00004040  40404040 40400000                                         *......      ..                 *
SECURITY
 0C65E150 000000                       006AC614 0902E4E2  D9E3F0F0 F3400903   *                ..F...USRT003 ..*
 0C65E160 000010    E2E8E2F1 40404040  00000000 00000000  00000000 00000000  00000000 00000000   *SYS1    .......................*
 0C65E180 000030    00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000   *...............................*
 0C65E1A0 000050    00000000 00000000  00000000 00000000  00000000 00000000  0000                 *.........................      *
USER SGM
 0C65E1BA 000000                                                              0100 0000C9D4        *                       ....IM*
 0C65E1C0 000006    E2F14040 4040C3D3  C9C5D5E3 F0F2F9F9  F9F94040 4040C5B8  1AA2754F D6660000   *S1    CLIENT029999    E..S.|O...*
 0C65E1E0 000026    00000000 00000000  00000C62 CAB00000  00000000 00001010  20000000 00004040   *...............................*
 0C65E200 000046    40404040 40400000  00000202 00000000  00000000 00004040  40404040 40400000   *      ................      ..*
 0C65E220 000066    00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000   *...............................*
 0C65E240 000086 TO 0C65E280 0000C6  SAME AS ABOVE
 0C65E2A0 0000E6    00000000 00000000  00000000 00000000  00000000 00000000  0000                 *.........................      *
APPL SGM
 0C24CF98 000000                                                             00680300 E896A440   *                        ....YOU *
 0C24CFA0 000008    A28595A3 7A40C8C9  40E3C8C5 D9C54040  40404040 40404040  40404040 40404040   *SENT: HI THERE                  *
 0C24CFC0 000028    40404040 40404040  40404040 40404040  40404040 40404040  40404040 40404040   *                                *
 0C24CFE0 000048                       SAME AS ABOVE
UDATA
 0C5EFD04 000000             003E0000  0C732060 0C1B1330  C9D4E2F2 40404040  0000002E 00000000   *      ........-....IMS2   ........*
 0C5EFD20 00001C    C9D4E2F1 40404040  C5B81AA3 68380E64  C9D4E2F1 40404040  C5B81AA3 68380E64   *IMS1    E..T....IMS1    E..T....*
 0C5EFD40 00003C    4000
```

314

This visual provides an example of the 6701 log record produced when AOSLOG=Y.

DFSMSCE0

- TM and MSC Message Routing and Control User Exit (DFSMSCE0)
  - Allows queuing the message either with the RRS option or the XCF option
    - On **input** to the exit, the following indicators are set

MSCE3XCF EQU	X'80'	- This is a global XCF enabled IMS system
MSCE3RRS EQU	X'40'	- This is a global RRS enabled IMS system

    - On **output**, the exit can set

MSTR2XCF EQU	X'02'	- Message is to be queued globally to the SQ with the XCF indicator. Only for synclevel of NONE or CONFIRM
MSTR2RRS EQU	X'01'	- Message is to be queued globally to the SQ with the RRS indicator. Only for synclevel of NONE or CONFIRM

    - If both options are set, IMS will use the XCF indicator option and ignore the RRS indicator option

315

Users can also use the DFSMSCE0 TM and MSC Message Routing and Control user exit to specify the APPC synchronous conversation or OTMA CM1 transaction with sync level of NONE or CONFIRM to be queued with the RRS or the XCF indicator option. When the exit is invoked, IMS provides the following indicators in the MSCEFL3 field of the DFSMSCEP parameter list: MSCE3XCF to indicate that the global XCF is active and MSCE3RRS to indicate that the global RRS is active. On output, the exit can set an indicator to request the use of either XCF or RRS. The XCF indicator option is the default value. If both options are set, IMS will use the XCF indicator option.

If the XCF indicator option is selected, IMS determines whether the message can be queued globally with the XCF indicator. If not, IMS determines whether the message can be queued globally with the RRS indicator. If not, IMS queues the message locally.

If the exit defaults to or sets the XCF indicator, IMS determines input transaction message queuing based on the following order:
   a. Queue the input globally to the SQ with the XCF indicator if eligible.
   b. Queue the input globally to the SQ with the RRS indicator if eligible.
   c. Queue the input locally with the affinity to the FE system where the input is received.
If the exit selects the RRS indicator option, the input transaction message queuing is determined as follows:
   a. Queue the input globally to the SQ with the RRS indicator if eligible.
   b. Queue the input locally with the affinity to the FE system where the input is received.

IBM

Errors

- Error conditions in this environment can produce either a message or an abend
  - DFS2088I - APPC/OTMA SMQ Enablement inactive. Reason = 040
    - Means AOS= B|S|X specified but MINVERS < 12.1
  - DFS2089I - APPC/OTMA SMQ ENABLEMENT ACTIVE [RRS IS USED| XCF IS USED. | XCF AND RRS ARE USED]

  - New: Abend U0109 in the back-end IMS
    - IMS did not deliver a response for the APPC synchronous| OTMA (CM1) transaction, or, the LU 6.2 device or OTMA client returned a negative acknowledgment (NAK) to the output message of the synclevel(confirm) synchronous|CM1 transaction using XCF communications
      - RC=01 means BE received a NAK from the FE
      - RC=02 means FE terminated or restarted
    - Transaction instance is ended but program and transaction are not stopped

  - Abends in the front-end continue to see U0119

316

DFS2088I is displayed when the APPC/OTMA Shared Message Queue Enablement support using XCF communications is inactive for standard transactions. Standard transactions are transactions with sync level of NONE or CONFIRM. This message could be issued with reason code=040 if an AOS option requesting XCF communications is specified but IMS is not defined with MINVERS of 12.1 designating the level of IMS that can process this function.

DFS2089I is produced to show the environment that has been enabled for APPC/OTMA SMQ enablement.

A new U0109 abend can occur if the LU 6.2 device or OTMA client returns a negative acknowledgement (NAK) to the output of send-then-commit (CM1) transactions with sync level of NONE or CONFIRM (SL0 and SL1) using XCF communications. The instance of the transaction is terminated but neither the program nor transaction are stopped. Reason code = 01 means that the BE received a NAK from the FE IMS, and Reason code = 02 means that the FE terminated or restarted. DFS554A and DFS555I will contain this reason code if issued.

IBM

IMS Commands

- Enhancements to the output of /DISPLAY commands
  - /DIS A DC, /DIS A REG, /DIS OTMA
    - New statuses in APPC/OTMA SHARED QUEUE STATUS –
      LOCAL=*status1* GLOBAL=*status2*

      - *status1* can be one of the followings
        - ACTIVE-RRS
        - ACTIVE-XCF
        - ACTIVE-RRS/XCF
        - FORCE-RRS
        - FORCE-RRS/XCF
        - INACTIVE
        - UNSUPPORTED

      - *status2* can be one of the followings:
        - ACTIVE-RRS
        - ACTIVE-XCF
        - ACTIVE-RRS/XCF
        - CHECK
        - INACTIVE

  - New text that shows the AOSLOG setting

317

Enhancements to the output of certain DISPLAY commands include information about the APPC/OTMA Shared Queues support:

- ACTIVE-RRS is changed from ACTIVE, where AOS=Y and RRS=Y are specified

- ACTIVE-XCF is a new status, where AOS=X is specified; or AOS=B and RRS=N are specified

- ACTIVE-RRS/XCF is a new status, where AOS=B and RRS=Y are specified

- FORCE-RRS is changed from FORCE, where AOS=F is specified

- FORCE-RRS/XCF is a new status, where AOS=S is specified

Additionally the APPC/OTMA Shared Queues Logging specification is displayed.

IBM

IMS Commands …

- **Example**

```
XX,/DIS A DC
DFS000I    VTAM STATUS AND ACTIVE DC COUNTS
DFS000I    VTAM ACB OPEN        -LOGONS DISABLED
DFS000I    IMSLU=N/A.N/A          APPC STATUS=DISABLED  TIMEOUT=   0
DFS000I    OTMA GROUP=N/A       STATUS=NOTACTIVE    IMS1
DFS000I    APPC/OTMA SHARED QUEUE STATUS - LOCAL=ACTIVE-XCF GLOBAL=ACTIVE-XCF
DFS000I    APPC/OTMA SHARED QUEUES LOGGING=Y
DFS000I    APPC/OTMA RRS MAX TCBS -   40 ATTACHED TCBS - 1 QUEUED  RRSWKS-   0
DFS000I    APPLID=APPL8    GRSNAME=        STATUS=DISABLED
DFS000I      LINE ACTIVE-IN   -   1 ACTIV-OUT -    0
DFS000I      NODE ACTIVE-IN -   0 ACTIV-OUT  -    0
DFS000I      LINK ACTIVE-IN -   0 ACTIV-OUT -    0
```

318

Enhancements to the output of certain DISPLAY commands include information about the APPC/OTMA Shared Queues support:

- ACTIVE-RRS is changed from ACTIVE, where AOS=Y and RRS=Y are specified

- ACTIVE-XCF is a new status, where AOS=X is specified; or AOS=B and RRS=N are specified

- ACTIVE-RRS/XCF is a new status, where AOS=B and RRS=Y are specified

- FORCE-RRS is changed from FORCE, where AOS=F is specified

- FORCE-RRS/XCF is a new status, where AOS=S is specified

Additionally the APPC/OTMA Shared Queues Logging specification is displayed.

IMS Commands …

- Example of /DIS ACTIVE REG output
  - New status of WAIT-XCF and TERM-WAIT XCF
  - New text: TMEM:tmember-name TPIPE:tpipe-name for OTMA and LUNAME:networkid.luname for APPC

```
24,/DIS A REG
DFS000I     REGID JOBNAME    TYPE  TRAN/STEP   PROGRAM  STATUS           CLASS   IMS1
DFS000I           JMPRGN     JMP   NONE    IMS1
DFS000I         1 IMSMPPA    TPI   APOL11 IMS1 APOL1    WAIT-RRS/PC      1,2,3,4
DFS000I     URID:   C2D6B6917DE820000000000001010000 ORIGIN: IMS2
DFS000I         2 IMSMPPB    TPI   APOL12 IMS1 APOL1    TERM-WAIT RRS    1,2,3,4
DFS000I     URID:   C2D6B6917DE830000000000001010000 ORIGIN: IMS2
DFS000I         3 IMSMPPC    TPI   APOL13 IMS1 APOL1    WAIT-XCF         1,2,3,4
DFS000I     TMEM:   HWS1           TPIPE: CLIENT01  ORIGIN: IMS2
DFS000I         4 IMSMPPD    TPI   APOL14 IMS1 APOL1    TERM-WAIT XCF    1,2,3,4
DFS000I     LUNAME: IMSNETWK.LU62IMS1 ORIGIN: IMS2
DFS000I           JBPRGN     JBP   NONE    IMS1
DFS000I           BATCHREG   BMP   NONE    IMS1
DFS000I           FPRGN      FP    NONE    IMS1
DFS000I           DBTRGN     DBT   NONE    IMS1
DFS000I           DBRICTAB   DBRC                                        IMS1
DFS000I           DLISDEP    DLS       IMS1
```

319

WAIT-XCF is a new status that is shown when an application program is the middle of processing a syncpoint for an APPC or OTMA transaction with sync level of NONE or CONFIRM. Sync point can continue after the client issues either a positive acknowledgement (ACK) or a negative acknowledgement (NAK). XCF is used to communicate between the back-end and the front-end IMS systems.

TERM-WAIT XCF is another new status that is displayed when a dependent region termination is in progress and the application in the region is still processing a syncpoint for an APPC or OTMA transaction with sync level of NONE or CONFIRM. Syncpoint can continue after the client issues either an ACK or NAK. When a dependent region is found in this state, a continuation line is inserted into the display, which shows either the transaction member (TMEMBER) and the transaction pipe (TPIPE) for OTMA client or the network id (NETWORKID) and the logic unit name (LUNAME) for APPC client as well as the originating IMS system ID (ORIGIN) that are associated with the transaction processing in the dependent region.

Setup Considerations

▪ Considerations

- Choice of AOS = B|S|X depends on syncpoint message processing
- Changing AOS= setting requires an IMS restart (NRE/ERE)
- DBRC MINVERS of 12.1 is required
  • All IMS systems in a SQ group must be at this level to enable the enhancement
    - IMS systems at lower version using the same RECON datasets will not be able to join the SQ group
    - IMS systems at lower versions using a different RECON dataset will still be able to join the SQ group, but may experience U711
- Different AOS parameter settings in different IMS systems
  • May complicate the environment
    - Understand the impact

320

From the XCF perspective for synchronous synclevel=NONE | CONFIRM requests, the AOS specifications of B, S, or X are identical. Where they differ is in the processing of synclevel=SYNCPOINT requests.

Note that and IMS restart is required to change either or both of the AOS and AOSLOG settings.

Additionally, the Database Recovery Control facility (DBRC) minimum version (MINVERS) value of 12.1 is required to enable the enhancement. This means that all of IMS systems in the same SQ group must have the same MINVERS value of 12.1 in order to activate the enhancement.

IBM

APPC and OTMA SQ Enhancements Benefits

- Benefit
  - Using XCF rather than RRS
    - Allow IMS to be the syncpoint manager
      - Enhances the performance of the commit processing by eliminating
        - RRS logging overhead
        - Potential RRS commit processing bottleneck
        - Overhead associated with communicating with an external syncpoint manager

321

The use of XCF rather than RRS in a Shared Queues environment for synchronous/CM1 synclevel(none|confirm) messages simplifies syncpoint processing and eliminates unnecessary RRS overhead.

IBM

LU 6.2 Input/Output Edit Exit (DFSLUEE0) Enhancement

- A new return code (RC=2) for asynchronous conversation requests
  – Requests that an undeliverable message be dequeued
    • Previously, IMS would requeue the message

- Benefit
  – Greater control over undeliverable asynchronous output

322

An enhancement to the LU6.2 Input/Output Edit Exit (DFSLUEE0) supports a new return code (RC=2) that requests the dequeuing of an undeliverable asynchronous output message. Previously, IMS would al ways requeue the message. The new return code is only valid for asynchronous conversations.

OTMA ACEE Reduction

IBM

OTMA ACEE Reduction for Multiple OTMA Clients

- New capability that creates, shares and **caches** a <u>single</u> ACEE associated with a RACF userid
  - Shared across multiple OTMA member clients (TMEMBER)

- New maximum ACEE aging value during client-bid
  - 999999 seconds (11.5 days)
    - Previously 68 years
    - Range: 300 seconds to 999999 seconds
      - If OTMA receives a value less than 300, the value is reset to 0 and OTMA will not refresh ACEEs

- A cached ACEE has an aging value based on the OTMA member client with the lowest value

324

To fully benefit from the availability, scalability, and performance advantages of the z/OS environment, multiple instances of OTMA member clients (TMEMBERs), e.g., IMS Connect, DB2 Stored Procedures, and WebSphere MQ (WMQ), are able to run in parallel on multiple servers or LPARs while still being able to connect to the same IMS. In previous releases, IMS OTMA isolated the security environment of one member client from another. This meant that an Access Control Environment Element (ACEE) for a userid had to be created for each OTMA member client instance if the same userid accessed IMS through more than one path, e.g., IMS Connect as well as WMQ. Creating multiple copies of the same ACEE resulted in increased storage usage in subpool 249 in EPVT. In IMS version 12, OTMA caches the ACEE so that only one copy exists for the same user even when messages from that user are sent in through different member clients. The cached ACEEs also reside in SP 249.

IMS version 12 also introduces a new maximum aging value for ACEEs. The ACEE aging value triggers when an ACEE is refreshed. This value is specified by the OTMA member client during the client-bid process. The maximum has been reduced from 68 years to 11.5 days. Each cached OTMA ACEE will have an aging value based on the OTMA member client using it that has specified the lowest number. For example, if WMQ sets its ACEE aging value to 5 days while IMS Connect sets its ACEE aging value to 1 day, any ACEEs that only WMQ uses will have an aging value of 5 days and any ACEEs that are only invoked through IMS Connect will have an aging value of 1 day. Once the ACEE which was previous only used through WMQ is invoked on behalf of the same user sending a message through IMS Connect, then the aging value is set to 1 day, the lower of the 5 days versus 1 day value. This enhancement is important because some OTMA clients, e.g., such as the DB2 Stored Procedure DSNAIMS, set the '7FFFFFFF'X seconds (68 years) as the ACEE aging value for users sending IMS transactions and commands. The enhancement also detects obsolete ACEEs thereby improving security.

To enable ACEE refreshes, the OTMA member client needs to request a minimum ACEE aging value of 300 seconds during the client-bid process. If the aging value specified by an OTMA member client is less than 300 then OTMA resets the value to 0 meaning ACEEs will not be refreshed. When an aging value OAAV value is sent to IMS version 12 that is greater than 999999 seconds (11.5 days), OTMA resets that value to 999999 seconds. IMS versions 10 and 11 allowed the value to be up to 2147483647 or 68 years.

Challenge Addressed: Multiple ACEEs for the same User
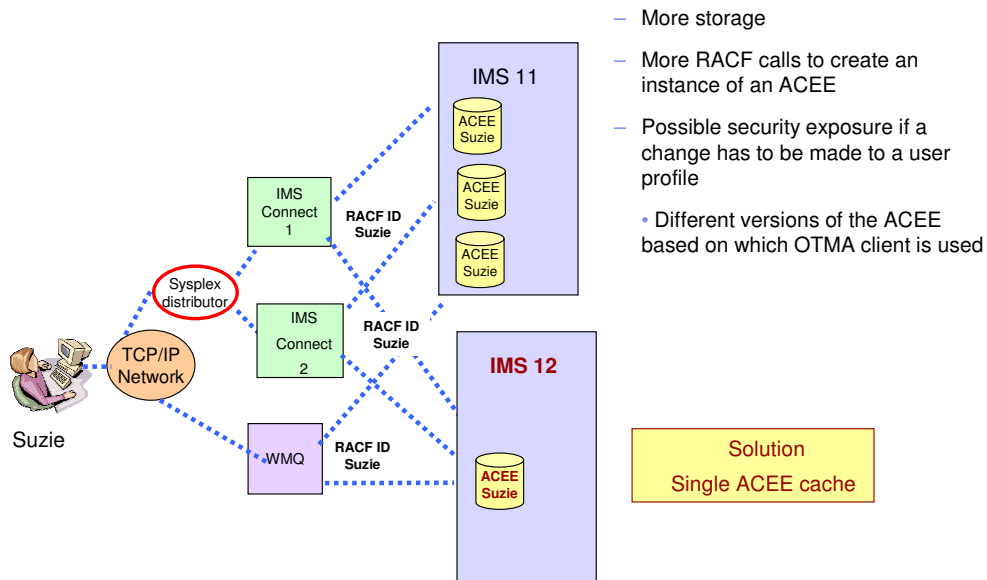
- More storage
- More RACF calls to create an instance of an ACEE
- Possible security exposure if a change has to be made to a user profile
  - Different versions of the ACEE based on which OTMA client is used

IMS 11

ACEE Suzie

ACEE Suzie

ACEE Suzie

IMS 12

ACEE Suzie

Solution
Single ACEE cache

Suzie

TCP/IP Network

Sysplex distributor

IMS Connect 1

IMS Connect 2

WMQ

RACF ID Suzie

RACF ID Suzie

RACF ID Suzie

325

The scenario in this visual referencing IMS 11 describes the issue when an end-user accesses IMS using several paths. With sysplex distributor and multiple servers, a connection can be routed through more than one IMS Connect for the same IMS system. Creating multiple ACEEs occupies storage in subpool 249 and requires calls to RACF to create each instance of an ACEE. For example, if we have 2 IMS Connect member clients connected to IMS and each one submitted IMS transactions on behalf of the same 10,000 RACF userids, then IMS OTMA would create 20,000 RACF ACEEs. As the same user sends in messages through WMQ then a third ACEE would be created for that user. The system storage for multiple copies of ACEEs on behalf of the same user can keep growing.

Additionally, a potential security exposure is introduced if a security change is made to the user profile but not all copies of the ACEE are refreshed. For example, an IMS input transaction for a userid via an instance of an OTMA member client application could be rejected by OTMA but the same userid sending in a message through a different path could be accepted by OTMA because the ACEE was created prior to the profile change and before a refresh of the ACEE is requested.

On the other hand, as shown with the connections to IMS 12, the issues described above no longer apply to this environment. The ACEE is created when a transaction message on behalf of a specific userid first comes in from an OTMA member client and then cached for reuse for subsequent transaction requests from the same user regardless of the communication path used.

Command Enhancement

- ▪ /DISPLAY OTMA command response shows the ACEE aging value for each OTMA client

```
/DIS OTMA

Response:
GROUP/MEMBER   XCF-STATUS  USER-STATUS    SECURITY   TIB  INPT SMEM DRUEXIT   T/O ACEEAGE

XCFGRP1
-IMS1          ACTIVE  SERVER         FULL        8000      N/A       0
-HWS1          ACTIVE  ACCEPT TRAFFIC FULL     0  5000      HWSYDRU0  239   3600
-HWS2          ACTIVE  ACCEPT TRAFFIC FULL     0  5000      HWSYDRU0  239   7200
-HWS3          ACTIVE  ACCEPT TRAFFIC FULL     0  5000      HWSYDRU0  239      0

*09121/172200*   IMS1
```

- – Cached ACEEs that are uniquely used by HWS1 and HWS2 have aging values of 3600 and 7200 respectively

- – Cached ACEEs that are used by both HWS1 and HWS2 have an aging value of 3600

- – HWS3 has no aging value and a non-cached ACEE is created for users that use this TMEMBER client

326

The /DISPLAY OTMA command can be issued to display the ACEE aging value for the OTMA client/member. OTMA uses this value to determine if a cached ACEE for a userid needs to be refreshed. In this example, the aging value for HWS1 is 3600 seconds. And the aging value for HWS2 is 7200 seconds. Any cached ACEE that only HWS1 uses has an aging value of 3600 seconds. Any cached ACEE that only HWS2 uses has an aging value of 7200 seconds. If a cached ACEE is used by both HWS1 and HWS2, then the aging value is 3600 seconds, which is the lower value between 7200 and 3600 seconds.

Since no aging value is specified for HWS3, IMS always creates a non-cached ACEE for users from HWS3.

Considerations

- Performance
  - For multiple OTMA clients, e.g., multiple copies of IMS Connects and WebSphere MQs
    - Potential reduction of total number of RACROUTE REQUEST=VERIFY (RACF I/Os) to create ACEEs for userids
  - When an expired RACF ACEE is detected for an input transaction
    - An asynchronous request updates the cached ACEE
    - The input transaction gets an ACEE for transaction authorization
      - Deleted when authorization is complete

- Operational
  - Refresh commands
    - /SECURE OTMA REFRESH TMEMBER *membername* and /SECURE OTMA REFRESH
      - Same effect
        - Both commands work on the one and only one OTMA ACEE table for all users.

327

Since cached ACEEs are applicable to all OTMA TMEMBER clients, the requirement to issue RACROUTE REQUEST=VERIFY requests to create ACEEs for userids for a new OTMA client instance is reduced thereby also reducing possible RACF I/Os.

When an input transaction message carries a userid whose ACEE has expired, OTMA uses an asynchronous request to create a new ACEE for this userid which will be kept in the cache for subsequent transaction requests. For the current transaction request, an ACEE is created for the authorization process and deleted when the authorization is complete.

From an operational perspective, the /SECURE OTMA REFRESH TMEMBER *membername* and /SECURE OTMA REFRESH commands have identical impacts because both commands work on the single OTMA ACEE table that

IBM

Considerations …

- Security
  - OTMA detects obsolete RACF ACEEs via an internal IMS timer
  - Every two minutes, OTMA performs an ACEE cleanup operation for 10 expired userids
  - Reduction of maximum Aging Value to 11.5 days could increase RACF I/Os if more refreshes are done

- Environments, e.g, IMSplex, with mixed IMS Versions
  - Could have wide differences in the maximum aging values for ACEE refreshes
    - IMS 12 – 999999 seconds
    - IMS 10 and IMS 11 - 2147483647 seconds

328

To reduce space consumption for obsolete ACEEs, OTMA performs a cleanup every two minutes.

IMS 12 changes the input ACEE aging value to 999999 seconds (11.5 days). This could increase the number of RACF I/Os and impact performance if more refreshes are done based on the aging value.

In a mixed version environment, the difference in maximum aging value could cause an ACEE to be refreshed in an IMS 12 system but not in the IMS 10 or IMS 11 environments.

IBM

Benefits

- Cached ACEEs
  - Reduce the system storage requirements while providing better security and performance
    - Only one copy of the ACEE instead of multiple per OTMA client
      - Reduced storage usage
      - Reduced security exposure
      - Improved performance
  - Provide consistency
    - Same security result regardless of which OTMA client is used
- Lower maximum ACEE aging value
  - Triggers faster ACEE cache refresh
    - Reduces security exposure, e.g., userid is revoked or access permissions are changed

329

IBM

OTMA DFS2082 for CM0

DFS2082I for CM0

- CM1 (Send-then-Commit) transactions rely on DFS2082
  - To end the outstanding wait if the IMS transaction does not send IOPCB reply
- Conversion from the use of CM1 to CM0 (Commit-then-send)
  - For remote programs waiting for a reply
    - May result in a hang until timeout if there is no IOPCB reply

- Enhancement
  - A new commit-then-send (CM0) optional flag to request DFS2082
    - Specified on an input CM0 transaction message
      - Triggers OTMA to send the DFS2082 message if
        - The IMS application does not reply to the IOPCB
        - Nor message switches to another transaction
    - Does not apply if the transaction is a switched-to program in a program-program switch scenario
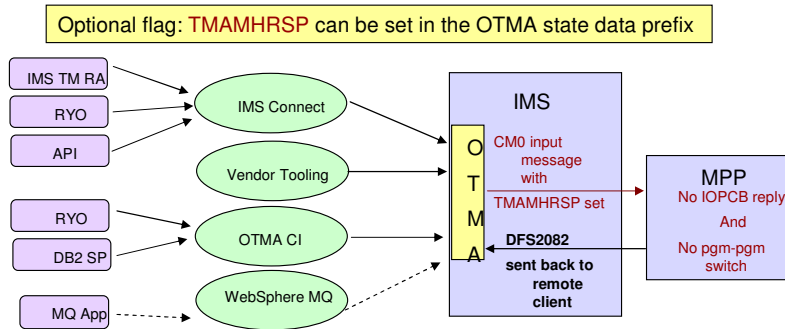
331

The DFS2082I (Response Mode transactions terminated without reply) message is a mechanism to release an outstanding wait for CM1 (Send-then-Commit) transactions when the IMS application flow (across program-to-program switches) terminates without inserting a reply to the IOPCB.

When converting remote programs that use CM1 to CM0 (Commit-the-Send), a problem can arise. Prior to IMS 12, a remote program that sends a CM0 transaction message then waits for a reply could possible wait a long time, until a timeout occurs, if the IMS application is one that does not reply to the IOPCB.

IMS OTMA in version 12 resolves this potential problem by introducing a new commit-then-send (CM0) optional flag to request the DFS2082 message. When this new flag is specified for an input commit-then-send transaction, and the IMS application does not reply to the IOPCB nor does it message switch to another transaction, OTMA will send a DFS2082 message to the client regardless of the IMS transaction response mode. This DFS2082 message for a commit-then-send transaction will only occur for the original input transaction and does not support the program-to-program switch. This restriction means that there will be no DFS2082 message for a switched to transaction, even if the switched to transaction fails to reply and the original transaction does not reply either.

Implementation

Optional flag: TMAMHRSP can be set in the OTMA state data prefix

IMS TM RA
RYO
API
→ IMS Connect →

Vendor Tooling →

RYO
DB2 SP
→ OTMA CI →

MQ App ⇢ WebSphere MQ ⇢

IMS

O T M A

CM0 input message with TMAMHRSP set

DFS2082 sent back to remote client

MPP
No IOPCB reply
And
No pgm-pgm switch

– IMS Connect exploitation
  • New IRM flag IRM_F3_DFS2082 allows CM0 client applications to request the DFS2082 msg
    – Connect sets the TMAMHRSP flag in the OTMA state data header
      • Message exit routines HWSSMPL0 and HWSSMPL1
– IMS TM Resource Adapter
  • New InteractionSpec property CM0Response
    – Internally, IMS TM RA sets the TMAMHRSP flag in the OTMA state data header which is passed to IMS Connect and on to IMS
– WebSphere MQ is looking at the new capability

332

The new optional flag TMAMHRSP is for OTMA input commit-then-send transactions (CM0) only. When TMAMHRSP is specified for an input send-then-commit (CM1) transaction, IMS OTMA ignores it. The DFS2082 message has been supported for OTMA send-then-commit messages without the need of setting any OTMA message prefix flag.

IMS Connect exploits this function by introducing a new IRM flag, IRM_F3_DFS2082, which allows the client applications to request the DFS2082 message for a CM0 input transaction. After the flag is set and the IMS application does not reply to the IOPCB nor message switch to another transaction, OTMA will send a DFS2082 message to the customer regardless of the IMS transaction response mode so that the customer application does not need to wait for the timeout. This DFS2082 message for a CM0 input will only occur for the original input transaction and would not support IMS program-to-program switches.

IMS TM RA also exploits this function by introducing a new InteractionSpec property, CM0Response, which allows customers to request the DFS2082 message for a CM0 input transaction. After the flag is set and the IMS application does not reply to the IOPCB nor message switch to another transaction, OTMA will send a DFS2082 message to the customer regardless of the IMS transaction response mode so that the customer application does not need to wait for the timeout. This DFS2082 message for a CM0 input will only occur for the original input transaction and does not support IMS program-to-program switches.

Benefit

- DFS2082 for CM0
  - Eases the CM1 to CM0 application conversion
    - Reduces the unnecessary timeout in remote applications
      - Which impacts the performance if the remote application has to wait for a timeout before continuing to the next request

This new enhancement reduces possible timeouts especially when migrating remote applications from the use of CM1 to CM0.

IBM

OTMA Performance Enhancement

OTMA Performance

- Reduced path length for OTMA transaction processing
  - Simplification in logic when validating a TPIPE name
    - Only when a new tpipe name is received on a message
      - Instead of when each message is received
  - APARs PM20292 (V10) / PM20293 (V11)
    - Shipped with the ICAL enhancements

- Benefit
  - Improved OTMA performance

335

Prior to this performance enhancement, OTMA experiences unnecessary CPU overhead because it always validates a tpipe name against keyword names when each message is received. With the enhancement, OTMA performs tpipe validation only when a new tpipe name is received. For the subsequent transactions using the same tpipe name, OTMA no longer does the tpipe name validation checking. Even for OTMA tmember clients like IMS Connect that only use 1 port tpipe for CM1 messages, validation checking will only be invoked once rather than for every message from that port tpipe.

IBM

V11 SPE – Transaction Expiration

and

WebSphere MQ V7.0.1 Support for
Message Expiry

And Enhancements in WMQ V7.1.0

336

IBM

DFS3688I

- IMS Transaction Expiration SPE

  - APARs PM05984 (IMS10) / PM05985 (V11) / PM46829 (V12)
    - Sends DFS3688I message <u>instead of</u> DFS555I or DFS2224I message for transaction expiration during application GU phase

      DFS3688I Transaction *aaaaaaaa* expired: EXPRTIME=*nnnnnn*, ELAPSE=*ssssss*
      Tmember xxxxx Tpipe xxxx

    - Enhancement only affects **OTMA** messages

  - Expired non-OTMA messages already receive DFS3688I
    - PK86426/UK47070 (V11) – non-OTMA transaction expiration is V11 only
      DFS3688I Transaction *aaaaaaaa* expired: EXPRTIME=*nnnnnn*, ELAPSE=*ssssss*

337

The transaction expiration support in IMS allows input messages whose time in IMS has exceeded an expiration value to be discarded prior to processing. The expiration value in IMS can either be set in the IMS definitions for the transaction (transaction-level) or, for OTMA interactions, through an override expiration value on a message by message basis (message-level).

The IMS Transaction Expiration SPE affects the IMS message that is sent out when an expired transaction message is detected during the application GU phase.   Instead of the traditional DFS555I message (DFS2224I if the situation is detected in a back-end Shared Queues IMS), message DFS3688I is sent. This enhancement adds the Tmember Tpipe information.

The DFS3688I message response for expired non-OTMA messages was included in a previous fix PK86426/UK47070 for IMS version 11.

IBM

MQ Message Expiration

- Extension of the WebSphere MQ (WMQ) Message Expiry facility to include the IMS transaction expiration function (WMQ 7.01)
  - A new service parameter
    - CSQ6SYSP SERVICE = 0000000001 or also specified through the SET SYSTEM SERVICE(0000000001) command

      - Used in conjunction with other queue manager service parameters
        - e.g. if queue manager already uses service parm 0040 then setting the new service would result in 0040000001

      - Provides toleration of an OTMA NACK_FOR_TRANS_EXPIRED response from IMS through the OTMA support
        - Leverages WMQ expiry processing as if the message had expired prior to sending the message to OTMA
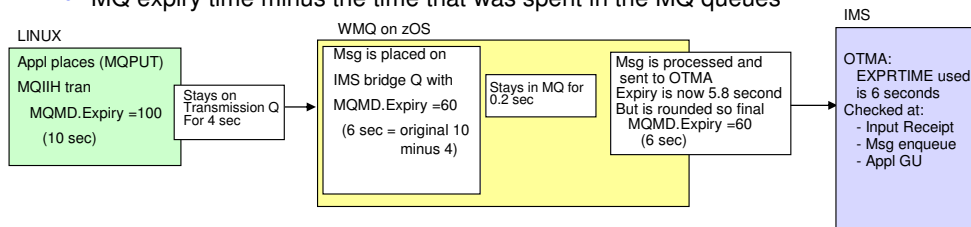
338

Support for WebSphere MQ (WMQ) includes enhancements to both IMS and WMQ.

In WMQ, the Message Expiry facility provides a new service parameter that indicates whether or not WMQ will take advantage of the OTMA message expiry interface. To enable passing the MQ expiration time to IMS, the service parameter value is represented by the hex value of 0000000001 which is set either by compiling into the queue manager's ZPARM parameter module via the CSQ6SYSP macro or by issuing the SET SYTEM SERVICE command. This service parameter setting allows WMQ to tolerate the receipt of an OTMA NACK_FOR_TRANS_EXPIRED response when WMQ passes a message via OTMA to IMS. This response can be received during times of stress and OTMA discovers the transaction message has expired in IMS.

MQ Message Expiration …

– User-Specified Expiry time (message-level)

- A value is passed to IMS if an MQ message expiry time (MQMD.Expiry) exists for the message AND the service parameter is set
  - Value is in 10ths of a second
  - **The residual expiry time for the message is built into the OTMA interface**
    - MQ expiry time minus the time that was spent in the MQ queues

LINUX

Appl places (MQPUT)
MQIIH tran
MQMD.Expiry =100
(10 sec)

Stays on Transmission Q For 4 sec

WMQ on zOS

Msg is placed on IMS bridge Q with MQMD.Expiry =60
(6 sec = original 10 minus 4)

Stays in MQ for 0.2 sec

Msg is processed and sent to OTMA Expiry is now 5.8 second But is rounded so final MQMD.Expiry =60 (6 sec)

IMS

OTMA:
EXPRTIME used is 6 seconds
Checked at:
- Input Receipt
- Msg enqueue
- Appl GU

From the remote application perspective (business as usual):
- The MQPUT application will be unaware of an expiry unless it specifies a Report option which can specify

–MQRO_EXCEPTION (Just the Expiration report)
–MQRO_EXCEPTION_WITH_DATA (First 100 bytes of the message)
–MQRO_EXCEPTION_WITH_FULL_DATA (All of the message)

IMS transaction expiration for an OTMA message Is checked in 3 phases:
- When the message is received (input receiving phase)
- When it is queued (enqueuing phase)
- When the IMS application issues GU (GU phase)

339

Any MQ message can specify a unique expiration time to be associated with it. The desired expiration interval is passed by the application to MQ using the MQMD.Expiry field. The time is expressed in 10ths of a second and can be thought of as a 'time to live' for the message. An attempt to perform an MQGET of a message later than the expiration time has expired results in the MQ message being removed from the queue and the execution of expiry processing. As an MQ message flows between queue managers in an MQ network, the 'time to live' expiration field is decremented even while the message waits on a transmission queue for movement between queue managers. MQ expiry processing is driven by the Report options specified by the application in the MQMD header of the message. Options include: MQRO_EXCEPTION (Just the Expiration report), MQRO_EXCEPTION_WITH_DATA (First 100 bytes of the message), MQRO_EXCEPTION_WITH_FULL_DATA (All of the message).

The putting application can request a report on expiration, and will get a message sent to the replyto queues with the expiration information

If the service parameter for OTMA transaction expiration is set, then WMQ also looks to see if an MQ expiry time has been defined for the message. If an MQ expiry time value exists, then WMQ calculates the residual value (MQ expiry time minus the time the message has already spent in MQ) and uses that value when building the OTMA interface. If no MQ expiry value has been set then WMQ does not send a user-specified value to IMS.

IBM

MQ Message Expiration …

- IMS Transaction Expiration (transaction-level)

    - Set in IMS
        - TRANSACT macro EXPRTIME value
        - IMS DFSINSX0 user exit routine
        - DRD commands: CREATE TRAN or UPDATE TRAN
    - Implication
        - WMQ application is not aware of an IMS expiration value or that the message may expire even without an MQMD.Expiry specification

    - Can be overriden by MQ Message Expiry value in the message

340

In IMS, the transaction may have been defined with an expiration time value on: the TRANSACT macro EXPRTIME specification, the IMS DFSINSX0 user exit or via IMS CREATE or UPDATE TRAN commands. This means that transaction expiry can be activated for a transaction without the MQ application being aware that the transaction may expire.

Message Expiry

- **Transaction (Input) Message Expiry in IMS (3 phases)**

  - During the first 2 phases: Input Receiving Phase or Enqueuing Phase
    - NAK is sent to the WMQ IMS Bridge with OTMA sense code x'0034'
      - Reason code
        - X'0001' - transaction was cancelled right after OTMA receives it from XCF.
        - X'0002' - transaction was cancelled before OTMA enqueues it.
      - WMQ treats this as if the message had expired before being sent to IMS
        - MQMD_REPORT options are honored

    - In IMS, the input message is flagged as expired and then discarded
      - A 67D0 log record that indicates OTMA detected an expired transaction and canceled the input message

341

IMS checks the eligibility for message expiry in three phases: These include the Input Receiving Phase, Enqueuing Phase, and GU Phase.

If a message expires during either the Input Receiving Phase when the input message is received or during the Enqueuing Phase which occurs before OTMA enqueues the transaction, then a NAK is sent back to WMP with an OTMA reason code of x'0034'. WMQ treats this response much as if the message had expired before being sent to IMS and MQMD_REPORT options that were originally set in the MQPUT are honored. Additionally, either a reason code of either x'0001' or x'0002' is sent to denote in which specific phase the message expired. Additionally, IMS writes a 67D0 log record.

IBM

Message Expiry …

- Transaction Input Message Expiry in IMS…

    - During GU Phase (IMS application issues GU,IOPCB)
        - Abend 0243, and
        - DFS3688I message with tmember and tpipe names is sent to the MQ IMS bridge
            - With flag TMAMEXPR EQU X'04', set in the byte 2 of OTMA control data prefix for the message, indicating 'message expiry'
            - The DFS3688I message is returned to the WMQ Reply_To Queue
        - 67D0 log record that indicates OTMA detected an expired transaction and canceled the input message

        - In WMQ
            - The MQMD_REPORT options are NOT honored

342

If the message expires during the GU phase, i.e., when the message is scheduled into a dependent region and the IMS application issues a GU to retrieve the message, then the application gets a 0243 abend, IMS sends a DFS3688 message back to WMQ and sets the TMAMEXPR flag in the OTMA control data prefix. A 67D0 log record is written to indicate that the input message was canceled.

Message Expiry …

- **WMQ 7.1.0** enhances the support for IMS Transaction Expiration
  - If the transaction expires at the GU to the IOPCB
    - OTMA returns the original input message to MQSeries instead of the DFS3688I message
    - The MQMD_REPORT options are honored
    - This enhanced functionality is also delievered in WMQ 7.0.1 via APAR PM47795

- Note: The Reply message can also Expire
  - MQIIH_FLAGS support a value of MQIIH_PASS_EXPIRATION
  - WMQ will pass the REMAINING expiry time in the OTMA header
  - The reply on the reply-to queue will start the Expiry process with that remaining time
    - Any time in IMS is NOT counted

343

With WMQ 7.1.0 and an APAR to WMQ 7.0.1 the WMQ expiry process has been enhanced to support the MQMD_REPORT options.

IBM

Migration and Benefits

- DFS3688I
  - Applications/users will see a DFS3688I message instead of DFS555I/DFS2224I when an input message is discarded in GU Phase

- WMQ support
  - When Enabled
    - Review the MQMD_REPORT options
    - Check the Reply_To queue for DFS3688I messages

- Benefits
  - Extends IMS transaction expiration function to WMQ
  - Standardizes the message (DFS3688I) that is sent out to remote clients when the transaction input message has expired