z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM Software Group

*TPF Users Group Fall 2005*

# z/TPF Dumps

Allan Feldman

**AIM Enterprise Platform Software**
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0
© IBM Corporation 2005

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

# Agenda

- Overview and Architectural changes
  - Regs, PSWs, Translation (Hex and EBCDIC/ASCII)
- ECB Trace Enhancements
  - C Function Trace
  - More Trace Entries
  - Trace Groups
  - Heap Trace
  - Data Level Trace
  - Trace Log
- z/TPFDF
- Dump Extensions
- Branch Trace (name/disp of pgm)
- Stack formatting (C and assembler)
- Owners
- Socket API Trace


© IBM Corporation 2005

# Agenda *(cont'd)*

- Named Manual Dumps
- Dump Groups - pgms which share zidots
- Dump Select by CP CSECT
- Dump by Owner
- Program Trace Name Enhancements
  - tpf_pnamc()
- CTL-C replacements
- Opr Dump Enhancements
- ZIDOT enhancements - FORCE
- Dump Message Enhancements


© IBM Corporation 2005

# Materials

- On USB Stick:
  - TPFUG October 2005\Wednesday Education Materials - Debugging and Dumps Class\TPFUG.sample.dump.txt
  - TPFUG October 2005\Wednesday Education Materials - Debugging and Dumps Class\zArchRefSummary\zArchRefSummary


© IBM Corporation 2005

# Dump Formatting Enhancements - Architectural Changes

- 64-bit Registers
- 128-bit PSWs
- 64-bit core addresses


© IBM Corporation 2005

# Dump Formatting Translation

- Storage formatting always has:
  - 8-byte core address on left
  - 16-byte hex display in middle
  - Translated display on right
- Translation can be
  - EBCDIC
  - ASCII
  - User defined code page
  - Specified on IDATG or LISTC macro


© IBM Corporation 2005

# ECB Trace Enhancements - C Function Trace

- C function trace is always available
  - Can be used in both test and production environments
  - Enabled by default compiler option
- Function entry and exit can be traced
  - Function entry and macro trace are equivalent
  - Function exit is a separate option
    - FUNCEXT parm on ZSTRC command
- Formatted macro and C function trace are collated together
- Function trace entries include:
  - Prototype
  - Parameter values
- Ability exists to add free-form text into function trace.


© IBM Corporation 2005

# ECB Trace Enhancements - More Trace Entries

- Number of trace entries per ECB can be set by the user
  - Maximum number of trace entries 65,535
  - TRENTRY parm on ZCTKA command
- Provides more diagnostic information for each ECB
  - Can see more of the flow through an application
- Makes debugging easier


© IBM Corporation 2005

# ECB Trace Enhancements - Trace Groups

- Problem:  middleware programs can dominate ECB trace
- Solution:  trace groups
  - Each ECB has multiple trace buffers
    - Trace buffers per ECB is user settable (min = 4, max = 254)
    - TRBUFFER parm on ZCTKA command
  - Trace Group = set of programs which share an ECB trace buffer
    - User settable by load  module
      - Control File (offline) PAT (online)
    - IBM use
      - MQ has its own trace group = IMQSGRP
      - z/TPFDF has its own trace group = DATABASE
- Makes debugging easier.


© IBM Corporation 2005

# ECB Trace Enhancements - New ECB Trace Capabilities

- ECB heap trace
  - Records all ECB heap requests (malloc, calloc, realloc, free)
  - Number of entries is user settable (max = 32,767)
    - EHTRACE parm on ZCTKA command
  - Ability to track heap usage even if ECB trace has wrapped
- ECB data level trace
  - Records name of load module
    - Obtained 4K frame in ECB private area
    - Released 4K frame in ECB private area
  - Ability to track ECB private area usage even if ECB trace has wrapped


© IBM Corporation 2005

# ECB Trace Enhancements - New ECB Trace Capabilities

- Trace log
  - Ability to record all macro and C function trace items for an ECB
  - Started by API call - TLOGC, tpf_tracelog_on()
  - Stopped by API call or ECB exit - tpf_tracelog_off()
  - Data written to tape or HFS file
  - Post processed on Linux
    - Output to HFS file


© IBM Corporation 2005

# z/TPFDF

- Current SW00SR block is formatted
- Pointer to Database Interface Block (DBIFB) in ECB is labeled
  - DBS is the tag name
  - DBIFB is in ECB Heap - included in the dump


© IBM Corporation 2005

# z/TPFDF SW00SR Example

```
*CURRENT SW00SR BLOCK
000000000C6001C0 BID E2E60000 PGM D8E7C4C8 KLS 00000000 KLS 036152AC  SW..QXDH........
000000000C6001D0 DET 00000000 WID B0750001 INB 001A00BF ILT 000100A5  ..............v
000000000C6001E0 EOR 0000004D     00320000 RBV 0001D980 OP2 06040000  ..........R.....
000000000C6001F0     00000000 ALG 0A30816C     D1000000     00000000  ......a.J.......
000000000C600200     00000000     00000000 FAD 00000000 FAD 435801EE  ................
000000000C600210 IPT 00000000 IPT 0A308150     0000001A     00000000  ......a.........
000000000C600220     00000000     00000000     00000000     00000000  ................
000000000C600230     00000000     00000000 UKY 00000000 ID1 08082600  ................
000000000C600240 ID5 00010000 ID9 00000000 IDF 00000000     00000050  ................
000000000C600250 RTN 00000400 PNB 001A0005     00000000 REC 0A30A61A  .............w.
000000000C600260 IPA 00000000     001AA000 NLR 00000000 ORD 0000003D  ................
000000000C600270     00000000     00000000 SRV 00000000 BWF 00000000  ................
000000000C600280 BPT 00000000     00000000     00000000     00000000  ................
000000000C6002A0     00000000     00000000     0000B075     00000CE7  ..............X
000000000C6002B0     C4C88001     FDAE0000     00000000     00000000  DH..............
000000000C6002C0     00000000     00000000     00000000     00000000  ................
000000000C600300     00000000     00000000     00000000     C4C6E3C4  ............DFTD
000000000C600310 CAL 00000004 RET 00000000 PM1 00000002 PM2 800200BE  ................
000000000C600320 PM3 0A308150     00000000     00000000     00000000  ..a.............
```


© IBM Corporation 2005

# Dump Extensions

- Dump processing has ability to call dump formatting extension
- Extension allows customization
  - What is to be dumped
  - Actual formatting of the data
- User extensions are available
- IBM uses extensions
    - MQ uses to dump queue and channel information
- IDATB.MAC used to determine formatting program


© IBM Corporation 2005

# Branch Trace

- Branch Trace is formatted for last 10 entries
  - Gives name of program and displacement into program
  - For CP, gives name of CP CSECT and displacement into CSECT


© IBM Corporation 2005

# Application Stack - C and Assembler

- Each in use stack frame is formatted in a dump
- Formatting contains:
  - Name of the function using the stack
    - C function name includes prototype
  - Tags to identify fields in the stack, for example:
    - BCH = Back Chain
    - PAT = PAT address
    - ...many more...


© IBM Corporation 2005

# Application Stack Assembler Example

```
*STACK FRAME FUNCTION- BXEY
0000000012C139C0 BCH 00000000     12C13C00      00000000     00000001  .....A.........
0000000012C139D0 R2  00000000     10A00000 R3   00000000     00135012  ..............
0000000012C139E0 R4  00000000     00000020 R5   00000000     000000F8  ..............8
0000000012C139F0 R6  00000000     12C13BC8 R7   0A320010     07F10008  .....A.H....1..
0000000012C13A00 R8  00000000     10A00008 R9   00000000     12C13BC8  ............A.H
0000000012C13A10 R10 00000000     000000F2 R11  00000000     0013506C  .......2........
0000000012C13A20 R12 00000000     00000008 R13  00000000     00000001  ..............
0000000012C13A30 R14 00000000     8D3C6B2A R15  00000000     0FA00000  ..............
0000000012C13A40 F0  00000000     12C13C00 F2   00000000     000000F8  .....A.........8
0000000012C13A50 F4  00000000     10A00000 F6   00000000     001351FE  ..............
0000000012C13A60 PAT 00000000     0FE030A0 CRE  00000000     00002000  ..............
0000000012C13A70 BAS 00000000     0D3C6AD0 TRN  C2E7C5E8     80000000  ........BXEY....
0000000012C13A80     00010000     00C13A08      00000000     0000E5F0  .....A........V0
```


© IBM Corporation 2005

# Dump Enhancements - Named Manual Dumps

- Problem: ZDUMP does not give enough information and ZDUMP ALL gives too much information
- Solution: Named manual dumps
  - ZDUMP has list of manual dumps
    - Predefined list targets specific functions
      - Example: Tape or SNA
    - ZIDOT can alter existing named manual dumps
    - IDOTM macro defines named manual dumps
  - Create your own manual dump
    - ZIDOT can create new named manual dumps
    - Or include IDOTM definitions in CUDP.CPY
    - You target the data that is dumped.


© IBM Corporation 2005

# Named Manual Dump Example

**Predefined Manual dump:**

```
ZDUMP IBMSNA
CSMP0097I 12.47.19 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0156I 12.47.19 DUMP SEQUENCE NUMBER 2778 STARTED ON VSN 000230+
CSMP0097I 12.47.19 CPU-B SS-BSS  SSU-HPN  IS-01
DUMP0000I 12.47.19 ZDUMP-OK+
CSMP0097I 12.47.19 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0115I 12.47.19 IS-0001 SS-BSS  SSU-HPN  SE-002778 MANUAL DUMP
010000B TRC-CVFM
CVFM    OBJ-cvfm.goff       0000024E  LOADSET-BASE
NAMED MANUAL DUMP-     IBMSNA   +
```


© IBM Corporation 2005

# Named Manual Dump Example (continued)

**Create my own Named Manual dump:**

```
ZIDOT INCLUDE MYMANUAL IMFST
CSMP0097I 12.54.45 CPU-B SS-BSS  SSU-HPN  IS-01
IDOT0002I 12.54.45 KEYWORD IMFST INCLUDED FOR NAMED MANUAL DUMP - MYMANUAL
IDOT0001I 12.54.45 COMPLETED+

ZDUMP MYMANUAL
CSMP0097I 12.55.44 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0156I 12.55.44 DUMP SEQUENCE NUMBER 2779 STARTED ON VSN 000230+
CSMP0097I 12.55.44 CPU-B SS-BSS  SSU-HPN  IS-01
DUMP0000I 12.55.44 ZDUMP-OK+
CSMP0097I 12.55.44 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0115I 12.55.44 IS-0001 SS-BSS  SSU-HPN  SE-002779 MANUAL DUMP
010000B TRC-CVFM
CVFM    OBJ-cvfm.goff       0000024E  LOADSET-BASE
NAMED MANUAL DUMP-     MYMANUAL +
```


© IBM Corporation 2005

# Dump Enhancements - Dump Groups

- Problem: If a core area is to be included in OPR dumps for 50 programs, doing 50 ZIDOTs is time consuming and error prone.
- Solution: Dump Groups
  - Set of programs which share ZIDOT dump characteristics
    - One ZIDOT command includes tables in OPR dumps for many programs
  - Defined in control file
  - Online maintained in PAT
    - ZDPAT to display
    - ZAPAT to change.


© IBM Corporation 2005

# Dump Enhancements - Select by CP CSECT

- Problem: If a core area is to be included in all dumps in a functional area of the CP, ZIDOTs for each dump is time consuming and error prone.
- Solution: Select core areas for dumps in the CP by CP CSECT
  - ZIDOT enhanced to support additions and alters
  - IDOTC macro created
  - Some examples...
    - CCTAPE dumps include ITAPE and ISWBUSE
    - CCSONS dumps include IMFST and IIOBS
    - CCSONP dumps include IPOOL and IRIAT
  - Selection used for CTL-1, CTL-2, CTL-3, OPR-4, and CPADDR=YES dumps.


© IBM Corporation 2005

# Owners

- A new mechanism to track users of system resources
  - 32-byte name is given to users of system resources
    - CMBs, SWBs, 4-KB Frames, 1-MB Frames
    - Name associated when resource obtained (GETBC, GSWBC)
  - The name is comprised of
    - 8-byte high level qualifier
    - 8-byte mid level qualifier
    - 16-byte low level qualifier
    - For example ITAPE.CCWTRANS.REAL_CCW_BLOCK.


© IBM Corporation 2005

# Owners (continued)

- ECBs can be registered to an owner
  - New API will register the ECB - EOWNRC, tpf_eownrc()
  - All resources obtained by the ECB after registration will use the registered owner name
- Online displays provided
  - ZSTAT OWNER.


© IBM Corporation 2005

# ZSTAT OWNER Example

```
ZSTAT OWNER BLOCK-FRM1MB
CSMP0097I 21.49.05 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0023I 21.49.05 BLOCK OWNER DISPLAY
                       IOB    FRAME   COMMON      SWB      ECB      FRM1MB
ALLOCATED             2704     5000      250     1416      300         300
AVAILABLE             2704     4952      247     1376      293         233
 _
                       IOB    FRAME   COMMON      SWB      ECB      FRM1MB
ICRPA                                                                   53
ISYSHEAP                                                                13
IECB                                                                     1
ITCPIP                                                                   1
END OF DISPLAY+
```


© IBM Corporation 2005

# Dump Enhancements - Dump by Owner

- Ability to dump blocks in a resource by Owner name
- Example, if ISWBUSE is selected...
  - A tape dump only needs SWBs which are owned by ITAPE
  - A MPIF dump only needs SWBs which are owned by MPIF
- ZIDOT provides ability to select by Owner
  - IDOTB provides same capability.


© IBM Corporation 2005

# ECB Socket API Trace

- Entries contain detailed information, including:
  - API input parameters
    - Includes implied parameters like time out values
  - Output, including the API return code
    - *sockerrno()* value included if error return code
  - How long it took the API to be completed
- Two trace entries created if ECB becomes blocked while processing the API:
  - First entry shows the input and when the API was issued
    - Allows you to determine if an API is pending
  - Second entry shows the output and when the API was completed


© IBM Corporation 2005

# ECB Socket API Trace *(cont'd)*

- ZSOCK TRACE command also available
  - Trace at ECB and per-socket level:
  - Useful because multiple ECBs can share a socket
  - Allows you to view recent history of the socket for faster online debugging
    - Can be used in conjunction with individual IP trace


© IBM Corporation 2005

# Dump Enhancements - No Core Dumps

- CTL-C dump number no longer used
- Unique dump number for each resource
  - 064C00 - System out of SWBs
  - 064C01 - System out of Common Blocks
  - 064C02 - System out of 4-KB frames
  - 064C03 - System out of 1-MB frames
  - 064C04 - System out of ECBs
- Allows customization
  - In TPF 4.1 need to dump all possible storage areas on a CTL-C
  - Ability to dump only what is needed
    - Example ... 064C00 - do not need to dump common blocks.


© IBM Corporation 2005

# Dump Enhancements - OPR dumps

- Enhanced ability to select core areas in an OPR dump
  - Ability to dump 4-KB of core around each register
  - Ability to include collated macro trace and I/O trace
  - Ability to include prefix pages of all I-streams
  - ZIDOT extended to OPR dumps
    - Ability to include core areas in OPR dumps
      - System tables
      - User tables.


© IBM Corporation 2005

# Dump Enhancements - Additional ZIDOT Enhancements

- Provide ability to take a dump if SERRC is in NODUMP table
  - ZIDOT FORCE
- Provide ability to take a dump if SNAPC is in NODUMP table
  - ZIDOT SNAP FORCE


© IBM Corporation 2005

# Program Trace Name Enhancements

- Standardized mechanism to modify program name used for tracing
  - No longer need to modify program name at 4(,R8)
  - PNAMC, tpf_pnamc() can be used to modify trace name
    - Set your own trace name
    - Use the caller of this routine
    - Use the name in the PAT
  - Automatically setup on enter
- Used:
  - Macro trace
  - Dump messages
  - File a record
  - Logging pool gets and releases.


© IBM Corporation 2005

# Dump Message Enhancements

- Message includes:
  - Trace name
  - Load module name
  - Object name
  - Displacement into the object where dump happened
  - If dump is in CP, CP CSECT name
  - If CTL-1, CTL-2, CTL-3, or OPR-4
    - Intruction length code + Program-interruption code
    - Failing instruction
  - If no core available dump
    - Summary of Owners information
- Dump Tape VSN where dump started is displayed.


© IBM Corporation 2005

# Dump Message Examples

**Standard OPR dump message:**

```
CPSE0152E 12.07.10 IS-0001 SS-BSS  SSU-BSS  SE-000499 OPR-IDBC161
010000A TRC-QXEN
CTDF    OBJ-ufgq.goff       000003E6  LOADSET-BASE
DFOPN -- REFERENCE NAME SHORTER THAN 8 BYTES +
```

**OPR-4 dump message:**

```
CSMP0097I 09.06.59 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0156I 09.06.59 DUMP SEQUENCE NUMBER 2380 STARTED ON VSN 000230+
CSMP0097I 09.07.56 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0152E 09.06.59 IS-0001 SS-BSS  SSU-HPN  SE-002380 OPR-I000004
010000B TRC-CVXS
CVXS    OBJ-cvxsf4.goff     00000164  LOADSET-BASE
PSW    07150001 80000000  00000003 8C5EF4CC  PIC 003B ILC 0004 I-50F02070
R0-2   00000000 00000008  00000003 8C5EF3BE  00000000 8CD04780
R3-5   00000000 0CD07028  BBF0067F ED5CAA2A  00000000 0AA0F000
R6-8   00000000 0AA101D8  00000000 0AA00390  00000000 0CC0E3E0
R9-11  00000000 0AA00000  00000000 00014BB0  00000000 00001000
R12-14 00000000 00002000  00000003 8C5EF4AC  00000000 00000000
R15    00000000 00000003
```


© IBM Corporation 2005

# Dump Message Examples (continued)

**No core dump message:**

```
CPSE0151T 09.21.21 IS-0001 SS-BSS  SSU-HPN  SE-002579 CTL-I064C02 CATASTROPHIC
                                      CSECT-CCNUCL    0000CA12
NO FRAMES AVAILABLE
                        IOB   FRAME  COMMON     SWB      ECB      FRM1MB
ALLOCATED           0002504 0001000 0000100 0001092 0000050 0000000220
AVAILABLE           0002502 0000000 0000096 0000065 0000047 0000000153
                        IOB   FRAME  COMMON     SWB      ECB      FRM1MB
ITAPE                       0000993
IECB                        0000004
ISYSTEM                     0000003
END OF OWNERS DISPLAY
```


© IBM Corporation 2005

# Trademarks

IBM is a trademarks of International Business Machines Corporation in the United States, other countries, or both.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both

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

Notes
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.


© IBM Corporation 2005