z/TPF V1.1

TPF Users Group – Fall 2012

z/TPF Debugger Update

Josh Wisniewski
Development Tools Subcommittee

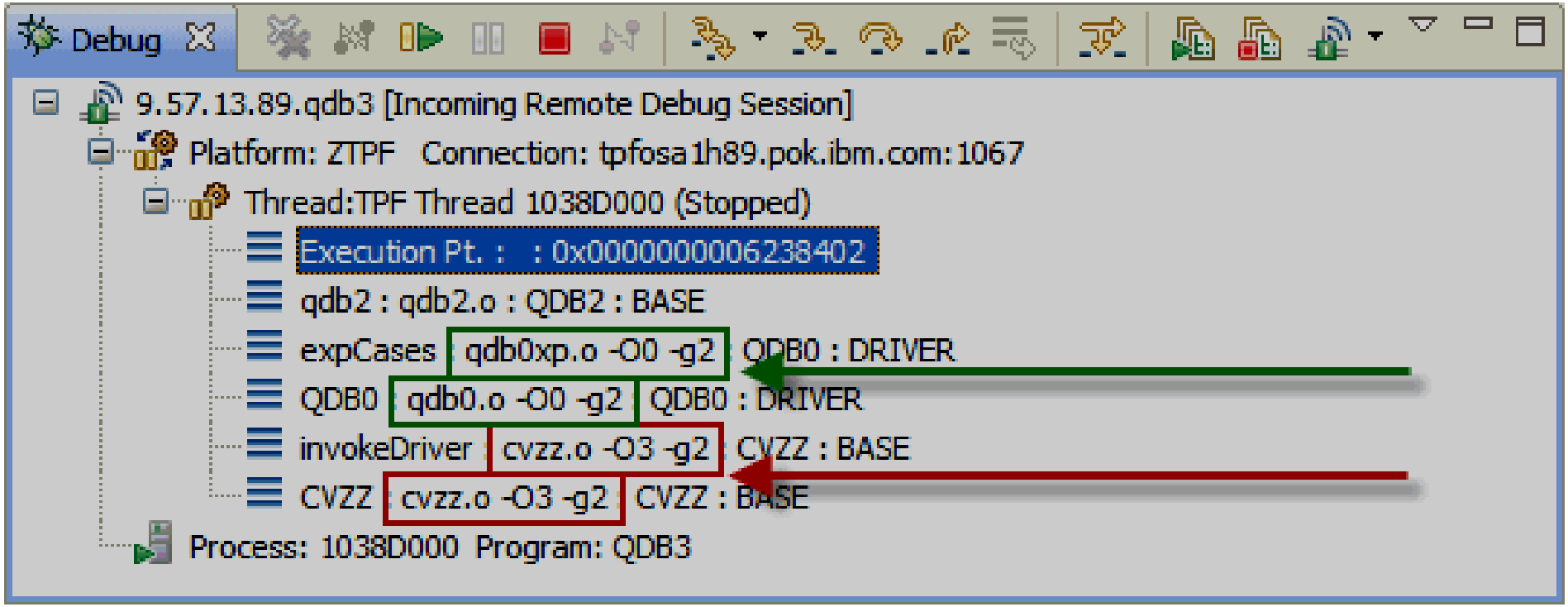
AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Agenda

- **New Features**
  - **Show Code Optimization Level**
  - **Show Loadset Name**
  - **Fork Interface Enhancement**
  - **Memory Search**
  - **Improve Remote Debug Information**
  - **Dump Viewer Data Scrubbing User Exit**
  - **CDB0 Interface**
  - **Prevent ZINET ADD S-DEBUG**
  - **Remote Debug Information Name User Exit**
- **Previously released features**
  - **Code Coverage Tool**
  - **Auto detect workstation IP**
  - **Improve debugger connection failure handling**

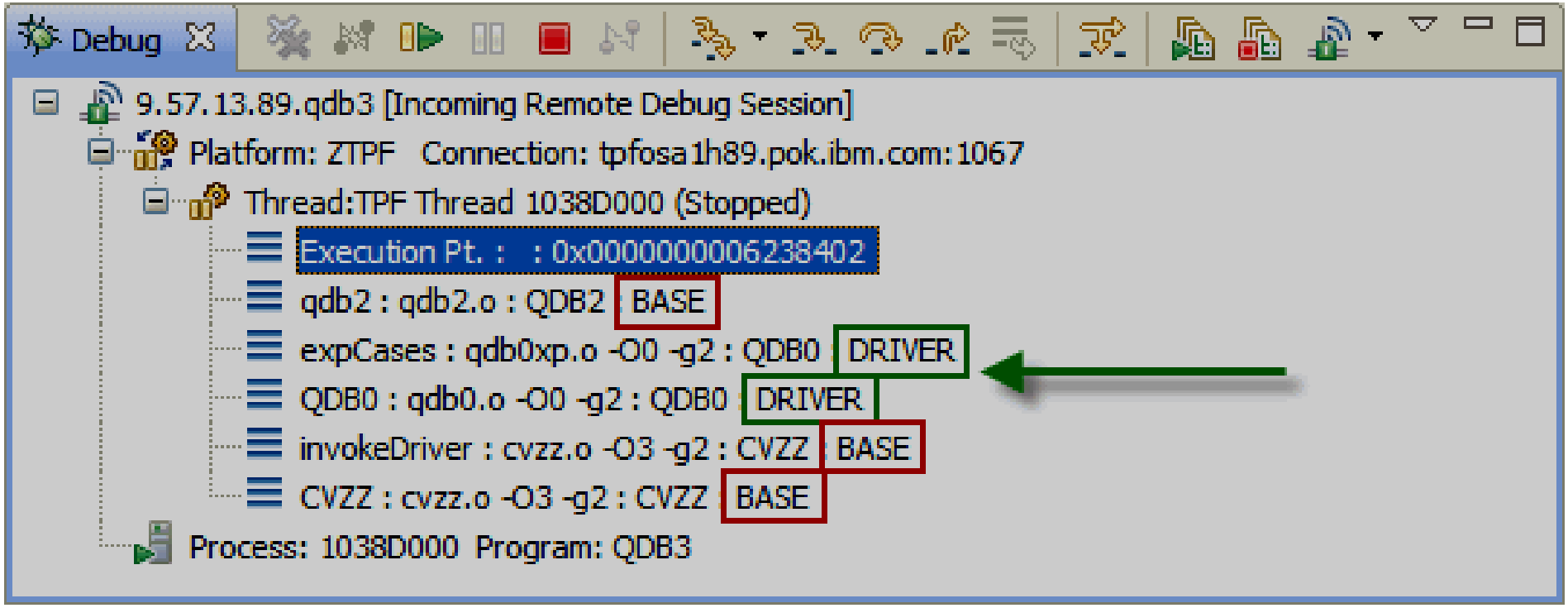
Show Code Optimization Level

- Debug view shows optimization level and debug information level (dwarf level) for each object in each stack frame.
  - Assembler objects do not show settings (for example qdb2.o).
  - C/C+ objects should be built with –O0 for optimal debugging.



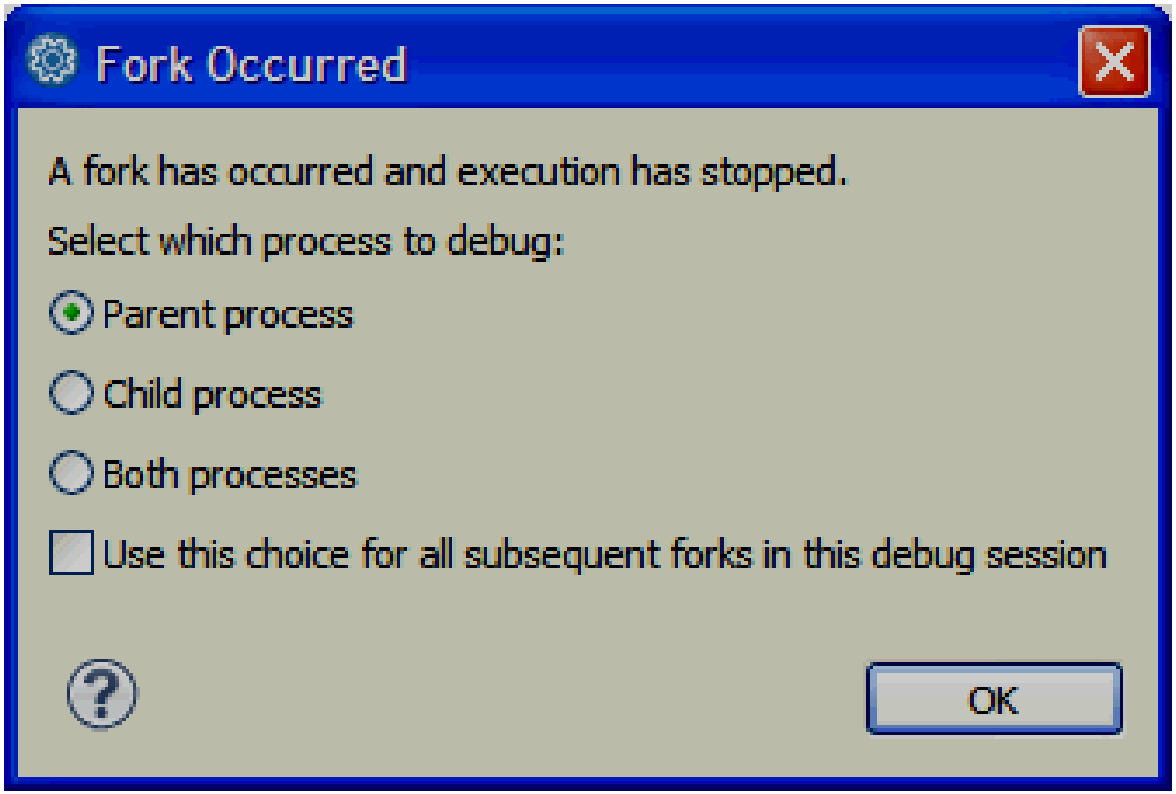
Show Loadset Name

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



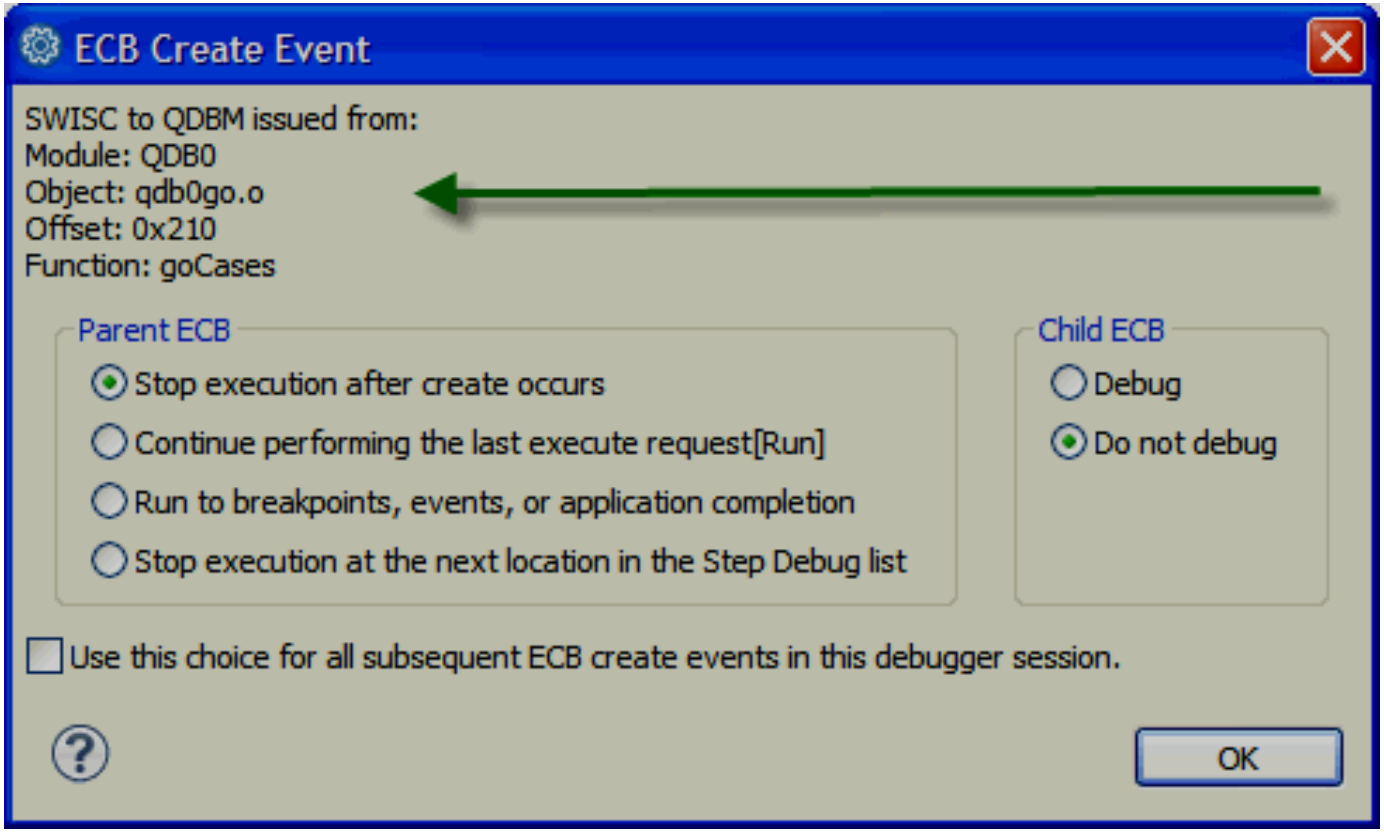
Fork Interface Enhancement

- The old fork interface presents options with ambiguous meaning.
- Details regarding the fork event are shown in the Debug Console which may be hidden.



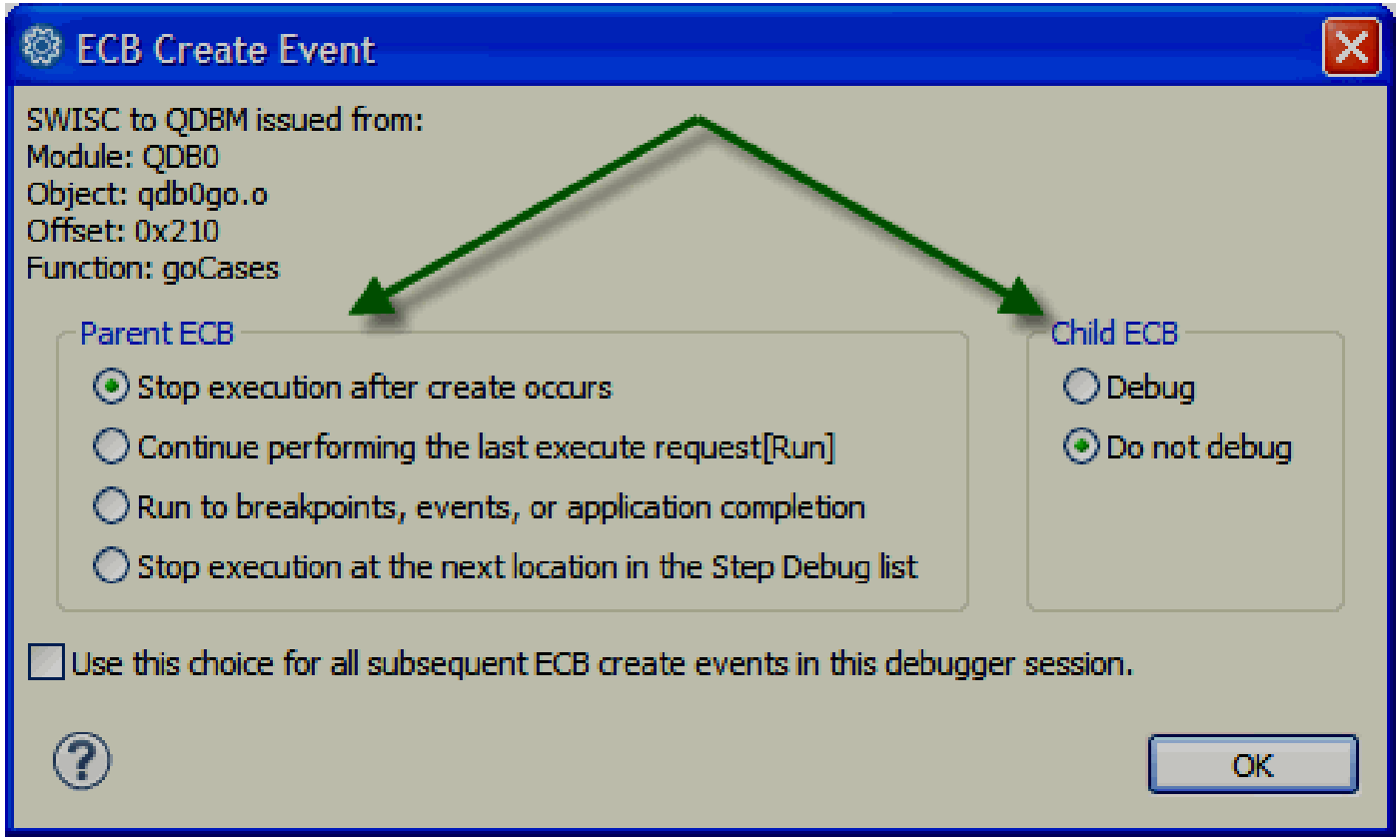
Fork Interface Enhancement

- 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.



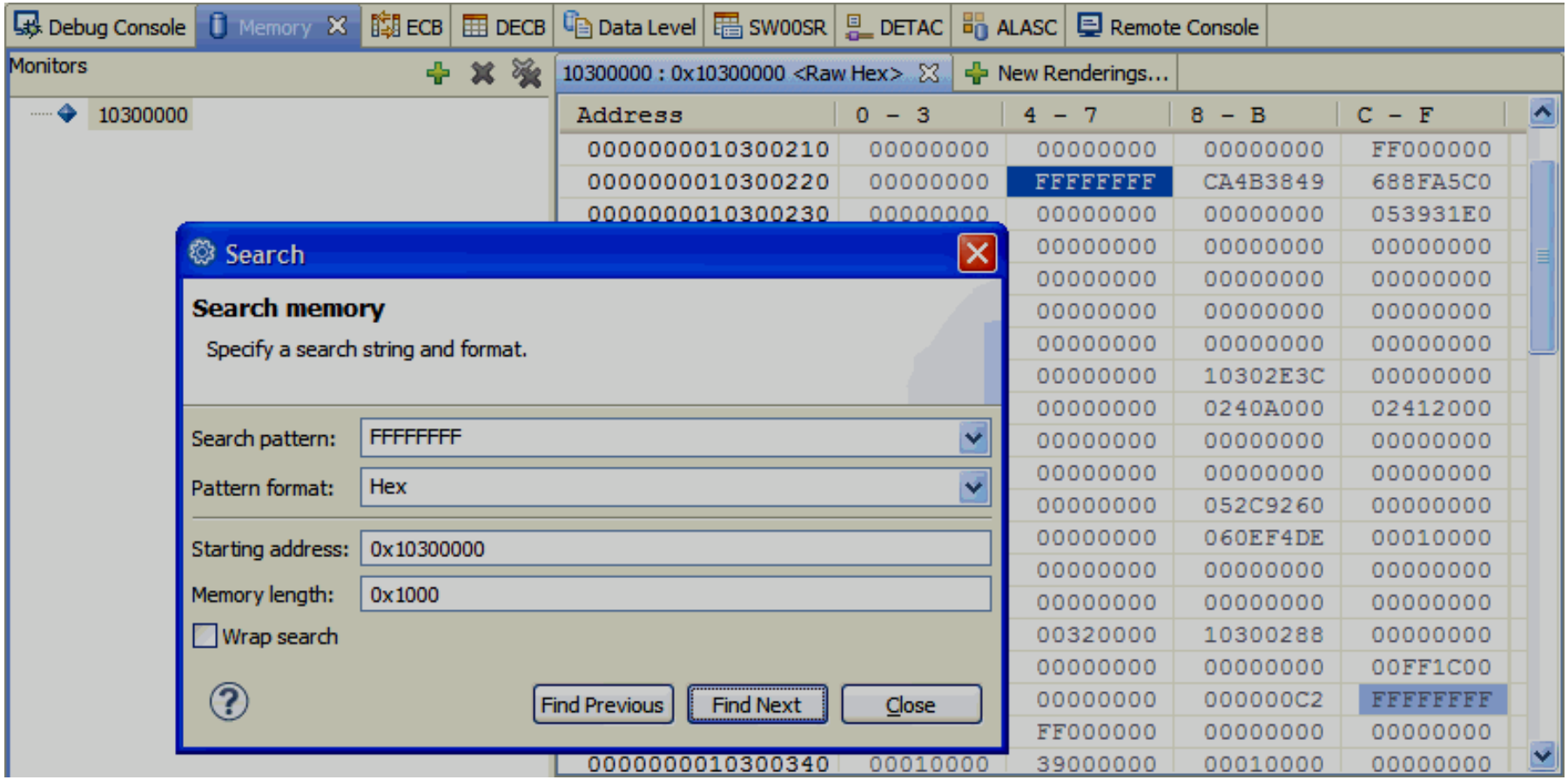
Fork Interface Enhancement

- 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.
- Hover over an option for more information.



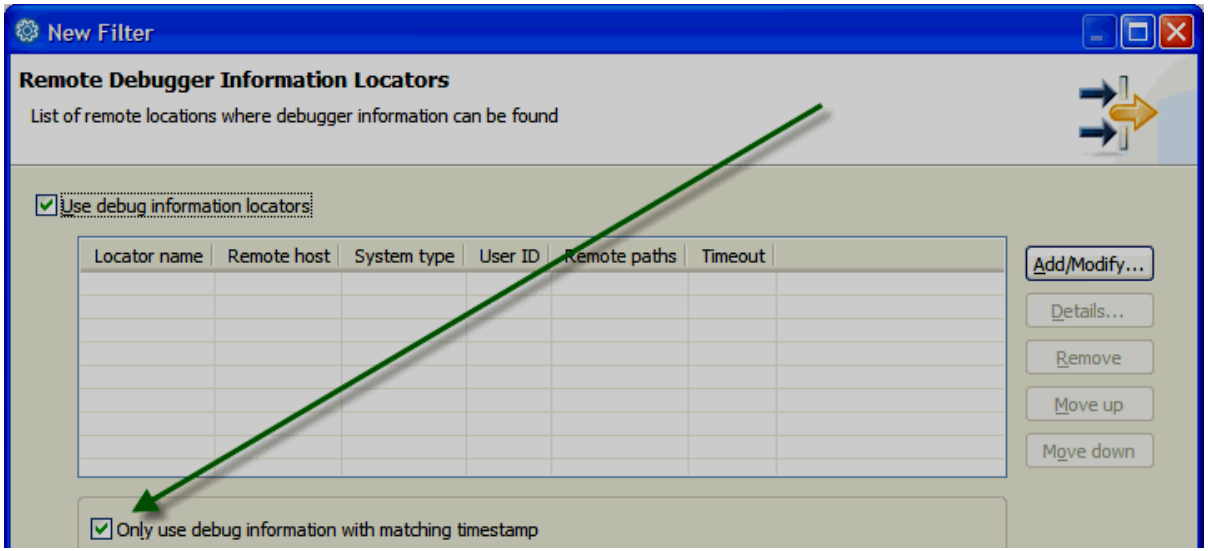
Memory Search

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



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.



Dump Viewer Data Scrubbing User Exit

- The debugger dump capture process has been enhanced to call user exit CDBX_ScrubDataUserExit in cdbxds.asm for all data areas captured to allow you to scrub any data from the dumps.
- This may be useful if you are not using the z/TPF Non-Displayable ECB Storage feature (NDSPC).

CDB0 Interface

- This new feature externalizes the interface to CDB0 as tpf_flag_for_debug which allows you to flag ECBs for debugging when an ECB is started from a custom Comms package.

Prevent ZINET ADD S-DEBUG

- This feature ensures that the z/TPF debugger daemon listener is defined on z/TPF as DBUG to ensure that the debugger daemon recycling (ZINET STOP and ZOLDR commands) occurs correctly.

Remote Debug Information Name User Exit

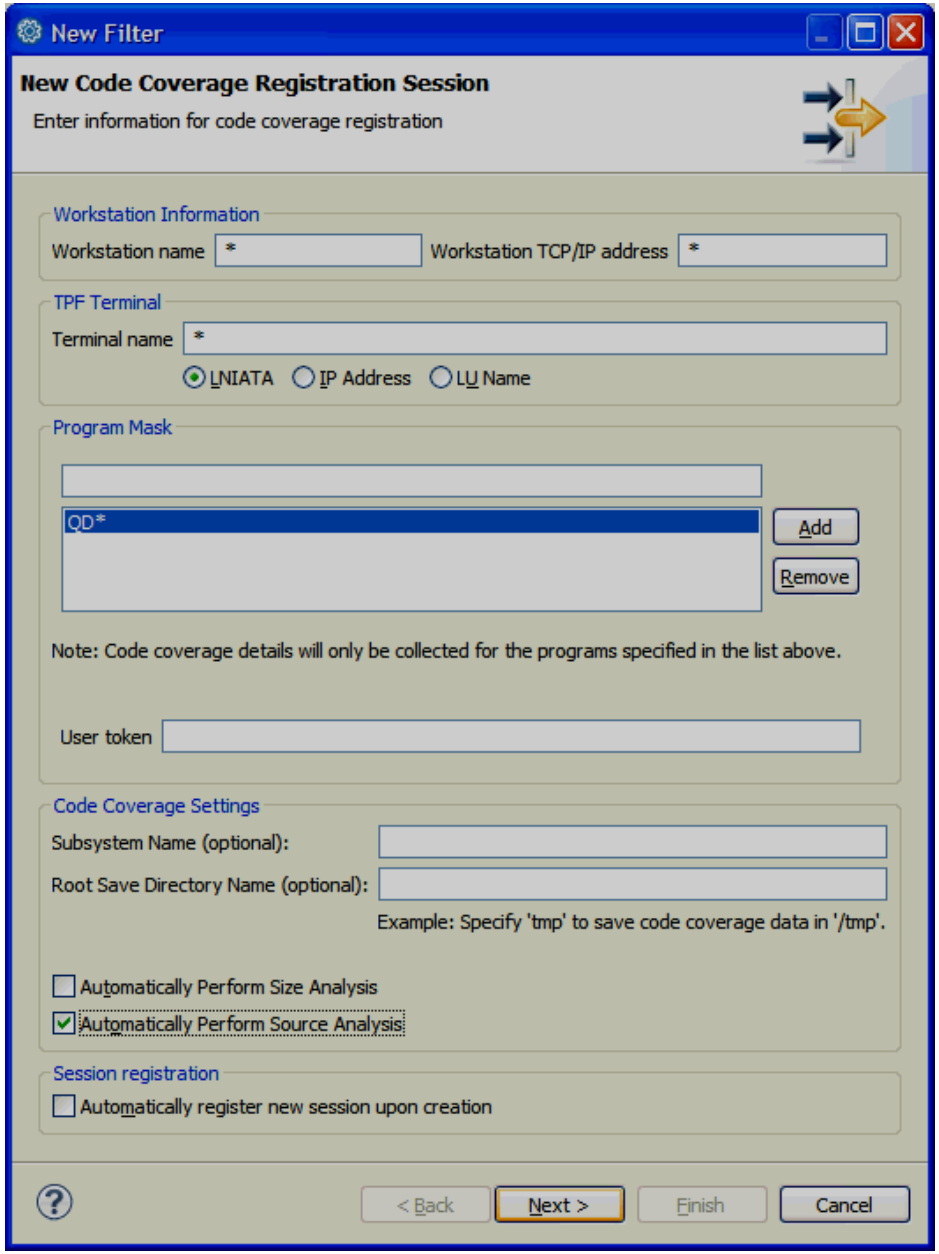
- This new user exit allows you to compose the name of shared objects to locate when using the Remote Debug Information feature.
- This may be useful if you do not use the maketpf build solution.

Code Coverage Tool

- Typical use cases of the code coverage tool:
  - A QA manager needs to know which modules of an application are not executed when a driver suite is run (hundreds or thousands of modules).
  - A tester needs to ensure that a test plan drives all modules, objects, and functions in an application (tens of modules).
  - A developer needs to ensure that all lines of a new application component have been tested (a few modules at most).
  - A QA regression test team is tasked with cooperatively writing a driver to test every line of an application.
- Diagnostic use cases of the code coverage tool:
  - Determine execution paths taken through ported code.

Code Coverage Tool

- Similar registration process to the z/TPF debugger.
- Program centric so multiple ECBs can cause data to be collected simultaneously in the same program.
- Size analysis provides instruction level statistics at the module, object and function level.
- Source analysis provides source line statistics at the source file and function level.



Code Coverage Tool

- Code coverage view shows execution statistics with sorting, filtering, export and etc capabilities.
- Editor view shows which lines have and have not been executed.
- Demo viewlet available at www.ibm.com/support/docview.wss?uid=swg21570467

Code Coverage | TPF Debug | Remote System Explorer | TPF Toolkit

Code Coverage

Filter is disabled. Configure filter.

	Size Percentage	Line Percentage
Host: 9.57.13.89, **Session:** qdb, **Timestamp:** October 9, 2012 9:4		
QDB0	8%	7%
iuddi.hpp [1 function covered. Results may be incomplete.]	can not be determined	not executed
qdb0.cpp	14%	18%
callJava	not executed	not executed
dispHelp	not executed	not executed
QDB0	16%	19%
viewDump	not executed	not executed
qdb0bp.cpp	not executed	not executed
qdb0er.cpp	not executed	not executed
qdb0ff.cpp	not executed	not executed
qdb0go.cpp	41%	51%
qdb0lk.cpp	not executed	not executed
qdb0ms.cpp	not executed	not executed
qdb0sk.cpp	not executed	not executed
qdb0sy.cpp	not executed	not executed
qdb0th.cpp	not executed	not executed
qdb0xp.cpp	30%	29%
string.h [2 functions covered. Results may be incomplete.]	can not be determined	2%
tpfregs.h [3 functions covered. Results may be incomplete.]	can not be determined	not executed
QDB2	23%	87%
QDB3	3%	100%
QDBA	not executed	not executed
QDBB	not executed	not executed

TPFXA1.hce | dialin.hce | qdb0.cpp

```
/* display help manual if parser error                          */
if (num_parms < 1)
{
    dispHelp();
}

/*****************************************************************/
/*                  */
/*                    */
/*****************************************************************/

 for ( i = 1, cur_parm = &parse_results; i <= num_parms;
        i++ , cur_parm = cur_parm->IPRSE_next )
 {
    if(strcmp(cur_parm->IPRSE_parameter,"ERR-d++")==0)
    {
        type = QDBGDRV_ERR;
        testcase=atoi(cur_parm->IPRSE_value);
        ecbptr()->ebw000 = testcase;
        ecbptr()->ebw001 = type;
        continue;
    } /* end of error */
    else if(strcmp(cur_parm->IPRSE_parameter,"GO-d++")==0)
    {
        type = QDBGDRV_GO;
        testcase=atoi(cur_parm->IPRSE_value);
```

Code Coverage Tool

- New Comparison feature to quickly see the differences between runs.
- Comparison feature includes an editor comparison tool to see the source analysis differences in a source file.

TPFXA1.hce | Compare Results

Compare code coverage results

	Size 1	Size 2
qdb0th.cpp	n/a	n/a
qdb0xp.cpp	6	+40
array_test()	n/a	+81
class_test()	n/a	+78
DataLevelFill_test()	n/a	n/a
expCases	26	+1
FillAndHold_test()	n/a	n/a
scalar_test()	n/a	n/a
struct_test()	n/a	+84
string.h	n/a	n/a
tpfregs.h	n/a	n/a
QDB2	23	0

	Line 1	Line 2
qdb0th.cpp	n/a	n/a
qdb0xp.cpp	5	+41
array_test()	n/a	+100
class_test()	n/a	+100
DataLevelFill_test()	n/a	n/a
expCases	21	+10
FillAndHold_test()	n/a	n/a
scalar_test()	n/a	n/a
struct_test()	n/a	+100
string.h	2	0
tpfregs.h	n/a	n/a
QDB2	87	0

Host: 9.57.13.89, **Session:** qdb, **Timestamp:** October 16, 2012 2:05:52 PM
Host: 9.57.13.89, **Session:** qdb, **Timestamp:** October 16, 2012 2:37:00 PM

Compare

Compare ('\\LINUXTPF.POK.IBM.COM\home\jwisnie\links\dbgdrvcom\qdb0xp.cpp' - '\\LINUXTPF.POK.IBM.COM\home\jwisnie...

C Compare (No Structural Differences)

Code Coverage Source Analysis Compare

```
440
441   QDB0_printf("executing debugger EX
442
443   switch(caseNum)
444   {
445      case 1:
446         QDB2(&reg);
447         break;
448
449      case 2:
450         array_test();
451         break;
452
453      case 3:
454         struct_test();
455         break;
456
457      case 4:
458         class_test();
459         break;
460
461      case 5:
462         scalar_test();
463         break;
464
465      case 6:
466         QDBDP1F3(type,caseNum);
```

```
440
441   QDB0_printf("executing debugge
442
443   switch(caseNum)
444   {
445      case 1:
446         QDB2(&reg);
447         break;
448
449      case 2:
450         array_test();
451         break;
452
453      case 3:
454         struct_test();
455         break;
456
457      case 4:
458         class_test();
459         break;
460
461      case 5:
462         scalar_test();
463         break;
464
465      case 6:
466         QDBDP1F3(type,caseNum);
```

Auto detect workstation IP

- The debugger attempts to automatically detect the workstation IP address and correct situations where the user has specified the workstation IP address incorrectly.
- Demo viewlet available at www.ibm.com/support/docview.wss?uid=swg21570467

Improve debugger connection failure handling

- On connection failures, the debugger attempts to notify the TPF Toolkit that registered the debugger of the debugger connection failure.
- ADB01 dumps indicate a connection failure occurred. Some ADB01 dumps did not include any information that was helpful in debugging the cause of the dump. As such, these ADB01 dumps were converted to WTOPC messages with additional information to aid in the diagnosis of the problem.

z/TPF Debugger Deliverable Details

Description	z/TPF APAR	z/TPF PUT Level	TPF Toolkit Level	TPFUG Requirement
CDB0 Interface	PJ39617	PUT9	None	Customer Request
Remote Debug Info Name			None	Customer Request
User Exit				
Dump Viewer Data Scrubbing			None	Customer Request
User Exit				
Show Loadset Name			None	Customer Request
Show Code Optimization Level			None	Customer Request
Prevent ZINET ADD S-DEBUG			None	Customer Request
Improve Remote Debug Info			V3.6.4	Customer Request
Memory Search			V3.6.4	V09114F

z/TPF Debugger Deliverable Details

Description	z/TPF APAR	z/TPF PUT Level	TPF Toolkit Level	TPFUG Requirement
Fork Interface Enhancement	PJ40255	PUT9	V.next	Customer Request
Code Coverage Tool:				
Size Analysis	PJ37973	PUT8	V3.6	
Source Analysis	PJ38995		V3.6.3	
Comparison Feature			V.next	
Auto detect workstation IP Improve debugger connection failure handling	PJ38995	PUT8	V3.6.3	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.**

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.**