



| z/TPF V1.1

2013 TPF Users Group

Title: z/TPF Debugger Update

Joshua Wisniewski
Development Tools Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1

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.

Agenda

- New education material
- New Features
 - Code Coverage Merge
 - LoadTPF Actions
 - Custom Terminal Token
 - Performance Improvement with Large Modules
- Previously Released Features
 - Show Code Optimization Level
 - Show Loadset Name
 - Fork Interface Enhancement
 - Memory Search
 - Improve Remote Debug Information

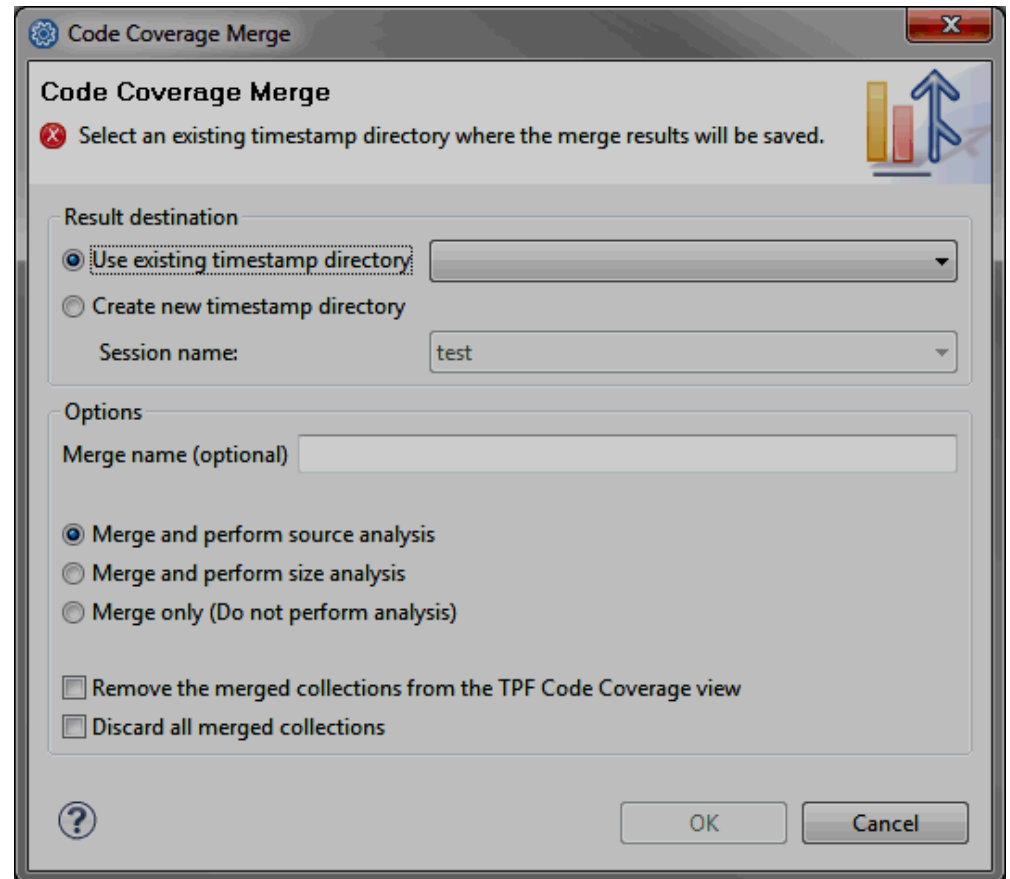
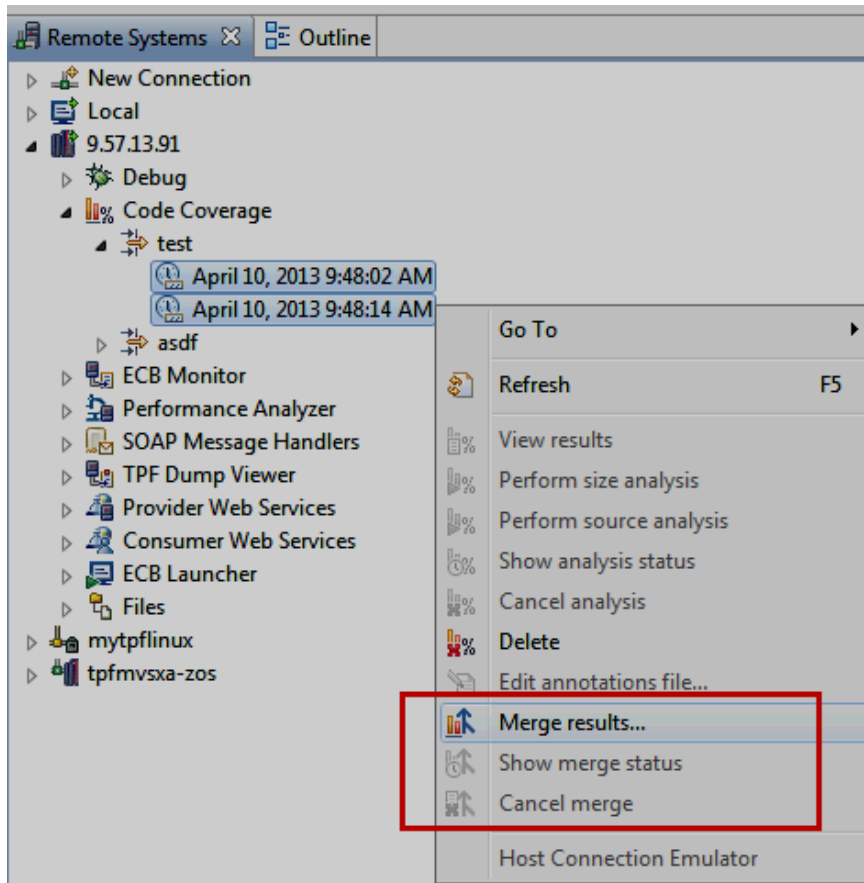
New education material

- New education materials are focused on practical application of the z/TPF debugger features. For example: if your application is corrupting your stack frame, what features of the debugger can be used to diagnose the problem.
- Available at

<http://www-01.ibm.com/software/htp/tpf/>. See the Fast links section on the lower left side. Select Tools -> z/TPF Debugger and then view the contents of the education material table.
- See debugger education on Wednesday for more information.

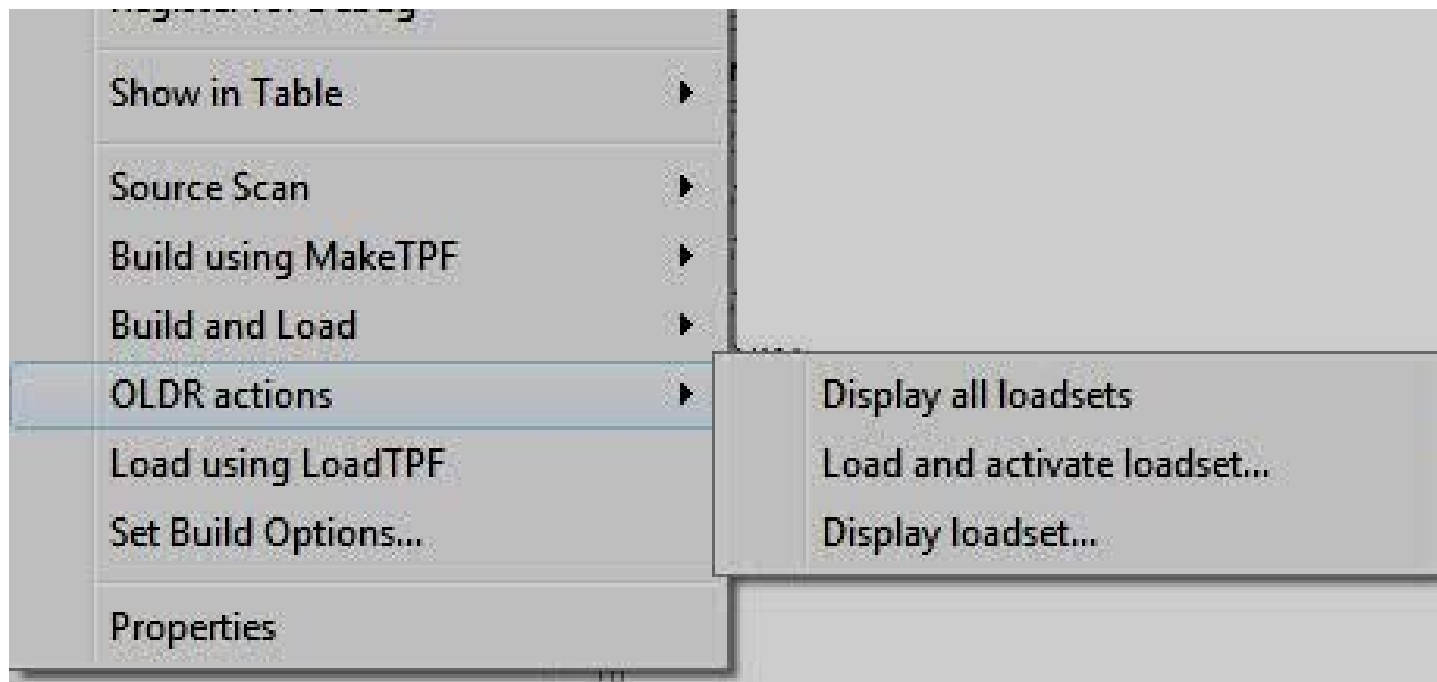
Code Coverage Merge

- Allows you to merge code coverage collections into a single collection.
- See TPF Toolkit presentation from TPF Toolkit Task Force.



LoadTPF Actions

- Allows you to issue OLDR load commands on TPF as part of your LoadTPF actions through the TPF Toolkit.
- LNIATAs need to be set up on TPF as described at http://publib.boulder.ibm.com/infocenter/tpfhelp/current/topic/com.ibm.ztpf-ztpfdf.doc_put.cur/gtpd3/d3tlniata.html
- See TPF Toolkit presentation from TPF Toolkit Task Force.



Custom Terminal Token

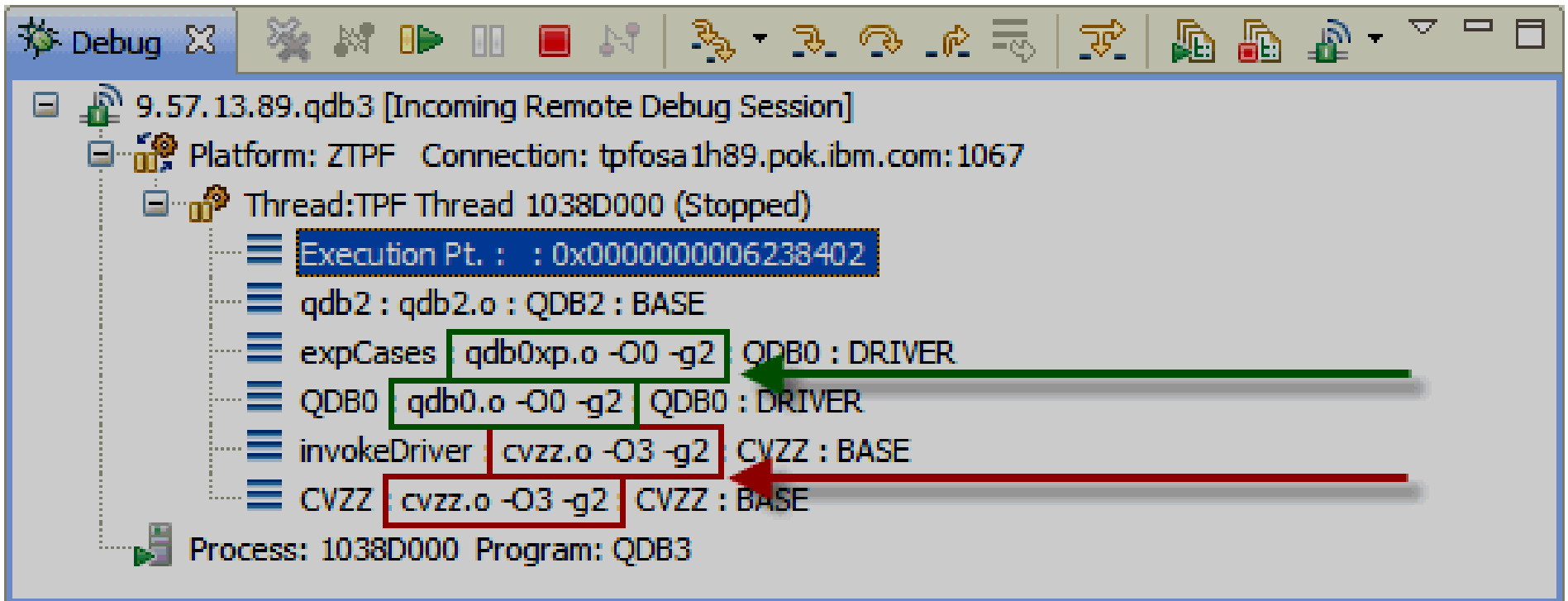
- Implemented for customers who have their own TN3270 support such that you can register for debugging by the logical LNIATA.
- Allows you to specify a custom terminal token for ECBs started from an unsupported communications protocol thereby allowing trace by terminal registration to be used.
- For example, suppose you implemented your own TN3270 support such that ECBs are created from a socket interface. You can use this user exit to return the LNIATA for the Terminal Token thereby allowing users to register for trace by terminal by LNIATA.
- Modify the new `CDBX_DebuggerTBTRegistrationTerminalUserExit` in `cdbuxt.c` to detect your ECB conditions and return the custom Terminal Token.
- See debugger education on Wednesday for more information.

Performance Improvement with Large Modules

- This enhancement provides a substantial performance improvement for debugging large modules or C/C++ modules with many global variables by deferring the loading of debug information to an object by object basis as needed.

Show Code Optimization Level

- Debug view shows optimization level and debug information level (dwarf level) for each object in each stack frame.
 - C/C++ objects should be built with `-O0` for optimal debugging.



The screenshot shows the Debug View interface with the following stack frames and their optimization levels:

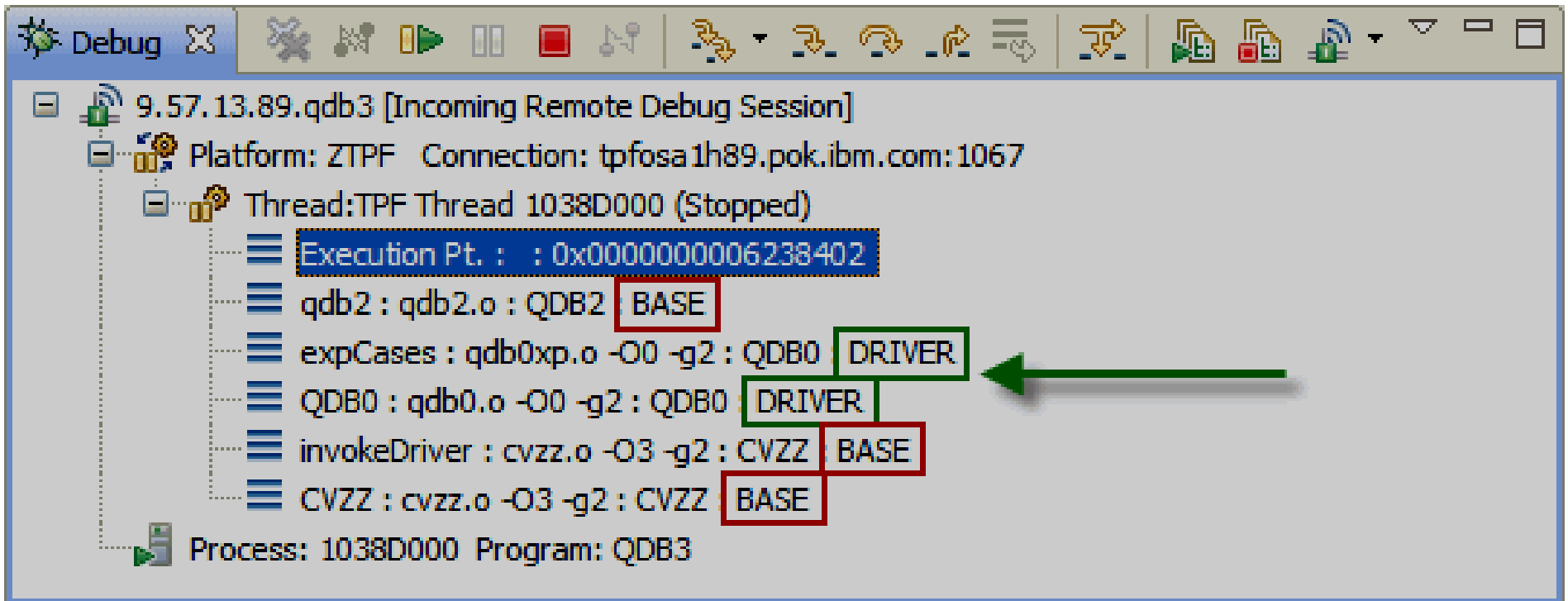
- Execution Pt. : : 0x00000000006238402
- qdb2 : qdb2.o : QDB2 : BASE
- expCases : qdb0xp.o -O0 -g2 QDB0 : DRIVER
- QDB0 : qdb0.o -O0 -g2 QDB0 : DRIVER
- invokeDriver : cvzz.o -O3 -g2 CVZZ : BASE
- CVZZ : cvzz.o -O3 -g2 CVZZ : BASE

Process: 1038D000 Program: QDB3

Annotations in the image include a blue dashed box around the execution point, green boxes around the `-O0 -g2` flags for `expCases` and `QDB0`, and red boxes around the `-O3 -g2` flags for `invokeDriver` and `CVZZ`. Green and red arrows point from these boxes to the corresponding object names in the stack frames.

Show Loadset Name

- Debug view shows the loadset name for each module in each stack frame.
 - Modules without a loadset name will show as BASE.



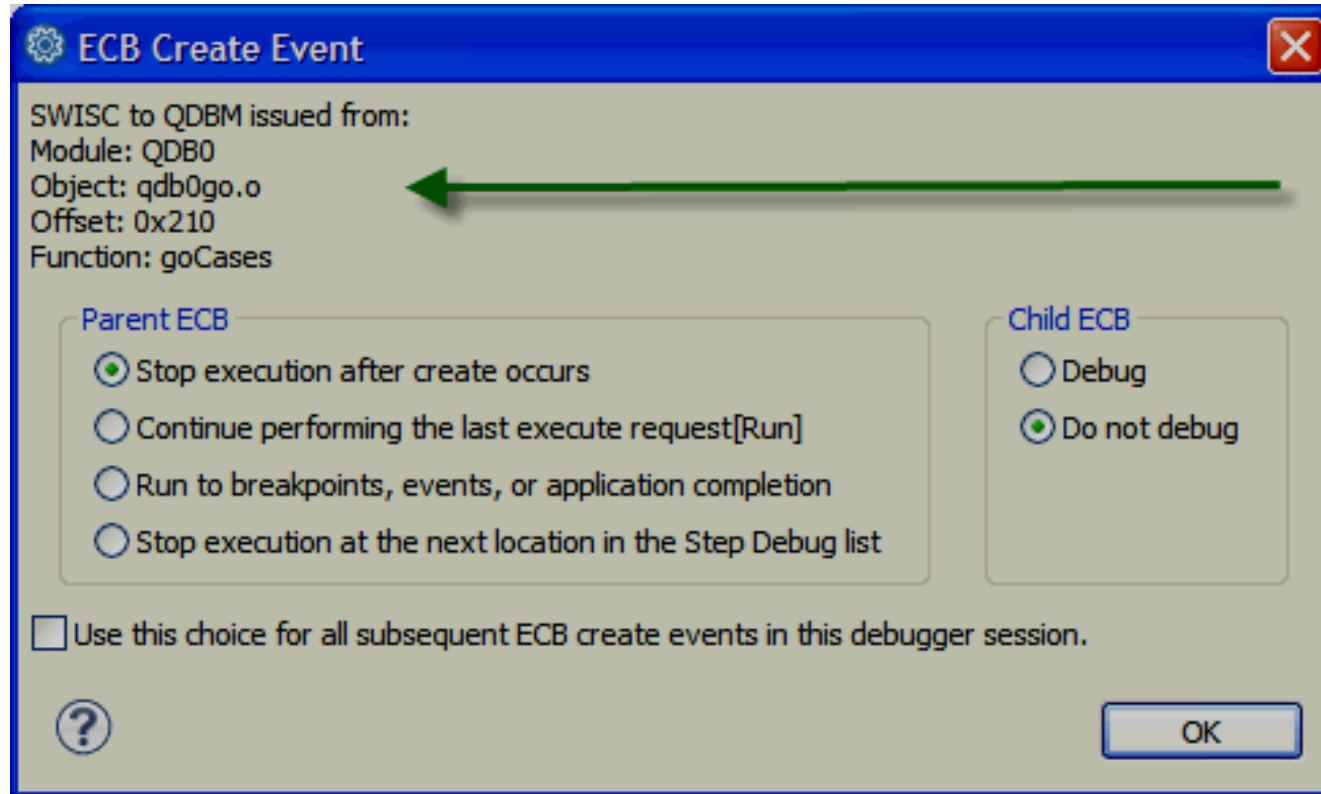
The screenshot displays the Debug View interface for a remote debug session. The stack trace shows the following frames from top to bottom:

- Execution Pt. : : 0x0000000006238402
- qdb2 : qdb2.o : QDB2 **BASE**
- expCases : qdb0xp.o -O0 -g2 : QDB0 **DRIVER**
- QDB0 : qdb0.o -O0 -g2 : QDB0 **DRIVER**
- invokeDriver : cvzz.o -O3 -g2 : CVZZ **BASE**
- CVZZ : cvzz.o -O3 -g2 : CVZZ **BASE**

At the bottom, the process information is shown: Process: 1038D000 Program: QDB3. A green arrow points to the 'DRIVER' label in the QDB0 frame.

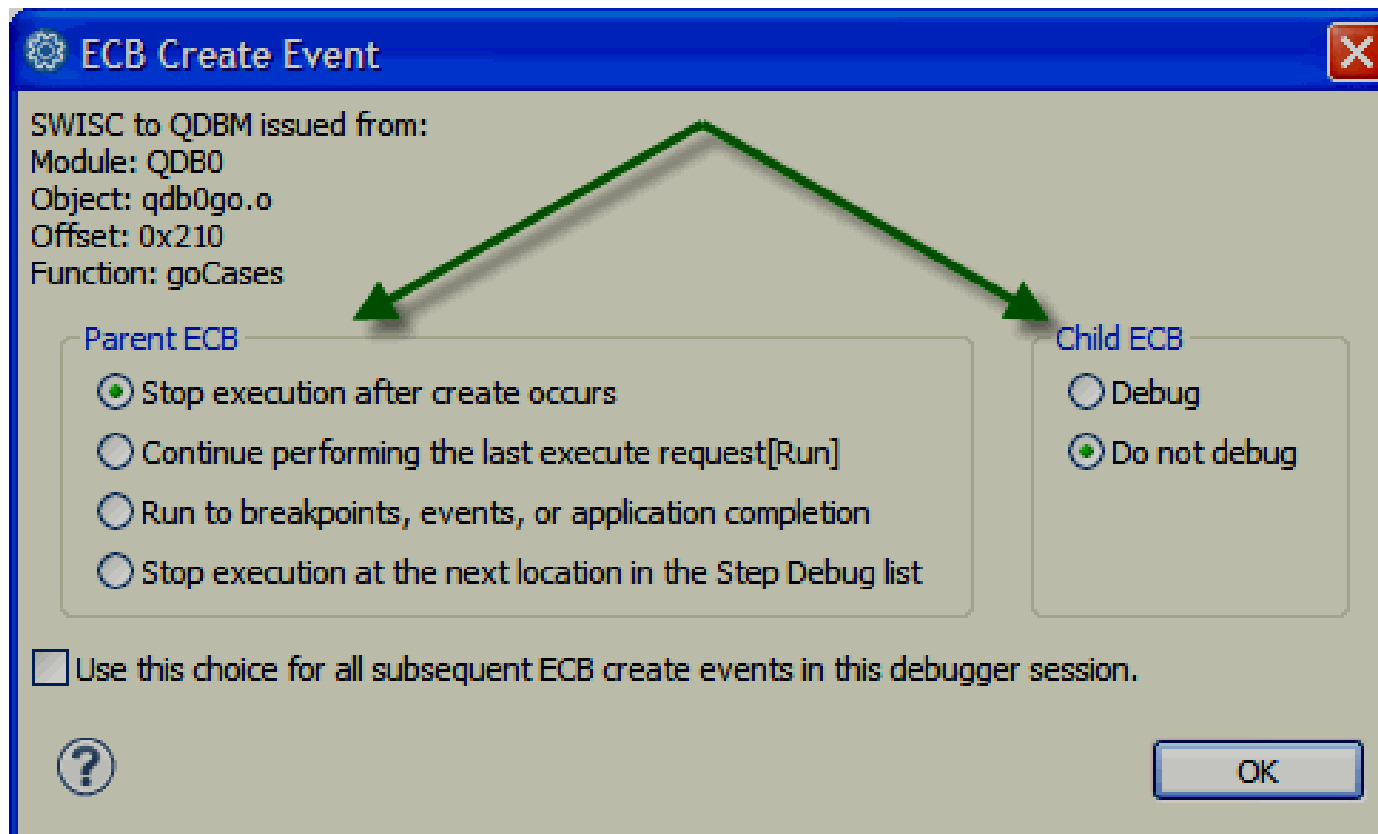
Fork Interface Enhancement

- The new ECB Create interface shows concise information.
 - What type of ECB Create event occurred.
 - Which module will be entered by the child ECB.
 - Where the ECB Create event occurred.



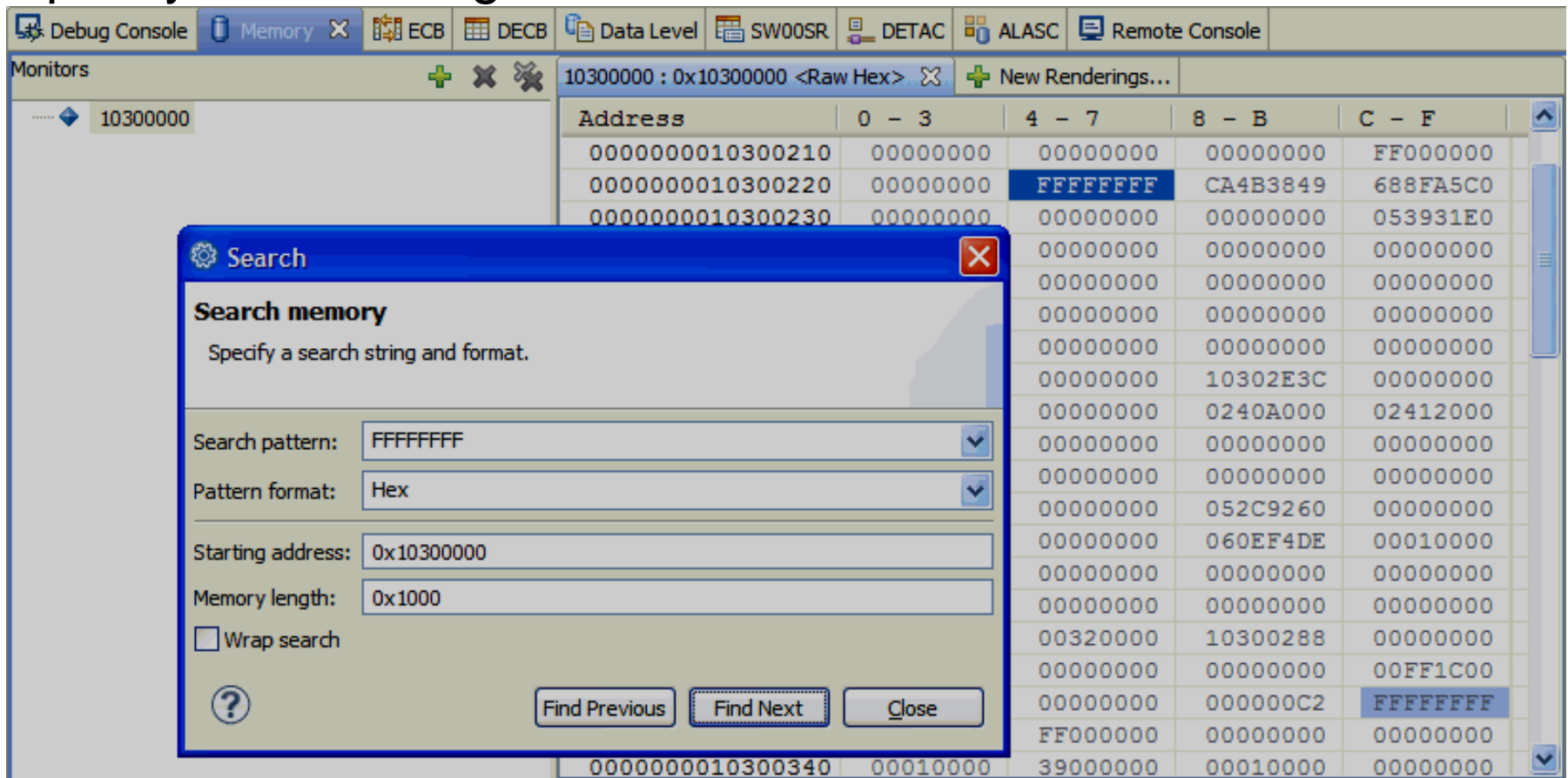
Fork Interface Enhancement

- The new ECB Create interface provides concise choices for both the parent and child ECBs separately.
- Two new options provided for parent ECB: Run and Step Debug.



Memory Search

- Right click and choose “Search Memory” in any memory view.
- Specify search pattern and type: HEX, ASCII, EBCDIC, UTF-8.
- Specify search range.



The screenshot shows the IBM z/OS Debug Console interface. A memory dump is displayed in the background, showing addresses and their corresponding hexadecimal values. A search dialog box is overlaid on the dump, allowing the user to specify search parameters.

Search Dialog Box:

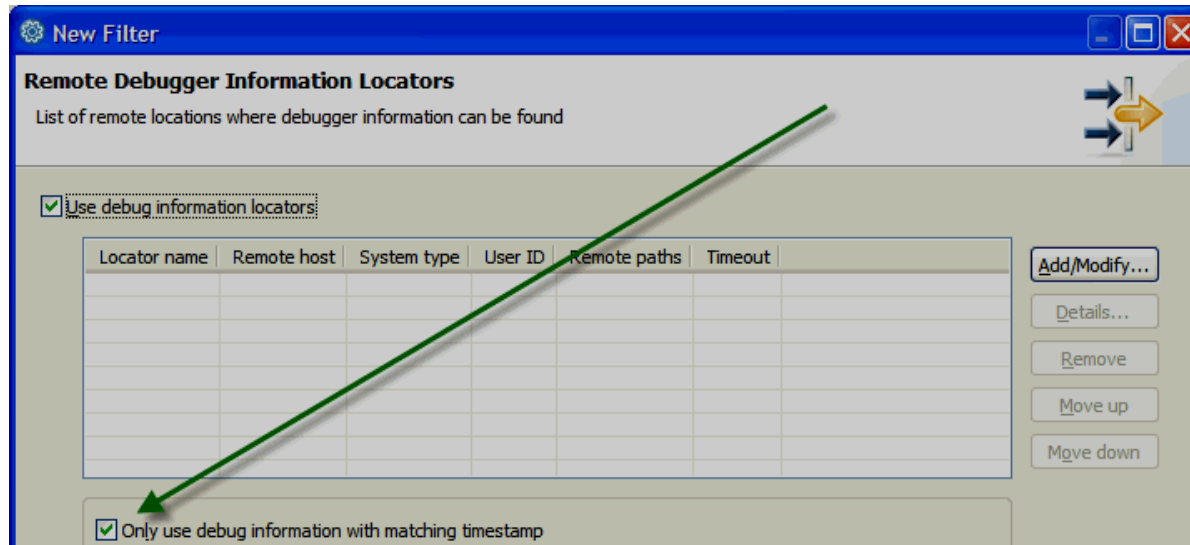
- Search memory**: Specify a search string and format.
- Search pattern:** FFFFFFFF
- Pattern format:** Hex
- Starting address:** 0x10300000
- Memory length:** 0x1000
- Wrap search
- Buttons:** Find Previous, Find Next, Close

Memory Dump (Address | 0 - 3 | 4 - 7 | 8 - B | C - F):

Address	0 - 3	4 - 7	8 - B	C - F
0000000010300210	00000000	00000000	00000000	FF000000
0000000010300220	00000000	FFFFFFFF	CA4B3849	688FA5C0
0000000010300230	00000000	00000000	00000000	053931E0
0000000010300240	00000000	00000000	00000000	00000000
0000000010300250	00000000	00000000	00000000	00000000
0000000010300260	00000000	00000000	00000000	00000000
0000000010300270	00000000	00000000	00000000	00000000
0000000010300280	00000000	00000000	10302E3C	00000000
0000000010300290	00000000	00000000	0240A000	02412000
00000000103002A0	00000000	00000000	00000000	00000000
00000000103002B0	00000000	00000000	00000000	00000000
00000000103002C0	00000000	00000000	00000000	00000000
00000000103002D0	00000000	00000000	052C9260	00000000
00000000103002E0	00000000	00000000	060EF4DE	00010000
00000000103002F0	00000000	00000000	00000000	00000000
0000000010300300	00000000	00000000	00000000	00000000
0000000010300310	00000000	00000000	00000000	00000000
0032000010300320	10300288	00000000	00000000	00000000
0000000010300330	00000000	00000000	00FF1C00	00000000
0000000010300340	000000C2	FFFFFFFF	00000000	00000000
FF00000010300350	00000000	00000000	00000000	00000000
0000000010300360	00010000	39000000	00010000	00000000

Improve Remote Debug Information

- The Remote Debug Information feature allows you to store your debug information on a remote system as selected at registration time for the debugger to dynamically load to z/TPF as needed.
- This new feature allows the debugger to search multiple remote paths for debug information that exactly matches the code loaded to the system. If an exact match is not found, no debug information is used to help ensure users do not have a debug information mismatch.
- Turning off this feature allows you to override the debug information as you can today.



z/TPF Debugger Deliverable Details: Available Soon

Description	z/TPF APAR	z/TPF PUT Level	TPF Toolkit Level	TPFUG Requirement
Code Coverage Tool Merge LoadTPF Actions	PJ40845	PUT10	V.next V.next	Customer Request

z/TPF Debugger Deliverable Details: Available

Description	z/TPF APAR	z/TPF PUT Level	TPF Toolkit Level	TPFUG Requirement
Custom Terminal Token	PJ40974	PUT10	None	Customer Request
Performance Improvements with Large Modules	PJ39869	PUT10	None	Customer Request
Show Loadset Name	PJ39617	PUT9	None	Customer Request
Show Code Optimization Level			None	Customer Request
Improve Remote Debug Info			V3.6.4	Customer Request
Memory Search			V3.6.4	V09114F
Fork Interface Enhancement	PJ40255	PUT9	V4.0.0	Customer Request



Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "[Copyright and trademark information](#)" at www.ibm.com/legal/copytrade.shtml.
- *(Include any special attribution statements as required – see Trademark guidelines on <https://w3-03.ibm.com/chq/legal/lis.nsf/lawdoc/5A84050DEC58FE31852576850074BB32?OpenDocument#Developing%20the%20Special%20Non-IBM%20Tr>)*

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.