

Operations

ON DEMAND BUSINESS™

© 2008 IBM Corporation



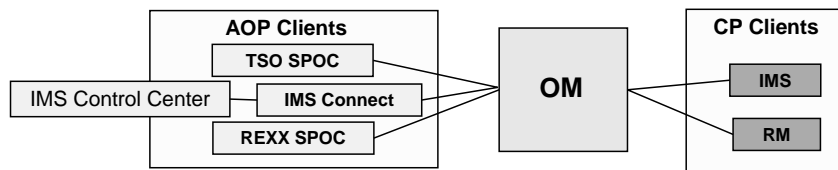
Operations Enhancements

- OM Audit Trail and Unsolicited Messages
- SPOC and REXX support
- Batch SPOC Utility
- REXX XML Parser

OM Audit Trail and Unsolicited Messages

OM Terminology Review

- A command processing (CP) client
 - ◆ An IMSplex member that accepts and processes commands entered by another member through the OM API
 - Examples are the IMS control region and RM
- An automated operator program (AOP) client
 - ◆ An IMSplex member that submits commands through the OM API to a CP client
 - Most common example is TSO SPOC
 - Other examples are REXX programs using the REXX SPOC API, IMS Connect (for IMS Control Center), vendor or user-written SPOC



4




As a review, OM has command processing clients and automated operator program clients.

Operations Manager Enhancements

- Operations Manager provides two enhanced functions in IMS V10
 - ◆ Create an audit trail of IMS commands entered through the OM interface by all AOP clients (e.g., TSO SPOC)
 - Audit trail
 - Input commands from AOP clients
 - Responses by CP clients to input commands
 - Unsolicited messages
 - Consists of log records written to z/OS System Logger log stream
 - ◆ Handle unsolicited messages from an IMSplex member (e.g., IMS)
 - Messages generated by IMSplex members can be sent to OM and delivered by OM to an OM “AOP Client”
 - AOP client must subscribe to OM for unsolicited messages
 - Shares log stream with audit trail

5

In IMS V10, OM provides two new functions: (1) an audit trail of commands and (2) support for unsolicited messages.

OM Audit Trail

- OM audit trail uses the z/OS System Logger
 - ◆ Specified in CSLOIxxx OM initialization proclib member
 - IMSPLEX(NAME=plexname,AUDITLOG=log stream name)
 - ◆ Can be CF structure log stream (multiple OMs) or DASD-only log stream (single OM)
 - ◆ Contains all OM log records from all OMs that are connected to this log stream within the IMSplex
 - Recommendation: no more than two OMs should share a log stream due to duplicate copies of unsolicited messages
- Command input logged before OM calls the OM input user exit
 - ◆ Logged again if text modified by OM input user exit
- Output logged before OM calls the OM output user exit
 - ◆ Logged again if text modified by OM output user exit

6




The OM Audit Trail is implemented with an MVS logger log stream. It is specified via the AUDITLOG=log stream name in CSLOIxxx. The log stream can be either in a CF (for multiple IMSs) or DASD-only (for a single IMS). OM does not require that the same MVS log stream be used within an IMSplex. You may have multiple OMs use the same log stream or they may each have their own log stream. When multiple OMs use the same log stream a CF is required. The audit trail will be maintained in chronological sequence.

Each OM will receive a copy of unsolicited messages. If multiple OMs share a log stream, it will contain multiple copies of unsolicited messages. For this reason, IBM recommends that no more two OMs share a log stream to avoid many duplicate messages. On the other hand, the advantage of a shared log stream is that all messages are in one stream.

The OM input user exit is an optional exit that allows a user to view and manipulate command input from an automation client. This exit is called before OM processes the command, which allows the command to be modified or rejected.

The OM output user exit is an optional exit that allows a user to view and manipulate output from OM. This exit is called when a command has been processed and is ready to be delivered to the originator of the command; the exit can modify the response text before it is delivered, or when an unsolicited message is received from a client using the CSLOMOUT API.

Based on whether or not the optional OM input and OM output exits modify the text, each input command or output response will have one or two entries in the audit trail.

OM Audit Trail

- Recommended CFRM and LOGR policy definition values:
 - ◆ CFRM policy
 - SIZE (8192)
 - ◆ LOGR policy
 - STRUCTURE
 - LOGSNUM (1)
 - AVGBUFZSIZE (4000)
 - MAXBUFSIZE (32760)
 - LOGSTREAM
 - LOWOFFLOAD (20)
 - HIGHOFFLOAD (50)

7

The recommended CFRM and LOGR policy definition values are shown.




The recommended structure size definition in the CFRM policy is SIZE(8192). This is specified in Kbytes, so it creates an 8M structure.

The DEFINE STRUCTURE statement in the LOGR policy should include:

```
LOGSNUM ( 1 )
AVGBUFZSIZE ( 4000 )
MAXBUFSIZE ( 32760 )
```

The DEFINE LOGSTREAM statement in the LOGR policy should include:

```
LOWOFFLOAD ( 20 )
HIGHOFFLOAD ( 50 )
```

Printing the new OM Audit Trail

- Use the IMS File Select and Formatting Print Utility (DFSERA10) with exit routine CSLULALE (formatted) or CSLOERA3 (dump format)
- Sample JCL

```

//CSLERA10      JOB      MSGLEVEL=1,MSGCLASS=A,CLASS=K
//STEP1        EXEC     PGM=DFSERA10
//STEPLIB      DD      DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT     DD      SYSOUT=A
//SYSUT1       DD      DSN=SYSLOG.OM.AUDIT.TRAIL.LOG,
//              SUBSYS=(LOGR,IXGSEXIT),
//              DCB=(BLKSIZE=32760)
//SYSIN        DD      *
CONTROL CNTL   H=EOF
OPTION         PRINT   EXITR=CSLULALE|CSLOERA3
END
//
  
```

8

This is an example of how to use DFSERA10, the IMS File Select and Formatting Print Utility, to print the OM Audit Trail. The JCL requirements are:

- STEPLIB - DSN points to IMS.SDFSRESL, which contains DFSERA10
- SYSUT1 - DSN= points to the log stream name that was specified in the CSLOlxxx PROCLIB member on the AUDITLOG parameter
- H= - specifies the number of log records to print (H=EOF to print all records)
- EXITR=CSLxxxxx – specifies the log record routine that is called to format each log record

OM log record formats are available by assembling mapping macro CSLOLGRC.

Each OM log record contains a log record prefix, followed by data that is unique to the record. Macro CSLZLGPF maps the log record prefix.

The audit trail log records are:

Log Record

Type	Subtype	Mapping Macro	Conditions for writing log record
X'06'	X'01'	CSLOLGCM	Command input, prior to calling the OM Input user exit
X'06'	X'02'	CSLOLGCM	Command input, after being modified by the OM Input user exit
X'08'	X'01'	CSLOLGCR	Command response output, prior to calling the OM Output user exit
X'08'	X'02'	CSLOLGCR	Command response output after being modified by the OM Output user exit
X'09'	X'01'	CSLOLGOU	Unsolicited output message to an AOP client before calling OM Output user exit
X'09'	X'02'	CSLOLGOU	Unsolicited output message to an AOP client after calling OM Output user exit and OM output user exit modified the unsolicited output message

The documentation for printing the OM audit trail using CSLULALE with DFSERA10 for formatted output is in the IMS 10 System Utilities Reference; the documentation for printing the OM audit trail using CSLOERA3 with DFSERA10 for dump output is in the IMS 10 Diagnosis Guide.

Information On Demand IMS Version 10

Sample Listing of OM Audit Trail using CSLULALE

IMSpIex
member

Date and time of
log record

Formatted
message text

```

OM1OM 2006.299 19:21:52.47 CSL0020I OM  READY OM1OM
RM1RM 2006.299 19:22:43.91 CSL0020I RM  READY RM1RM
IMS1 2006.299 19:23:20.43 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1 2006.299 19:23:20.86 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1 2006.299 19:23:20.86 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1 2006.299 19:23:23.45 DFS3613I - LUM TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.61 DFS3613I - RLM TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.61 DFS3613I - XCF TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.61 DFS3613I - RLM TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.62 DFS3613I - ALC TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.62 DFS3613I - ALM TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.62 DFS2088I APPC/OTMA SMQ Enablement inactive. Reason
IMS1 2006.299 19:23:23.62 DFS2088I APPC/OTMA SMQ Enablement inactive. Reason
IMS1 2006.299 19:23:23.65 DFS3613I - RRS TCB INITIALIZATION COMPLETE IMS1
IMS1 2006.299 19:23:23.66 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1 2006.299 19:23:23.66 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM

```

9

This shows the listing created by DFSERA10 with exit routine CSLULALE (formatted output) for messages in the OM audit trail.

Information On Demand IMS Version 10

Sample Listing of OM Audit Trail using CSLOERA3 (command input/output)

```

0601 RECORD - 2006-09-15 22:34:01.721325 UTC - CSLOMCM UNMODIFIED BY OM INPUT EXIT

000000 00000187 06010000 BF695E7B 429ED597 .... *...g.....;#..NpCSLPLEX10130....*
000020 0B9153 4E78001 0A9A5288 BF695E7B .... *.j..67.....h..;#....USRT011 ....*
000040 00084040 40404040 40404040 00000000 .... *..          USRT011 .....USRT011 *
000060 00000000 00000000 00000000 00000000 .... *.....*
      LINES 00000000 00000000 00000000 00000000
0000C0 00FD4B98 00000000 7FFFF000 7FFFF000 .... *...q....".0.".0.....;#*
0000E0 429ED597 40404040 40404040 00000000 .... *.Np.....*
000100 00000000 00000000 00000000 00000027 .... *.....QRY TRAN NAME(AP*
000120 D6D35C5D 40E2C8D6 E64DC3D3 C1E2E26B .... *OL*) SHOW(CLASS,STATUS).....*
000140 00000000 00000000 00000000 00000000 .... *.....*
      LINES 000160 TO 00017F SAME AS ABOVE
000180 00000000 00000000 ..... *.....*

0801 RECORD - 2006-09-15 22:34:01.721325 UTC - CSLOMRSP UNMODIFIED BY OM OUTPUT EXIT

000000 00000A9C 08010000 BF695E7B 429ED597 .... *.....;#..NpCSLPLEX10130....*
000020 0B9153 4E78001 0A9D3A08 BF695DF3 .... *.j..67.....)3....USRT011 ....*
000040 00084040 40404040 40400000 00000000 .... *..          .....*
000060 E4E2D9E3 F0F1F1 F0F1F5 F3F4F0F1 .... *USRT011 10153401..)3|...OMIOM *
000080 00000000 00000000 ..... *.....*
      LINES 0000A0 TO 0000
0000C0 00000000 00000000 ..... *.....;#..Np*
0000E0 40404040 40404040 00000000 00000000 .... *.....*
000100 00000000 00000000 00000990 4C6FA794 .... *.....<?xml version="1.0"?*
000120 6E4C5AC4 D6C3E3E8 D7C54089 94A296A4 .... *><!DOCTYPE imsout SYSTEM "imsout*
000140 4B84A384 7F6E4C89 94A296A4 A36E4C83 .... *.dtd"><imsout><ctl><omname>OMIOM*
000160 4040404C 61969495 8194856E 4C9694A5 .... *  </omname><omvsn>1.3.0</omvsn>*
000180 4CA79493 A5A2956E F2F04040 4C61A794 .... *<xmlvsn>20  </xmlvsn><stotime>20*
0001A0 F0F64BF2 F5F840F2 F27AF3F4 7AF0F14B .... *06.258 22:34:01.721325</stotime>*
0001C0 4CA2A396 A3899485 6EF2F0F0 F64BF2F5 .... *<stotime>2006.258 22:34:01.72226*

```

10

This shows the listing created by DFSERA10 with exit routine CSLOERA3 (dump format) for a QRY TRAN command where the OM input and OM output exits did not modify the messages. Since this is in XML format, it is very large and only the beginning of the output record is shown here.

Unsolicited Message Support

- An unsolicited message is a message that is issued by a CP client such as IMS but is not associated with a specific command input

- OM support for unsolicited messages
 - ◆ Logs unsolicited messages to “audit trail log stream” defined in CSLOIxxx
 - ◆ If requested, will send these messages to a user-provided AOP client

- AOP client must use new OM requests to Subscribe and Unsubscribe to unsolicited messages
 - ◆ Implemented by REXX SPOC API
 - ◆ Not yet implemented by IMS Control Center

11

A command processing client will use the CSLMOUT API to send an unsolicited message.

The IMS-provided TSO SPOC does not use Subscribe/Unsubscribe support.

Command responses are sent as unsolicited messages to any AOP clients that have asked for them (subscribed), in addition to sending the command response back to the client that issued the command.

Unsolicited Message Support




- Two new OM requests for assembler language AOP clients
 - ◆ CSLOMSUB (subscribe)
 - Issued by an AOP client to subscribe to unsolicited messages
 - OM will send unsolicited messages only to subscribed clients
 - AOP client must have an SCI Input Exit to receive the unsolicited output message
 - Messages go to OM Output User Exit (if present) first which can edit the message or indicate not to send the unsolicited output
 - ◆ CSLOMUSB (unsubscribe)
 - Issued by an AOP client to unsubscribe from unsolicited messages
 - OM no longer sends unsolicited messages to this AOP

**Format and parameters for above macros
shown in System Programming API Reference**

12

OM provides two new requests for unsolicited message support for assembler language programs: CSLOMSUB and CSLOMUSB.

REXX support is shown in the next section.



Unsolicited Message Support - Tailoring

- DFSDFxxx or DFSCGxxx PROCLIB member UOM= parameter
 - ◆ Controls which IMS unsolicited output messages are sent to OM
 - UOM=MTO - only unsolicited messages destined for MTO or console are sent
 - UOM=ALL - all unsolicited messages are sent
 - UOM=NONE - no unsolicited messages are sent
- Optional user modifiable tables
 - ◆ Indicates whether or not to send a message to OM as unsolicited output
 - Default is to send all messages
 - ◆ CSLZUMTU – CSL Unsolicited Output Messages
 - ◆ CQSUOMTU – CQS Unsolicited Output Messages
 - ◆ DFSUOMTU – IMS Unsolicited Output Messages
- Benefits
 - ◆ May not need messages for automation
 - ◆ May not need messages for auditing

13

There are two ways to limit the unsolicited messages sent to OM.

IMS messages may be limited by using the UOM= parameter in the DFSDFxxx or DFSCGxxx PROCLIB member. UOM=MTO is the default. It specifies that the only unsolicited messages sent to OM from IMS are those that IMS sends to the MTO or system console. UOM=ALL specifies that all unsolicited messages from IMS are sent to OM. UOM=NONE specifies that no unsolicited messages are sent to OM from IMS.

There are optional user modifiable tables for IMS, CSL, and CQS which may be used to send or not send messages based on their message number. You may specify whether messages whose numbers are specified should be sent or not sent. You may also specify whether messages whose numbers are not specified in the table should be sent or not sent.

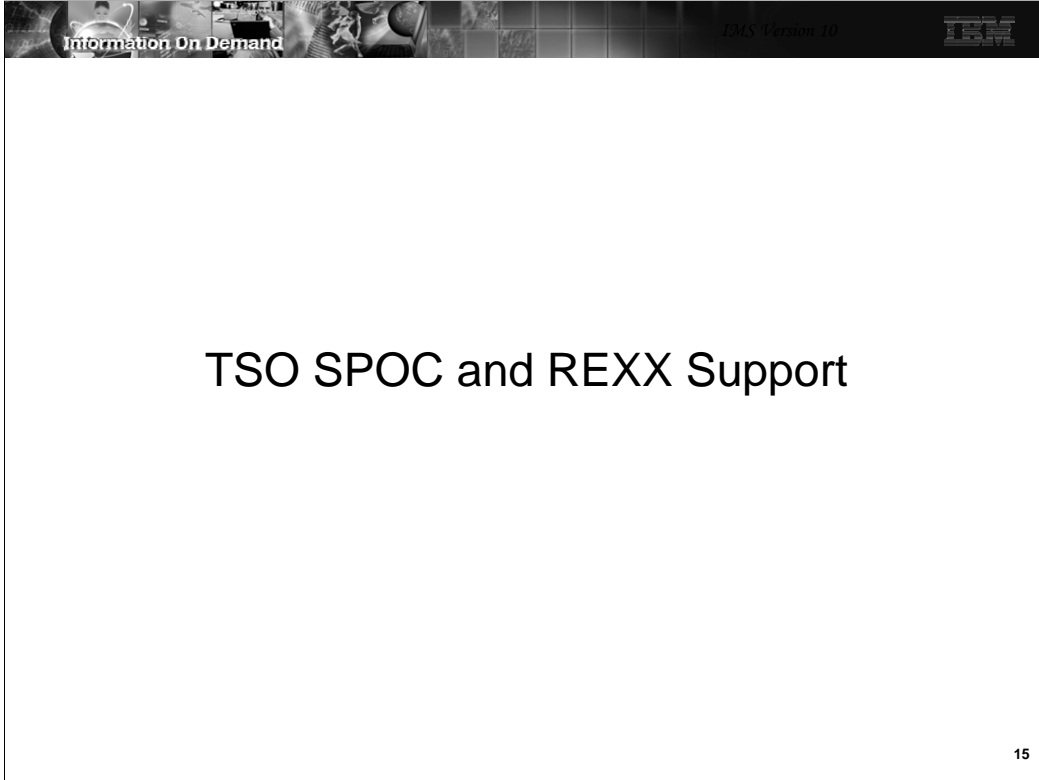
Of course, if messages are not sent to OM they will not appear in the audit trail.

You may want to limit the unsolicited output messages sent to OM for several reasons. You may only need some messages from to AOP processing. You may only need some for auditing purposes. You may not need any messages.

OM Audit Trail Migration Considerations

- OM audit trail
 - ◆ Enabled by specifying new **AUDITLOG=log-stream-name** parameter in the CSLOIxxx (OM Initialization) PROCLIB member
 - ◆ Must define log stream in LOGR policy
 - ◆ Must define logger structure in CFRM policy (if using CF)
 - ◆ For OMs using the same AUDITLOG parameter (same log stream)
 - Log records are merged in chronological sequence
 - Warning: log stream will contain duplicate copies of unsolicited output messages
 - ◆ Consider user modifiable tables and UOM= parameter to limit output sent to OM

- Processing the audit trail
 - ◆ All types of messages can be retrieved by TSO SPOC or REXX SPOC
 - ◆ Should consider procedures to handle archiving z/OS log stream output if you need to retain its information



This section addresses enhancements to the SPOC ISPF application and enhancements to REXX support for the OM interface.



TSO SPOC Support for OM Enhancements

- TSO SPOC enhanced to display audit trail log stream
 - ◆ “SPOC” function on Action Bar
 - Replaces “Display” function of previous releases
 - New option (6) added to display messages in “Audit Trail” log stream
 - Reads directly from log stream
 - Does not use OM services for this function
 - Requires local IXGLOGR address space
 - Commands entered through OM and their responses
 - Includes “non-deliverable” command responses received late by OM (WAIT time expired)
 - Unsolicited messages received by OM
- TSO SPOC Preferences
 - ◆ New Audit Trail Preferences allow user to set defaults for IMSplex member name or IMSplex member type

More information and examples are in the TSO SPOC Appendix

16

The TSO SPOC program has been enhanced to support the new OM “audit trail” of command input/response through the OM and unsolicited messages sent by IMSplex members to OM. Note that the function of SPOC reads directly from the Audit Trail logstream. It does NOT retrieve any of these messages directly from OM. This allows the user to scroll back and forth through the entire logstream, going as far back in time as those log records still in the logstream. This does require that the SPOC be on an LPAR which has an ISGLOGR address space to which it can connect and request log records.

REXX Support for Audit Trail

- Users can write REXX programs to retrieve messages from the “Audit Trail” log stream
 - ◆ Commands and responses using OM interface
 - ◆ Unsolicited messages sent to OM by IMSplex members

- User programs must
 - Subscribe to OM for messages
 - Retrieve and process messages
 - Unsubscribe from OM

- Only messages received by OM after subscription are sent
 - ◆ Unlike SPOC which can look at messages in the past

17

You can also write a REXX program to

-join the IMSplex

-subscribe to OM for unsolicited messages (actually, for any logstream message that arrives after you subscribe, including command input and response)

-retrieve the messages into a REXX stem variable

-unsubscribe from OM when you want to exit

Unlike the TSO SPOC application, this feature receives its messages from OM directly – not from the system logger. OM will only send messages received AFTER the program subscribes.

REXX Functions

- Three new REXX functions are provided
 - ◆ CSLULSUB
 - Subscribe to OM for messages
 - ◆ CSLULGUM
 - Get one or more messages from OM (since last CSLULGUM)
 - Repeat as needed
 - ◆ CSLULUSB
 - Unsubscribe from OM

18

There are three new REXX functions provided by IMS to help you with these functions.

-CSLULSUB is a function used to subscribe to OM

-CSLULGUM retrieves messages from OM and puts them into a REXX stem variable

-CSLULUSB is used to unsubscribe to OM

Information On Demand
IMS Version 10

Sample REXX Code

```

/* rexx */
interval = 5
Call syscalls 'ON'
Address LINK 'CSLULXSB'
If rc = 0 Then
  Do
    subrc = CSLULSUB('PLEX1','IMS2')
    If subrc = 0 then
      Do a = 1 To 25
        Address syscall "sleep" interval
        results = CSLULGUM('PLEX1','xml.')
        say 'Message = ('results')'
        If xml.0 /= '' Then
          Do
            say 'xml.'0' = ('xml.'0)''
            Do idx = 1 To xml.0
              say 'xml.'idx'=('xml.'idx)''
            End
          End
        End
      End
    usbrc = CSLULUSB('PLEX1', 'IMS2')
  "END"
End
Exit

```

This program gets a message from OM every five seconds. After 25 messages, it terminates.

Only messages received by OM after subscription are sent to the program.

19

This is a sample of a REXX program which sets up the REXX SPOC environment, subscribes to OM for audit trail messages from IMS2 which is a member of CSLPLEX1.

You invoke the subscription service by coding **subrc = CSLULSUB** and its parameters. subrc will be set to the return code from this function. subrc=0 means it worked.

You retrieve the messages into the stem variable by coding - **results=CSLULGUM(plexname,stem-variable)**. The stem variable in the example is xml. but you can use anything you like.

The results are returned in a stem variable named "xml."

- xml.0 is the number of rows (lines) returned
- xml.1 is the first row
- xml.2 is the second row
- etc

It is up to the programmer to parse the responses and take action (if necessary).

Full details on coding these REXX functions is in the Systems Programming API Reference manual.

Batch SPOC Utility

Batch SPOC Utility (CSLUSPOC)

- Provides a capability to submit IMS commands from a batch job
 - ◆ Uses the Operations Manager (OM) interface
 - ◆ Supports both Type-1 and Type-2 commands
- IMSplex environment defined in execution parameters
 - ◆ IMSplex name
 - ◆ Command routing
 - ◆ Wait time
- Commands defined in SYSIN file
 - ◆ Multiple commands allowed
 - ◆ Commands executed serially
- Output to SYSPRINT
 - ◆ Responses formatted to look like SPOC screen format

21

You can now invoke a batch SPOC application and submit IMS commands from a batch job using the OM interface. Like the online SPOC, both Type-1 and Type-2 commands are supported. Execution parameters include the IMSPLEX name, the routing (default routing is all IMSs), and how long you want OM to wait for IMS to respond before returning a negative response to the SPOC. Commands are coded in a SYSIN file with each command executed serially – that is, OM submits the first command and waits for a response before submitting the second command. The output goes to SYSPRINT and looks like a formatted SPOC screen.



CSLUSPOC Parameters

- IMSPLEX= (required)
 - ◆ 5 character IMSplex name suffix
- ROUTE= (optional)
 - ◆ Defines routing for all commands in SYSIN file
 - ◆ Examples
 - ROUTE=IMS1
 - ROUTE=(IMS2,IMS4)
 - ◆ Default is all IMSs registered to OM
- WAIT= (optional)
 - ◆ Time OM waits for IMS to respond
 - ◆ Minutes and seconds (mmm:ss) or seconds (sssss)
 - ◆ WAIT=0
 - Submit next command immediately – don't wait for response
 - ◆ Default is WAIT=5:00

22

-IMSPLEX name is required. (Note: you must have a SCI address space on the image you are running on, but IMS can be on any image in the sysplex.

-ROUTE defines the routing for the command. It is optional. If not coded, the default routing will be all IMSs.

-WAIT is also optional with a default of 5 minutes (probably way too long). Specifying WAIT=0 will cause the SPOC to submit commands immediately without waiting for a response to the previous command from OM. When you specify WAIT=0, OM will wait up to 5 minutes for an IMS to respond.

Information On Demand IMS Version 10

Sample Batch SPOC Utility JCL

```

//SPOCJOB      JOB      ...
//STEP1 EXEC PGM=CSLUSPOC,PARM=( ` IMSPLEX=PLEX1,ROUTE=IMS2,WAIT=30' )
//STEPLIB      DD      DSN=IMSV10.SDFSRESL,DISP=SHR
//SYSPRINT     DD      SYSOUT=*
//SYSIN DD      *
  QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)
  QRY TRAN NAME(ACCT*) SHOW(ALL)
  /DIS DB ACCTMSTR
/*EOF

```

23

This is the JCL for the batch SPOC. The program name is CSLUSPOC. The input parameters are the IMSPLEX name, where you want the commands routed to, and how long you want OM to wait for a response from IMS before returning to the SPOC. The SPOC will then submit the next command in the SYSIN file.

The example shows the SYSIN to be inline, however, you could put the commands into a data set and identify the data set in the SYSIN DD statement. This would allow you to just update the contents of the file and resubmit the job rather than change the job input stream

If the continuation character is a plus (+) sign, then the next line is concatenated to this one without any blanks. This would be required when you are (for example) coding the SHOW parameters and they don't all fit on one line. A minus (-) sign will insert a blank before whatever is coded on the next line. For example, the next keyword should be separated by a blank from the previous one.



//SYSIN File

- Commands are read from a SYSIN file
 - ◆ Utility supports all commands supported by OM
 - ◆ Commands are executed serially
 - When OM responds to one command, the next is submitted
 - Exception: if WAIT=0, commands are submitted without waiting for response from OM

24

The utility supports all commands supported by OM. They are executed serially, waiting for the response to one before submitting the next. If WAIT=0 is coded, commands are submitted without waiting for the response.

Command continuation characters allow commands to span multiple lines. A plus (+) sign means no blank between this line and the continuation line. A minus (-) sign means to insert a blank between lines.

//SYSPRINT Output

- Contains the formatted command response from OM
 - ◆ Formatted to look like the response on a TSO SPOC display

```
Log for. . . : QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS1,IMS3
Start time. . . . : 2005.132 15:36:28.11
Stop time . . . . : 2005.132 15:36:29.17
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IMS1

IMSpIex  MbrName      CC Member   Type  Status
CSLPLEX1 OM1          0 IMS1      DBDC  ACTIVE
CSLPLEX1 OM1          0 IMS3      DBDC  ACTIVE
```

The output goes to SYSPRINT. If you print it, it will look almost like the output to the same command submitted from the TSO SPOC (with a few things omitted, like function keys).

//SYSPRINT Output

- If OM does not respond before WAIT interval, or if WAIT=0
 - ◆ Response is not written to SYSPRINT
 - ◆ Short summary information written

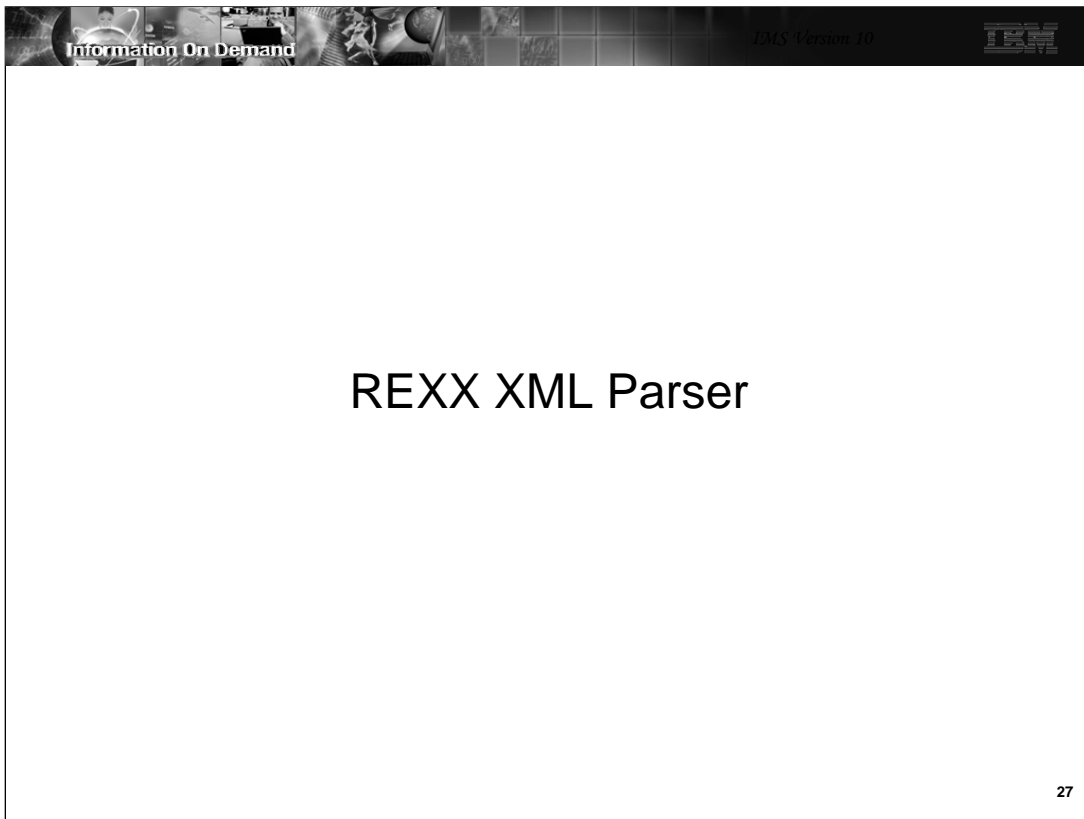
```
LOG FOR... : QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)
```

```
IMSpIex .....: PLEX1
```

```
Routing .....: IMS1,IMS3
```

```
Start time ...: 2006.185 15:36:22.11
```

If OM does not respond within the wait interval time, but responds later, then that response does not go to sysprint. Instead, a short summary page is printed as shown. This is also true if you code WAIT=0.



This section addresses an XML parser that can be used with REXX programs to process command responses from IMS.

Prior to V10

- Responses to OM-submitted commands always encapsulated in XML
 - ◆ Commercial REXX XML parser does not exist
 - ◆ Makes interpreting command responses more difficult
 - Programs must look for XML tags
- Example of QRY TRAN command using the original function

```
...  
"QRY TRAN NAME(CUS*)  
results = CSLULGTS('qryinfo.',cartid,'1:30')"  
...
```

- ◆ REXX stem variable is "qryinfo."
 - qryinfo.0 = number of rows returned
 - qryinfo.1 = 1st row
 - qryinfo.2 = 2nd row
 - etc.

28

IMS has provided support for REXX programs to join an IMSplex and submit commands to IMS using the OM interface. However, IMS always responds to these commands by encapsulating the response in XML. This made it difficult for the programmer to analyze and take any appropriate action.

The example shows a REXX function (CSLULGTS) used to retrieve the IMS response and put the "output lines" into a REXX stem variable. For example "qryinfo.1" would be the first line, "qryinfo.2" the second line, etc. "qryinfo.0" is the number of lines (or rows) returned.

Information On Demand IMS Version 10

Sample XML Output

```

<imsout>
<ctl>
<omname>OM10M </omname>
...
</ctl>
<cmd>
...
<verb>QRY </verb>
<kwd>TRAN </kwd>
<input>QRY TRAN NAME(CUS*)SHOW(PGM,QCNT) </input>
</cmd>
<cmdrsphdr>
<hdr slbl="TRAN" llbl="Trancode" ... />
<hdr slbl="MBR" llbl="MbrName" ... />
<hdr slbl="CC" llbl="CC" ... />
...
</cmdrsphdr>
<cmdrspdata>
<rsp>TRAN(CUSA ) MBR(IMS1) CC(0) CUSTADD 4 </rsp>
<rsp>TRAN(CUSQ ) MBR(IMS1) CC(0) CUSTQRY 0 </rsp>
</cmdrspdata>
</imsout>

```

Your REXX code must parse XML response to find fields of interest.

29

This slide shows a sample response to a QRY TRAN command with the XML tags. It is difficult for the average programmer to write code to analyze this response.

-qryinfo.1 = <imsout>

-qryinfo.2 = <ctl>

Somewhere down the line (line n) is the actual response that you are interested in:

- qryinfo.n = <rsp>TRAN*CUSA) MBR(IMS1) CC(0) CUSTADD 4 </rsp>

Using REXX Parser Function



- New REXX XML parser function
 - ◆ CSLULGTP
 - Use instead of CSLULGTS
 - ◆ Parses XML output directly into a “stem variable” and “suffix”
 - ◆ Makes interpreting command responses easier for REXX programmer
- Same command using the new function

```
...  
"QRY TRAN NAME(CUS*)"  
results = CSLULGTP('qryinfo.',cartid,'1:30')  
...
```

- ◆ Stem variable is “qryinfo.”
- ◆ Stem variable suffix is “xmltag1.xmltag2”

30

IMS now provides a REXX parser function that makes it much easier. The new function is CSLULGTP and has the same format as CSLULGTS. The first parameter is the stem variable (qryinfo). What is new is that the function assign values to stem variable suffixes which the programmer can use to find the pertinent information in the response. The suffix is always the xmltag1.xmltag2 where xmltag1 is a high level tag and xmltag2 is an imbedded tag within xmltag1. This will easier to understand with an example.

Sample XML Output with REXX Stem Variables

```

<imsout>
<ctl>
<omname>OM10M </omname>
...
</ctl>
<cmd>
...
<verb>QRY </verb>
<kwd>TRAN </kwd>
<input>QRY TRAN NAME(CUS*) SHOW(PGM,QCNT) </input>
</cmd>
<cmdrsphdr>
<hdr slbl="TRAN" llbl="Trancode" ... />
<hdr slbl="MBR" llbl="MbrName" ... />
<hdr slbl="CC" llbl="CC" ... />
<hdr slbl="PSB" llbl="..." />
<hdr slbl="LQ" llbl="..." />
</cmdrsphdr>
<cmdrspdata>
<rsp>TRAN(CUSA ) MBR(IMS1) CC(0) PSB(CUSTADD) LQ(4) </rsp>
<rsp>TRAN(CUSQ ) MBR(IMS1) CC(0) PSB(CUSTQRY) LQ(0) </rsp>
</cmdrspdata>
</imsout>

```

31

Here is that same XML output again. Note that there are xmltags imbedded within xmltags. For example, with the <cmd> ... </cmd> tags are several other tags - <verb>...</verb>, <kwd>...</kwd>, etc. The suffixes assigned by CSLULGTS are, for example:

- qryinfo.cmd.verb
- qryinfo.cmd.kwd
- qryinfo.cmd.input

Likewise

-qryinfo.ctl.omname is the name of the OM that forwarded the response from IMS

qryinfo is just an example of a stem variable name. You can choose anything you like as the name.

Using REXX Parser Function

- Using previous example
 - ◆ qryinfoctl.omname = OM10M
 - ◆ qryinfo.cmd.verb = QRY
 - ◆ qryinfo.cmd.kwd = TRAN
 - ◆ qryinfo.cmd.input = QRY TRAN NAME(CUS*)

- For repeating tags (e.g., the <rsp> tag represents 1 “row” from IMS)
 - ◆ qryinfo.rsp.0 = number of rows = 2
 - ◆ qryinfo.rsp.1 = 1st row = TRAN(CUSA) MBR(IMS1) CC(0) CUSTADD 4
 - ◆ qryinfo.rsp.2 = 2nd row = TRAN(CUSQ) MBR(IMS1) CC(0) CUSTQRY 0

- Within each row
 - ◆ qryinfo.rsp.1.0 = number of columns = 5
 - ◆ qryinfo.rsp.1.1 = 1st column = TRAN(CUSA)
 - ◆ qryinfo.rsp.1.5 = 5th column = 4

32

Things get a little trickier when there are repeating tags. For example, the query in the previous example resulted in IMS returning several lines of response, each of which is imbedded within the <rsp> ... </rsp> tags. In this case:

- qryinfo.rsp.0 would be the number of rows
- qryinfo.rsp.1 would be the entire 1st row

If you want to parse each “column” in a “row”, then:

- qryinfo.rsp.1.0 is the number of columns in that row
- qryinfo.rsp.2.3 would be the 3rd column in row 2

Summary

- OM Audit Trail and Unsolicited Messages
 - ◆ OM auditing capability for commands, responses and unsolicited messages
- SPOC and REXX support
 - ◆ For audit trail and unsolicited messages
- Batch SPOC Utility
 - ◆ Batch utility for submitting commands
- REXX XML Parser
 - ◆ Makes writing REXX programs and execs to interface with OM much easier

33

Enhancements in SPOC make the use of CSL and OM much more attractive to even the single IMS user.

Enhancements in IMS support for REXX makes writing your own AO programs much more feasible.



This section addresses an XML parser that can be used with REXX programs to process command responses from IMS.

Action Bar Enhancements

- Options
 - ◆ New preferences
 - Command confirmation for commands that create, update, or delete resources
 - Confirm changes before submitting
 - Don't confirm changes before submitting
 - Exit confirmation
 - Do you want to exit
 - Do you want to keep or erase command responses
 - Storage management
 - Maximum number of commands in status list
 - Delete command after "n" minutes
 - Command view
 - List view
 - Syntax view
 - Query exceptions
 - Allows user to identify "exceptions (unusual status)" when submitting QRY exceptions command

35

There are new "Preferences" that can be chosen. Command Confirmation, Exit Confirmation, Storage Management, Command View, and Query Exceptions. The impact of each of these will be shown in later slides.

Action Bar Enhancements

- SPOC (renamed from Display)
 - ◆ Audit trail (new)
 - Displays messages sent to OM as command responses or unsolicited messages

- Manage resources (new)
 - ◆ Allows user to go to Manage Resource application directly from SPOC

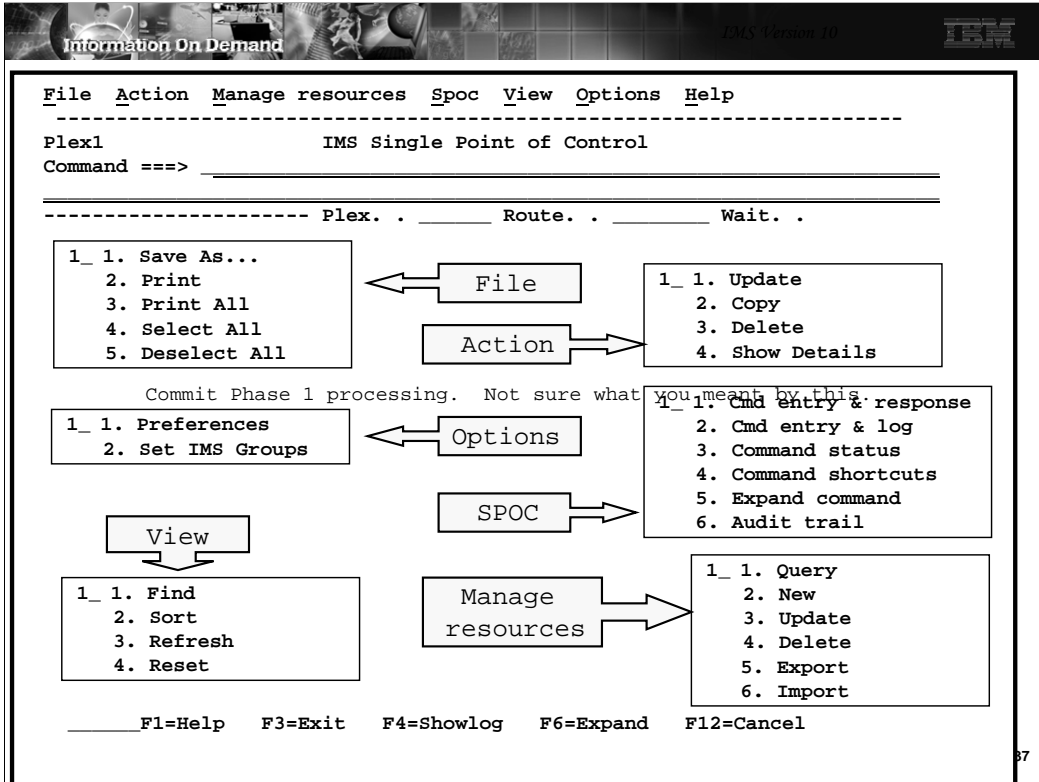
- View
 - ◆ Refresh (new)
 - Resubmits previous command

36

-SPOC is renamed from DISPLAY of earlier releases of SPOC. There is one new option to display the Audit Trail created by OM for command input and responses and unsolicited messages.

-Manage resources is new. It invokes an ISPF application similar to SPOC which can be used to manage resources using DRD. This will also be discussed in detail later.

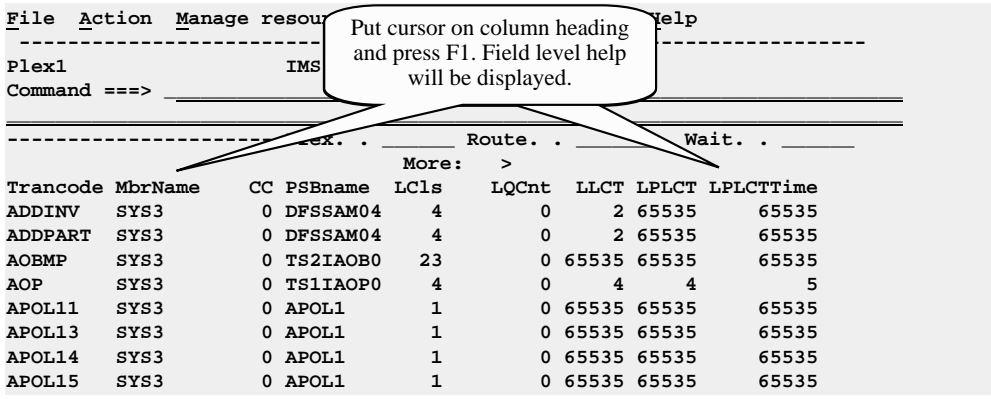
-View has a new option. "Refresh" resubmits the previous command.



This screen shot shows what each of these pull down menus from the Action Bar would look like.

Improved Help

- Field level help added to QRY display screen
 - ◆ Put cursor on column heading and press F1



Trancode	MbrName	CC	PSBname	LCLs	LQCnt	LLCT	LPLCT	LPLCTTime
ADDINV	SYS3	0	DFSSAM04	4	0	2	65535	65535
ADDPART	SYS3	0	DFSSAM04	4	0	2	65535	65535
AOBMP	SYS3	0	TS2IAOB0	23	0	65535	65535	65535
AOP	SYS3	0	TS1IAOP0	4	0	4	4	5
APOL11	SYS3	0	APOL1	1	0	65535	65535	65535
APOL13	SYS3	0	APOL1	1	0	65535	65535	65535
APOL14	SYS3	0	APOL1	1	0	65535	65535	65535
APOL15	SYS3	0	APOL1	1	0	65535	65535	65535

Field level help is now available for columns on the QRY result screen. Place the cursor on the column heading and press F1. A description of the meaning of that field will be displayed.

Support for OM Audit Trail

- Operations Manager provides support in V10 for an “audit trail” of
 - ◆ Commands and responses entered through OM interface
 - ◆ Unsolicited messages from IMSplex members

- SPOC provides support for displaying this audit trail
 - ◆ Preference screen
 - IMSplex member names or types
 - ◆ Display
 - Message written to “audit trail log stream” by OM
 - Read directly from log stream
 - ◆ Details earlier in this section of class

Choosing Option 6 in the SPOC pull down menu invokes the SPOC Audit Trail display function.

Information On Demand IMS Version 10

File Action Manage resources Spoc View Options Help

Plex1 IMSplex Audit Trail

Command ==> _____

----- Members. . IMS1,IMS2,IMS3__ Type. . _____

MbrName	Time	Message
IMS1	2006.227 14:00:01.00	DFSxxxxW message text
IMS1	2006.227 14:00:02.05	DFSxxxxI message text
IMS2	2006.227 14:00:02.16	DFSxxxxI message text
IMS1	2006.227 14:01:02.92	DFSxxxxW message text
IMS3	2006.227 14:03:01.37	DFSxxxxI message text
SPOCUSR1	2006.227 14:04:15.22	Response for QRY IMSPLEX
IMS2	2006.227 14:04:47.00	DFSxxxxW message text

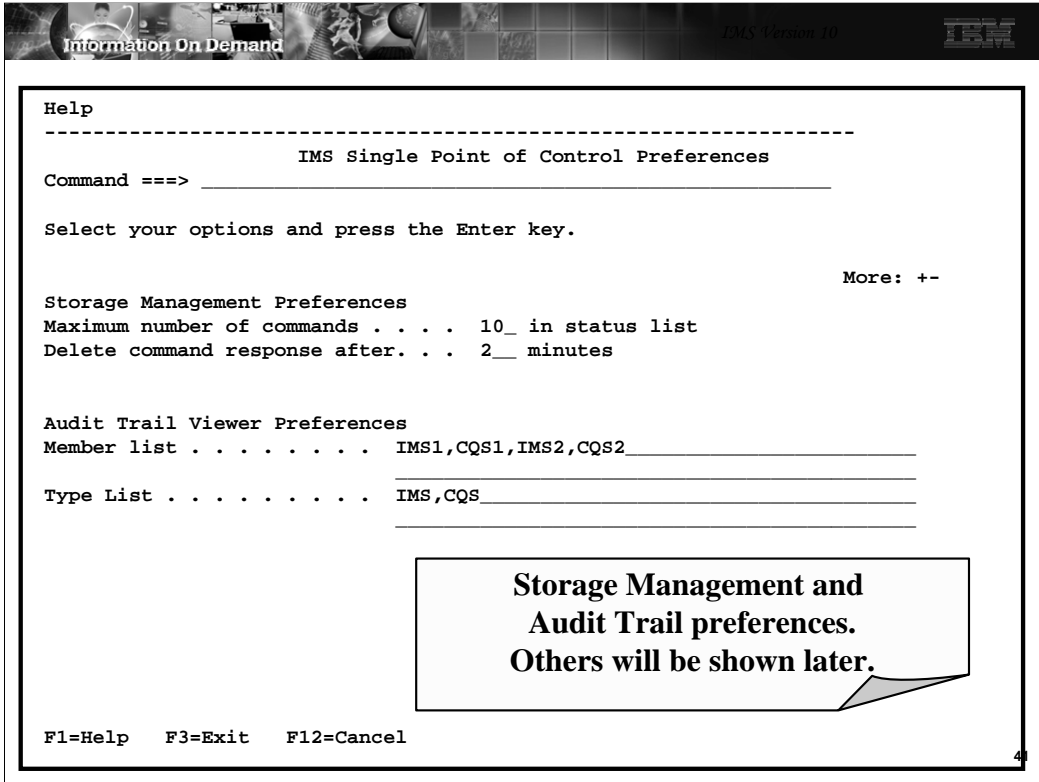
Use F7 and F8 to scroll up and down.

Press Enter to refresh screen with new messages.

To display responses for Type-2 commands, put cursor on line and press Enter

F1=Help F7=Up F8=Down F12=Cancel 40

This is an example of a screen shot of an Audit Trail display. Note that Type-2 command responses may be seen by placing the cursor on the line and pressing Enter.



This screen shows the Storage Management Preferences

- What is the maximum number of commands and their responses to keep in the status list
- How long should a command/response stay in the status list before being deleted

Audit Trail Preferences

- Member list of messages to display
 - Type list of messages to display
- If both have values, the member list takes precedence