



TPF Toolkit for WebSphere Studio V3.2

TPF Users Group Fall 2007 TPF Debugger Update

Name: Josh Wisniewski
Venue: Development Tools
Subcommittee

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

Any reference 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.

TPF Debugger Update

- ECB, DECB, and Data Level views
- Toolkit XML Map Enhancements
- TPF XML Map Generator
- SW00SR View
- #define Support
- TPF Debugger Thread Support
- APAR and Toolkit Level Information

ECB View

The screenshot displays the ECB View in the TPF Debugger, showing three panels:

- ECB Tree:** A hierarchical view of the ECB structure. The selected path is `ECB : Layout ztp` > `CHW` > `BAD` > `W00-103` > `W00`. Other visible nodes include `004`, `008`, `012`, `016`, `020`, `024`, `028`, `032`, and `036`.
- ECB Dump:** A table showing the contents of the 16 data levels. The address is `0xB100000`. The table has columns for Address, Data Level, and four data levels (0-3, 4-7, 8-B, C-F).

Address		0 - 3		4 - 7		8 - B		C - F	
B100000	CHW	00000000		BAD	00000000	W00	C4C2E4C7	004	C3E5E9E9
B100010	008	80B00000		012	00000000	016	00000000	020	00000000
B100020	024	00000000		028	00000000	032	00000000	036	00000000
B100030	040	01000000		044	00000000	048	00000000	052	E2D4D7C2
B100040	056	010000C2		060	80B00000	064	00000000	068	00000000
B100050	072	00008400		076	04000000	080	E3C5E2E3	084	00000000
B100060	088	00000000		092	037C58C8	096	00000000	100	00000000
B100070	SW1	00000000		CM1	01000000	FA0	00000000		00000000
B100080	FA1	00000000			00000000	FA2	00000000		00000000
B100090	FA3	00000000			00000000	FA4	00000000		00000000
B1000A0	FA5	00000000			00000000	FA6	00000000		00000000
B1000B0	FA7	00000000			00000000	FA8	00000000		00000000
B1000C0	FA9	00000000			00000000	FAA	00000000		00000000
B1000D0	FAB	00000000			00000000	FAC	00000000		00000000
- ECB EBCDIC:** A table showing the EBCDIC representation of the data levels. The address is `0xB100000`. The table has columns for Data Level and four data levels (0-3, 4-7, 8-B, C-F).

	0 - 3		4 - 7		8 - B		C - F
	rrrr		rrrr		DEBUG		CVZZ
	ø^rr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		SMPB
	rrrB		ø^rr		rrrr		rrrr
	rrdr		ørrr		TEST		rrrr
	rrrr		L@iH		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr
	rrrr		rrrr		rrrr		rrrr

- Data Level view shows the contents of the 16 data levels
- Rec Held indicates the file address is held in the record hold table for this ECB
- Backed by XML map for customization
- For more information see <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm> (TPF Toolkit Task Force – TPF Debugger TPF Views)

Data Level View

Console Tasks ECB DECB Data Level Debug Console Memory

Data Level Table

ecbptr : 0xB100000 <Data Level Table>

Name	Blk Addr	Blk Type	Blk Size	RID	RCC	File Addr	File Ext	SUD	Rec Held	DETAC Cnt
D0	0B106E80	0021	017D	0000	00	00000000	0000000000000000	00	No	00
D1	0B108000	0001	0FFF	0000	00	00000000	0000000000000000	00	No	00
D2	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
D3	0B106480	0021	017D	E7C9	00	102905A4	0000000000000000	00	Yes	00
D4	0B106C00	0021	017D	E7C9	00	102905A4	0000000000000000	00	Yes	00
D5	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
D6	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
D7	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
D8	0B106C00	0001	017D	0000	00	00000000	0000000000000000	00	No	00
D9	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
DA	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
DB	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
DC	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
DD	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
DE	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00
DF	00000000	0001	0000	0000	00	00000000	0000000000000000	00	No	00

- Data Level view shows the contents of the 16 data levels
- Rec Held indicates the file address is held in the record hold table for this ECB
- Backed by XML map for customization
- For more information see <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm> (TPF Toolkit Task Force – TPF Debugger TPF Views)

DECB View

Console Tasks ECB DECB Data Level Debug Console Memory

DECB Table

0xB108D20 : 0xB108EAD <DECB Table>

Addr	Name	Blk Addr	Blk Type	Blk Size	RID	RCC	File Addr	File Ext	SUD	Rec Held	DETAC Cnt
0B108020		0B106600	0021	017D	E7C9	00	00000000102905A3	0000000000000000	00	Yes	0000
0B108D20	QDCA.FIWHC.FIND	0B106C00	0001	017D	E7C9	09	00000000102905A1	0000000000000000	00	No	0000
0B108DA0	QDCA.FIWHC.FILE	0B106A80	0001	017D	E7C9	09	00000000102905A2	0000000000000000	00	No	0000
0B108E20	QDCA.COPY.LEVEL	0B106900	0021	017D	0000	00	0000000000000000	0000000000000000	00	No	0000
0B108EAD	QDCA.COPY.DECB	0B106780	0021	017D	0000	00	0000000000000000	0000000000000000	00	No	0000
0B108F20		0B106A80	0021	017D	E7C9	00	00000000102905A3	0000000000000000	00	Yes	0000

- DECB view shows the contents of the created DECBs
- Like the Data Level view in terms of functionality provided
- Backed by XML map for customization
- For more information see <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm> (TPF Toolkit Task Force – TPF Debugger TPF Views)

Toolkit XML Map Enhancements

- Memory view XML map usability enhancements
 - Choosing columns to display
 - Click and drag columns to reorder
 - Click and drag to resize columns
 - Choosing offset display type (decimal or hex)
 - Representation of data (hex, ASCII, or EBCDIC)
 - Edit data
 - Show types (from XML)
 - Ability to modify the XML maps (XML editor)
 - **Find field**
- Memory view XML ORG_GROUP support
- For more information see <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm> (TPF Toolkit Task Force – TPF Debugger TPF Views)

TPF XML Map Generator

- V03001F – Requested automatic generation of XML maps of C/C++ headers and ASM DSECTs
- TPF XML Map Generator automatically generates XML maps of ASM DSECTs
- XML maps for TPFDF LRECs contain an additional generated ORG statement for each primary key section. (In the following explanation XXXXXX and XXXX represent a portion the LREC DSECT name)
 - TPFDF LREC DSECTs are identified by the presence of #XXXXXXS and #XXXXXXI fields
 - Beginning of key section is assumed to be XXXXORG
 - Keys are identified by using #XXXXKYY and XXXXEYY fields where YY is a primary key identifier
- C/C++ header files (structures, classes, etc) are planned to be supported in the future for z/TPF

TPF XML Map Generator

- XML maps contain the following fields
 - Typical fields in the DSECT (EQU generally not handled)
 - ORG statements
 - XML maps for TPFDF LRECs contain an additional generated ORG statement for each primary key section
- XML maps contain the following information regarding DSECT fields
 - Field name as declared in the DSECT
 - Field size as declared in the DSECT
 - Field type as declared in the DSECT
 - Field description as declared in the DSECT

TPF XML Map Generator Usage

TPF 4.1 and z/TPF recommended usage

1. Include `ASMFLAGS_USER += XMLGEN` in your Toolkit configuration to generate XML maps automatically whenever you build an ASM object

TPF 4.1 manual usage

1. Assemble a segment to produce `SYSADATA`
2. Run `TPFSYM` against the `SYSADATA` to produce `ADATA`
3. Run `XMLAGGR` against the `ADATA` to produce the XML maps for every data macro included in the object

z/TPF manual usage

1. Include `ASMFLAGS_USER += XMLGEN` in your `maketpf.cfg`
2. Build the BSO
 - `XMLAGGR` is run against resulting object

TPF XML Map Generator Usage

1. To create an XML map for a specific data macro, create a trivial program to build

```
BEGIN NAME=QZZ2,VERSION=40,IBM=YES  
DR12ED REG=R1  
FINIS  
END
```

2. Follow the previously outlined build steps

Example DSECT

```

#DR12EDS EQU    &SW00WRS          BLOCK SIZE
#DR12EDI EQU    X'&SW00WID'      FILE ID
DR12REC&CG1 DS   0CL1            1ST RECORD START (1=VARIABLE,ELSE SIZE)
DR12SIZ&CG1 DS   H              SIZE OF LOGICAL RECORD
DR12KEY&CG1 DS   X              LOGICAL RECORD IDENTIFIER
*          EQUATE OF LOGICAL RECORD KEYS (KEY AND LENGTH)          *
#DR12KD3 EQU    X'D3'           LOGICAL RECORD KEY X'D3'
#DR12LD3 EQU    DR12ED3&CG1-DR12REC&CG1 LENGTH OF LOGICAL RECORD X'D3'
#DR12KD7 EQU    X'D7'           LOGICAL RECORD KEY X'D3'
#DR12LD7 EQU    DR12ED7&CG1-DR12REC&CG1 LENGTH OF LOGICAL RECORD X'D7'
DR12ORG&CG1 EQU *              START VARIABLE DATA PER LREC
*          AUTHORIZED USERS LREC          *
DR12ND3&CG1 DS   CL20           USER NAME
DR12SSN&CG1 DS   CL9           SOCIAL SECURITY NUMBER
DR12ADR&CG1 DS   CL40          USER ADDRESS
DR12ED3&CG1 EQU *              END
          ORG DR12ORG&CG1
*          PURCHASE HISTORY LREC          *
DR12ND7&CG1 DS   CL20           USER NAME
DR12VID&CG1 DS   PL5           VENDOR ID
DR12ED7&CG1 EQU *              END
          ORG DR12ORG&CG1

```

Example XML Map

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<LAYOUT Header = "DR12ED" length="81">
<ORG_GROUP FIELD="DR12HDR" Header="DR12REC">
<FIELD offset="0" Type="EBCDIC" Header="DR12REC" length="1" description="1ST RECORD START"></FIELD>
<ORG_GROUP FIELD="*NONE" Header="DR12SIZ">
<FIELD offset="0" Type="HEX" Header="DR12SIZ" length="2" description="SIZE OF LOGICAL RECORD"></FIELD>
<FIELD offset="2" Type="HEX" Header="DR12KEY" length="1" description="LOGICAL RECORD IDENTIFIER"></FIELD>
</ORG_GROUP>
</ORG_GROUP>
<ORG_GROUP FIELD="0" Header="#DR12_KEY_D3">
<FIELD offset="3" Type="EBCDIC" Header="DR12ND3" length="20" description="USER NAME"></FIELD>
<FIELD offset="23" Type="EBCDIC" Header="DR12SSN" length="9" description="SOCIAL SECURITY NUMBER"></FIELD>
<FIELD offset="32" Type="EBCDIC" Header="DR12ADR" length="40" description="USER ADDRESS"></FIELD>
</ORG_GROUP>
<ORG_GROUP FIELD="0" Header="#DR12_KEY_D7">
<FIELD offset="3" Type="EBCDIC" Header="DR12ND7" length="20" description="USER NAME"></FIELD>
<FIELD offset="23" Type="HEX" Header="DR12VID" length="5" description="VENDOR ID"></FIELD>
</ORG_GROUP>
</LAYOUT>
```


SW00SR View

ECB DECB Data Level SW00SR Memory

SW00SR Summary

D38FA00 : 0xD38FA00

SW00SR Address	WID	REF	DSECT	CMD	RTN	PCA	REC	FAD8
0D38FA00	B221	DR21ED	DR21ED	01	00	0A90A000	0A90A01A	00000000C022E005
0D390100	B226	DR26ED	DR26ED	01	00	0A90E000	0A90E025	000000008075E62F
0D390800	B223	DR23ED	DR23ED	01	00	0A90C000	0A90C01A	00000000C0230005
0D390F00	B222	DR22ED	DR22ED	01	00	0A916000	0A9161FE	000000008073C00F
0D391600	B221	@IDX	DR21ED	01	00	0A914000	0A91401A	00000000C022E005

Details of the selected SW00SR

DBIFB Info File Info Context Keys Core Block SW00SR <D390100>

Description	Variable	Value
USER PROGRAM STAMP (4,RAP)	SW00PGM	QXHF
ID OF REQUIRED FILE	SW00WID	B226
LAST TPFDF COMMAND	SW00CMD	01
RETURN CODE	SW00RTN	00
2ND RETURN INDICATOR	SW00RT2	00
ERROR COUNTER (# OF ERRORS SINCE OPEN)	SW00RT1	00
BAM RETURN INDICATORS	SW00RT3	00
8-BYTE CURRENT DATA BLOCK ADDRESS (FILE)	SW00CFA8	000000008075E62F
4-BYTE CURRENT DATA BLOCK ADDRESS (CORE)	SW00CCA	0A90E000
CURRENT NAB USED BY ACCESS ROUTINE	SW00NAB	0025

- Split into two panes: Summary & Details
 - Summary shows all active SW00SRs for the debugged ECB
 - Details shows content for the selected SW00SR in the Summary pane

SW00SR View: Summary

SW00SR Summary								
D38FA00 : 0xD38FA00								
SW00SR Address	WID	REF	DSECT	CMD	RTN	PCA	REC	FAD8
0D38FA00	B221	DR21ED	DR21ED	01	00	0A90A000	0A90A01A	00000000C022E005
0D390100	B226	DR26ED	DR26ED	01	00	0A90E000	0A90E025	000000008075E62F
0D390800	B223	DR23ED	DR23ED	01	00	0A90C000	0A90C01A	00000000C0230005
0D390F00	B222	DR22ED	DR22ED	01	00	0A916000	0A9161FE	000000008073C00F
0D391600	B221	@IDX	DR21ED	01	00	0A914000	0A91401A	00000000C022E005

- This pane shows all active SW00SRs for the debugged ECB
- Backed by two XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr.xml for location and
 config/TPFSHARE/map files/display/tpf41/SW00SRSummaryTable.xml for the layout or
 config/TPFSHARE/map files/memory/ztpf/sw00sr.xml for location and
 config/TPFSHARE/map files/display/ztpf/SW00SRSummaryTable.xml for the layout

SW00SR View: DBIFB Info

DBIFB Info		
File Info		
Context		
Keys		
Core Block		
SW00SR <DB95000>		
Description	Variable	Value
FILES OPEN	CE1DBO	5
DBIFB ADDRESS	CE1DBS	DB93830
HIGHEST SLOT NUMBER USED	SW00SWN	0005
NUMBER OF ACTIVE SLOTS	SW00SWT	0005
SW00SR BASE ADDRESS OF LAST USED FILE	SW00SWB	0DB95700
LAST USED FILE REFERENCE NAME	SW00LRF	@IDX#]
DECB LEVEL-DD COPY AND WORK BLOCK	SW00DWD	0B108D20
DBIFB INTERNAL SAVE AREA	SW00RS2	00000001
DBIFB INTERNAL SAVE AREA	SW00RS3	0DB93870
DBIFB HEADER FLAG	SW00HID	00

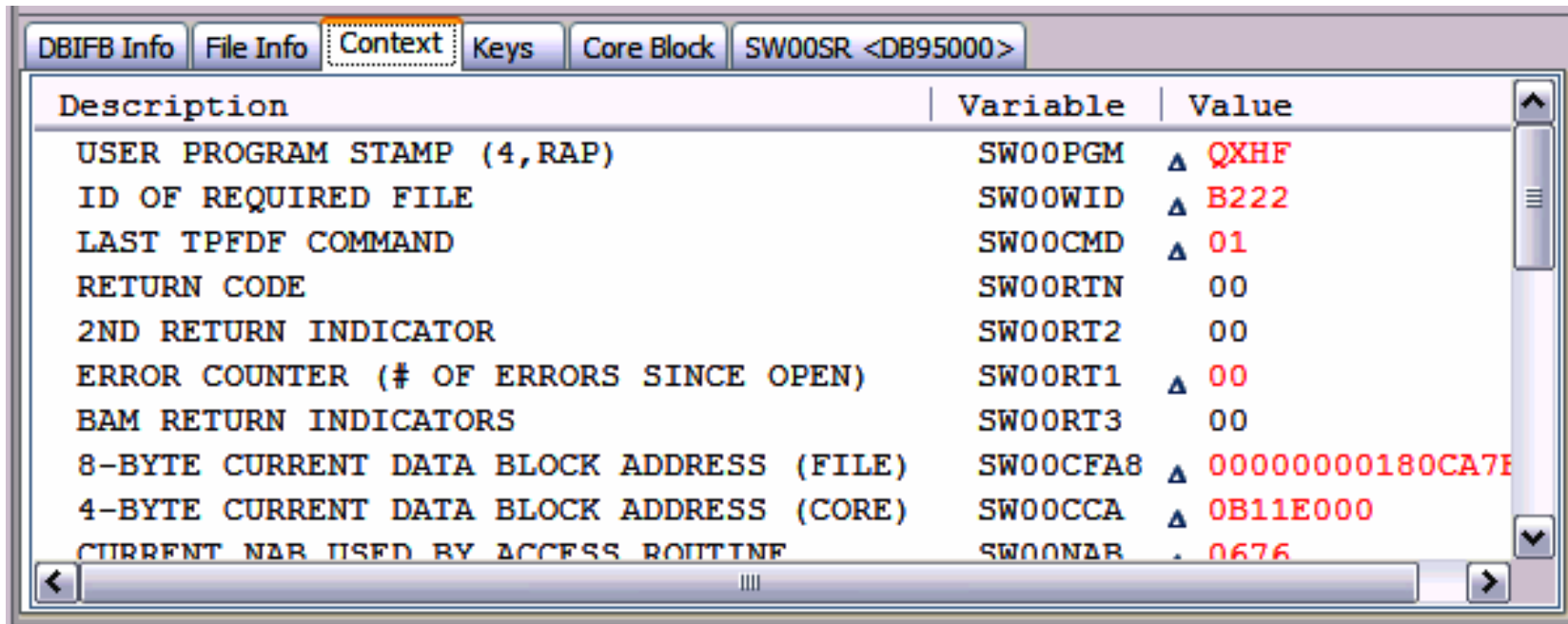
- This tab in the details pane shows the DBIFB Information for the selected SW00SR
- Backed by two XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr.xml for location and
 config/TPFSHARE/map files/display/tpf41/SW00SRDBIFBHeaderTab.xml for the layout or
 config/TPFSHARE/map files/memory/ztpf/sw00sr.xml for location and
 config/TPFSHARE/map files/display/ztpf/SW00SRDBIFBHeaderTab.xml for the layout

SW00SR View: File Info

Description	Variable	Value
ID OF REQUIRED FILE	SW00WID	B222
FILE TYPE FOR FACE AS GIVEN IN SW02SR	SW00RCT	0000
CURRENT ORDINAL NUMBER	SW00ORD	00000000
FILE TYPE	SW00TYP	R
PRIMARY BLOCK TYPE	SW00WRS	51
ALTERNATE BLOCK TYPE	SW00ARS	00
DEFAULT POOL TYPE (0)	SW00PFO	02
RETR. ALGORITHM BV	SW00RBV	00FF
RETRIEVAL ALGORITHM ARGUMENTS	SW00ALG	0DA0F3FF000000000000
PATH NUMBER FOR INDEX ACCESS	SW00PTH	0002

- This tab in the details pane shows a logical grouping of data for the selected SW00SR
 - Backed by two XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr.xml for location and
 config/TPFSHARE/map files/display/tpf41/SW00SRFileInfoTab.xml for the layout
- config/TPFSHARE/map files/memory/ztpf/sw00sr.xml for location and
 config/TPFSHARE/map files/display/ztpf/SW00SRFileInfoTab.xml for the layout

SW00SR View: Context



Description	Variable	Value
USER PROGRAM STAMP (4,RAP)	SW00PGM	Δ QXHF
ID OF REQUIRED FILE	SW00WID	Δ B222
LAST TPFDF COMMAND	SW00CMD	Δ 01
RETURN CODE	SW00RTN	00
2ND RETURN INDICATOR	SW00RT2	00
ERROR COUNTER (# OF ERRORS SINCE OPEN)	SW00RT1	Δ 00
BAM RETURN INDICATORS	SW00RT3	00
8-BYTE CURRENT DATA BLOCK ADDRESS (FILE)	SW00CFAB	Δ 00000000180CA7E
4-BYTE CURRENT DATA BLOCK ADDRESS (CORE)	SW00CCA	Δ 0B11E000
CURRENT NAR USED BY ACCESS ROUTINE	SW00NAR	- 0676

- This tab in the details pane shows a logical grouping of data for the selected SW00SR
- Backed by two XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr.xml for location and
 config/TPFSHARE/map files/display/tpf41/SW00SRContextTab.xml for the layout or
 config/TPFSHARE/map files/memory/ztpf/sw00sr.xml for location and
 config/TPFSHARE/map files/display/ztpf/SW00SRContextTab.xml for the layout

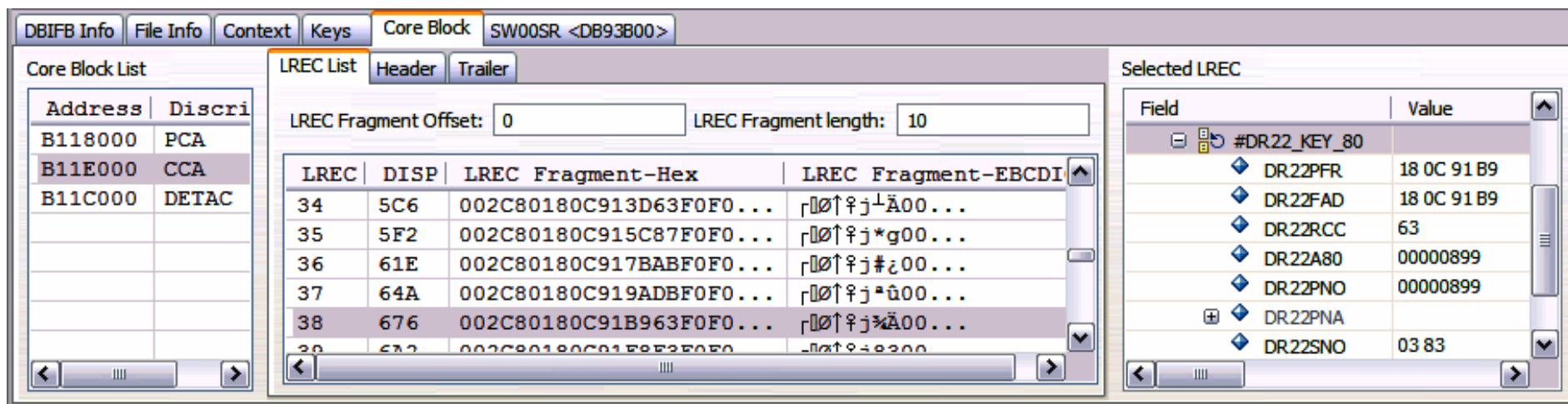
SW00SR View: Keys

Description	Variable	Value	Key Instructions			
CORE ADDRESS OF KEY BLOCK	SW00KEY	00000000	App Key	Instructions	Byte Code	Search Argumen
KEY INDICATORS	SW00NKY	A0	KEY 001	L R6, 376 (R3)	58603178	ODAF407
ADDRESS OF KEY LIST	SW00KLS8	000000000DAF8500		CLC 2 (1, R4), 0 (R6)	D50040026000	
				BL 4 (R14, R0)	474E0004	
				BNZ 24 (R14, R0)	477E0018	
			KEY 002	L R6, 380 (R3)	5860317C	ODAF6005
				CLC 8 (3, R4), 0 (R6)	D50240086000	
				BL 4 (R14, R0)	474E0004	
				BNZ 24 (R14, R0)	477E0018	

- This tab in the details pane shows a logical grouping of key data for the selected SW00SR
 - Backed by two XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr.xml for location and
 config/TPFSHARE/map files/display/tpf41/SW00SRKeyTab.xml for the layout or

 config/TPFSHARE/map files/memory/ztpf/sw00sr.xml for location and
 config/TPFSHARE/map files/display/ztpf/SW00SRKeyTab.xml for the layout
- This tab also shows the disassembled key instructions and the corresponding search arguments
 - The disassembled key instructions are either the keys that the application has setup (if any exist) or the default keys from the DBDEF

SW00SR View: Core Block



The screenshot displays the SW00SR <DB93B00> interface. The 'Core Block List' pane shows the following data:

Address	Discri
B118000	PCA
B11E000	CCA
B11C000	DETAC

The 'LREC List' pane shows the following data:

LREC	DISP	LREC Fragment-Hex	LREC Fragment-EBCDI
34	5C6	002C80180C913D63F0F0...	⌈⊘↑∫j⌞Ä00...
35	5F2	002C80180C915C87F0F0...	⌈⊘↑∫j*g00...
36	61E	002C80180C917BABF0F0...	⌈⊘↑∫j#z00...
37	64A	002C80180C919ADBFOF0...	⌈⊘↑∫j*â00...
38	676	002C80180C91B963F0F0...	⌈⊘↑∫j*Ä00...

The 'Selected LREC' pane shows the following data:

Field	Value
#DR22_KEY_80	
DR22PFR	18 0C 91 B9
DR22FAD	18 0C 91 B9
DR22RCC	63
DR22A80	00000899
DR22PNO	00000899
DR22PNA	
DR22SNO	03 83

- This tab in the details pane shows a list of all core blocks in memory that can be viewed in the Core Block tab (note DETAC blocks)
- After selecting an address from the Core Block List, the center pane is filled in with the corresponding content.
 - The core block containing SW00REC is selected by default

SW00SR View: Core Block

ext Keys Core Block SW00SR <DB93B00>

LREC List Header Trailer

LREC Fragment Offset: 0 LREC Fragment length: 10

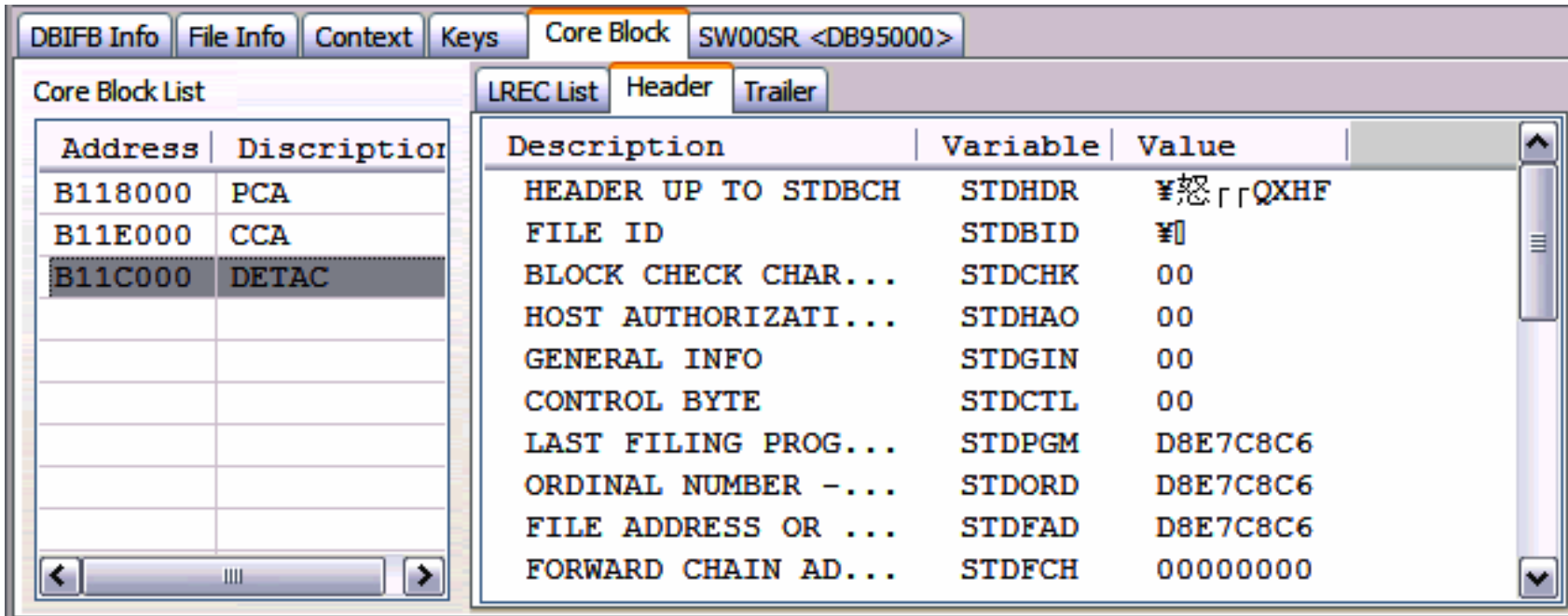
LREC	DISP	LREC Fragment-Hex	LREC Fragment-EBCDI
34	5C6	002C80180C913D63F0F0...	┌┐┐↑↑↑j┐┐Ä00...
35	5F2	002C80180C915C87F0F0...	┌┐┐↑↑↑j*g00...
36	61E	002C80180C917BABF0F0...	┌┐┐↑↑↑j#z00...
37	64A	002C80180C919ADBFF0F0...	┌┐┐↑↑↑j^û00...
38	676	002C80180C91B963F0F0...	┌┐┐↑↑↑j%Ä00...
39	6A2	002C80180C91E8F3F0F0...	┌┐┐↑↑↑j=9200

Selected LREC

Field	Value
#DR22_KEY_80	
DR22PFR	18 0C 91 B9
DR22FAD	18 0C 91 B9
DR22RCC	63
DR22A80	00000899
DR22PNO	00000899
DR22PNA	
DR22SNO	03 83

- Upon selecting an LREC in the LREC List tab, if available, the LREC XML map is opened and expanded to the corresponding primary key mapping (this is the generated primary key ORG section discussed previously in the TPF XML Map Generator portion of this presentation)
 - The LREC pointed to by SW00REC is selected by default
- These LREC XML maps must be provided by the user and located in Config\TPFSHARE\map files\memory\tpf41\sw00sr\ or Config\TPFSHARE\map files\memory\ztpf\sw00sr\ (This is a good use of the XML generator!)

SW00SR View: Core Block

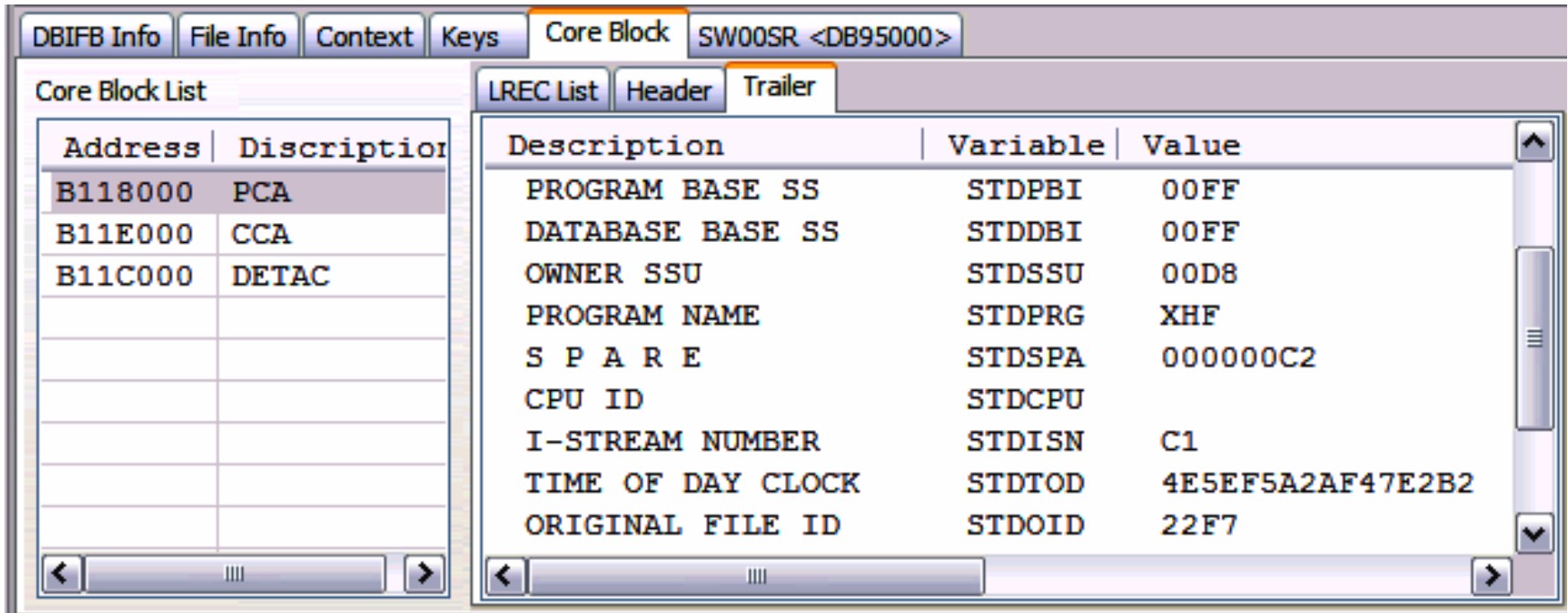


Address	Description
B118000	PCA
B11E000	CCA
B11C000	DETAC

Description	Variable	Value
HEADER UP TO STDBCH	STDHDR	¥怒「「QXHF
FILE ID	STDBID	¥
BLOCK CHECK CHAR...	STDCHK	00
HOST AUTHORIZATI...	STDHAO	00
GENERAL INFO	STDGIN	00
CONTROL BYTE	STDCTL	00
LAST FILING PROG...	STDPGM	D8E7C8C6
ORDINAL NUMBER -...	STDORD	D8E7C8C6
FILE ADDRESS OR ...	STDFAD	D8E7C8C6
FORWARD CHAIN AD...	STDFCH	00000000

- This tab in the Core Block tab shows the formatted Header information for the currently selected core block
- The header is backed by one of the XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr/STDHD_STDHD.xml or
 config/TPFSHARE/map files/memory/tpf41/sw00sr/ISTD8_ISTD8_HDR.xml for the layout or
 config/TPFSHARE/map files/memory/ztpf/sw00sr/STDHD_STDHD.xml or
 config/TPFSHARE/map files/memory/ztpf/sw00sr/ISTD8_ISTD8_HDR.xml for the layout

SW00SR View: Core Block



Address	Description
B118000	PCA
B11E000	CCA
B11C000	DETAC

Description	Variable	Value
PROGRAM BASE SS	STDPBI	00FF
DATABASE BASE SS	STDDBI	00FF
OWNER SSU	STDSSU	00D8
PROGRAM NAME	STDPRG	XHF
S P A R E	STDSPA	000000C2
CPU ID	STDCPU	
I-STREAM NUMBER	STDISN	C1
TIME OF DAY CLOCK	STDTOD	4E5EF5A2AF47E2B2
ORIGINAL FILE ID	STDOID	22F7

- This tab in the Core Block tab shows the formatted Trailer information for the currently selected core block
- The Trailer tab is backed by one of the XML maps:
 config/TPFSHARE/map files/memory/tpf41/sw00sr/STDHD_STDBTL.xml or
 config/TPFSHARE/map files/memory/tpf41/sw00sr/ISTD8_ISTD8_BTL for the layout or
 config/TPFSHARE/map files/memory/ztpf/sw00sr/STDHD_STDBTL.xml or
 config/TPFSHARE/map files/memory/ztpf/sw00sr/ISTD8_ISTD8_BTL for the layout

SW00SR View: Full XML map

DBIFB Header				
File Info				
Context				
Keys				
Core Block				
SW00SR <DD95000>				
Field	Value	Offset	Description	
◆ HEX SW00LVL	70	0x7F	LEVEL INFORMATION	
◆ HEX SW00RTN	00	0x80	RETURN CODE	
◆ HEX SW00RT1	00	0x81	ERROR COUNTER (# OF ERRORS SINCE OPEN)	
◆ HEX SW00RT2	00	0x82	2ND RETURN INDICATOR	
◆ HEX SW00RT3	00	0x83	BAM RETURN INDICATORS	
◆ HEX SW00SP8	00	0x84	SPARE, SAVE FOR SW00RT EXPANSION	
◆ HEX SW00DSC	00 00 00 00 00 00	0x85	DUMP SEGMENT CODE (IF FILLED IN)	
◆ HEX SW00DDC	00	0x8B	DUMP DETAIL CODE	
◆ HEX SW00DRC	00 00 00 00	0x8C	DUMP RETURN CODE	
◆ HEX SW00SP7	00 00 00 00	0x90	SPARE, SAVE FOR SW00REC EXPANSION	

- This tab in the details pane shows all fields for the selected SW00SR
- Backed by one XML maps:
<config/TPFSHARE/map files/memory/tpf41/sw00sr.xml> for location and layout or
<config/TPFSHARE/map files/memory/ztpf/sw00sr.xml> for location and layout

#define Support

In z/TPF, compiling C/C++ programs with `-g3` will generate debug information for the debugger to evaluate `#defines` in expressions.

Similar support has been retro-fitted to TPF 4.1 such that `#defines` will now evaluate in C/C++ expressions.

In order to use this TPF 4.1 debugger `#define` support, you must add your `#define` statements to `usot.cpp`, rebuild USOT, and load it to your system.

#define Support Examples

USOT example #define table entries:

```
extern const TPF_EEDEFINES user_eeDefines[] = {
    {"ebw000", "ce1wka[0]"},
    {"ebw001", "ce1wka[1]"},
    {"ip",     "input_ptr"},
    {"increment(a)", "a+1"},
    {NULL, NULL} /* end table with NULL entry */
};
```

With the above defined in the USOT table, the user can enter the following expression:

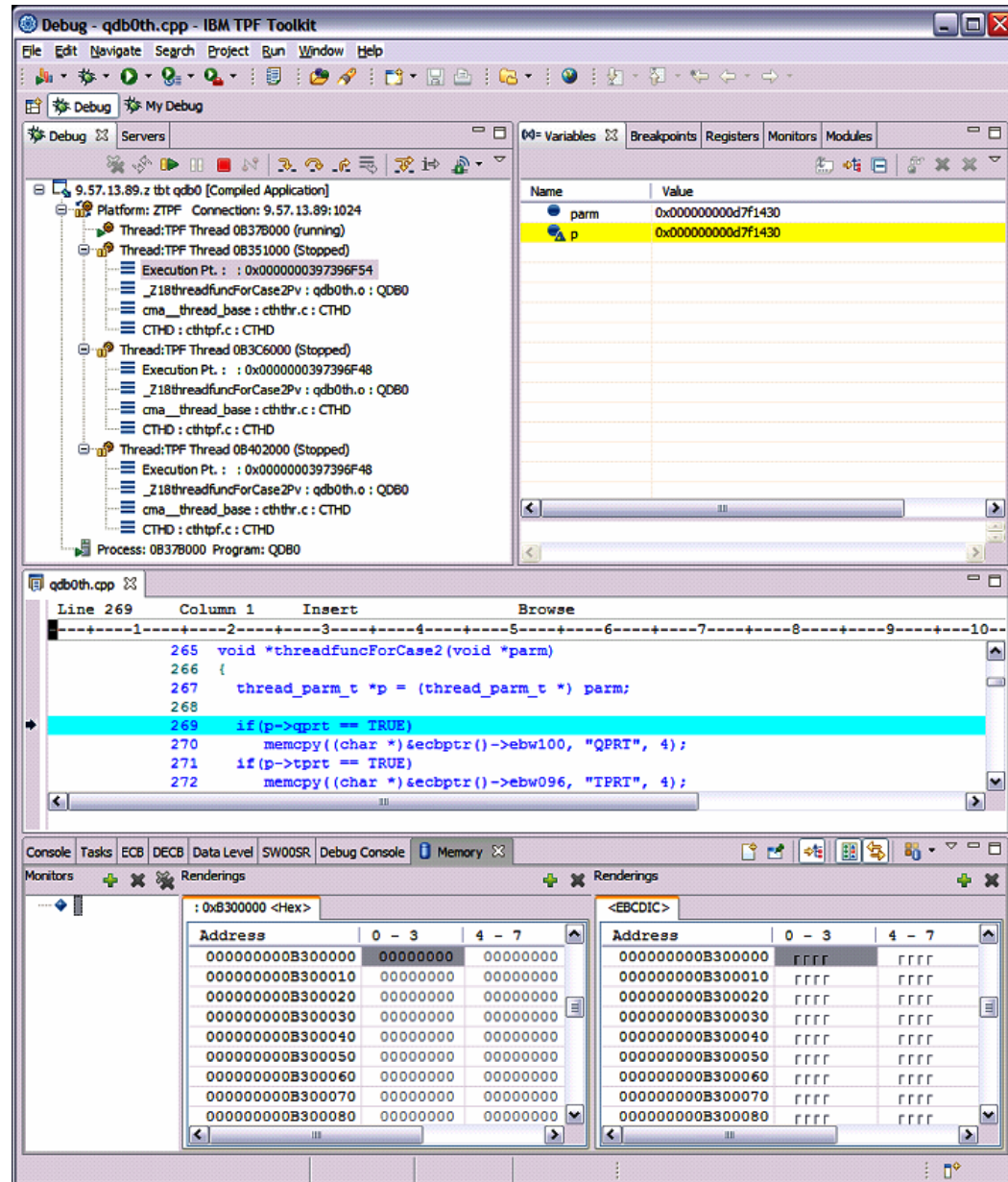
```
ecbptr()->ebw000
```

The debugger will automatically substitute the #define value during expression evaluation, such as:

```
ecbptr()->ce1wka[0]
```

TPF Debugger Thread Support

- z/TPF fully supports the threaded environment with PUT 04 and the z/TPF Debugger has been updated to provide a threaded debugging environment.



The screenshot displays the IBM TPF Debugger interface for a compiled application named 'qdb0th.cpp'. The top panel shows a tree view of threads, including 'Thread:TPF Thread 0B378000 (running)'. The middle panel shows the source code for 'qdb0th.cpp' with line 269 highlighted. The bottom panel shows memory renderings for addresses 0xB300000 and 0xB300080.

Thread List:

- 9.57.13.89.z tbt qdb0 [Compiled Application]
 - Platform: ZTPF Connection: 9.57.13.89:1024
 - Thread:TPF Thread 0B378000 (running)
 - Execution Pt. : 0x0000000397396F54
 - _Z18threadfuncForCase2Pv : qdb0th.o : QDB0
 - cma_thread_base : cththr.c : CTHD
 - CTHD : cthtpf.c : CTHD
 - Thread:TPF Thread 0B351000 (Stopped)
 - Execution Pt. : 0x0000000397396F54
 - _Z18threadfuncForCase2Pv : qdb0th.o : QDB0
 - cma_thread_base : cththr.c : CTHD
 - CTHD : cthtpf.c : CTHD
 - Thread:TPF Thread 0B3C6000 (Stopped)
 - Execution Pt. : 0x0000000397396F48
 - _Z18threadfuncForCase2Pv : qdb0th.o : QDB0
 - cma_thread_base : cththr.c : CTHD
 - CTHD : cthtpf.c : CTHD
 - Thread:TPF Thread 0B402000 (Stopped)
 - Execution Pt. : 0x0000000397396F48
 - _Z18threadfuncForCase2Pv : qdb0th.o : QDB0
 - cma_thread_base : cththr.c : CTHD
 - CTHD : cthtpf.c : CTHD

Source Code (qdb0th.cpp):

```

265 void *threadfuncForCase2(void *parm)
266 {
267     thread_parm_t *p = (thread_parm_t *) parm;
268
269     if(p->qprt == TRUE)
270         memcpy((char *)&cbpnr()->ebw100, "QPRT", 4);
271     if(p->tprt == TRUE)
272         memcpy((char *)&cbpnr()->ebw096, "TPRT", 4);
  
```

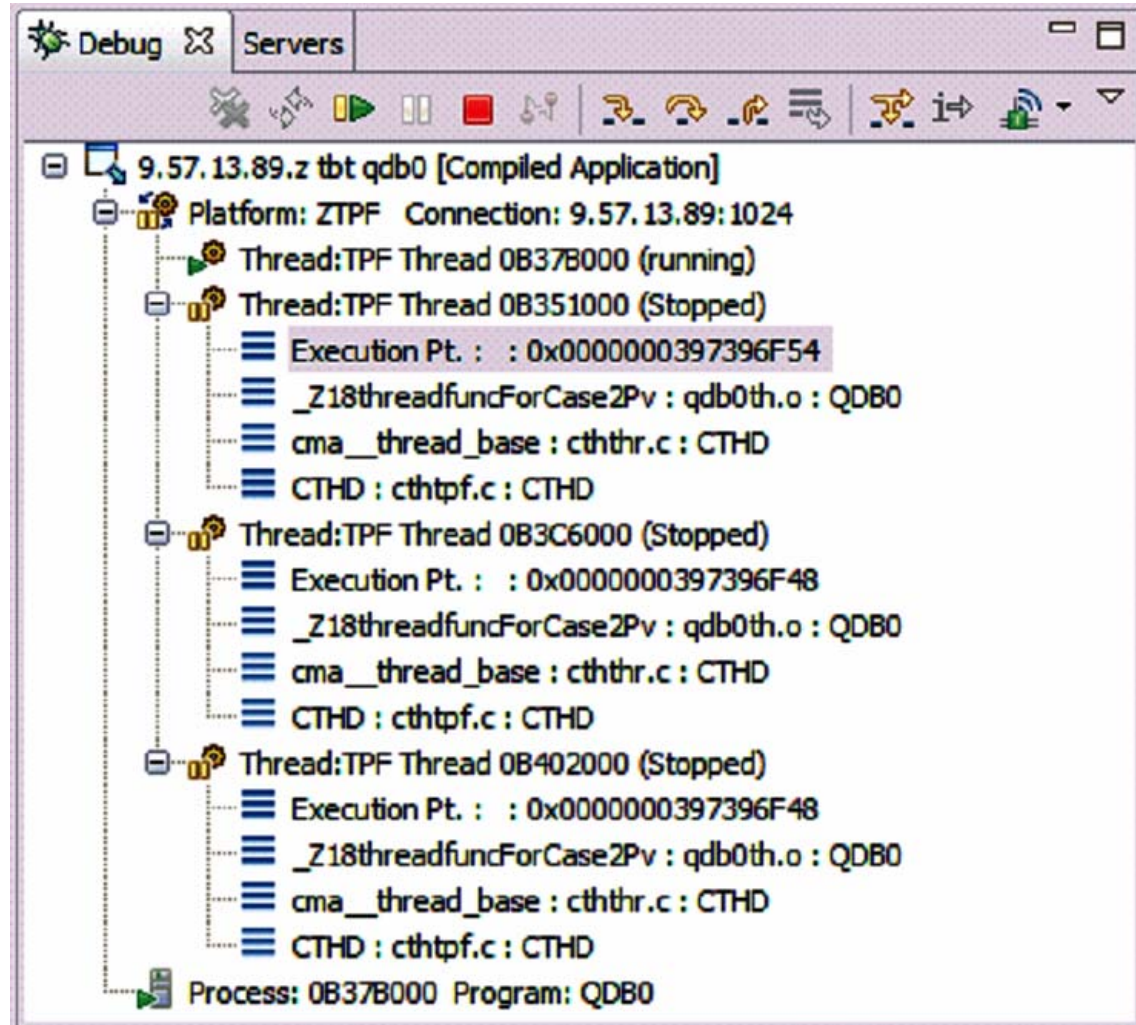
Memory Renderings:

Address	0 - 3	4 - 7
00000000B300000	00000000	00000000
00000000B300010	00000000	00000000
00000000B300020	00000000	00000000
00000000B300030	00000000	00000000
00000000B300040	00000000	00000000
00000000B300050	00000000	00000000
00000000B300060	00000000	00000000
00000000B300070	00000000	00000000
00000000B300080	00000000	00000000

Address	0 - 3	4 - 7
00000000B300000	ffff	ffff
00000000B300010	ffff	ffff
00000000B300020	ffff	ffff
00000000B300030	ffff	ffff
00000000B300040	ffff	ffff
00000000B300050	ffff	ffff
00000000B300060	ffff	ffff
00000000B300070	ffff	ffff
00000000B300080	ffff	ffff

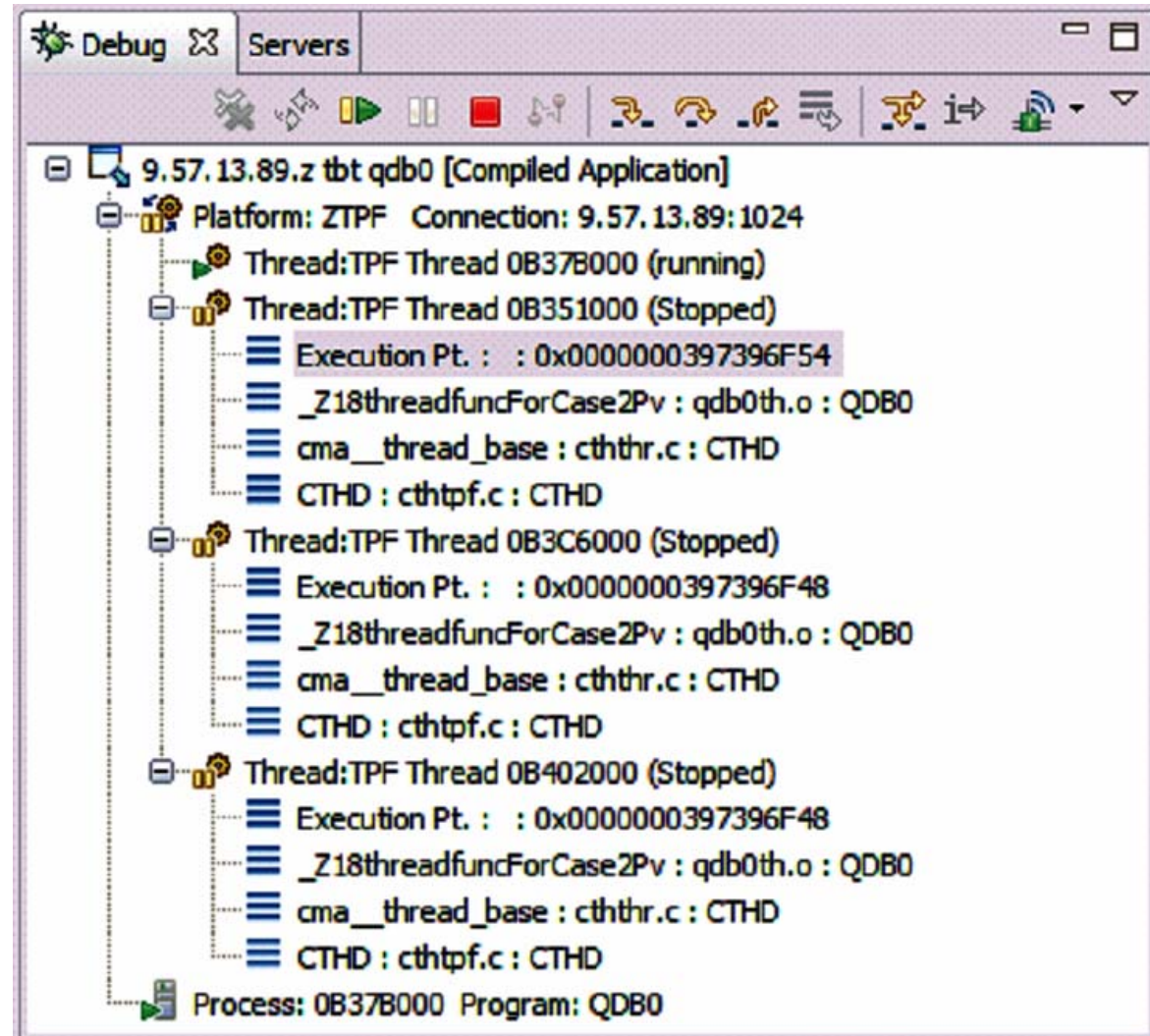
TPF Debugger Thread Support

- z/TPF Debugger takes advantage of a new asynchronous protocol to the TPF Toolkit. This means you control each thread independent of all other threads.



TPF Debugger Thread Support

- The key to using the debugger in the threaded environment is to click on the thread you are interested in, prior to performing an action (Debug Console commands, setting breakpoints, stepping, expression evaluation, register viewing, and etc).



APAR and Toolkit Level Information

- ECB, DECB, and Data Level Views
 - TPF Toolkit V3.2
 - TPF 4.1 PJ31891
 - z/TPF PJ31890
- Toolkit XML Map Enhancements
 - TPF Toolkit V3.2
- TPF Debugger XML Map Generator
 - TPF 4.1 PJ32242
 - z/TPF PJ31440
- SW00SR View
 - TPF Toolkit V3.2.3
 - TPF 4.1 PJ32240
 - z/TPF PJ32221
- #define Support
 - TPF4.1 PJ32070
- z/TPF Debugger Thread Support
 - TPF Toolkit V3.2
 - z/TPF PJ31357

Trademarks

- IBM and TPF Toolkit V3.2 are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.