



IBM Software Group

26th ALCS User Group Meeting, Lisbon, 2-4 December 2008

ALCS 2.4.1

Mike Hannaford
alcs@uk.ibm.com



Legal Notices

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS);
- OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

RACF*

DB2*

Hiperspace

IBM*

IBM logo*

MVS

Websphere*

VTAM*

z/Architecture

z/OS*

zSeries*

OS/390*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Other company, product and service names may be trademarks or service marks of others.



ALCS 2.4 announcement letter – May 20, 2008

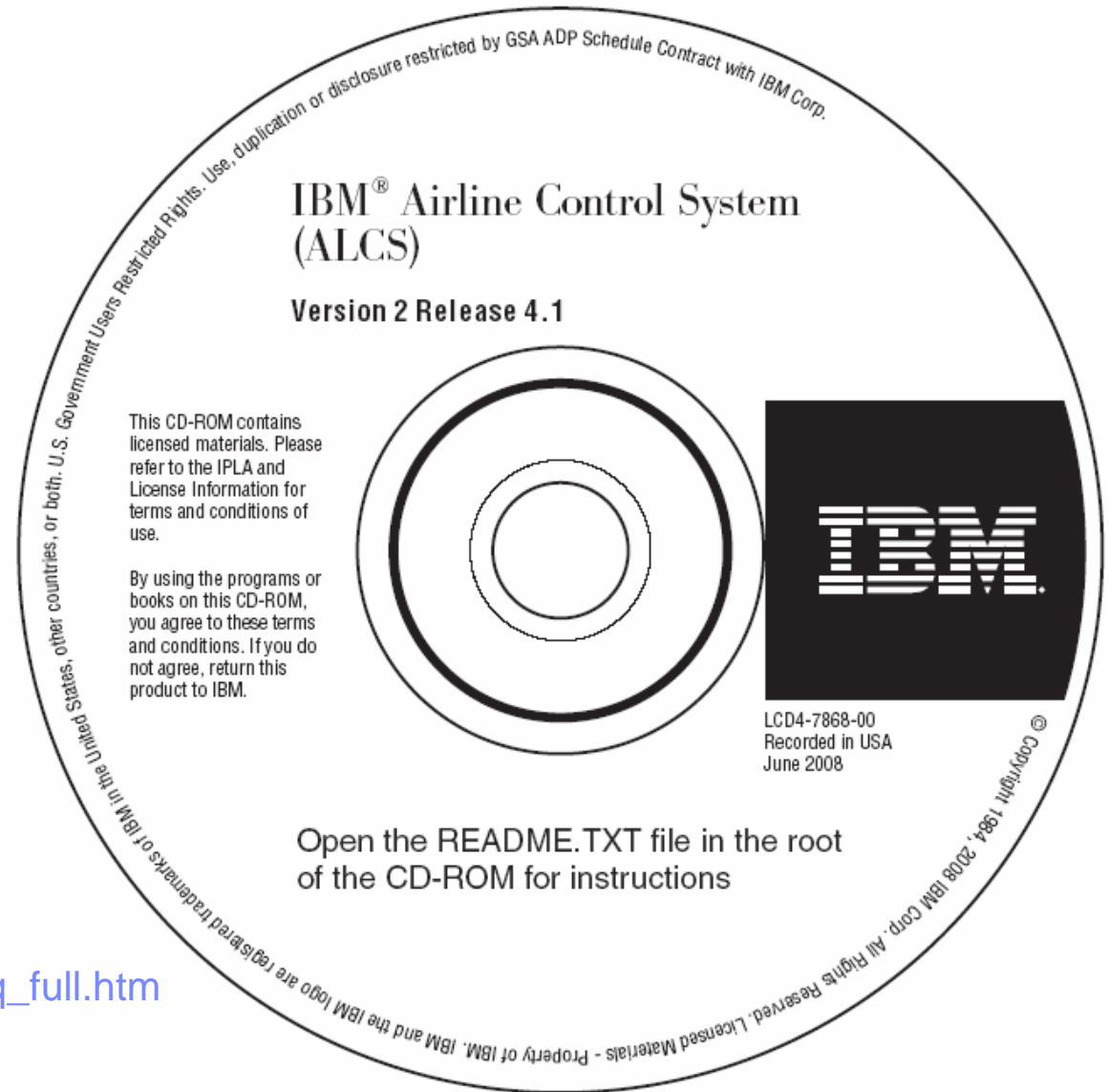
ALCS V2.4.1 is designed with the following new facilities and enhancements:

- Exploits virtual storage above 2 GB
- Moves Virtual File Addressing above 2 GB
- Removes message size constraint of 32 KB
- Optimizes MQ Bridge facility
- Provides the ability to display TCP/IP Trace
- Enhances the display of load module status
- Allows an application to check availability of a given pool
- Uses filing stamps that are now application, not TPF Database Facility program names
- Determines whether storage belongs to an ECB
- Adds program version to duplicate dump table
- Changes the minimum operating system prerequisite from OS/390 to z/OS
- Consolidates the program temporary fixes since the previous release



Product Distribution

- CD
 - If you have one of these then please treasure it as no one in the ALCS development team has ever seen one
- Download
 - Recommended method
 - e-delivery to current license holders
 - Instructions at ...



http://www.ibm.com/tpf/pages/Req_full.htm

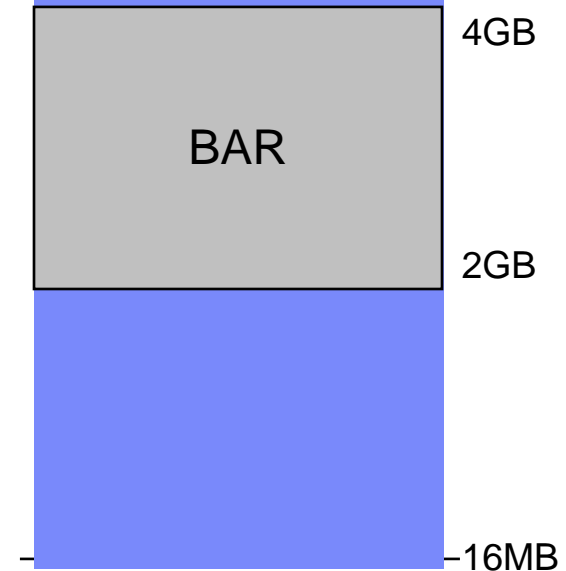
Prerequisite Software

- Current supported z/OS release
 - At time of announcement / availability this was z/OS 1.7 and above
 - z/OS 1.7 end of service was September 30, 2008
 - z/OS 1.8 end of service will be September 2009
 - z/OS 1.9 end of service is planned for September 2010
 - z/OS 1.10 available now



64 bit Virtual

- OS/390 (R10) (real address > 31 bit)
- z/OS 1.2 (virtual address above the BAR)
 - Virtual Bar
 - 000000007FFFFFFF
 - 0000000100000000
- IARV64
 - IARV64 GETSTOR service creates 'memory objects' and that storage is **cleared** to zeros.
 - Each storage area is a multiple of 1MB in size and begins on a megabyte boundary.
 - The area above the bar is intended to be used for data only, not for executing programs.
 - Your installation can set a limit on the use of the address space above the bar for a single address space. The limit is called the MEMLIMIT.



VFA above 2GB Virtual

- VFA buffers can be moved above the 2GB bar
 - Allows more memory for working storage
- Optional – new SCTGEN parameter AMODE64=VFA
 - Uses IARV64 to acquire page fixed memory objects for VFA
- Use z/OS termination resource manager to free the memory objects
 - Avoids memterm
 - After RTM has terminated all I/O and detached the CPULoop tasks
 - New routine runs under initialiser task (DXCABR)
- If AMODE64=VFA specified then
 - MEMLIMIT must be specified on JCL or PARMLIB SMFPRMxx member
 - Default MEMLIMIT is 0 – no storage above bar
 - DXC104T Initialization failed - Available storage....

VFA above the bar - 64 bit considerations

- Installation-wide monitor exits USRGFS, USRLOG, USRPIDC, and USRVFA must be reviewed and modified if necessary as they have extra parameters.
 - Same MVS call mechanism to pass information to the user exits as in ALCS 2.3.1.
 - 64 bit addresses are passed as extra parameters in order to provide smooth migration
 - DXCSAVE mechanism not modified
 - ALCS restores registers after the invocation of the user exits and after a few Monitor routines
- Test database and monfind/monfile locate mode
 - VFA buffer copied to/from a staging buffer below the bar (there are 50 page fixed)
 - Avoids changing current 31-bit code
- 64 bit - analog instructions not allowed in application ECB code
 - Marked "invalid" in conversational trace

Displaying storage above 2GB

- ZHCOR is provided to display addresses above 2GB
 - Not formally documented
- Can be used to display addresses below 2GB
- Storage areas to which application programs do not have read access will not be displayed
- ZHCOR xxxxxxxxxxxxxxxxxxxx nn
 - xxxxxxxxxxxxxxxxxxxx – up to 16 hex digits of address – leading zeros can be omitted
 - nn - number of fullwords to display



Ensure enough storage above 2GB is available

- Determine how much not-reconfigurable storage available above 2 GB.
- Consult MVS system programmer to obtain this with the help of
 - the D M=STOR command and
 - the RMF VSTOR report to see if ALCS or other systems currently running use real storage above 2 GB. This may happen when there is not enough real storage available below 2 GB.
- Exclude 20% to allow for z/OS DAT structures
- If DB2, or other ALCS systems (automatic takeover) are planned to run in the same LPAR then adjust the amount to account for these systems.
- To compute the VFA storage required for ALCS use the ALCS system generation and add a margin of 10%
- Verify there is enough real storage for pageable requests.
 - If z/OS cannot pagefix real storage above 2 GB then it will attempt to steal a frame above 2 GB before taking a frame below 2 GB. This avoids pageable storage shortages caused by fixing too much real storage below 2 GB but impacts system performance.



Example of how much storage above 2GB required

- Assume there is 1 GB not-reconfigurable storage available above 2 GB.
- Exclude 20% to calculate the amount of real storage available for VFA buffers. So there is 819 MB available.
- Assume that neither DB2, or other ALCS systems, run in the same LPAR. Otherwise will need to adjust the figure obtained above.
- Compute the VFA storage required for ALCS. Use the ALCS sysgen:
 - If VFABUF=(50000,150000,1000,100000) and CISIZE=(512,1536,4096,4608). The storage required is:
 - $(50000 \times 512) + (150000 \times 1536) + (1000 \times 4096) + (100000 \times 4608) =$ approx 688 MB.
 - Add a reasonable margin, let us say 10%, so 757 MB is needed.
 - There is no problem as 819 MB real storage is available above 2 GB.
- Verify with MVS system programmer there is enough real storage for pageable requests.



Large message support

- Solicited messages up to 2MB in size are now supported for –
 - ALC terminals connected through TCP/IP
 - TCP/IP connections
- New COMDEF/COMDFLT parameter IPMGSZ= for IP connections
 - Value is from 32K up to 63K for MATIPA
 - Value is from 32K up to 2048K for non-MATIP
 - Default is 0 (no large TCP/IP large messages)
 - Not allowed for HTTP, ACSA or MATIPB
- Input messages will be put in storage blocks if they fit
 - Otherwise on input D0 will contain address of message in heap storage



Large Messages - Heap storage use

- Heap storage is used for messages that do not fit in a storage block
- Allocate type 3 storage units using SCTGEN NBRSU and SUSIZE.
 - SUSIZE must be at least IPMGSZ + 21 bytes for the message header
 - Number required will depend on
 - Rate of messages
 - Existence time
- When IPMGSZ is used then specify a MEMLIMIT for the ALCS job
- Review some installation-wide monitor exits and modify if necessary
 - USRCOM2, USRCOM4, USRDCR1, USRTCP2 and USRTCP3



Large Messages - Extended message block

- Extended message block similar to existing CM1CM, except
- Size of message field
 - CM1CCX fullword at location 14 (extended format)
 In place of
 - CM1CCT halfword at location 16 (standard format)
 - Location 14 previously used for BSC, SLC and raw VTAM 3270
 - New section XSOPZRO in DXCOPZ used for extended message instead of HSOPZRO
- There is no forward or backward chaining

Large Messages - api

- SENDC has a new parameter - X for large messages
 - Data level contains the start address of the heap storage where the message is stored. Extended message format is used
 - The heap storage will not be released by SENDC X
- ROUTC has new option bit RCPL0EXT
 - On means extended message format and the address of the heap storage is stored in the data level
 - Off means standard format, does not use heap
 - The heap storage will not be released by ROUTC
 - For C/C++ or assembler application programs



Large Messages – other changes

- Data Collection logging item now variable length
 - If DCMLNG/DC0LNG is X'FFFF' indicates extended format
 - Logging item size is four bytes larger
 - Real message length in new four byte field DCMLNGE appended to the original item
 - Changes to DXCDCR and DXCSRGO
- New extended input buffer for each resource
 - Pageable above the bar storage
- Output sent between ALCS and TCPIP in 32K chunks
 - EWOULDBLOCK retry faster



Optimize MQ Bridge - move from ECB to monitor

- The MQ bridge has moved from ECB controlled code to monitor code
- Efficiency is improved as fewer ECBs are created
- MQ Bridge exits moved from ECB to Monitor exits
- Some dumps are obsolete - one new dump added
- Some text in displays and dumps has been changed
 - from MQSeries to MQ

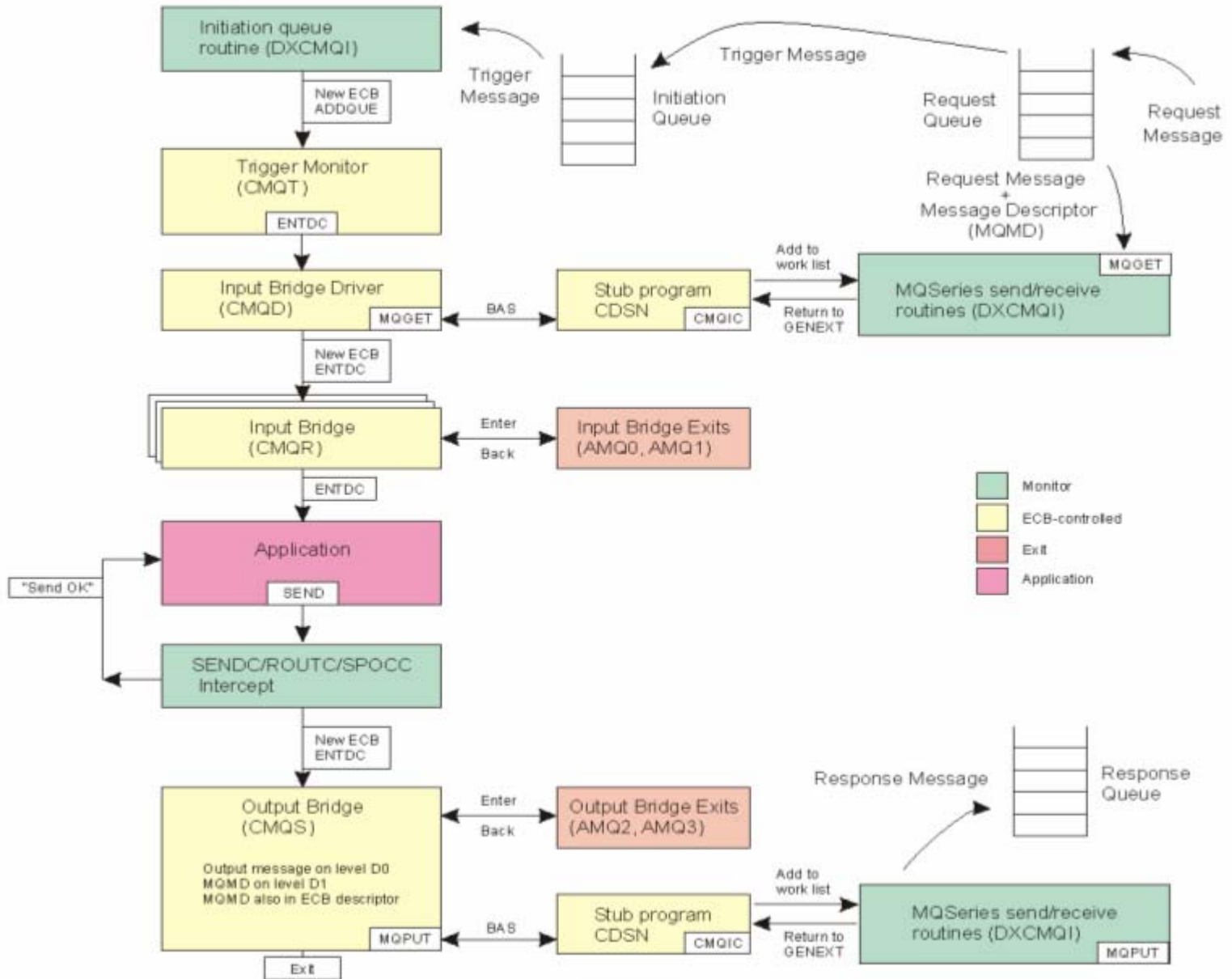


Optimize MQ Bridge - move from ECB to monitor

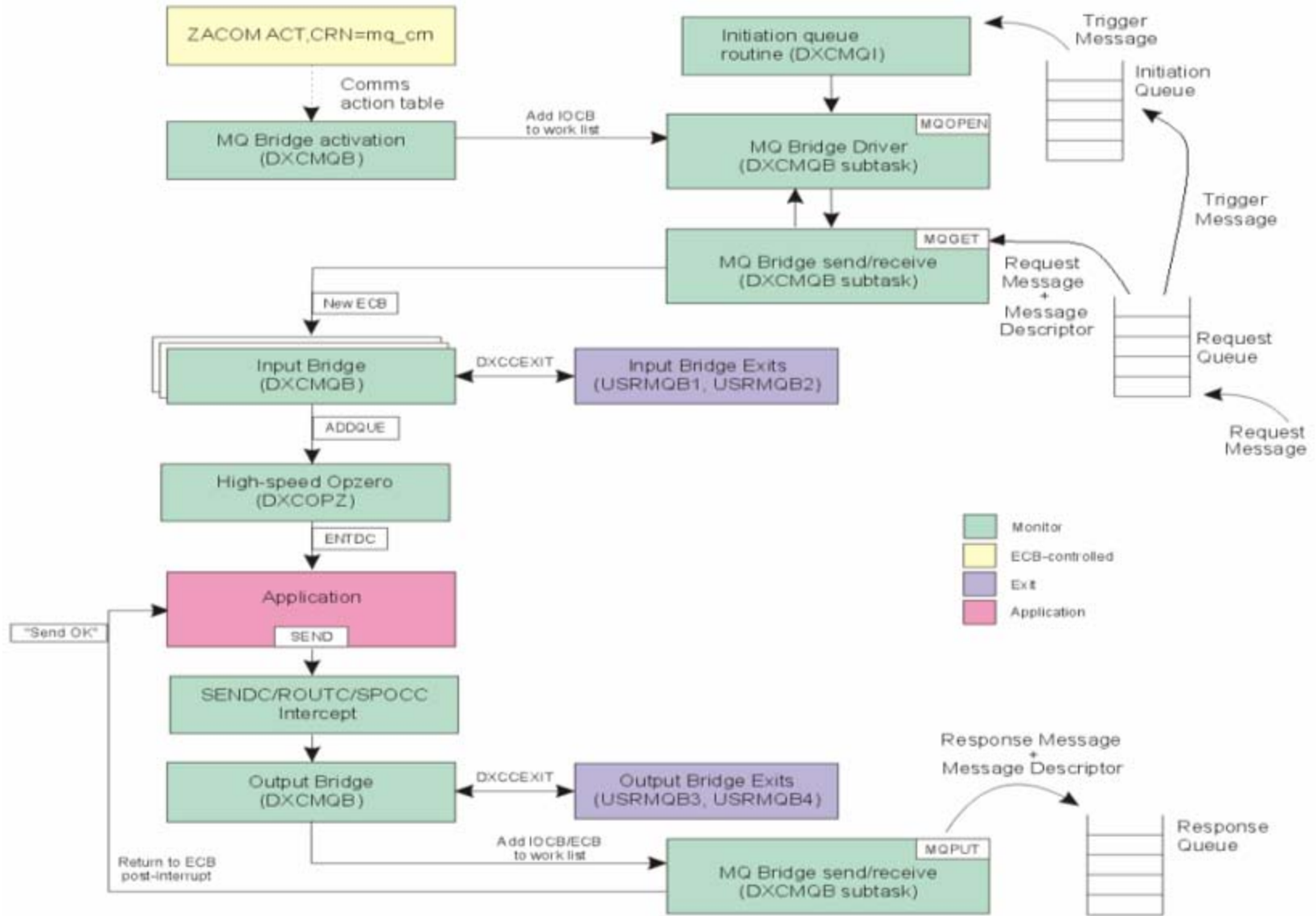
- Improved performance on input
 - No longer create an extra ECB for each trigger message.
- Improved performance on output
 - no longer create an extra ECB for each output message.
- Resolves issue where normal opzero processing is missing from input messages from MQ Bridge terminals.
- Resolves User Group requirement
 - ATC0266 - Allow PF Keys from MQ Bridge terminals.



ALCS 2.3.1



ALCS 2.4.1



Optimize MQ Bridge - move from ECB to monitor

- MQ Bridge trigger monitor routine scans the MQ communications resources (this runs under control of the MQ attached subtask).
- When it finds a match for the application queue specified in the trigger message, it gets an IOCB to invoke the MQ Bridge driver routine.
- MQ Bridge driver routine opens the request queue for the MQ communication resource and issues MQGET with signal to retrieve the next message from the request queue.
- The object handle for the open queue will be stored in the MQ communication table entry.



Optimize MQ Bridge - move from ECB to monitor

- Input messages pass through full “opzero” processing
 - Process PF keys
 - Process context-sensitive help
 - Process ZRETR command
 - Process ZSCRL LOG command
 - Process ZLOGN/ZLOGF commands
 - Do special processing for ALC input (backspace, erase, FID)
 - Support mixed case applications
 - Set end-user security environment
 - Trace initial entry from a terminal
 - Call USRCOM2 installation-wide exit for input message
 - Call USRCOM8 installation-wide exit for printer ACK

Optimize MQ Bridge - move from ECB to monitor

- New monitor CSECT called DXCMQB containing routines:
 - MQ Bridge trigger monitor (CMQT in ALCS 2.3.1)
 - MQ Bridge activation and deactivation routines (was CMQQ and CMQC)
 - MQ Bridge driver routines (was CMQD)
 - MQ Bridge receive routines (was CMQR)
 - MQ Bridge send routines (was CMQS)
- New monitor save area for DXCMQB
- New "MQ Bridge" IOCB type



Optimize MQ Bridge - move from ECB to monitor

- MQ Bridge Installation Wide Exits
 - USRMQB0 - MQ input bridge address exit (was AMQ0)
 - USRMQB1 - MQ input bridge format exit (was AMQ1)
 - USRMQB2 - MQ output bridge format exit (was AMQ2)
 - USRMQB3 - MQ output bridge message descriptor exit (was AMQ3)
- Shells DXCUMQB0/1/2/3 in DXCIWE1 library

- No impact on existing applications
- No impact on Communication Definitions
- MQ definitions remain the same
- Trigger monitor remains CMQT



Detect and throttle input messages

- Detect abnormally high message rates from a terminal
 - Reduce the number of input messages passed from a terminal to the applications
 - Robot detection mechanism approach similar to current CI solution
- 3 new fields provided in terminal section of COORE
 - RETDMTC – 4 byte count of messages received on this CRN – incremented each time a message is received
 - RETDMTD – 4 byte count of messages rejected on this CRN
 - RETDMTT – 8 byte TOD clock value when last message was accepted
 - The counts are 4 byte logical fields – they will wrap around to zero
- These 3 fields are passed to USRCOM2 as param_2 is the COORE address

Detect and throttle input messages – USRCOM2

- USRCOM2 can examine the fields -
 - accept the input message by setting RC=0
 - discard the input message by setting RC=4
 - Issue messages/warnings to the operator
- On return from USRCOM2 DXCOPZ will check return code
 - 0 – message accepted.
 - The RETDMTT field will be updated with the current TOD clock value
 - 4 – message discarded.
 - The RETDMTD field will be incremented
- Sample code is provided in USRCOM2
- The value in RETDMTT gives the time of the last accepted message. This can be used to calculate a message rate per second



Detect and throttle input messages – Sample code

- Sample USRCOM2 in DXCIWE tagged with eye-catcher *SAMPLE*
- Uses Communication data table user area to define
 - Indicator that terminal should be subject to throttling
 - Number of delay units between input messages (1 delay unit = 50 msec)
- The code then
 - Checks if input is from a terminal ?
 - If so, is the terminal subject to throttling ?
 - If so, check time from last message is greater than delay ?
 - If so, accept message
 - If not, request message is discarded

Enhance display of load module status

- Comes from a 2007 User Group requirement
 - Really 4 separate requirements
- In ALCS 2.4.1 base
 - Display the test load modules loaded by a certain terminal
 - Display all of the terminals which own a certain test module
- Still outstanding - requires a rewrite of DXCPGC
 - Display the program version information in a load module
 - Display all of the load modules which include a certain program, and the program version in each load module



Enhance display of load module status

- Requirement – Part 1
 - Display the test load modules loaded by a certain terminal. For example,
ZPCTL DISPLAY CRN=PYEYTC02

```
DXC8345I  CMD M   18.52.49 PCTL
Module    Type    CRN/Status  Load Date/Time  Unload Date/Time
MQMH      ECB    PYEYTC02   2008.329 13.25.34
JELLO     ECB    PYEYTC02   2008.329 13.25.40
```

Enhance display of load module status

- Requirement – Part 2
 - Display all of the terminals which own a certain test module. For example,
ZPCTL DISPLAY JELLO

```
DXC8345I CMD M 18.48.51 PCTL
Module      Type      CRN/Status  Load Date/Time      Unload Date/Time
JELLO      ECB      PYEYTC02   2008.329 13.25.40
JELLO      ECB      PYEYTC09   2008.329 13.27.27
JELLO      ECB      PYEYTC08   2008.329 13.27.37
```

Suppress duplicate nodump messages

- New mechanism to restrict the number of nodump messages.
 - When a series of duplicate dumps occurs in quick succession, a lot of messages are produced to the Diagnostic File and to RO CRAS.
 - This can produce problems due to the high volume of messages generated.
- New SCTGEN parameter - DDTIME (Duplicate Dump Time)
 - System wide value, the same for all dumps
 - Value 0. (Default) means no suppression of messages
 - Value 1 – 254. Time interval in seconds during which duplicate nodump messages are suppressed
 - Value 255. Suppress all nodump messages



Suppress duplicate nodump messages

- **Processing**
 - Once a nodump message is issued, over the next DDTIME seconds any nodump messages (for the same dump) are suppressed
 - A count is kept of suppressed messages
 - When the DDTIME interval has expired, the count of suppressed messages is reset and a nodump message can be issued.
 - So the DDTIME value provides a filter such that only 1 NODUMP message per DDTIME seconds will be issued.
- **Display**
 - ZDSER DDT shows the number of suppressed NODUMP messages per dump



Allow application to check availability of a given pool type

Callable ECB-type programs have been provided

- CVEP – return pool type information for a pool interval
 - Input
 - R14 - Pool type number (L1STPOOL, L1LTPOOL,....)
 - Output
 - R14 - Number of available records in this pool type
 - R15 - Pool dispense rate per second for this pool type
 - R14=0 then R15 contains completion code



Allow application to check availability of a given pool type

- CVEQ – return pool type information for a record id
 - Input
 - R14 - Record id in low order 2 bytes
Record id qualifier (0-9) in high order byte
 - Output
 - R14 - Number of available records in this pool type
 - R15 - Pool dispense rate per second for this pool type
 - R14=0 then R15 contains completion code



Use application program name instead of TPFDF

- Choose to have the application program that requested TPFDF to file the record named in the control stamps (RS0RS field RS0PFSN).
- New SCTGEN subparameter
 - TPFDF={YES,.....,TSTAMP} – use TPFDF program name
 - Default – works as now
 - TPFDF={YES,.....,ASTAMP} – use requesting application program name
- This information is located at the end of the physical record and can be displayed with the ZDFIL I command
- If ASTAMP is used, only new file updates will use the application program name – no retroactive conversion



Determine whether storage belongs to an ECB

- New installation-wide user exits callable service, USTRECB, is used to validate if a storage address and length is owned by an ECB
- Storage is valid if it is within any storage unit allocated to the ECB
 - Primary Storage Unit or Overflow Storage Unit
 - Unhooked blocks are considered not valid
 - Type 2 or Type 3 Storage Units
- Input parameters to DXCSERV USTRECB
 - Storage address
 - Address of fullword containing the length of storage area
 - ECB address
- USTRECB returns with the following
 - R15 = 0 The storage belongs to the ECB
 - R15 = 4 The storage does not belong to the ECB
 - R15 = 8 The ECB address is invalid

Program version added to duplicate dump table

- When the same dump occurs at the same offset in more than one version of a program, a separate entry is placed in the duplicate dump table for each version that takes a dump.

```

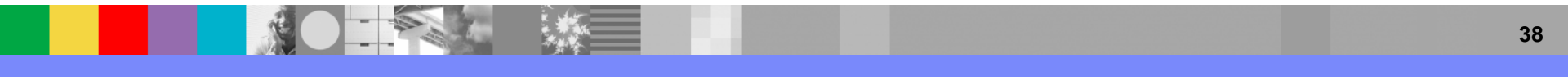
DXC8275I  CMD M  11.12.42 DSER  Duplicate Dump Table
Nbr Prog Ver Error  Off  Sup          Nbr Prog Ver Error  Off  Sup
001 PPPP 00  BBBB 002C 0000          002 PPPP 00  CCCCC 0036 0000
003 PPPP 00  DDDDD 0040 0000          004 PPPP 00  AAAAA 0062 0000
005 PPPP 00  000017 0088 0000          006 PPPP 99  BBBB 002C 0000
007 PPPP 99  CCCCC 0036 0000          008 PPPP 99  DDDDD 0040 0000
009 PPPP 99  AAAAA 0062 0000
  
```

- When new version of a program loaded, makes it easier to track dumps from the new program version



Display TCP/IP trace online

- In ALCS 2.3.1 the TCP/IP trace was written to the diagnostic file
- Now it written to the system TCP/IP trace block in core
 - Current contents of block can be viewed online with ZCTCP
 - Can be written to the diagnostic file
 - By using the ALCS web server can be viewed in a web browser
- Trace data stored in the ALCS TCP/IP sockets data area DXCSOCKA
 - 64 entries in wrap around table, each entry contains
 - IN / OUT indicator
 - TOD clock
 - CRN
 - data block size and first 32 bytes of data block
- ZACOM command to activate and deactivate the TCP/IP trace facility for each TCP/IP communication resource



Display TCP/IP trace online

- ZCTCP TRACE CLEAR clears the system trace block
- ZCTCP TRACE START starts tracing to Diagnostic file
- ZCTCP TRACE STOP stops tracing to Diagnostic file
- ZCTCP TRACE SHOW displays the trace block contents

- Block can be displayed by ALCS Web Server
 - Use the HFS interface to create a directory entry -
 - `mkprg tcpiptrace.cgi ctw`
 - Use the corresponding URL on your web browser
 - <http://alcswebserver.com/tcpiptrace.cgi>
 - Most recent trace data is shown
 - See *ALCS World Wide Web Server User Guide* for more information about the ALCS Web server and the ALCS HFS.



ZALCS command – Diagnostic Aid

- Collection of system programmer's aids
 - We do not intend to document in the operations guide
 - We do expect to add to this command over time
 - Use ZALCS ? For help with format of command

Use ZALCS to display ALCS diagnostic information or convert a TOD clock value.

```
Format:  ZALCS mmmmmmmm
         Or:  ZALCS SAVEAREA=sss
         Or:  ZALCS SAVEAREA=sss,CPULoop=ll
         Or:  ZALCS DISPLAY=MOMB
         Or:  ZALCS DISPLAY=VFASTAGE
         Or:  ZALCS DISPLAY=VFA
         Or:  ZALCS TOD=tttttttt
```

```
Where:  mmmmmmmm   Monitor storage area (default is DXCNUC)
        sss       Monitor save area name
        ll        CPU loop number -- 1-2 decimal digits
                Default is 0 (initializer task)
        tttttttt  TOD clock -- 1-16 hexadecimal digits
```

ZALCS command – Diagnostic Aid

- Implements what you may have as ZDMON
 - ZALCS computes address to pass to ZDCOR
- Display system areas pointed to by CPGEN
 - ALCS monitor CSECTs
 - ZALCS ADXCSEQT Display monitor CSECT using CPGEN label
 - ZALCS DXCSEQT or synonym CSECT name
 - Useful to verify APAR level in eyecatcher
- ALCS control blocks
 - ZALCS ADXCCDT Display tape status table
- See ECB controlled program CAT0 for list of labels and synonyms

ZALCS SAVE and ZALCS TOD command

- Display monitor save areas
 - ZALCS SAVE=NUC,CPU=1
 - Default CPU=0 is initialiser task
 - ZALCS computes address to pass to ZDCOR
- Convert TOD clock value
 - ZALCS TOD=xxxxxxxxxxxxxxxx

```
zalcs tod=C362107000000000
DXC5230I CME M 14.08.49 ALCS
      TOD = C3621070 00000000
Julian Day = 2008.337
      Time = 14.00.00
```

ZALCS DISPLAY command

- ZALCS DISPLAY=VFA shows VFA control area and buffer counts

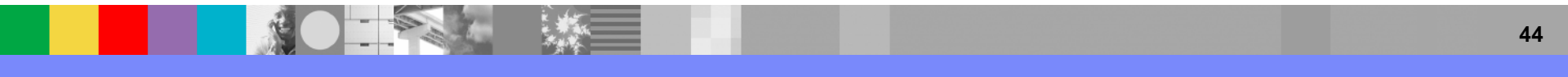
```

DXC5240I CME M 14.27.01 ALCS
VFA control table at 7F521638
7F418008 00001645 7F42E500 FFFFFFFF5 7F4448A8 00001645 00000000 7F521680
7F521740 7F521800 7F5218C0 7F521980 00000000 00000000 00000000 00000000
VFA buffer counts
  Total      Free Age-List File-Pending
    800      788     12         0
    800      725     75         1
    100         2     97         1
   4000     3990     10         0
  
```

- ZALCS DISPLAY=VFASTAGE
 - Display VFA staging buffers
- ZALCS DISPLAY=MOMB
 - Display memory object (MOMB) control table and entry table

System dumps – Additional systems areas

- Additional control/data areas included in system error dumps:
 - APPC area (mapped by DXCAPPCA)
 - CPU loop TCB table (mapped by TC0TB)
 - Monitor exits table (mapped by IE0TB)
 - MQ area (mapped by DXCMQIA)
 - OCTM area (mapped by DXCTMCT)
 - PDU area (mapped by DXCPDUA)
 - SQL area (mapped by DXCSQLA)
 - TCP/IP area (mapped by DXCSOCKA)
- TABLES print option in DXCDTP will show these areas



ALCS Throttle

- THTRC macro controls resources used by batch-like applications
 - Designed by Jorge Widmar distributed to some ALCS 2.3.1 customers
 - Uses the performance monitor values to determine system load
- Throttle Control Table
- Macro THTRC results in ENTRC to Throttle control program
- 4 ECB controlled Throttle programs
 - CTH1 - Command Interface
 - CTH2 - THTRC Macro Service
 - CTH3 - Initialisation
 - CTH4 - Police
- Command ZCTHR



Throttle Table

- In core table read from file at restart
- Updated via Z command
 - ZCTHR ADD NAME=RECP, NLOOP=50
 - ZCTHR ALT NAME=RECP, NLOOP=5
 - ZCTHR THRS NAME=RECP,IOQ=5
 - ZCTHR DISP

		Normal Set			Threshold			Restrictive Set		
	State	nloop	wait1	wait2	cpu	i/o	usr	nloop	wait1	wait2
RECP	Norm	50	0	0	0	5	0	5	0	0

Throttle Table

- THTRC LOAD
 - Returns value 50

		Normal Set			Threshold			Restrictive Set		
	State	nloop	wait1	wait2	cpu	i/o	usr	nloop	wait1	wait2
RECP	Norm	50	0	0		5		5	0	0

- SYSTEM LOAD INCREASES
 - I/O queue time exceeds 5 ms
 - Throttle police routine updates table

		Normal Set			Threshold			Restrictive Set		
	State	nloop	wait1	wait2	cpu	i/o	usr	nloop	wait1	wait2
RECP	Restrict	50	0	0		5		5	0	0

- THTRC LOAD
 - Returns value 5

Throttle Table

		Normal Set			Threshold			Restrictive Set		
	State	nloop	wait1	wait2	cpu	i/o	usr	nloop	wait1	wait2
APPL	Norm	10	1	2	10	0	0	4	3	9

- THTRC WAIT
 - Application waits 1 second
 - After 10 THTRC WAIT macros application waits 2 seconds
- SYSTEM LOAD INCREASES
 - ALCS CPULOAD exceeds 10% of partition
 - Throttle police routine updates table

		Normal Set			Threshold			Restrictive Set		
	State	nloop	wait1	wait2	cpu	i/o	usr	nloop	wait1	wait2
APPL	Restrict	10	1	2	10	0	0	4	3	9

- THTRC WAIT
 - Application waits 3 seconds
 - After 4 THTRC WAIT macros application waits 9 seconds

Throttle facility – USR threshold

- CPU and IOQ thresholds are continuously compared with values from the performance monitor
- The USR threshold is compared with a value obtained from the ECB controlled exit program ATH1
 - ALCS conditionally enters ATH1 when updating the throttle table.
 - ATH1 must return with a value in R15 (range is 0 through 99).
 - If this value exceeds the non-zero USR threshold of a throttle application, then the restricted (and not the normal) throttle values defined for this throttle application will be used by the throttle.



Why use the throttle?

- Performance Monitor identified 'batch-type' problem programs
 - LODIC used in past to throttle programs
 - Not satisfactory
 - LODIC ECBCREATE uses single check level for all appls
 - LODIC CONDITIONAL DEFER is fixed
 - ALCS throttle is dynamic and can be defined for particular applications



Use Type 3 storage units for C/C++ heap storage

C/C++ heap moved from SU type 2 to SU type 3

- API routines added for malloc(), calloc(), realloc(), free() C functions
 - Routines are now wrappers that call the assembler routines internally
 - Allows assembler and C/C++ programs to share the same heap storage
 - This makes mixing assembler and C/C++ programs easier
- Type 2 storage units used for C/C++ stack
- Review SCTGEN SUSIZE= values
 - For Type 2 specify a value of 7 (508K) or greater when applications use LE support.
 - For Type 3 specify a value so the size exceeds the largest possible heap storage request.
- Set SCTGEN ENTSTOR1, ENTSTOR2, ENTSTOR3, rather than ENTSTOR
- Review assembler SLIMC STORAGE=... use STORAGE1, STORAGE2, STORAGE3 instead
- Review C/C++ function slimc(SLIMC_STORAGE,...) use SLIMC_STORAGE1, SLIMC_STORAGE2, SLIMC_STORAGE3 instead

Performance

- For typical entries the performance in terms of path length and CPU utilization is very similar to release 2.3.1.



Migration

- MQ Bridge
 - Code new IWE exits to replace defunct ECB exits
- VFA above 2GB
 - Ensure MEMLIMIT specified
 - Review exits USRGFS, USRLOG, USRPIDC, and USRVFA and modify if necessary
- Large message support
 - Ensure MEMLIMIT specified
 - Review exits USRCOM2, USRCOM4, USRDCR1, USRTCP2 and USRTCP3 and modify if necessary



Migration

- Understand storage unit requirements type 2 / type 3
 - Review SCTGEN and SLIMC values
- For TPFDF DLI users (TPFDF PUT 15, 16 or 17)
 - Usermod D500063
 - ALCS 2.3.1
 - AQ38882 added DLI TPFDF CEP support
 - AQ85391 removed the DLI code
- If UK28709 not applied on ALCS 2.3.1
 - Use IDCAMS job to change the share options for your ALCS data sets from the default (1,3) to (3,3)
 - Update DFSMSdss backup/dump jobs to specify the TOLERATE(ENQF) option
 - UK28709 NOTES file for details



User Group requirements included in 2.4.1

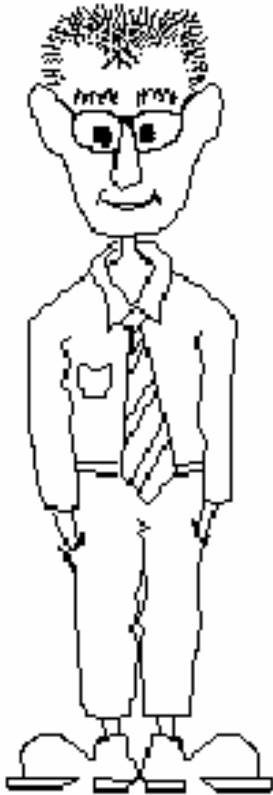
Develop. code tag	Requirement number	Description
D500015	ATC0278	Detect and slow down robot CRNs
D500005	ATC0288	Enhance display of load module status
D500016	ATC0292	Suppress Duplicate Dump Messages
D500002	ATC0286	Enable ALCS data transfer up to 2 MB
D500006	ATC0127	Allow an application to check availability of a given pool
D500007	ATC0291	Filing stamps should be Application not TPFDF program names
D500008	ATC0238	Determine whether storage belongs to an ECB
D500009	ATC0284	Add program version to duplicate dump table
D500001	ATC0285	Place VFA above 2GB Virtual Bar

Other updates included in 2.4.1

Development Code Tag	Description
D500003	Move MQ bridge from ECB code to Monitor code. Includes ATC0266 – allow PF keys from MQ bridge terminals
D500004	Enhance TCP/IP trace display. On console or web browser
D500012	New ZALCS command to display diagnostic information
D500013	Add ALCS control areas to dumps
D500014	Performance Throttle facility – based on Jorge Widmar's design
D500030	C/C++ heap now in Type 3 SU



Any Questions ?



Mike Hannaford

alcs@uk.ibm.com

mike_hannaford@uk.ibm.com