



TPF Toolkit V3.2

TPF Users Group Spring 2008 TPF Debugger

Name: Isa Torres
Venue: Hot Topics

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.

© 2008 IBM Corporation

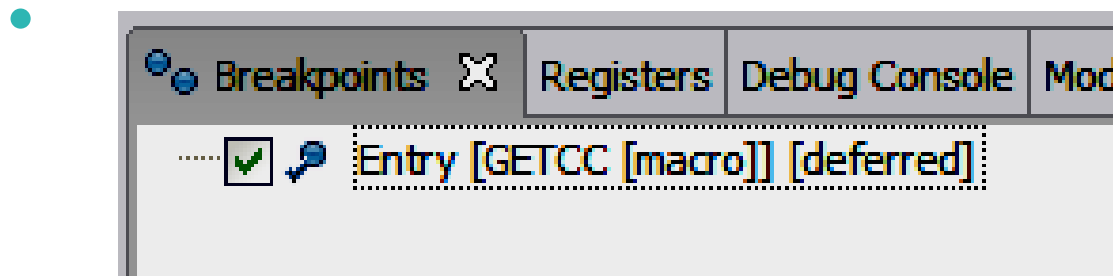
Agenda

- **Existing Features**
 - Breakpoints
 - SW00SR, ECB, DECB, Data Level Views
- **Demos**
- **Open discussion**

Existing Features

- **Macro breakpoints**

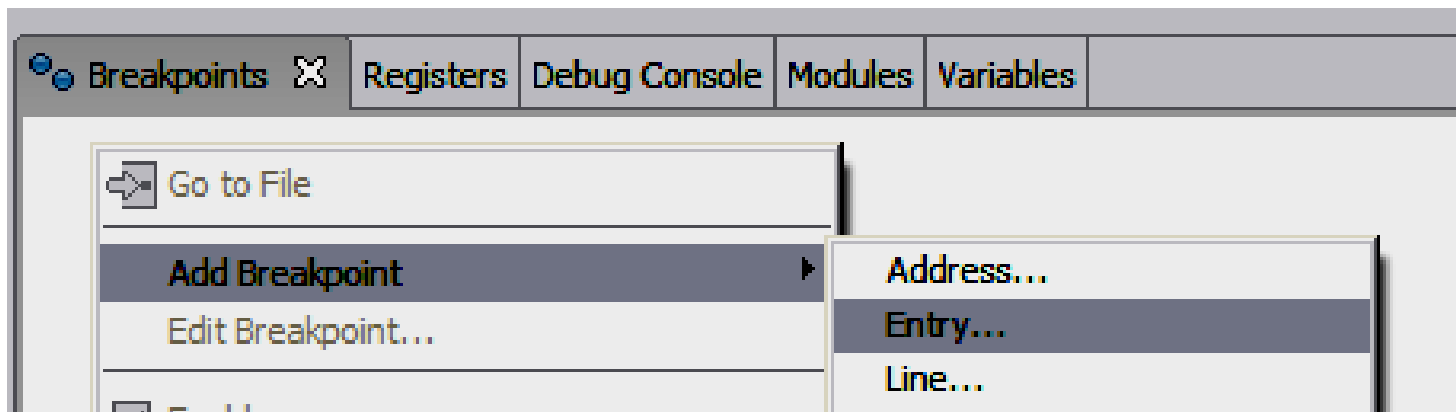
- Debugger stops at macro call



(GETCC Macro)

- Setting a macro breakpoint

1. Right-click on breakpoints view and select “Entry...”



Existing Features

- **Macro breakpoints (Cont.)**

- Setting a macro breakpoint
 2. Select the breakpoint to be deferred
 3. Enter the module name where the breakpoint would apply, "*" for all modules
 4. Enter macro name in Function or Entry Point field
 5. Click Finish

Add an Entry Breakpoint

Required information
Sets a breakpoint to stop execution at a specific function or entry point

Defer breakpoint until executable is loaded
 Show items with debug info only

Executable
*

Object

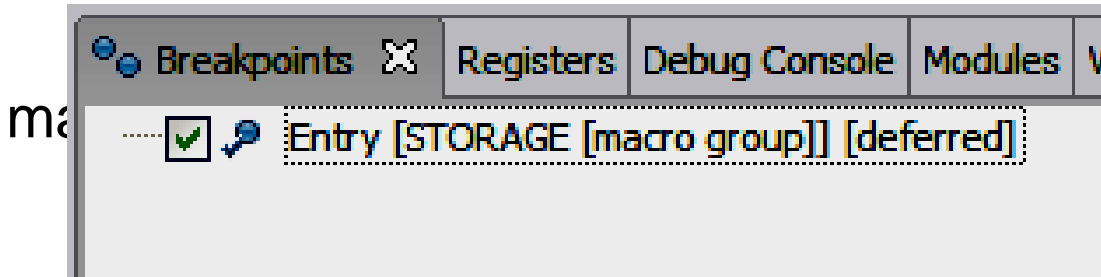
Function or Entry point
GETCC

Case sensitive

< Back Next > Finish Cancel

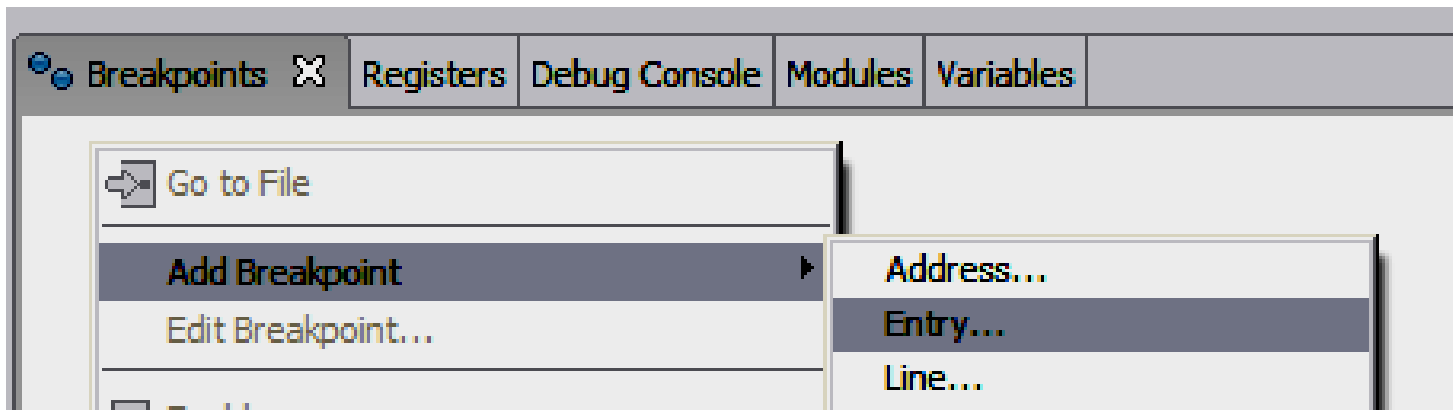
Existing Features

- **Macro group breakpoints**
 - Debugger stops at macro group call



(STORAGE
group)

- Setting a macro group breakpoint
 1. Right-click on breakpoints view and select “Entry...”

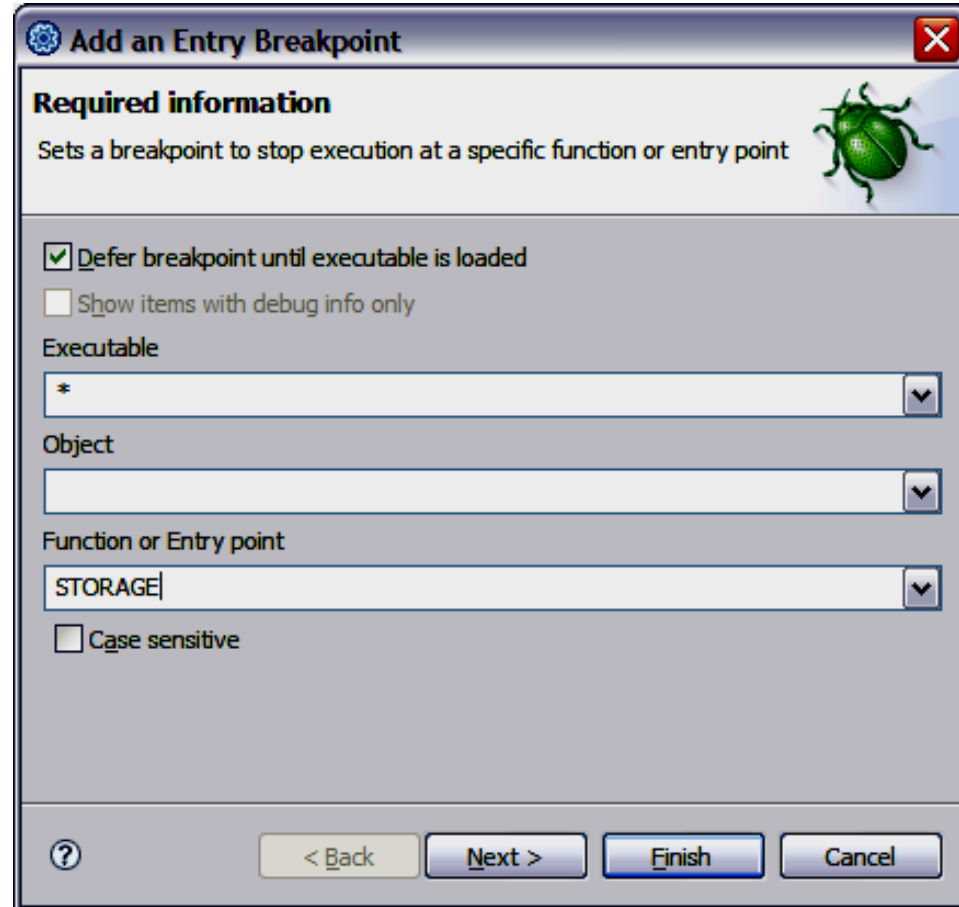


Existing Features

• Macro group breakpoints (Cont.)

• Setting a macro group breakpoint

2. Select the breakpoint to be deferred
3. Enter the module name where the breakpoint would apply, "*" for all
4. Enter macro group name in Function or Entry Point field
5. Click Finish



Add an Entry Breakpoint

Required information

Sets a breakpoint to stop execution at a specific function or entry point

Defer breakpoint until executable is loaded

Show items with debug info only

Executable

*

Object

Function or Entry point

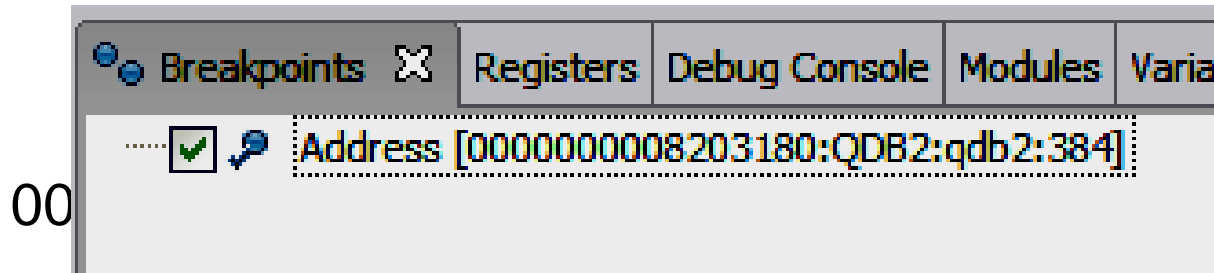
STORAGE

Case sensitive

< Back Next > Finish Cancel

Existing Features

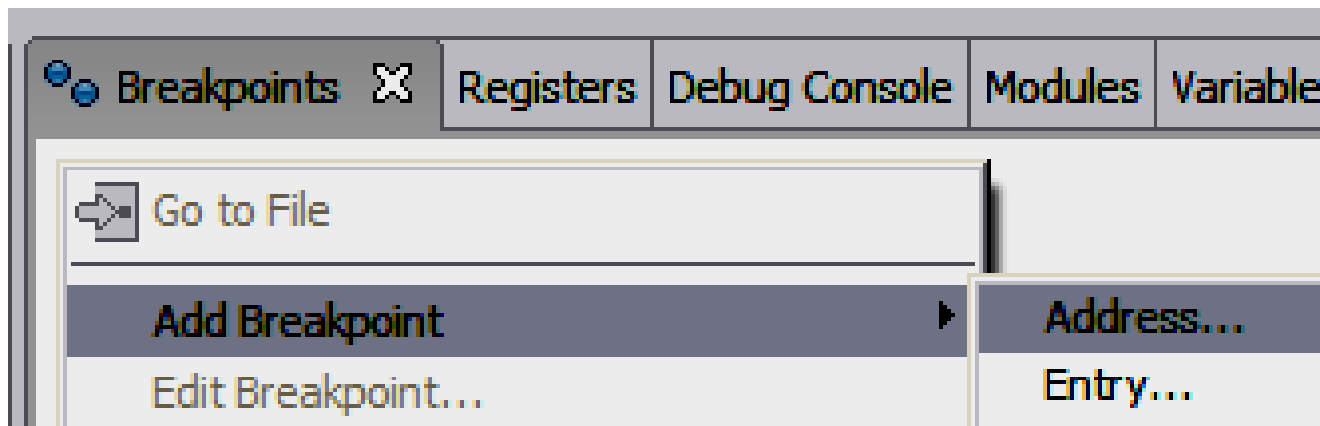
- **Address breakpoints**
 - Debugger stops at address



(Address

- Setting an address breakpoint

1.



Existing Features

- **Address breakpoints (Cont.)**

- Setting an address breakpoint
 2. Enter the address where the breakpoint is to be set
 3. Click finish

Add an Address Breakpoint

Required information

Sets a breakpoint to stop execution at a specific address

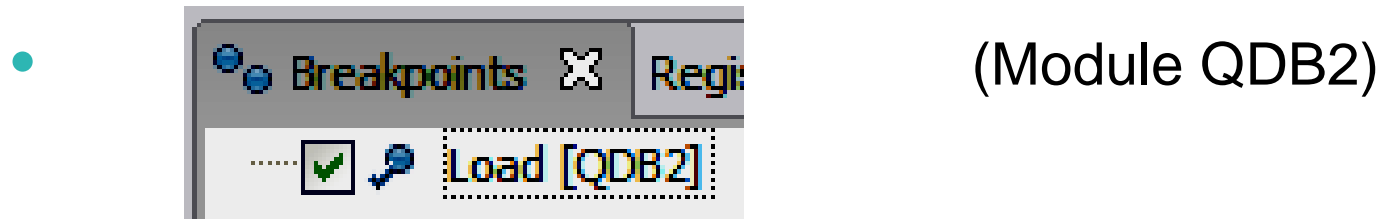
Address or expression: 0000000008203180

? < Back Next > Finish Cancel

Existing Features

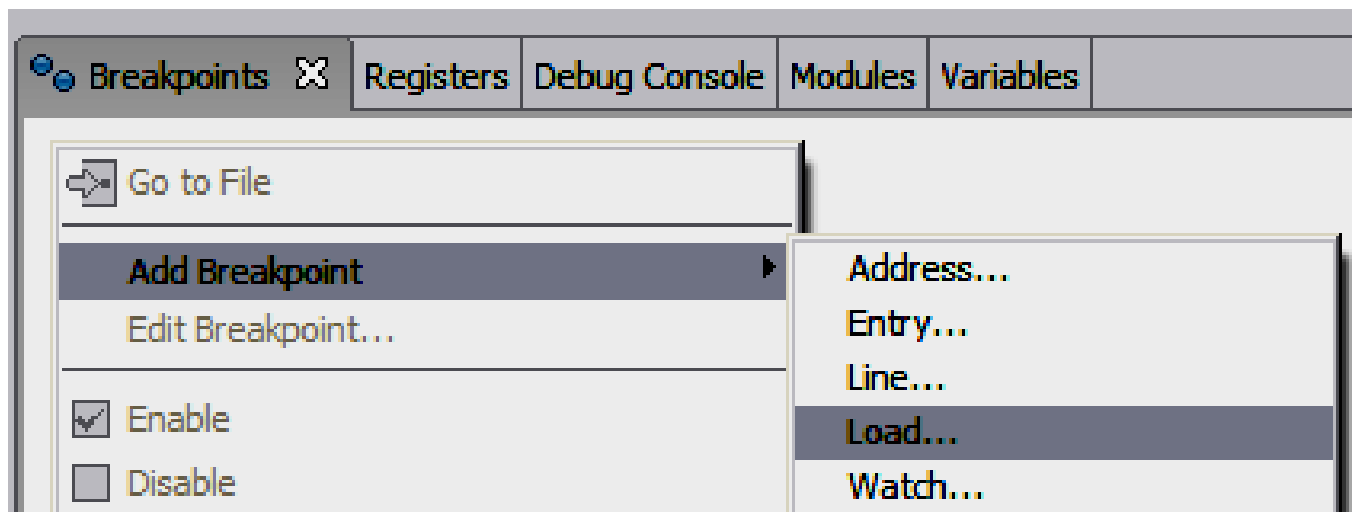
- **Load breakpoints**

- Debugger stops when entering module



- Setting a load breakpoint

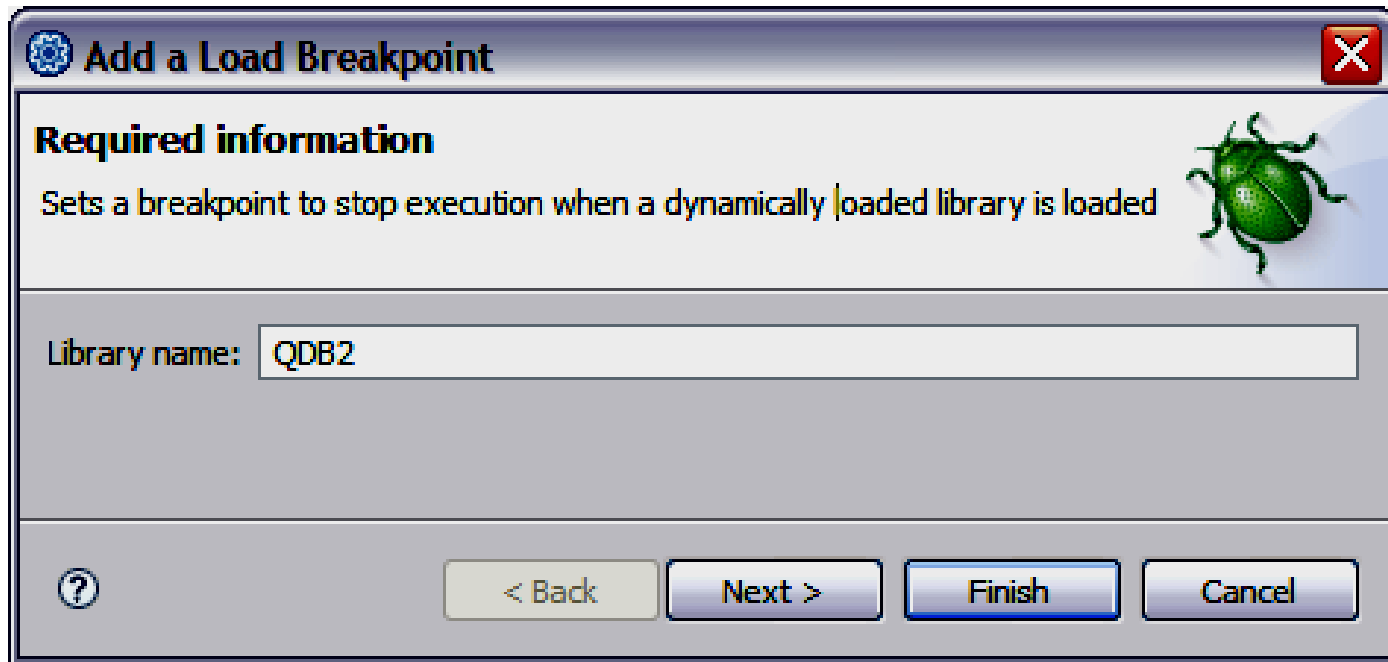
1. Right-click on breakpoints view and select “Load...”



Existing Features

- **Load breakpoints**

- Setting a load breakpoint
 2. Enter the module name where you wish to stop
 3. Click finish



Existing Features

- **Watchpoint breakpoints**

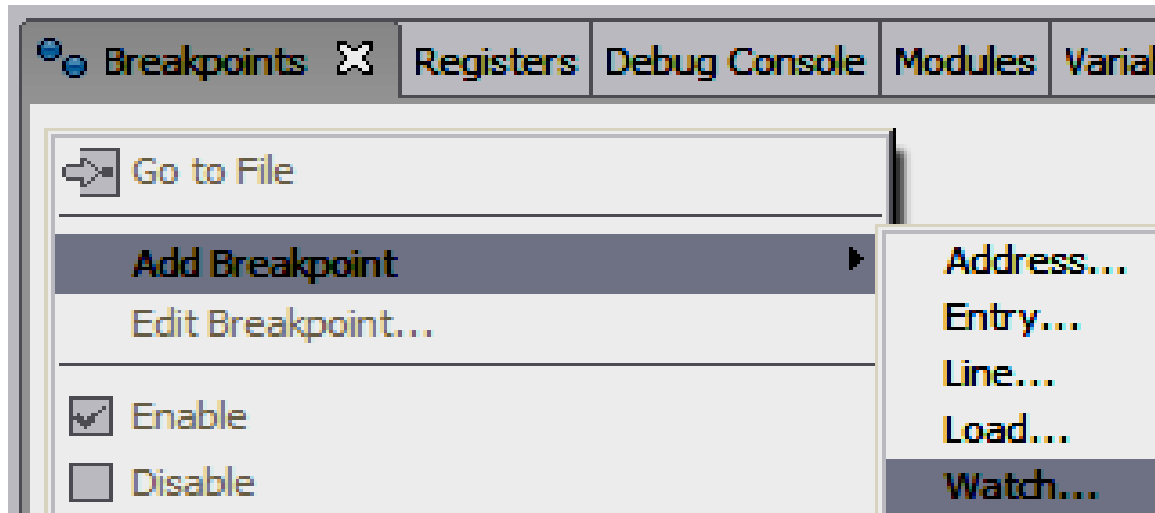
- Debugger stops when storage is altered



(Variable block_ptr)

- Setting a watchpoint breakpoint

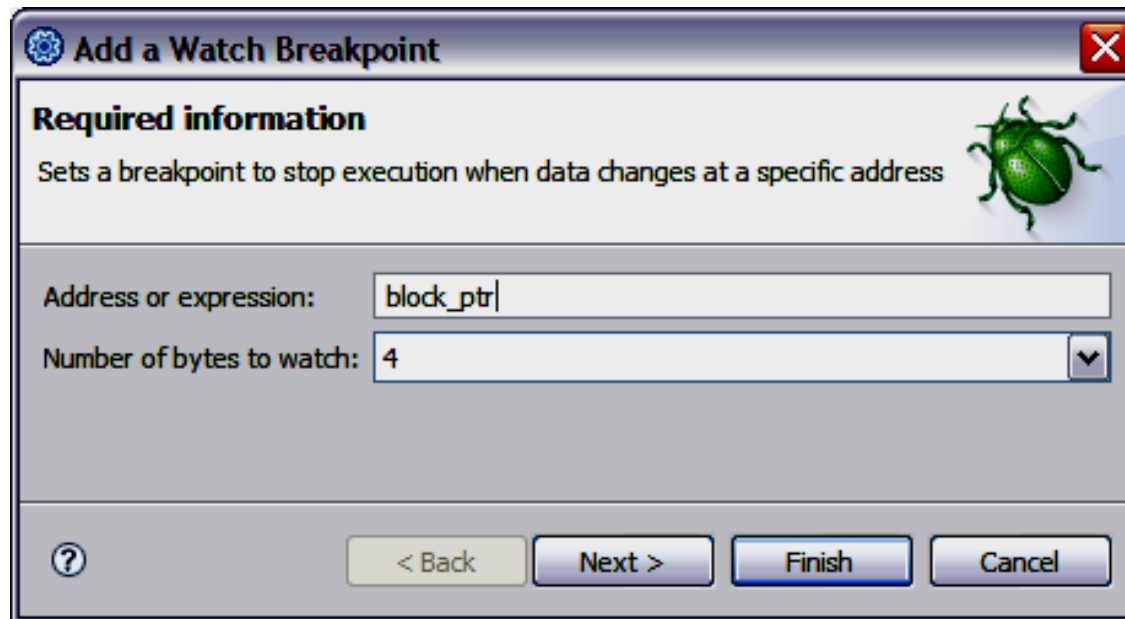
1. Right-click on breakpoints view and select “Watch...”



Existing Features

- **Watchpoint breakpoints**

- Setting a watchpoint breakpoint
 2. Enter the name of the address or expression where the breakpoint is to be set.
 3. Enter the number of bytes to watch
 4. Click finish



Add a Watch Breakpoint

Required information
Sets a breakpoint to stop execution when data changes at a specific address

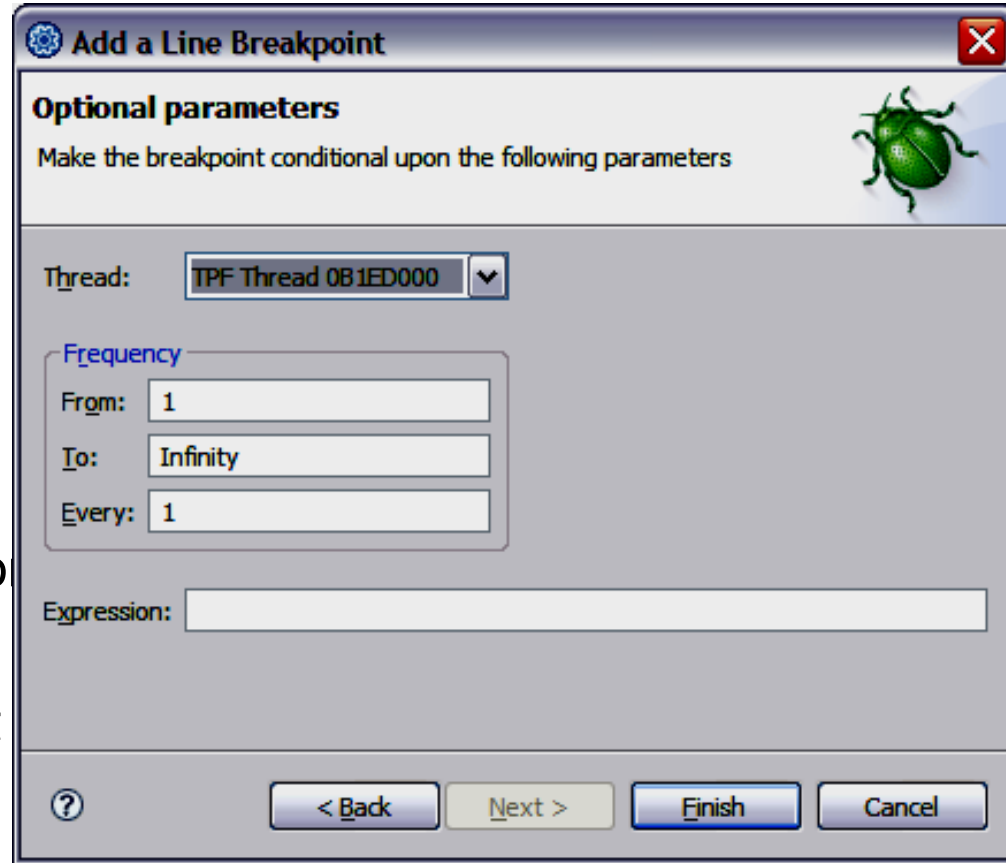
Address or expression:

Number of bytes to watch:

Existing Features

- **Frequency and expression support**

- Frequency support
 - Breakpoint stops only
 - when the occurrence condition is met
 - All breakpoint types
- Expression support
 - Breakpoint stops only
 - when expression condition is met
 - All breakpoint types except
 - Load breakpoint



Add a Line Breakpoint

Optional parameters
Make the breakpoint conditional upon the following parameters

Thread: TPF Thread 0B1ED000

Frequency

From: 1

To: Infinity

Every: 1

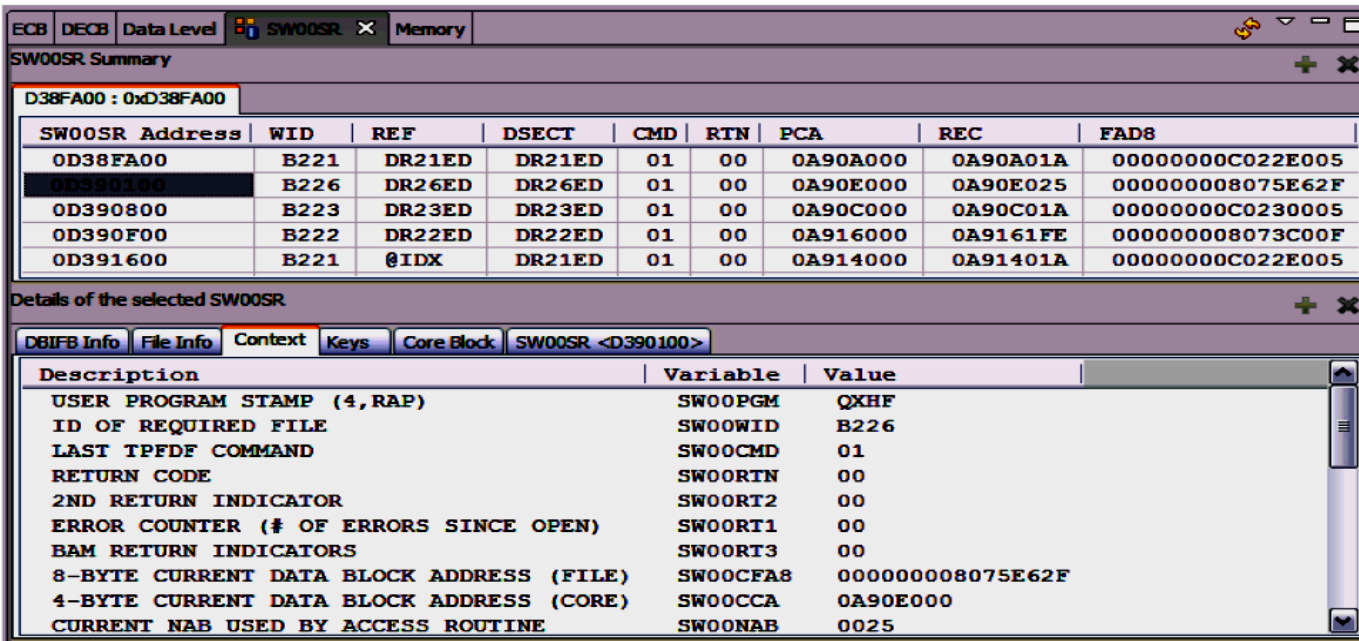
Expression:

< Back Next > Finish Cancel

Existing Features

• SW00SR View

- Displays the SW00SR slots for a debugged ECB
- It contains information on items such as core blocks and LRECs.
- For more information, reference TPF Debugger Update <http://ibm.com/tpf/tpfug/tgf07/tgf07.htm>



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

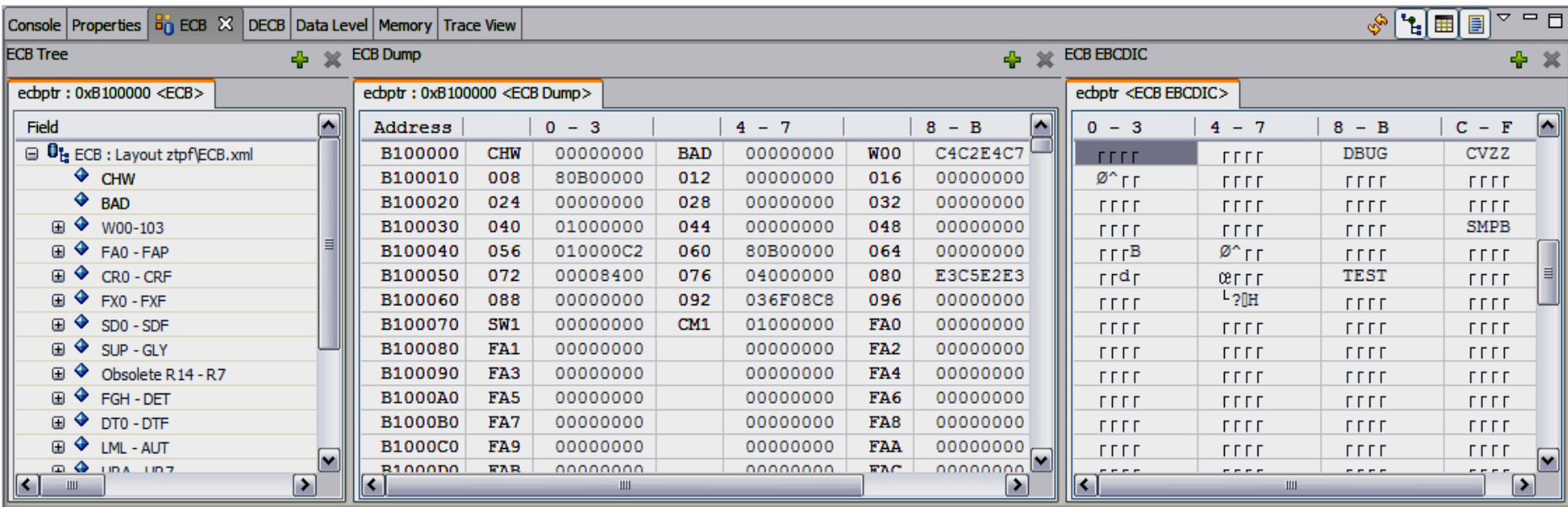
D390100

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)	SW00CFAB	000000008075E62F
4-BYTE CURRENT DATA BLOCK ADDRESS (CORE)	SW00CCA	0A90E000
CURRENT NAB USED BY ACCESS ROUTINE	SW00NAB	0025

Existing Features

• ECB View

- Displays page 1 of the ECB memory block
- For more information, reference TPF Debugger TPF Views <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm>



The screenshot displays the TPF Debugger interface with three main panels:

- ECB Tree:** Shows a hierarchical view of the ECB structure. The root is "ECB : Layout ztpf\ECB.xml", which contains several fields including CHW, BAD, W00-103, FA0 - FAP, CR0 - CRF, FX0 - FXF, SD0 - SDF, SUP - GLY, Obsolete R14 - R7, FGH - DET, DT0 - DTF, LML - AUT, and LPA - LPT.
- ECB Dump:** A table showing the memory dump for the selected ECB. The columns are Address, 0 - 3, 4 - 7, 8 - B, and C - F. The data is as follows:

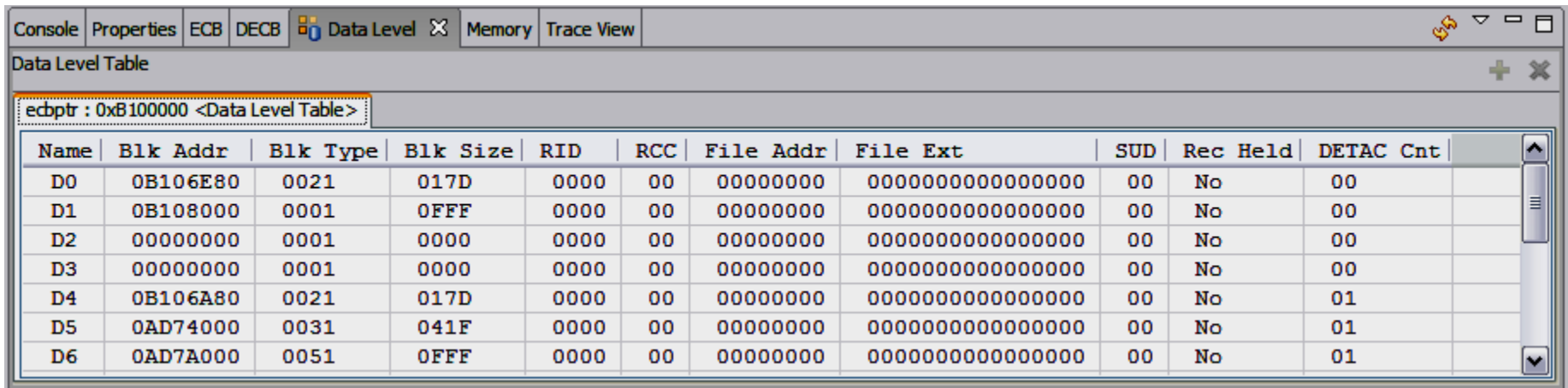
Address	0 - 3	4 - 7	8 - B	C - F
B10000	CHW	00000000	BAD	00000000
B100010	008	80B00000	012	00000000
B100020	024	00000000	028	00000000
B100030	040	01000000	044	00000000
B100040	056	010000C2	060	80B00000
B100050	072	00008400	076	04000000
B100060	088	00000000	092	036F08C8
B100070	SW1	00000000	CM1	01000000
B100080	FA1	00000000		FA2
B100090	FA3	00000000		FA4
B1000A0	FA5	00000000		FA6
B1000B0	FA7	00000000		FA8
B1000C0	FA9	00000000		FAA
B1000D0	FAB	00000000		FAC
- ECB EBCDIC:** A table showing the EBCDIC representation of the ECB data. The columns are 0 - 3, 4 - 7, 8 - B, and C - F. The data is as follows:

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
rrd	ørrr	TEST	rrrr
rrrr	L?[]H	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
rrrr	rrrr	rrrr	rrrr

Existing Features

- **Data Level**

- Data Level view shows the contents of 16 data levels
- Rec Held indicates the file address is held in the record hold table for this ECB
- For more information, reference TPF Debugger TPF Views <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm>



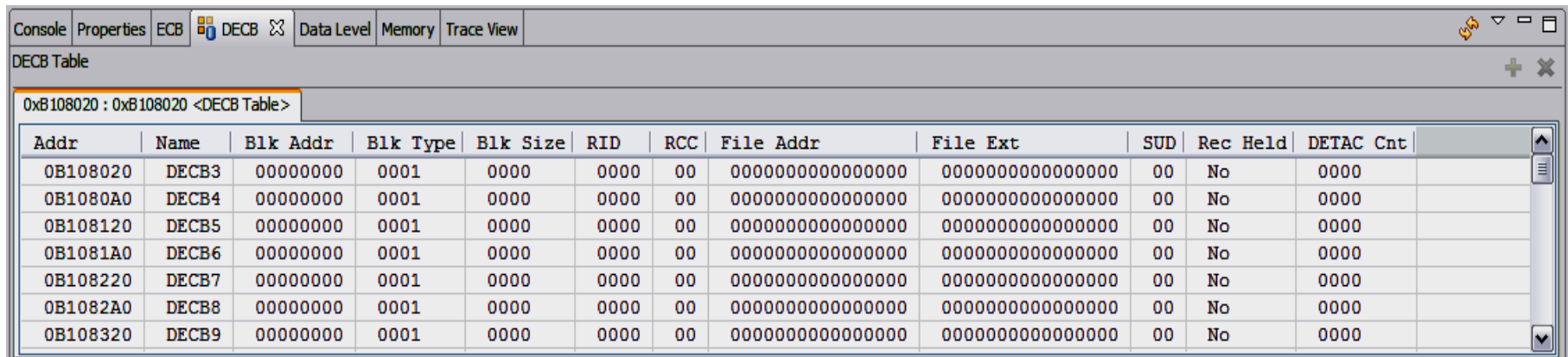
ecbptr : 0xB100000 <Data Level Table>

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	00000000000000000000	00	No	00
D1	0B108000	0001	0FFF	0000	00	00000000	00000000000000000000	00	No	00
D2	00000000	0001	0000	0000	00	00000000	00000000000000000000	00	No	00
D3	00000000	0001	0000	0000	00	00000000	00000000000000000000	00	No	00
D4	0B106A80	0021	017D	0000	00	00000000	00000000000000000000	00	No	01
D5	0AD74000	0031	041F	0000	00	00000000	00000000000000000000	00	No	01
D6	0AD7A000	0051	0FFF	0000	00	00000000	00000000000000000000	00	No	01

Existing Features

- **DECB**

- DECB view shows the contents of the created DECBs
- Same functionality provided as Data Level view
- For more information, reference TPF Debugger TPF Views <http://ibm.com/tpf/tpfug/tgs07/tgs07.htm>



Addr	Name	Blk Addr	Blk Type	Blk Size	RID	RCC	File Addr	File Ext	SUD	Rec Held	DETAC Cnt
0B108020	DECB3	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000
0B1080A0	DECB4	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000
0B108120	DECB5	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000
0B1081A0	DECB6	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000
0B108220	DECB7	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000
0B1082A0	DECB8	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000
0B108320	DECB9	00000000	0001	0000	0000	00	0000000000000000	0000000000000000	00	No	0000

APARS

- **ECB, DECB, and Data Level Views**
 - TPF Toolkit V3.2
 - TPF 4.1 PJ31891
 - z/TPF PJ31890
- **SW00SR**
 - TPF Toolkit V3.2.3
 - TPF 4.1 PJ32240
 - z/TPF PJ32221

Trademarks

- **IBM** are trademarks of International Business Machines Corporation 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.