z/TPF Debugger Update

Josh Wisniewski
TPF Development Lab

Development Tools Subcommittee
March 11, 2014

Disclaimer

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.

TPFUG Washington, D.C.

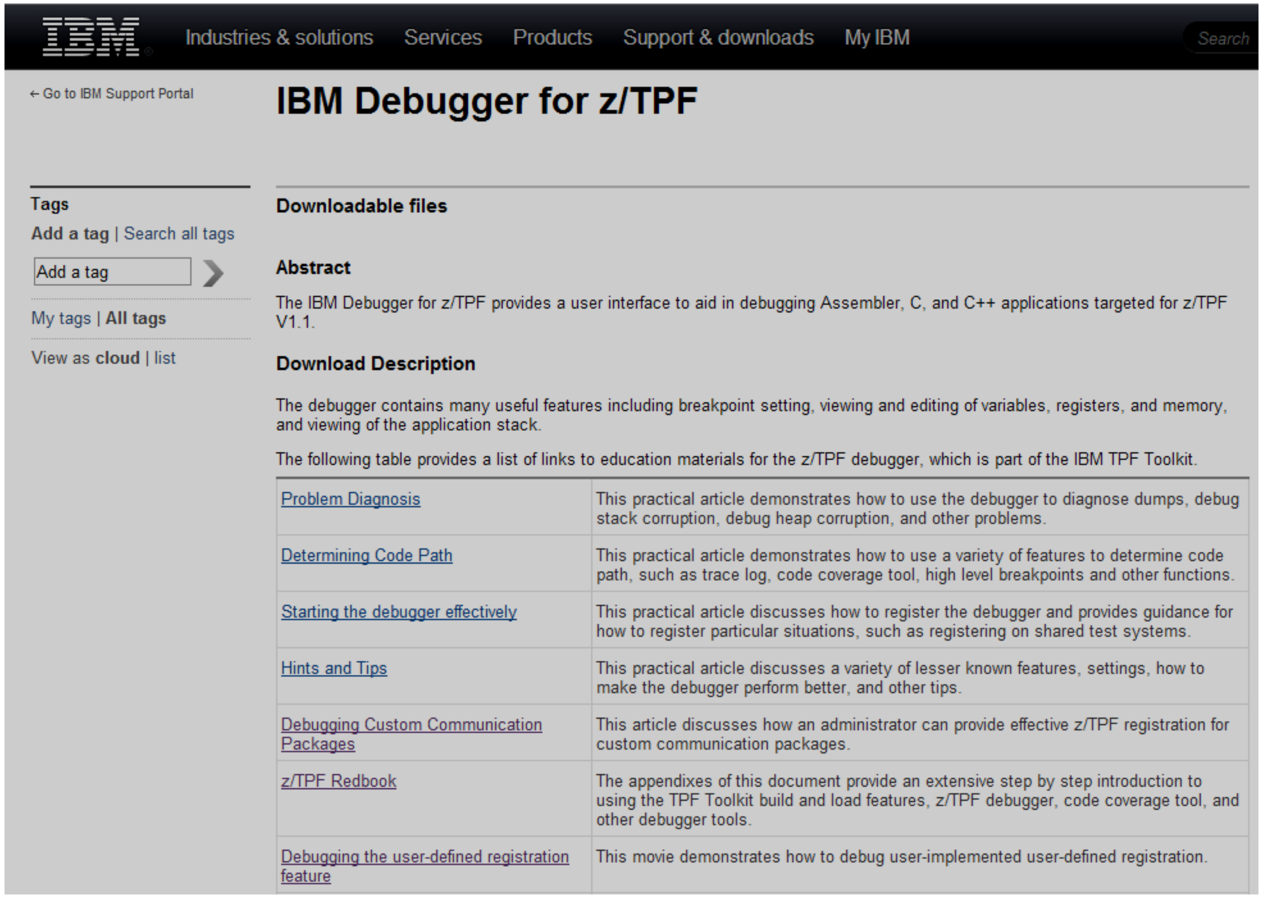
Agenda

- Education materials
  - Previously Announced.
  - Three New Videos.
- Recently delivered function
  - Mixed Source View.
- New function (coming soon...)
  - Highlight Registered Sessions.
  - TPF File View.
  - User Summary View.
  - ECB Trace View.
  - Disconnect Debugger.
  - Set ECB Debuggable State.
  - Active USINGs in the Variables View.
  - Registers in 31 bit Addressing Mode.
  - Button to Trace Created Entries.
  - Default Hex and Char Rendering.
  - Registers View Go To Address Menu.

TPFUG Washington, D.C.

Education Materials: Previously Announced

- http://www.ibm.com/tpf/ Choose Downloads at the bottom. Choose Tools on the left. Then choose z/TPF Debugger.



IBM Debugger for z/TPF

Downloadable files

Abstract

The IBM Debugger for z/TPF provides a user interface to aid in debugging Assembler, C, and C++ applications targeted for z/TPF V1.1.

Download Description

The debugger contains many useful features including breakpoint setting, viewing and editing of variables, registers, and memory, and viewing of the application stack.

The following table provides a list of links to education materials for the z/TPF debugger, which is part of the IBM TPF Toolkit.

Problem Diagnosis	This practical article demonstrates how to use the debugger to diagnose dumps, debug stack corruption, debug heap corruption, and other problems.
Determining Code Path	This practical article demonstrates how to use a variety of features to determine code path, such as trace log, code coverage tool, high level breakpoints and other functions.
Starting the debugger effectively	This practical article discusses how to register the debugger and provides guidance for how to register particular situations, such as registering on shared test systems.
Hints and Tips	This practical article discusses a variety of lesser known features, settings, how to make the debugger perform better, and other tips.
Debugging Custom Communication Packages	This article discusses how an administrator can provide effective z/TPF registration for custom communication packages.
z/TPF Redbook	The appendixes of this document provide an extensive step by step introduction to using the TPF Toolkit build and load features, z/TPF debugger, code coverage tool, and other debugger tools.
Debugging the user-defined registration feature	This movie demonstrates how to debug user-implemented user-defined registration.

Education Materials: Previously Announced

- Practical Articles
  - Problem Diagnosis
  - Determining Code Path
  - Starting the debugger effectively
  - Hints and Tips
  - Debugging Custom Communication Packages
- New User Resources
  - z/TPF Redbook (appendices – web service oriented examples)
  - Debugger for z/TPF Demo Movie (dated but still relevant)

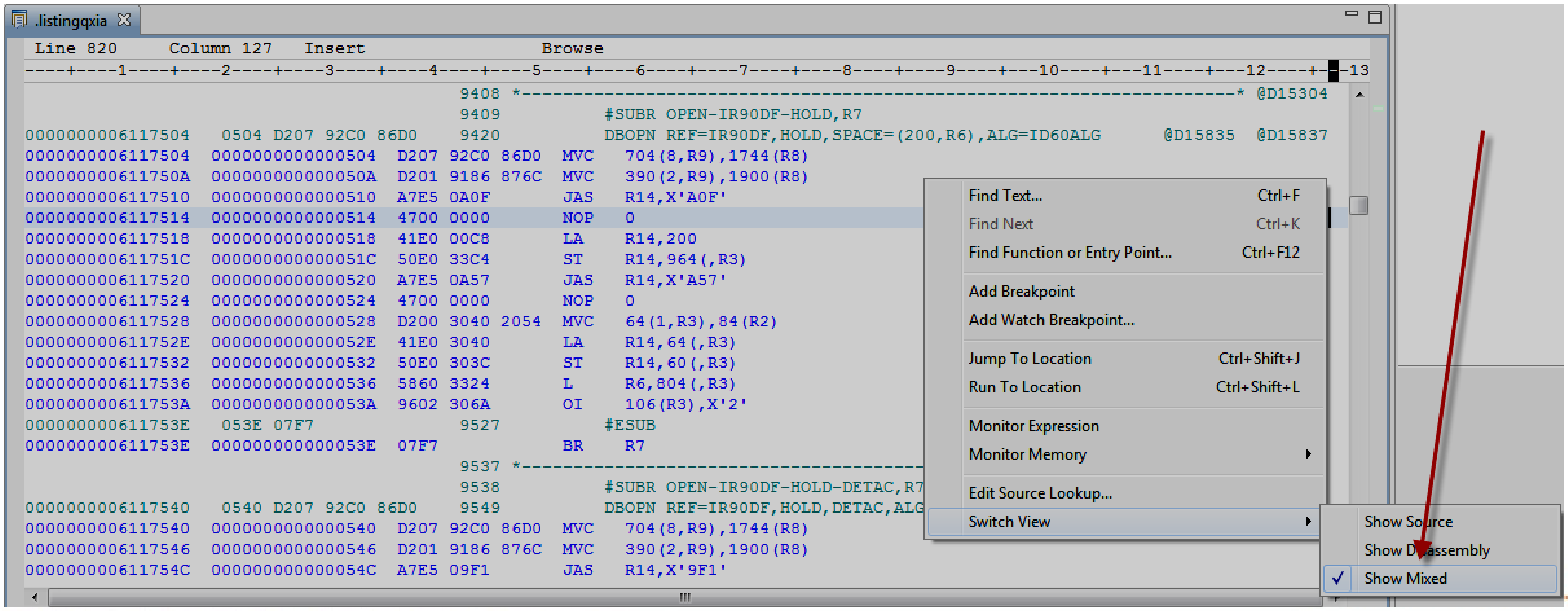
TPFUG Washington, D.C.

Education Materials: Three New Videos

- The z/TPF Debugger Webinar Recording and Presentation.
- The z/TPF Code Coverage Webinar Recording and Presentation
  - Code coverage tool.
  - Debugger – Hex and char memory rendering – very feature rich rendering.
  - Debugger – User defined registration.
- Debugging the user-defined registration feature.
- Links to all three videos can be found on the z/TPF Debugger page previously mentioned.

Recently Delivered Function: Mixed Source View

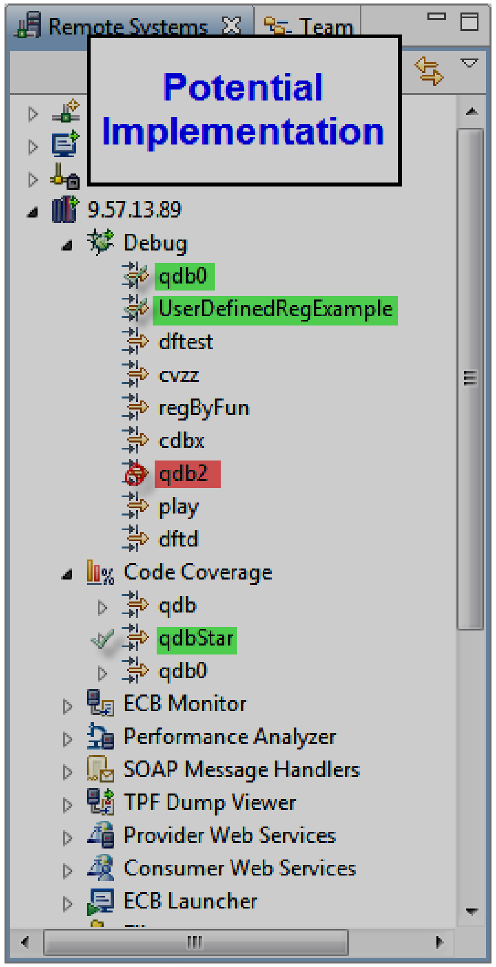
- The mixed source view shows you the assembler instructions that implement a macro with the source lines inserted as comments.
- This feature may be particularly useful debugging SPMs, TPFDF code, and etc.
- Currently, only assembler code is supported.



```
 Line 820      Column 127    Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10----+---11----+---12----+--█-13
                                          9408 *----------------------------------------------------------------------* @D15304
                                          9409          #SUBR OPEN-IR90DF-HOLD,R7
0000000006117504   0504 D207 92C0 86D0    9420          DBOPN REF=IR90DF,HOLD,SPACE=(200,R6),ALG=ID60ALG      @D15835  @D15837
0000000006117504  0000000000000504  D207 92C0 86D0  MVC   704(8,R9),1744(R8)
000000000611750A  000000000000050A  D201 9186 876C  MVC   390(2,R9),1900(R8)
0000000006117510  0000000000000510  A7E5 0A0F       JAS   R14,X'A0F'
0000000006117514  0000000000000514  4700 0000       NOP   0
0000000006117518  0000000000000518  41E0 00C8       LA    R14,200
000000000611751C  000000000000051C  50E0 33C4       ST    R14,964(,R3)
0000000006117520  0000000000000520  A7E5 0A57       JAS   R14,X'A57'
0000000006117524  0000000000000524  4700 0000       NOP   0
0000000006117528  0000000000000528  D200 3040 2054  MVC   64(1,R3),84(R2)
000000000611752E  000000000000052E  41E0 3040       LA    R14,64(,R3)
0000000006117532  0000000000000532  50E0 303C       ST    R14,60(,R3)
0000000006117536  0000000000000536  5860 3324       L     R6,804(,R3)
000000000611753A  000000000000053A  9602 306A       OI    106(R3),X'2'
000000000611753E   053E 07F7                  9527          #ESUB
000000000611753E  000000000000053E  07F7        BR    R7
                                          9537 *--------------------------------------------
                                          9538          #SUBR OPEN-IR90DF-HOLD-DETAC,R7
0000000006117540   0540 D207 92C0 86D0    9549          DBOPN REF=IR90DF,HOLD,DETAC,ALG
0000000006117540  0000000000000540  D207 92C0 86D0  MVC   704(8,R9),1744(R8)
0000000006117546  0000000000000546  D201 9186 876C  MVC   390(2,R9),1900(R8)
000000000611754C  000000000000054C  A7E5 09F1       JAS   R14,X'9F1'
```

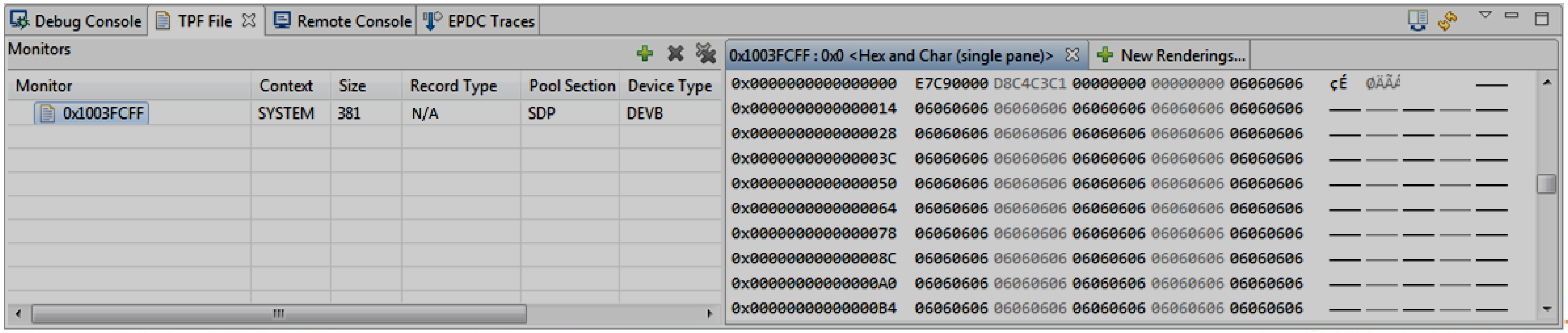
New Function: Highlight Registered Sessions

- Upon registering the debugger, code coverage tool or performance analyzer, a heartbeat will be started by the TPF Toolkit. A request will be periodically sent from the TPF Toolkit to the TPF system to verify the registration entries still exist. The registered entries will be marked up to indicate that they are registered. If an entry was registered but is found to no longer be registered, it will be marked up differently.
- This heartbeat mechanism will also be used to update the registered workstation IP address to help ensure sessions will continue to function properly if the workstation's IP address changes.
- System administrators will be able to disable the heartbeat mechanism from within the TPF Toolkit or on TPF using the ZDBUG ACCESS command.
- The timestamp when the register occurred will also be added to the registration entry displays (ZDBUG DISP and ZDDBG DISP).



New Function: TPF File View

- Clicking green plus allows you to monitor a file by file address, record type and ordinal, or expression.
- Left pane provides details about a file. Right pane shows content in a memory like view.
- Files can be viewed from system context (ZDFIL equivalent) or from the ECB context (commit scopes are honored). The ECB context shows the file content that would be retrieved if a FINDC was performed by the application at that point in the code. The ECB context does not show the contents of a file read into memory.
- Registers, data level, SW00SR and other views allow you to right click and monitor a file.

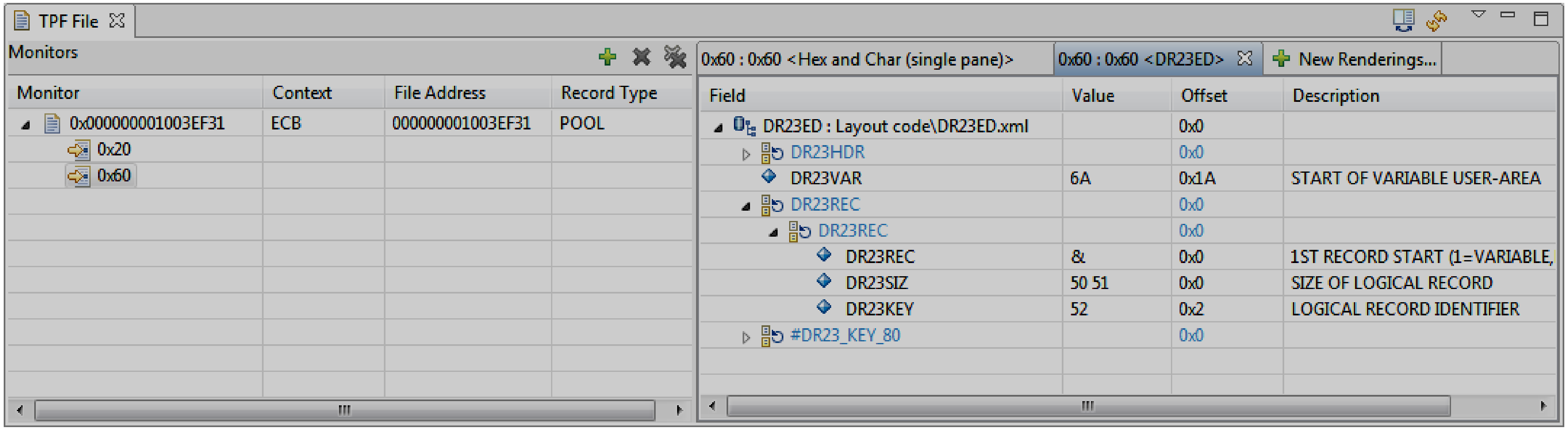


Monitor	Context	Size	Record Type	Pool Section	Device Type
0x1003FCFF	SYSTEM	381	N/A	SDP	DEVB

```
0x0000000000000000  E7C90000 D8C4C3C1 00000000 00000000 06060606   çÉ  ØÄÃÁ
0x0000000000000014  06060606 06060606 06060606 06060606 06060606
0x0000000000000028  06060606 06060606 06060606 06060606 06060606
0x000000000000003C  06060606 06060606 06060606 06060606 06060606
0x0000000000000050  06060606 06060606 06060606 06060606 06060606
0x0000000000000064  06060606 06060606 06060606 06060606 06060606
0x0000000000000078  06060606 06060606 06060606 06060606 06060606
0x000000000000008C  06060606 06060606 06060606 06060606 06060606
0x00000000000000A0  06060606 06060606 06060606 06060606 06060606
0x00000000000000B4  06060606 06060606 06060606 06060606 06060606
```

New Function: TPF File View

- Since the TPF File view is built upon the memory view base, all memory view renderings can be applied to files including XML maps.
- Right clicking on a monitor allows you to add an offset. Multiple offsets can be added. And the data renderings can be applied separately to each.

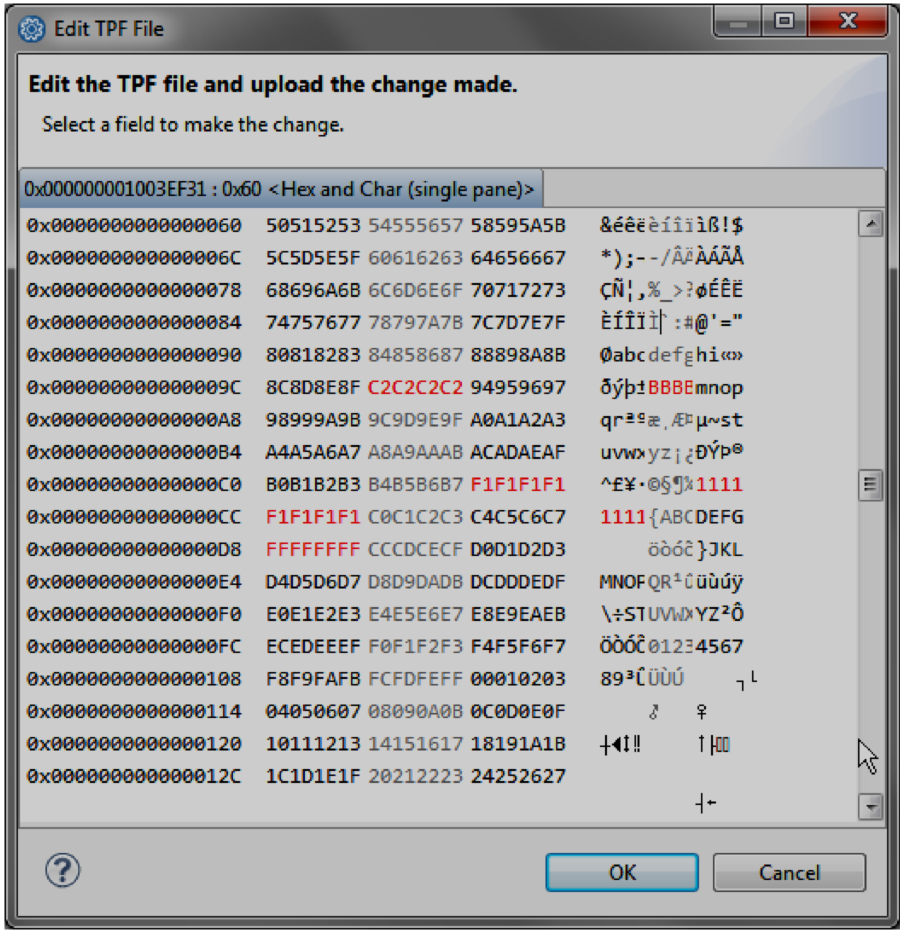


Monitor	Context	File Address	Record Type
0x000000001003EF31	ECB	000000001003EF31	POOL
0x20			
0x60			

Field	Value	Offset	Description
DR23ED : Layout code\DR23ED.xml		0x0	
DR23HDR		0x0	
DR23VAR	6A	0x1A	START OF VARIABLE USER-AREA
DR23REC		0x0	
DR23REC		0x0	
DR23REC	&	0x0	1ST RECORD START (1=VARIABLE,
DR23SIZ	50 51	0x0	SIZE OF LOGICAL RECORD
DR23KEY	52	0x2	LOGICAL RECORD IDENTIFIER
#DR23_KEY_80		0x0	

New Function: TPF File View

- Right clicking on the entry in the monitor pane provides the option to edit the content of the file. All changes to the file are made in the pop up window. Choosing ok writes the entire file out to disk. This edit feature differs from most other views in that changes are not made in line but is intended to help ensure the integrity of the file contents.
- The ZDBUG ACCESS command allows you to prohibit viewing and/or the editing of files on your system.



Edit the TPF file and upload the change made.

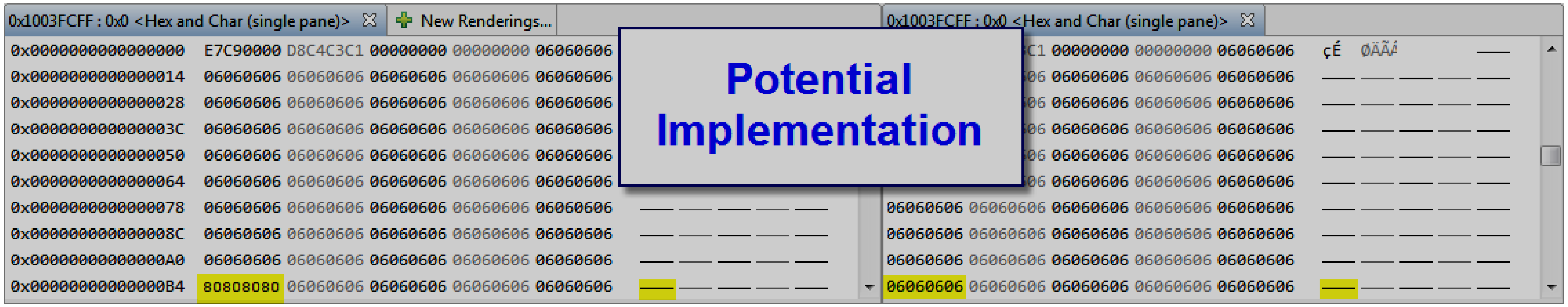
Select a field to make the change.

0x000000001003EF31 : 0x60 <Hex and Char (single pane)>

```
0x0000000000000060   50515253 54555657 58595A5B   &éêëèíîïìß!$
0x000000000000006C   5C5D5E5F 60616263 64656667   *);--/ÂÄÀÁÃÅ
0x0000000000000078   68696A6B 6C6D6E6F 70717273   ÇÑ¦,%_>?øÉÊË
0x0000000000000084   74757677 78797A7B 7C7D7E7F   ÈÍÎÏÌ`:#@'="
0x0000000000000090   80818283 84858687 88898A8B   Øabcdefghi«»
0x000000000000009C   8C8D8E8F C2C2C2C2 94959697   ðýþ±BBBBmnop
0x00000000000000A8   98999A9B 9C9D9E9F A0A1A2A3   qrªºæ¸Æ¤µ~st
0x00000000000000B4   A4A5A6A7 A8A9AAAB ACADAEAF   uvwxyz¡¿ÐÝÞ®
0x00000000000000C0   B0B1B2B3 B4B5B6B7 F1F1F1F1   ^£¥·©§¶¼1111
0x00000000000000CC   F1F1F1F1 C0C1C2C3 C4C5C6C7   1111{ABCDEFG
0x00000000000000D8   FFFFFFFF CCCDCECF D0D1D2D3       öòóõ}JKL
0x00000000000000E4   D4D5D6D7 D8D9DADB DCDDDEDF   MNOPQR¹ûüùúÿ
0x00000000000000F0   E0E1E2E3 E4E5E6E7 E8E9EAEB   \÷STUVWXYZ²Ô
0x00000000000000FC   ECEDEEEF F0F1F2F3 F4F5F6F7   ÖÒÓÕ01234567
0x0000000000000108   F8F9FAFB FCFDFEFF 00010203   89³ÛÜÙÚ
0x0000000000000114   04050607 08090A0B 0C0D0E0F
0x0000000000000120   10111213 14151617 18191A1B
0x000000000000012C   1C1D1E1F 20212223 24252627
```

New Function: TPF File View

- The file view will also provide the ability to do data comparisons
  - Memory contents in a data level against the contents of the file in system or ECB context.
  - File contents in the ECB context against the file contents of the system context.



0x1003FCFF : 0x0 <Hex and Char (single pane)> New Renderings...

```
0x0000000000000000  E7C90000 D8C4C3C1 00000000 00000000 06060606
0x0000000000000014  06060606 06060606 06060606 06060606 06060606
0x0000000000000028  06060606 06060606 06060606 06060606 06060606
0x000000000000003C  06060606 06060606 06060606 06060606 06060606
0x0000000000000050  06060606 06060606 06060606 06060606 06060606
0x0000000000000064  06060606 06060606 06060606 06060606 06060606
0x0000000000000078  06060606 06060606 06060606 06060606 06060606
0x000000000000008C  06060606 06060606 06060606 06060606 06060606
0x00000000000000A0  06060606 06060606 06060606 06060606 06060606
0x00000000000000B4  80808080 06060606 06060606 06060606 06060606
```

0x1003FCFF : 0x0 <Hex and Char (single pane)>

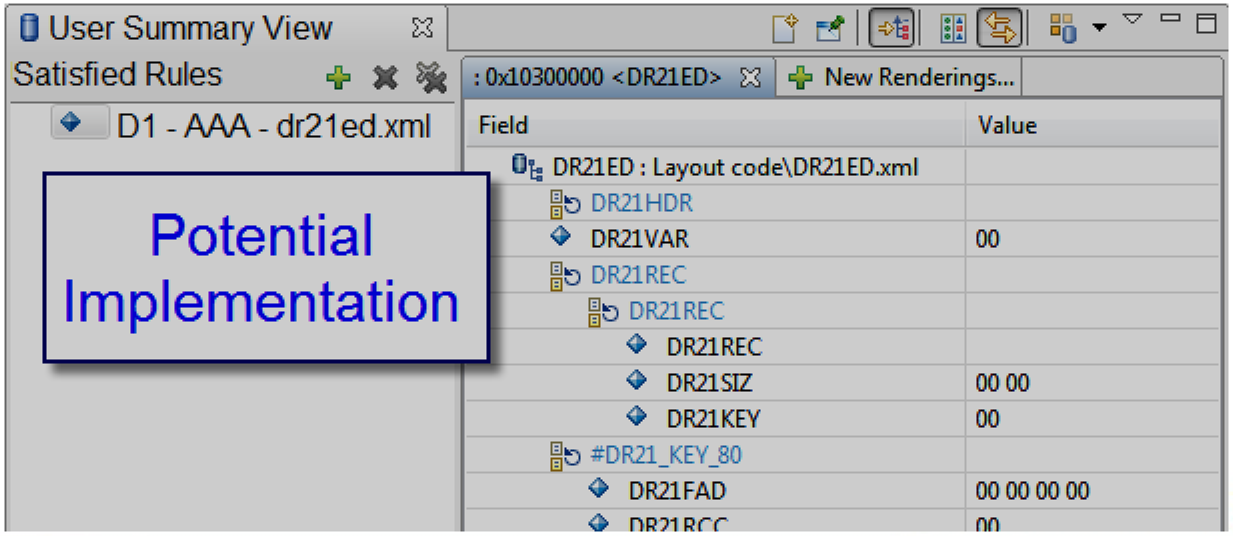
```
     C1 00000000 00000000 06060606   çÉ  ØÄÃÁ
     06 06060606 06060606 06060606
     06 06060606 06060606 06060606
     06 06060606 06060606 06060606
     06 06060606 06060606 06060606
     06 06060606 06060606 06060606
06060606 06060606 06060606 06060606 06060606
06060606 06060606 06060606 06060606 06060606
06060606 06060606 06060606 06060606 06060606
06060606 06060606 06060606 06060606 06060606
```

Potential Implementation

TPFUG Washington, D.C.

New Function: User Summary View

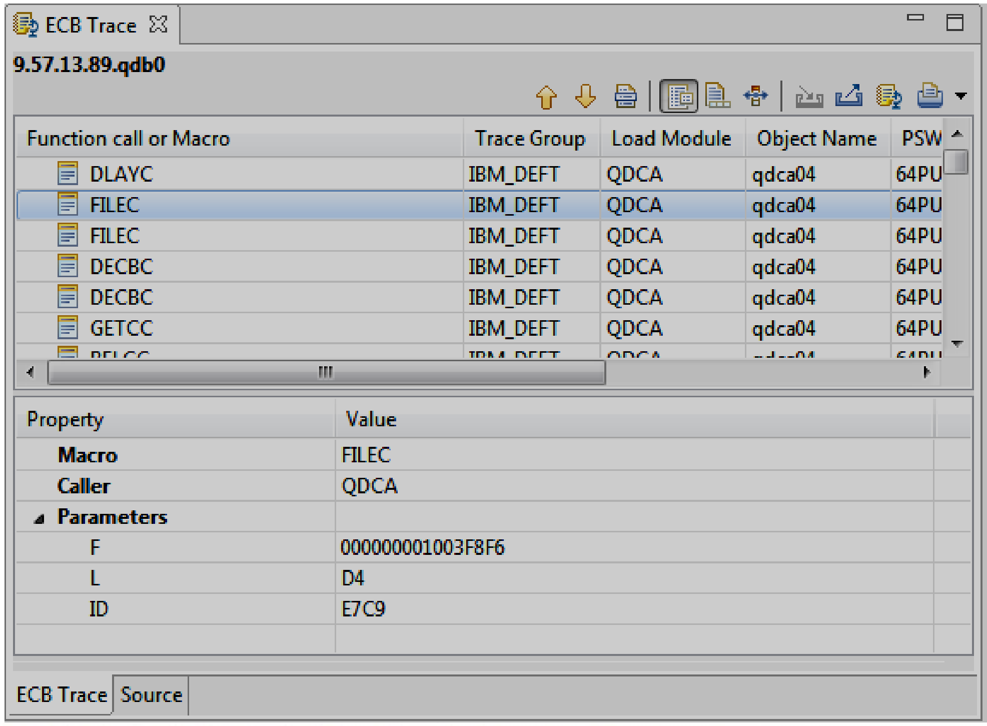
- This feature provides automatic memory views based upon a mapping provided by the user. For example: record id XXXX in a data level is mapped by XML map XX00XX.xml, when program ABCD is selected on the stack EBW060 is mapped by XML map wxyz.xml, and so on.
- The first pane will show a list of the rules that have been satisfied. When the user clicks on an entry, the formatted contents will be shown.
- Rules can be dynamically added to the list.
- TPF Toolkit administrators can deploy default rules.



Field	Value
DR21ED : Layout code\DR21ED.xml	
DR21HDR	
DR21VAR	00
DR21REC	
DR21REC	
DR21REC	
DR21SIZ	00 00
DR21KEY	00
#DR21_KEY_80	
DR21FAD	00 00 00 00
DR21RCC	00

New Function: ECB Trace View

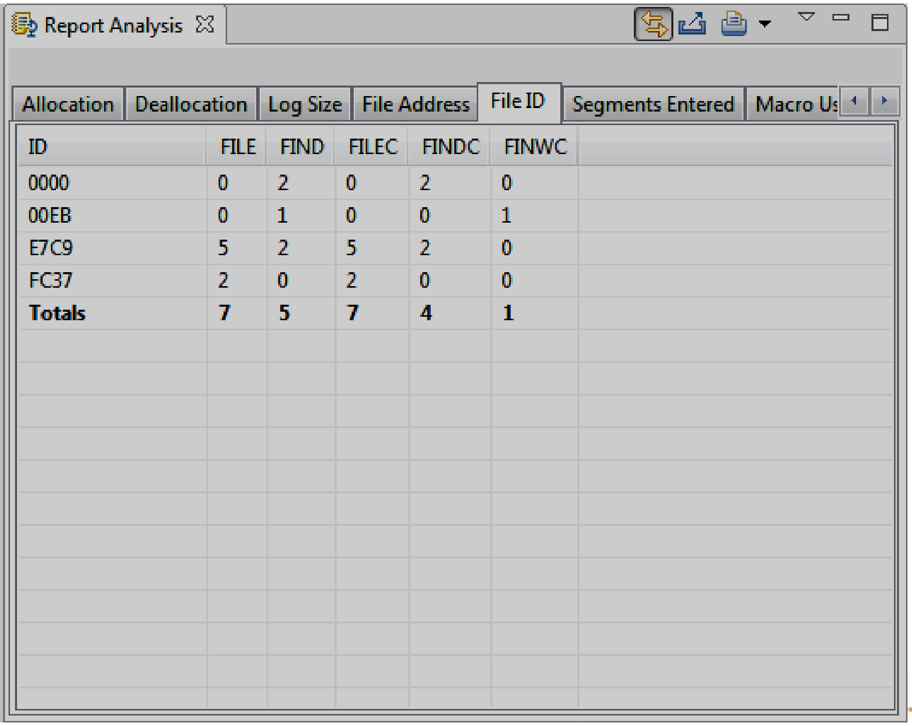
- This view shows the content of the ECB trace in a format that is similar to the trace log editor. The indentation, analysis and etc makes the ECB trace easier to consume than the textual versions previously available through the debug console.
- This view can be used in the debugger and dump viewer.



9.57.13.89.qdb0

Function call or Macro	Trace Group	Load Module	Object Name	PSW
DLAYC	IBM_DEFT	QDCA	qdca04	64PU
FILEC	IBM_DEFT	QDCA	qdca04	64PU
FILEC	IBM_DEFT	QDCA	qdca04	64PU
DECBC	IBM_DEFT	QDCA	qdca04	64PU
DECBC	IBM_DEFT	QDCA	qdca04	64PU
GETCC	IBM_DEFT	QDCA	qdca04	64PU

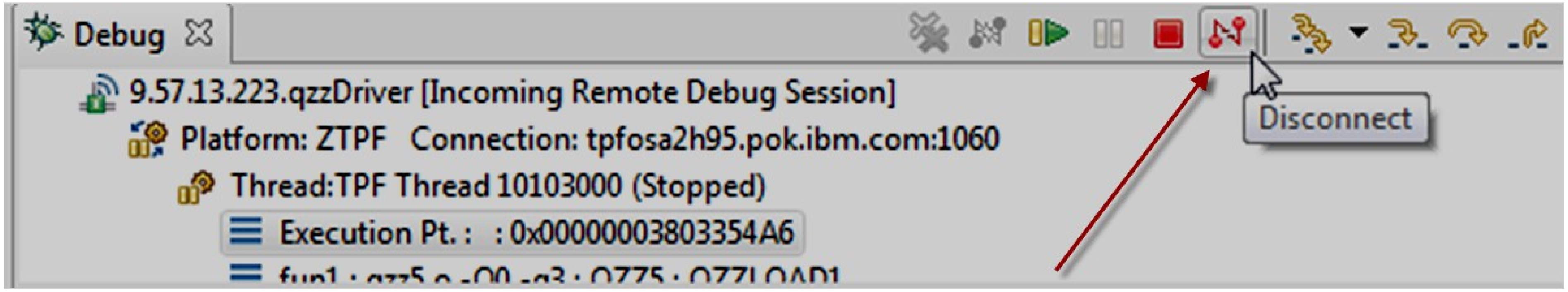
Property	Value
Macro	FILEC
Caller	QDCA
Parameters	
F	000000001003F8F6
L	D4
ID	E7C9



ID	FILE	FIND	FILEC	FINDC	FINWC
0000	0	2	0	2	0
00EB	0	1	0	0	1
E7C9	5	2	5	2	0
FC37	2	0	2	0	0
Totals	**7**	**5**	**7**	**4**	**1**

New Function: Disconnect Debugger

- A new “Disconnect” button will be located in the Debug view. Clicking this button will cause the debugger to set the application running without any further debugger intervention. The user will not need to disable breakpoints or etc. The debugger will exit. The application ECB will not be able to be debugged again.



9.57.13.223.qzzDriver [Incoming Remote Debug Session]
Platform: ZTPF   Connection: tpfosa2h95.pok.ibm.com:1060
Thread:TPF Thread 10103000 (Stopped)
Execution Pt. : : 0x00000003803354A6

TPFUG Washington, D.C.

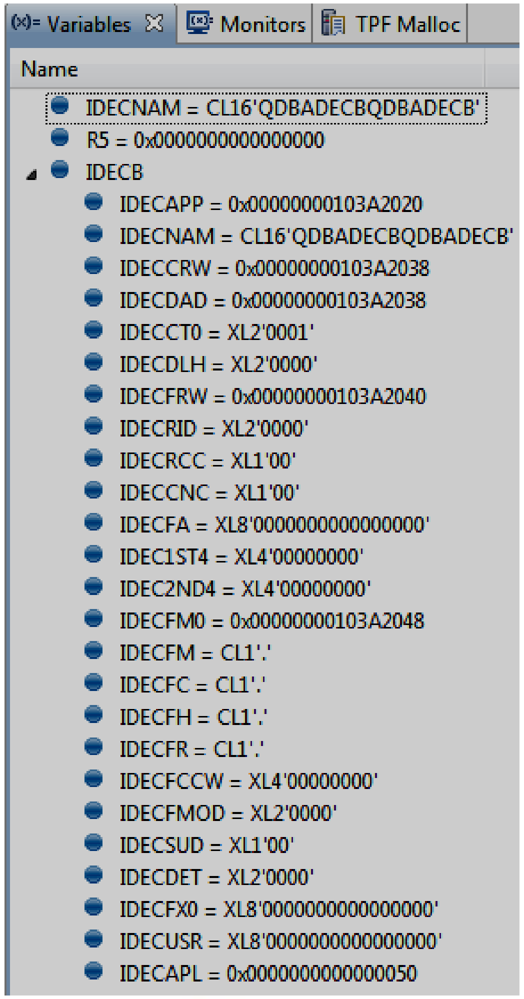
New Function: Set ECB Debuggable State

- A new z/TPF API is provided to prevent an ECB from being debugged: tpf_setECBDebuggableState. If TPF_ECB_IS_NOT_DEBUGGABLE or 1 is passed as the parameter, the debugger will not start for that ECB. Further, if a debugger session is already active for the ECB, the debugger will force the ECB to continue executing until the ECB becomes debuggable again (breakpoints, ECB create events, and etc are ignored).
- One circumstance where this may be useful is to set the ECB as not debuggable before locking a resource and then setting the ECB as debuggable after the lock is released.
- The tpf_setECBDebuggableState API must be coded by application or embedded in a macro called by the application. For example, in the lock illustration above, tpf_setECBDebuggableState could be coded in the lock and unlock macros.

TPFUG Washington, D.C.

New Function: Active USINGs in the Variables View

- When debugging assembler code, the active USINGs will be shown in the variables view as the name of the DSECT. From this location, the DSECT can be expanded to see the values.

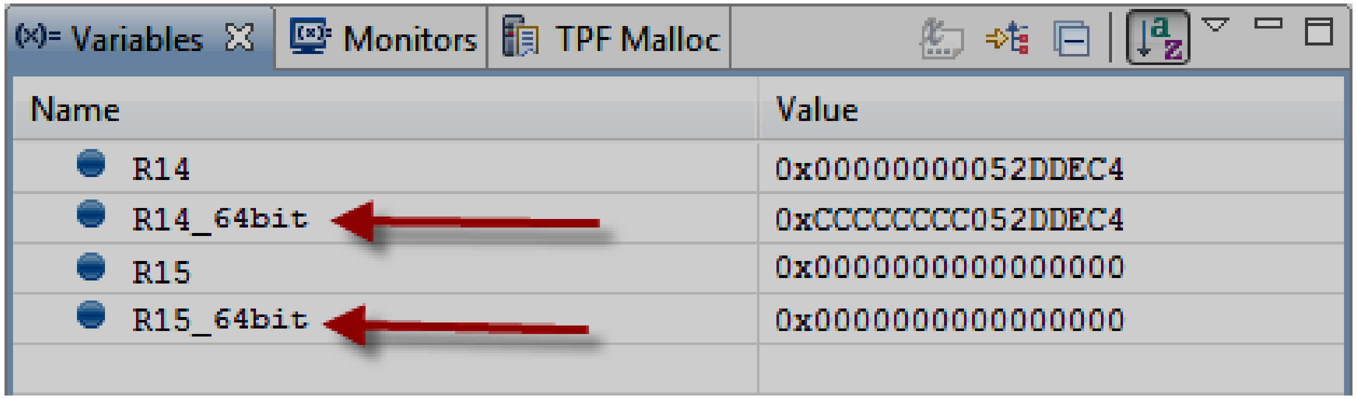


```
IDECNAM = CL16'QDBADECBQDBADECB'
R5 = 0x0000000000000000
IDECB
    IDECAPP = 0x000000000103A2020
    IDECNAM = CL16'QDBADECBQDBADECB'
    IDECCRW = 0x000000000103A2038
    IDECDAD = 0x000000000103A2038
    IDECCT0 = XL2'0001'
    IDECDLH = XL2'0000'
    IDECFRW = 0x000000000103A2040
    IDECRID = XL2'0000'
    IDECRCC = XL1'00'
    IDECCNC = XL1'00'
    IDECFA = XL8'0000000000000000'
    IDEC1ST4 = XL4'00000000'
    IDEC2ND4 = XL4'00000000'
    IDECFM0 = 0x000000000103A2048
    IDECFM = CL1'.'
    IDECFC = CL1'.'
    IDECFH = CL1'.'
    IDECFR = CL1'.'
    IDECFCCW = XL4'00000000'
    IDECFMOD = XL2'0000'
    IDECSUD = XL1'00'
    IDECDET = XL2'0000'
    IDECFX0 = XL8'0000000000000000'
    IDECUSR = XL8'0000000000000000'
    IDECAPL = 0x0000000000000050
```

TPFUG Washington, D.C.

New Function: Registers in 31 bit Addressing Mode

- When a 31 bit application is being debugged, the register values shown in the variables view are purified to only show 31 bits. A second register value will be added to the variables view to show the full 64 bit value in the register.



Name	Value
R14	0x00000000052DDEC4
R14_64bit	0xCCCCCCCC052DDEC4
R15	0x0000000000000000
R15_64bit	0x0000000000000000

TPFUG Washington, D.C.

New Function: Button to Trace Created Entries

- Currently, the trace created entries checkbox in the debugger registration entry must be selected at the time you register the debugger in order to debug ECBs that will be created by the debugged application.
- This new functionality will provide a button that can be selected at any time to dynamically turn on or off the trace created entries feature.

New Function: Default Hex and Char Rendering

- This new functionality will allow users to set the “hex and char” memory rendering as the default.

TPFUG Washington, D.C.

New Function: Registers View Go To Address Menu

- This new functionality will allow users choose a “go to address” menu action from the registers view. “go to address” is different from “monitor memory” in that “go to address” adds the hexadecimal address to the memory view while “monitor memory” adds the “register” expression (ie R14) to the memory view such that whenever the value in the register changes, the new location is shown.

TPFUG Washington, D.C.

z/TPF Debugger Deliverable Details: Available

Description	**z/TPF APAR**	**z/TPF PUT Level**	**TPF Toolkit Level**	**TPFUG Requirement**
Mixed Source View	PJ41281	PUT10	N/A	V09113F

z/TPF Debugger Deliverable Details: Coming Soon...

Description	z/TPF APAR	z/TPF PUT Level	TPF Toolkit Level	TPFUG Requirement
Highlight Registered Sessions (show registration timestamp)	PJ41688	PUT11	V.next	V12129 RFE 44588
TPF File View (display) (modify) (monitor from views) (compare data level contents)	PJ41688	PUT11	V.next	V08024F V08033F V08040F V08042S
User Summary View	N/A	N/A	V.next	V09108S
ECB Trace View	N/A	N/A	V.next	

TPFUG Washington, D.C.

z/TPF Debugger Deliverable Details: Coming Soon...

Description	**z/TPF APAR**	**z/TPF PUT Level**	**TPF Toolkit Level**	**TPFUG Requirement**
Disconnect Debugger	PJ41688	PUT11	N/A	Customer Request
Set ECB Debuggable State	PJ41820	PUT11	N/A	RFE 38517
Active USINGs in the Variables View	TBD	PUT11	N/A	Customer Request
Registers in 31 bit Addressing Mode	TBD	PUT11	N/A	Customer Request
Button to Trace Created Entries	N/A	N/A	v.Next	Customer Request

TPFUG Washington, D.C.

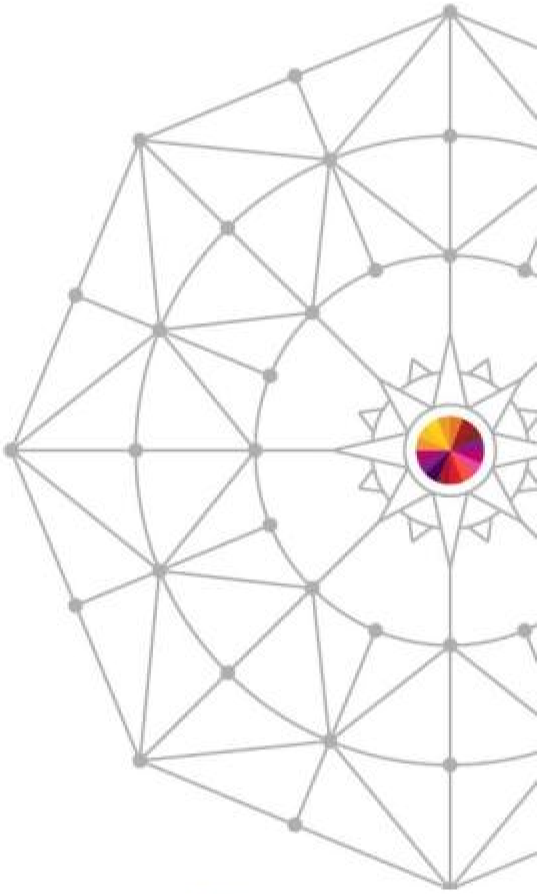
z/TPF Debugger Deliverable Details: Coming Soon...

Description	z/TPF APAR	z/TPF PUT Level	TPF Toolkit Level	TPFUG Requirement
Button to Trace Created Entries	N/A	N/A	v.Next	Customer Request
Default Hex and Char Rendering	N/A	N/A	v.Next	Customer Request
Registers View Go To Address Menu	N/A	N/A	v.Next	V12128

TPFUG Washington, D.C.

Trademarks

Notes



TPFUG Washington, D.C.