IBM Software Group

TPF Users Group Spring 2007

z/TPF Features

Name : Michael Shershin
Venue : Main Tent

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Agenda

- SOA Enabled
- Additional new features in development
- Constraint relief
- More Memory
- Greater security
- Reduce costs
- Increased productivity
- Improved tuning
- Improved availability
- Migration to z/TPF
- Details

SOA Enabled

z/TPF enables SOA/Web services exploitation

- More memory per ECB helps to deal with memory usage characteristics of ported SOA/Web services components
  - Apache
  - B2B XML Scanner
  - XML4C
- Software license costs addressed
  - As MIPs increase due to use of SOA components: Apache, SOAP, XML
  - Software license MSUs increase at a slower rate when using Workload License Charging (WLC)

SOA Enabled - New functions

New SOA/Web services infrastructure and functionality targeting PUT 4

- HTTP Client
- SOAP enhancements to help support the suite of SOAP feature specifications (WS-*)
- Dynamic Deployment Mechanism instead of the current user exit based mechanism
- Web Services Interoperability profile support
- TPF Toolkit enhancements for creating/managing SOA/Web services artifacts

Additional Features in Development

- GCJ - Java
- MySQL compatibility
- Service Data Objects - SDO - for databases in TPFDF
- Threads

Constraint Relief

Do not let TPF infrastructure be a reason to prevent business growth

- More than 2GB of memory
- Up to 255 SSUs
- SDA addresses up to FFFF for all devices
- Up to 40,000 DASD modules
- >16 I-Stream support

More Memory per ECB

- Physical 12 K ECB allocated below 2 GB bar
- Most ECB memory not limited to below 2 GB bar allowing significant expansion
  - Pre-allocated ECB private area
  - Pre-allocated 31-bit ECB heap
  - Pre-allocated ECB application stack
  - ECB trace
    - Define number of trace entries
    - Macro trace
    - C function trace
    - Heap trace
  - 64-bit ECB heap
  - Memory backing non pre-allocated 31-bit ECB heap and application stack is allocated above 2 GB

More Memory per ECB (continued)

- Larger 31-bit ECB heap can be used
- ECB private area size
  - Minimum = 4 meg
  - Maximum = 16 meg
- ECB private area mapping improves core corruption detection

More System Memory

- IP Message Table
- Socket block table
- Socket trace
- Dump buffer
- 64-bit Core Resident Program Area
- 64-bit System heap
  - Preallocation System heap
- Larger 31-bit System heap can be used
- Larger VFA can be used
- I/O (or LDEV) trace
  - Define number of entries

Greater Security

- Crypto Express2 accelerator (CEX2A) support
- AES cipher suites for SSL
- User APIs to encrypt data using AES
- Hardware acceleration for AES
- New APIs for shared SSL sessions
- Secure FTP client
- SHA-1 APIs for data integrity
- File system security
- Secure key management
- Tape hardware encryption support
- (in development) Protecting data in use

Reduce Costs

- Workload License Charging support
  - z/TPF software license fees based on amount of processing used
  - Check processor utilization
    - Use LODIC to manage processor utilization for long running batch type utilities
- Additional way to reduce costs
  - Increased productivity
  - Tuning

Increased Productivity

Reduce time to market and reduce costs by increasing productivity

- Improved development environment
- Assembler language enhancements
- C / C++ language enhancements
- Improved diagnostics
- Large programs

Development Environment

- Use of Open tools
- Use of Linux
  - make
  - find, grep
- GNU Compiler Collection (GCC)
  - cross compiler for z/TPF
  - binutils: as, ld
  - Extended Link Format (ELF)
- TPF build enhancements
  - maketpf
  - bldtpf
  - loadtpf
- TPF Toolkit
  - IDE seamless integration with TPF build tools
  - Single source migration rules

Assembler Language Enhancements

- Program packaging
- Ability to execute in 31-bit mode and 64-bit mode
- Greater than 4K support
- Baseless support
- Multiple base registers
- R8 saved / restored across all SVCs - can use like R1 - R7
- R10, R11, R12, R13 can be used as scratch registers
- Extended Register Save support
  - Saves / Restores R10, R11, R12, R13 across general macros
- Subroutine linkage support - CLINKC / SLINKC / RLINKC / ELINKC
- Ability to call C functions - CALLC
- Application stack available to assembler programs
- Increased instruction set in z/Architecture
- ENTRC SAVEREGS=YES - saves R0 - R8 and restores on return

C / C++ Language Support

- C environment exists for all ECBs
- Support for GCC built as a cross-compiler for z/TPF
- Support for glibc and libstdc++ libraries
- Support for the GCC Standard Template Library (STL)
- Combined macro and function trace
- The entrc function now will work in a C++ segment.
- Added 3 new data types that represent data (not function) pointers that point to 32-bit addresses
  - ► __ptr32_t 32-bit void pointer
  - ► __chptr32_t 32-bit char pointer
  - ► __uiptr32_t 32-bit unsigned int pointers
- Added macro definition PTR32ATT to assist in the declaration of the data types. Use the PTR32ATT macro to declare explicit 32-bit pointers for any other pointer type.
- Provided definitions for 32-bit fields time_t32, size_t32, and ssize_t32. Note: time_t, size_t, or ssize_t will be 64-bit fields on z/TPF

C / C++ Language Support (continued)

- alloca function -- New C/C++ function that obtains memory from the stack.
- Assembler macros to process C/C++ library functions
  - CSTKC -- obtains or saves the address of the current C stack frame
  - EPLGC -- generates epilog code in library functions written in assembler, similar to the TMSEC macro
  - PBASC -- gets or saves the address of the previous program base
  - PRLGC -- generates prolog code in library functions written in assembler, similar to the TMSPC macro
- Assembler macros used to set up the compiler interface between an assembler program and a C/C++ program being called by the assembler program, and enables an assembler program to pass parameters to a C/C++ program without having to set up the appropriate compiler interface with the C/C++ program .
  - CPROC -- defines the C language data type of the parameters
  - CALLC -- generates the code needed to enter the C/C++ function

C / C++ Language Support (continued)

- floating point conversion routines
  - __fp_htob convert from hexadecimal floating point to IEEE floating point
  - __fp_btoh convert from IEEE floating point to hexadecimal floating point
  - __fp_bton convert from IEEE floating point to native floating point
  - __fp_ntob convert from native floating point to IEEE floating point
  - __fp_hton convert from hexadecimal floating point to native floating point
  - __fp_ntoh convert from native floating point to hexadecimal floating point
- System scope initializer functions
  - __tpf_module_init
  - __tpf_module_term
- Support for 4-byte wide character support in UCS-4 (unicode format)

Improved Diagnostics

- ECB Heap Check Mode
- Branch Target Check Mode
- Debugger dump viewing
- ZDMAP
  - E-type program link map: BSO and CSO
  - CP link map
  - ZDMAP ADDR to get program name based on address
- ZSPER enhancement to trace store of a specified data into a specified location
- SNAPC ability to include last 40 ECB trace items
- TPFDF: ZUDFM MLS offline processing replaced by online usage of Debug Data

Tracing at the ECB level

- C function trace always on
  - No special compiles required
  - Captures input parameters as well as function name and load module name
  - C function extended trace is optional
    - Includes where C function was called
    - C function exit trace - captures return parameters
- C function trace collated with macro trace
- Trace Groups
- Multiple trace buffers
  - 1 trace buffer = macro trace + C function trace
- Ability to add free form text to ECB trace - tpf_trace_info()
- Ability to turn on / off register trace in macro trace without requiring an IPL
- Trace log
- ECB Heap trace
- ECB data level trace
- Socket trace
  - Trace all socket APIs issued by ECB

Tracing at the System level

- I/O trace
  - Command to define number of trace entries to use
  - Command to change number of trace entries for a specific SDA
  - Command to display I/O trace
- Socket trace
  - Trace all socket APIs issued for a given socket
- DEBUGV trace - VFA internal trace
  - No reassembly required
- DEBUGC
  - Can be used to identify when specific conditions occur in frequently activated routines

Dump tailoring

- Ability to tailor OPR dumps
  - Add / delete memory tables
    - System tables
    - User tables
  - Dump 4K around registers on all dumps
  - Include collated trace on all dumps
  - Include all I-Streams prefix page on all dumps
- Ability to dump only blocks with a specific owner name
- Ability to tailor dumps taken in the control program by CP CSECT
- Dump extensions
  - Ability to selectively dump areas of memory needed
  - Ability to format these areas

Debugger

- Dump Viewing - use debugger to view an ECB dump
- ECB Monitoring - use debugger to view a snapshot of a long running ECB
- C/C++ macro support - compile program with -g3 option
  - Debugger can resolve expression like 'ecbptr()->ce1cr0', where ecbptr() is a C macros.
- New Debug console commands (online help, from debug console, is available)
  - ECBTRACE - display the ECB trace in the debug console
  - TRACELOG - start/stop trace log
  - ECBHEAP - display ECB heap usage
- Memory View performance enhancement
- Display C++ object initialization variables in the variable pane
- Registration with a condition clause
- Data level operation (allow user GETCC, RELCC, FLIPC, ATTAC, DETAC on a data level from debug console)

Large programs

Large amounts of memory allow larger programs

- Do not need to split programs
- Use subroutines in assembler
- Use of fork()
- Threads
- SWISC TYPE=IMMEDIATE
- More inline service routines

File System Enhancements

- Virtual File System (VFS) architecture and mountable file systems
- Memory File System (MFS)
- Fixed File System (FFS)
- Pool File System (PFS)
- File service levels
- File attributes
- File system utilities (dspsys, fsck, pax, tar, view)
- File system security (protecting files and commands)
- Shared memory extensions
- Pseudo-file systems (PROCFS and SYSFS)

File System Enhancements (continued)

- New device drivers
  - General data set
  - Tape
  - Virtual reader
  - Sockets
  - User written driver
- Full file and byte locks

Improved Tuning Capabilities

- BAL repackaging
- ECB heap lists
- Preallocated storage
  - ECB private area
  - ECB Heap
  - Application stack
- More in core tables - trade memory for MIPs
  - Move mostly read only file records into memory
  - Format 2 globals
  - System heap

Improved Availability and Operations

- Improved Availability
  - Norm State Pool Reallocation
  - FCTB load in Norm
  - Dump buffer
  - Scheduler changes
  - RIAT - Dynamically add record IDs to RIAT
  - ZAPAT option to allocate new programs online
- Improved Operations
  - ECB resource monitor (aka Resource policeman)
  - Memory management
  - Recoup deferred lost
  - (in development) Selective Recoup by SSU
  - SDA addresses up to FFFF
  - Ability to load from file system
    - FTP load to TPF
    - ZOLDR LOAD
    - ZTPLD

Migration to z/TPF

- Single source
  - TPF Toolkit automates most changes
- Coexistence in a loosely coupled complex when migrating to z/TPF
- Installation of many user enhancements
- Education
- Services

User Enhancements

- ECB resource monitor (aka Resource Policeman)
- APIs
  - LODIC enhancements
    - Additional 4 user resource classes
    - When marked ECBs create child ECBs, allow the child ECBs to use a different class
    - Count create requests (CREMC, ... , CXFRC) as ECBs
  - SNAPC to include ECB trace
  - SWISC TYPE=IMMEDIATE - move ECB to another I-Stream and start processing immediately after the SWISC
  - SYNCC WAIT=YES - global synchronization wait for all processors to be updated

User Enhancements (continued)

- Commands
  - Display memory based on CINFC tag - ZDCNF
  - Display event table - ZDEVN
  - Display I/O (LDEV) trace - ZIOTR
  - DIsplay link map for CP - ZDMAP CP
  - Display record hold table - ZDRHT
  - Display TOD isynchronization information - ZPSMS D TOD
  - ECB Resource monitor - ZECBM
  - FCTB load in Norm state - ZFCTB
  - Pool directory empty - ZPOOL EMPTY
  - Pool directory force reorder - ZPOOL FORCE REORDER
  - Software profiler - ZTRAP
  - ZDSVC to display macro name based on SVC number
  - ZDSYS to display system state for all subsystems

User Enhancements (continued)

- New CP user exits
  - Duplicate dump - UCCSDUP
  - Looping dump support - UCCSEM
  - Online database reorganization - UCCADBR
- New ECB user exits
  - Define ECB labels - ueqce1.cpy & tpf/c_ueqce1.h
  - Midnight processing - udt1.asm & udt2.asm & udt3.asm
  - RLCHA - urc4.asm & urc8.asm
  - ZSTAT - usta.asm
- TPFDF user exits
  - Following a successful FINWC/FIWHC macro for a prime block
  - Allow bypassing OPR-IDB011B system errors
  - Allow bypassing authorization checks from ZUDFM RESTRICT table
  - Allow system-wide equates for z/TPFDF C/C++ applications

User Enhancements (continued)

- General
  - Input list bypass
  - Prevent looping catastrophic dumps
  - Shutdown values are 4 bytes
  - Improved CXFRC parent processing
  - Improved lock release routine in dumps

User Enhancements (continued)

- DASD
  - Prime / dupe module pairing
  - New DASD least queuing option
  - RHT enhancements
  - Module copy DASD VSN change
- Pools
  - RLCHA internals enhancements
  - Short term pool logging
  - FC33 and CA blocks can be 1055 or 4K is size
  - More data is captured on GETFC / RELFC
- Performance tools
  - Software profiler
  - Collect DASD device measurements
- Tape: Repeat tape mount request in restart

Additional Details

DASD Support Enhancments

- 40,000 DASD
- SDAs up to FFFF
- Prime / dupe module pairing
- New DASD least queuing option
  - When prime and dupe queue length is equal always go to dupe
- ZPATH DOWN
- After module copy DASD VSN changed to use common prefix (SP) and the number part of the previous VSN.
- RHT enhancements
  - If overflow is full, use system heap to obtain more overflow entries
  - Display record hold table - ZDRHT
  - Level that Wait Queue Threshold warning message triggers is customizable
  - Record hold table monitor
- Ability to set lost interrupt timeout value - ZSONS ALTER LOSTINT
- Validate physical DASD format against FCTB expectations - ZSVTT

Dump management

- Dump groups
- Dump controls
  - Maximum number of bytes to dump in an ECB heap buffer
  - Maximum number of CPSE messages within 1 minute
  - Maximum number of control dumps within 1 minute
  - Define percentage of a work block included in a dump
  - Define skip factor to skip a percentage of blocks in a dump

Dump management (continued)

- Named manual dumps
- Duplicate dump table changes
  - Increased size
  - Ability to remove dumps from duplicate dump table
- Dump suppression
  - Ability to not dump (suppress) a SERRC or SNAPC
  - Ability to remove dumps from suppressed dump table
  - Display all suppressed dumps
  - Ability to force a dump
  - Display all forced dumps
- Dump messages
  - More information on SERRC message
  - No core errors send owner information to console
- No core dumps have unique numbers for each type of physical block

Dump format changes

- Collated macro and C function trace
- Application stack
- ECB Heap
  - Data about the buffer: address, size, obtaining load module and displacement
  - ECB heap trace
- ECB data level trace
- Hex on left, translation on right
  - Ability to do ASCII or EBCDIC translations or have a user defined code page
- Link map of failing program on CTL-3 and OPR-4 dumps
- Last 10 branch trace items show program / CP csect name and displacement into the program
- SW00SR formatted
- Owner name displayed for physical block

Dump format changes (continued)

- Memory configuration name
- Architectural changes
  - 64-bit core addresses
  - 64-bit registers
  - 128-bit PSWs
  - Floating point control regiser
  - Breaking event register

General

- Scheduler changes - use of common ready / input / defer lists
- Improved positive feedback
- FTP client
- Owners
  - Physical block owners
  - ECB Owner names
- Versionless support
- Trace name support
- RIAT - Dynamically add record IDs to RIAT
- CTL-10 processing enhancements
  - Timeout specified by program
  - Timeout is enabled in restart and 1052 state

General (continued)

- API enhancements
  - ECB Heap API enhancements
    - Tag an ECB heap buffer
    - Identify largest ECB heap buffer that can be obtained
  - DETAC on a DECB supports detaching more than 255 blocks
  - ERRNOC - assembler interface to errno value
  - Test addressing mode - TAMCC
  - Time Slice (TMLSC) restrictions removed
    - Can turn on time slice and call system services and other programs
  - Storage protection override - GLMOD / STPOC

- API enhancements
  - LODIC enhancements
    - Additional 4 user resource classes
    - Check processor utilization
    - When marked ECBs create child ECBs, allow the child ECBs to use a different class
    - Count create requests (CREMC, ... , CXFRC) as ECBs
    - Enhanced lose of control support
      - Save and restore new floating point registers.

General (continued)

- Commands
  - Data definitions (ZDMSG DEFINE) can be in the file system
  - Enhanced disassembler
  - ZDPGM program listing view
  - ZDECK
    - PAT - based on online PAT create file that can be used to build PAT
    - RIAT - based on online RIAT create file that can be used to build RIAT
- Reduce VM impact by keeping working set size to what is used
  - IPLB does not TB every 4 KB block
    - Done when frames are first dispensed
    - Done in VFA when buffer is used
  - Use of Available and Allocated lists when dispensing blocks
- 48 K Keypoints
- Ability to selectively not file VFA delay file short term pool records

Globals

- Format 2 Globals
- Format 1 global enhancements
  - Storage protection override
    - Easier for C programs to update globals
  - SYNCC option to return control once global is updated on all processors
    - SYNCC WAIT=YES
  - Modification of format 1 global restart to run concurrently with other parts of restart

Loaders

- Alternate FCTB Load
  - FCTB load in Norm state - ZFCTB
- Load from hfs on linux or zOS
- New more flexible load deck format
- Improved diagnostics and reports from offline loader
- Elimination of SALTBL; no need to maintain compatible online & offline tables
- IPAT 'merge' allows reloads without reloading programs
- ZAPAT option to allocate new programs online
- New mechanism to feed back onlne PAT changes back into control file (ZDECK command online and pat2ctl utility offline)
- Backup copy of keypoints made during LGF IPL (ACPL load)
- Support for named BSS
- BAL repackaging support (>4K linked BAL programs)
- Ability to alter all programs on file (ZAPGM)
- CP link map available via ZDMAP
- Support for large keypoints
- ZDMAP ADDR (find program containing specified address)

Pools support

- Norm State Pool Reallocation
- Ability to have GFS active in 1052 state
- Force reorder of a pool directory - ZPOOL FORCE REORDER
- Empty a pool directory - ZPOOL EMPTY
- Short term pool logging
- FC33 and CA blocks can be 1055 or 4K is size
- More data is captured on GETFC / RELFC
- RLCHA enhancements
  - More efficient queueing
  - Chain chasing changed to minimize impacts of a single large chain on short term releases
  - RLCH uses 4K queuing block

Performance Tool Enhancments

- Software Profiler
  - EI / EA / MA
- I/O Measurements
  - FICON Measurements
- CDC and Data Collection / Reduction Enhancements
  - LPAR utilizations
  - I/O Measurements
  - DASD I/O service time
  - 1 meg frame support
  - Unique collection frequency for each type of data

Tape Support Enhancements

- Tape hardware encryption support
- Dump buffering
- Large tape blocking up to 128 K
- SDAs up to FFFF
- TGETC / TPUTC API for tape operations
- Repeat tape mount request in restart

TCP / IP

- Can dynamically increase socket block and IP message table (IPMT) sizes
- High priority messages
- Faster network recovery after IPL
- Socket API traces (ECB level and socket level)
- Display socket exceptions
- Ability to analyze IP trace data using standard tools like Ethereal
- Enhanced socket sweeper diagnostics
- Sockets in 1052 state
- New INETD server models

TPFDF

- Code ships as full source
  - Part of z/TPF hierarchy
  - No more sequence numbers
- Central DB routines use standard Enter/Back
  - Allows use of I-stream scheduler and other z/TPF features
  - Eliminates E-type loader restrictions loading z/TPFDF programs
  - Eliminates special considerations for using Debugger
- New user exit for configuration values such as for user-defined algorithms
- ZUDFM MLS offline processing replaced by online usage of Debug Data
- Eliminate automatic display of entire subfile for various ZUDFM commands
- Recoup displays messages when starting/completing each DBDEF
- DBDEF segments can exceed 4K in size
- Formatted SW00SR in dumps
- Return-Optional System Errors

TPFDF User Exits

- Following a successful FINWC/FIWHC macro for a prime block
- Allow bypassing OPR-IDB011B system errors
- Allow bypassing authorization checks from ZUDFM RESTRICT table
- Allow system-wide equates for z/TPFDF C/C++ applications

New User Exits - Control Program

- About to start execution of BSO: UCCBSOS
- About to return from CSO: UCCCSOR
- About to start execution of CSO: UCCCSOS
- BACKC macro entry point: UCCBSOR
- Debugger dump selection: UCCDBDS
- Duplicate dump: UCCSDUP
- ECB resource monitor count of resource: UCCERM0
- ECB resource monitor first limit: UCCERM1
- ECB resource monitor second limit: UCCERM2
- Enter macro entry point: UCCENTM
- File address decode - online reorganization exit: UCCDDBR
- Get 1-MB frame macro: UCCGLF
- Logging for file-type operations - online reorganization exit: UCCBDBR
- Looping dump support: UCCSEM
- Queue data record for file-type operations - online reorganization exit: UCCADBR

New User Exits - Control Program (continued)

- Release 1-MB frame macro: UCCRLF
- Return pool file address: UCCRPFA
- SERRC dupl dump processing: UCCSDUP
- Trace log session: UCCTLG
- VFA delay file: UCCVFAD
- z/TPF locking on coupling facility - convert user lock name: UCCCFCL
- z/TPF locking on coupling facility - validate user lock name: UCCCFVL

New User Exits - ECB programs

- Debug file management: uelj.c
- Define user ECB labels (Assembler): ueqce1.cpy
- Define user ECB labels (C/C++ language): tpf/c_ueqce1.h
- Define user specific errno values: custer.c
- Encrypt passwords for file system security processing: ufve.c
- Format-2 global alter: ugla.cpp
- Format-2 global alter data: uglz.cpp
- Format-2 global define: ugld.cpp
- Format-2 global delete: uglt.cpp
- Format-2 global initialization: ugli.cpp
- Format-2 global keypoint: uglk.cpp
- Format-2 global load: ugll.cpp
- Format-2 global migration: uglm.cpp
- Format-2 global restart: uglr.cpp
- Format-2 global synchronization: ugls.cpp
- Format-2 global undo: uglu.cpp

New User Exits - ECB programs (continued)

- Include a specific ECB in a ZDECB summary display: uvxs.c
- Log ECB trace (macro and function trace) information: utlg.c
- Midnight processing calendar updates: udt1.asm
- Midnight processing GDATX macro: udt2.asm
- Midnight processing post calendar updates: udt3.asm
- Provide a copy of the input message of the ECB to the ZDECB command: uvxs.c
- Release chain processing for 4-byte file addresses (RLCHA HDR=4): urc4.asm
- Release chain processing for 8-byte file addresses (RLCHA HDR=8): urc8.asm
- Secure Sockets Layer (SSL) application configuration file (tpf_SSL_getConfig): uscf.c
- System error terminal response message: upsa.asm
- Unplanned module down: uyen.asm
- User command: additional validation and authorization of z/TPF commands: umex.asm
- VFA delay file: cvft.asm

New User Exits - ECB programs (continued)

- ZFCTB LOAD command compatibility processing: ufct.c
- ZFCTB command user data relocation processing: uftr.cpy
- ZFCTB command processing: uftz.c
- ZSTAT command display user exit: usta.asm

New Commands

- ZACNF - alter data referenced by CINFC label
- ZAGBL - alter the contents of a format 2 global
- ZAPFS - alter positive feedback
- ZAVFS - alter file system information
- ZDBAI - manage dump buffer area
- ZDCNF - display the main storage address of a CINFC label
- ZDDMP - manage dumps captured by the debugger
- ZDECK - create offline input deck from online table
- ZDEVN - display event table
- ZDGBL - display contents or characteristics of a format 2 global
- ZDPFS - display positive feedback information
- ZDRHT - display the record hold table
- ZDVFS - display file system configuration and trace information
- ZECBM - resource manager
- ZFCTB - FCTB load in Norm State
- ZFTPC - manage FTP client

New Commands (continued)

- ZGLBL - manage format 2 globals
- ZILGF - respond to ACPL prompt to backup keypoints
- ZIOTR - display and manage I/O trace
- ZIPDB - manage TCP/IP network services database
- ZMEAS DISPLAY - display current values of data reduction parameters
- ZOODB REUSE - set or display options for collections pool reuse table
- ZOVFS - manage file system users and groups
- ZPVFS - log in to and log out of the file system
- ZSUBC - manage subcapacity reporting
- ZSVTT - verify pool section or fixed file records
- ZTRAP - software profiler

Enhanced commands

- ZACLV - alter CPU loop and create macro levels
- ZACOR - alter core
- ZADCA - alter data referenced by dump label
- ZAPAT - alter program attribute table
- ZAPGM - alter program
- ZASER - alter system error options
- ZBROW - manage collections
- ZCACH - manage logical cache records
- ZCDCO - manage continuous data collection
- ZCTKA - display and alter keypoint A values
- ZDBUG - display and clear the debug server
- ZDCLV - display CPU loop and create macro levels
- ZDCOR - display core
- ZDDBG - display debug server
- ZDDCA - display main storage address of a dump label
- ZDDSI - display I/O device status information

Enhanced commands (continued)

- ZDECB - display in-use ECBs
- ZDMAP - display link map data
- ZDPAT - display program attribute table
- ZDPGM - display program
- ZDPLT - display program linkage type
- ZDSER - display system error options
- ZDSMG - define a data definition
- ZDSVC - display SVC code
- ZDSYS - display system operating state
- ZDTCP - TCP/IP connectivity diagnostic tools
- ZDTOD - display date, time, and TOD clock
- ZDUMP - manual dump
- ZFCAP - capture
- ZFECB - display active ECB information
- ZFILE - manage file system
- ZFKPA - Reply to change in memory configuration

Enhanced commands (continued)

- ZFRST - restore
- ZGFSP - set pools controls
- ZIDOT - display or modify dump overrides
- ZINET - manage internet server applications
- ZNKEY - display or alter SNA keypoint
- ZPATH - DASD path management
- ZPOOL - pool support
- ZPROT - utility and tape ownership
- ZPSMS - processor status management services
- ZPTCH - maintain memory patch decks
- ZRBKD - recoup descriptor functions
- ZRECP - manage recoup
- ZRPGM - retrieve a program
- ZRTDM - manage RIAT
- ZSOCK - TCP/IP tools
- ZSONS - manage DASD support controls

Enhanced commands (continued)

- ZSPER - alter and display per options
- ZSTAT - display system status
- ZSTRC - alter and display system trace options
- ZSYSG - alter and display system generation options
- ZSYSL - display or change priority class shutdown levels
- ZTDEV - modify and display tape device status for automatic tape mounting
- ZTICL - emergency tape removal
- ZTINT - initialize tape
- ZTLBL - tape label maintenance
- ZTMNT - mount a tape
- ZTOCU - dismount tapes by logical control unit
- ZTOFF - dismount tape
- ZTPLD - active image loader
- ZTPLF - manage tape library
- ZTRMT - remount tape
- ZTSTB - display tape status table entry

Enhanced commands (continued)

- ZTTCP - manage IP
- ZTVAR - configure tape devices
- ZTWTM - write tape mark
- ZVFAC - manage VFA

New Assembler General Macros

- ALLOC - allocate space on the application stack
- APSTKC - define user application stack area
- CALLC - call a C function
- CLINKC - call a BSO application subroutine
- CPROC - define a C function prototype for the CALLC macro
- CRYPC - Encrypt and decrypt data
- DEBUGC - debug facility
- DEFBC - define macro code generation options
- ECBMC - adjust the resource limits in the current ECB
- EHEAPC - manage ECB Heap storage
- EISAC - ECB I-Stream affinity set
- ELINKC - mark the end of a block of code in a subroutine
- EOWNRC - register ECB owner
- ERRNOC - set or retrieve the errno value
- ERRWPC - restore the extended register save registers
- ERSWPC - save the extended register save registers

New Assembler General Macros (continued)

- GETKC - get a keypoint record
- GLOBLC - manage format-2 global records
- ILSDC - input list shutdown test
- LBASEC - load program base of BAL shared object
- LREGSC - restore registers from the ECB register save area
- PDIRC - get file pool directory record type
- RELKC - release a keypoint record
- RLINKC - restore saved registers
- SLINKC - save registers and setup the subroutine base register
- SREGSC - store registers into the ECB register save area
- TAGDFC - generate an equate or literal for a CINFC tag
- TAMCC - test addressing mode
- TGETC - read a record from tape
- TLOGC - enable or disable ECB trace logging
- TPUTC - write a record to tape
- UPDKC - update a keypoint record

Enhanced Assembler General Macros

- BACKC - return to previous program
- BEGIN - begin assembler program
- CIFRC - cipher program interface
- CINFC - control program interface
- CM0PR - Scan input message for keywords
- CORHC - define and hold a resource
- CORUC - unhold resource
- CRESC - create new sychronous ECB
- CRETC - create time initiated entry
- DBSAC - attach TPFAR database support structure
- DBSDC - detach TPFAR database support structure
- DECBC - manage data event control blocks
- DETAC - detach an ECB working storage block
- ENQC - define and enqueue a resource
- ENTNC - enter a program with no return expected
- ENTRC - enter program with return expected

Enhanced Assembler General Macros (continued)

- EPLGC - epilog for C functions written in assembler
- EVNTC - define internal event
- FINDC - find a file record
- FINIS - finish program assembly and define program end
- GETPC - fetch program into memory
- GFSCC - initiate GFS control
- GLMOD - change global storage protection
- GLOBZ - define format-1 globals
- ITRPC - send simple network management protocol user trap
- LISTC - dump facility list generator
- LODIC - check system load and mark ECB
- MALOC - reserve a storage block
- PBASC - load previous program base
- PNAMC - find, save, or modify a program name
- PRLGC - prolog for C functions written in assembler
- RALOC - change reserved storage block size

Enhanced Assembler General Macros (continued)

- RCATC - find an RCAT entry
- RIDCC - RID conversion
- SAWNC - wait for event completion, signal aware
- SERRC - system error
- SIZBC - obtain logical size
- SNAKEY
- SNAPC - snapshot dump
- SWISC - switch entry to another I-Stream
- SYNCC - synchronize format-1 globals
- TMSPC - prolog for C functions calling TPF macro services
- TPPCC
- VIPAC - move a VIPA to another processor
- WGTAC - locate terminal entry
- WTOPC - edit and send system message

Trademarks