TPF Toolkit Enhancements

2023 TPF Users Group Conference
April 24-26, Dallas, TX
Development Tools Subcommittee

—

Gerry Strait

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.

Personas

Jeremy

QA Manager

My role is ensuring product quality.

Christine

Security Admin

My role includes ensuring adherence to business security requirements.

Arthur

Toolkit Administrator

My role is to make TPF Toolkit available for install, get fix packs from IBM, create customizations, assist users with problems, maintain the Toolkit web application, etc…

Zach

Toolkit User

My role is TPF development.
That includes writing and debugging code.
That also includes developing automated test framework testcases, running code coverage, etc…

What’s New in TPF Toolkit Fix Packs since 2022 TPFUG:

4.6.14 – Released 04/07/22

4.6.15 – Released 08/29/22

4.6.16 – Released 03/31/23

Summary of Enhancements

- **For the Toolkit Administrator**
  Custom Menu Sample and the Problems View
  Pushing Preferences to Users

- **Security**
  Secure HTTP

- **Support for Automated Test Framework on TPF**
  Automated Test Framework View
  Load, Activate and Test

- **For the Developer**
  Debugger Null Pointer/Address Detection
  Run C/C++ Code Analysis
  DFDL Validation
  Remote File Search
  Trace Log Collection
  TPF LPEX Editor

For the Toolkit Administrator

Custom Menu Sample and the Problems View

The **Custom Menu Sample plug-in project,** included with each Fix Pack, demonstrates how to develop a custom menu item on a TPF project that executes a remote maketpf command on the linux build machine.

For example: Right-click a source file and execute a maketpf command: **maketpf –f cdco.c**

Pain Point: When the developer clicks the custom menu item, any error/warning/info messages that are produced appear in the Console view as text only.

Solution: New **MaketpfErrorParser** class
The administrator uses the MaketpfErrorParser in custom menu handlers to automatically add markers to the Problems View for any command response lines containing error/warning/info messages.

Value: Better productivity.
Developers can easily understand source code problems by looking at the Problems View and from the view jump to a source file to begin fixing problems.

Delivered in Fix Pack 4.6.16

Learn more… TPF Toolkit Knowledge Center
Developing custom menu items that run commands

Custom Menu Sample and the Problems View

Arthur

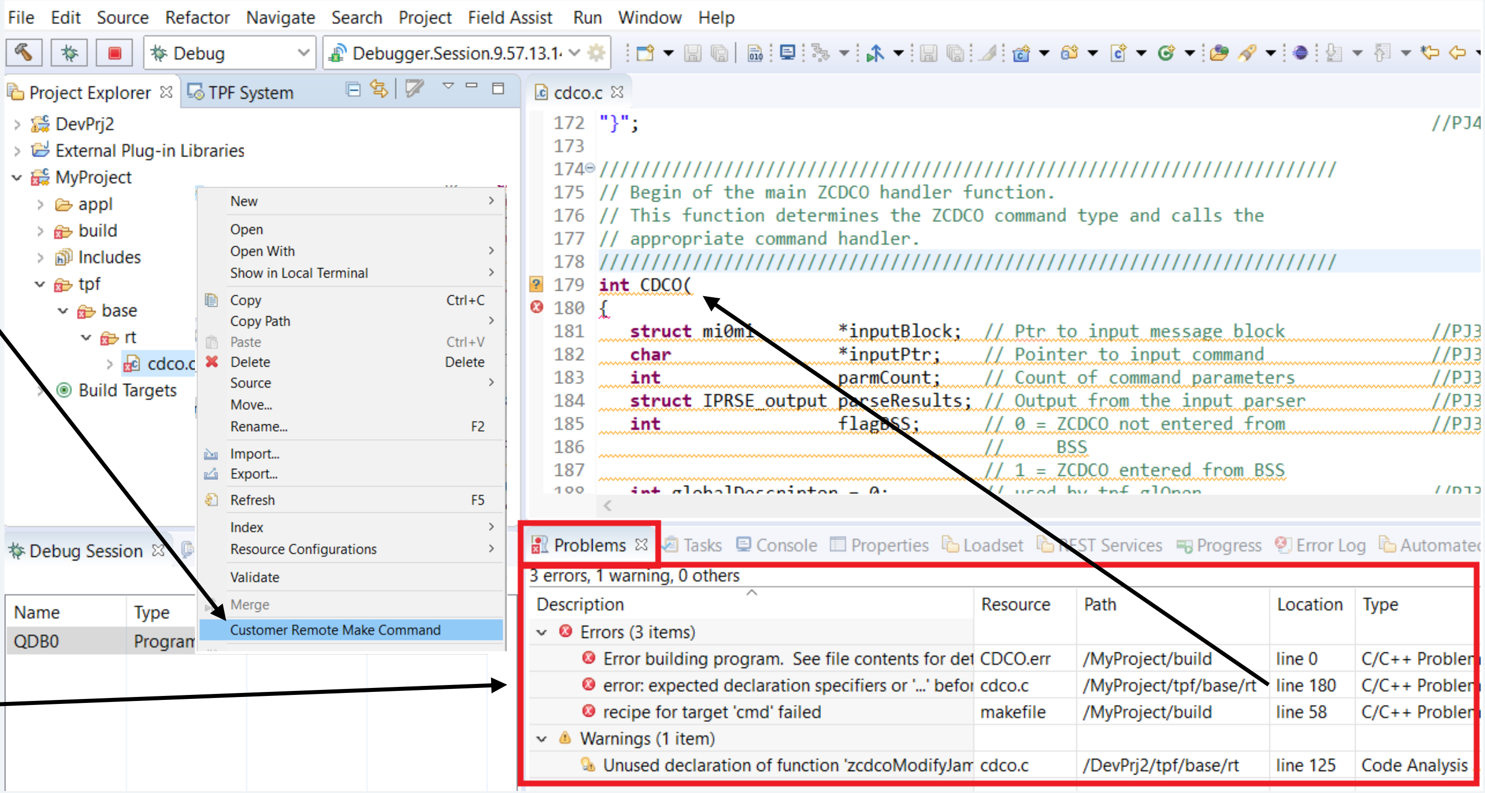
Toolkit Administrator

Delivers the custom menu

Zach

Toolkit User

Quickly locates and fixes errors in his code



Name	Type
QDB0	Program

```
"}";                                                                  //PJ4

//////////////////////////////////////////////////////////////////////
// Begin of the main ZCDCO handler function.
// This function determines the ZCDCO command type and calls the
// appropriate command handler.
//////////////////////////////////////////////////////////////////////
int CDCO(
{
   struct mi0mi        *inputBlock;   // Ptr to input message block      //PJ3
   char                *inputPtr;     // Pointer to input command        //PJ3
   int                 parmCount;     // Count of command parameters     //PJ3
   struct IPRSE_output parseResults;  // Output from the input parser    //PJ3
   int                 flagBSS;       // 0 = ZCDCO not entered from      //PJ3
                                      //     BSS
                                      // 1 = ZCDCO entered from BSS
   int globalDescriptor = 0;          // used by tpf_glOpen              //PJ3
```

3 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
Errors (3 items)				
Error building program. See file contents for det	CDCO.err	/MyProject/build	line 0	C/C++ Problem
error: expected declaration specifiers or '...' befor	cdco.c	/MyProject/tpf/base/rt	line 180	C/C++ Problem
recipe for target 'cmd' failed	makefile	/MyProject/build	line 58	C/C++ Problem
Warnings (1 item)				
Unused declaration of function 'zcdcoModifyJam	cdco.c	/DevPrj2/tpf/base/rt	line 125	Code Analysis

Pushing Preferences to Users

Toolkit behavior can be customized by using **Window>Preferences**

Pain Point: For some preferences, administrators want all users to have the same preference setting. But directing users to manually set certain preferences is not a reliable solution.

Solution: Eclipse Installer Preference Recorder
The administrator sets preferences while the recorder is active. The recorder saves the preferences into the Eclipse Installer .setup file accessed by all Toolkit installations. The next time users restart Toolkit, those recorded preferences will be set in the users Toolkit.
** If you install Toolkit from the product archive then this option is not available.*

Value: The administrator sets base Toolkit behavior.
Consistent Toolkit behavior across the organization.
Less hunting for preferences by users.

***Has been available since 4.6 was released!**

Learn more... Toolkit Knowledge Center
[Setting up default preferences]

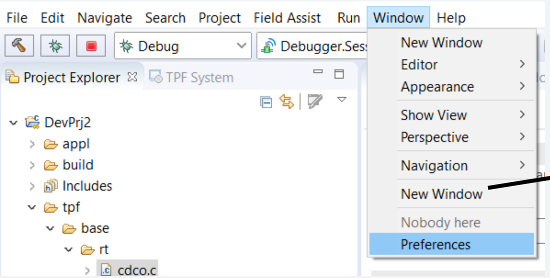
Pushing Preferences to Users

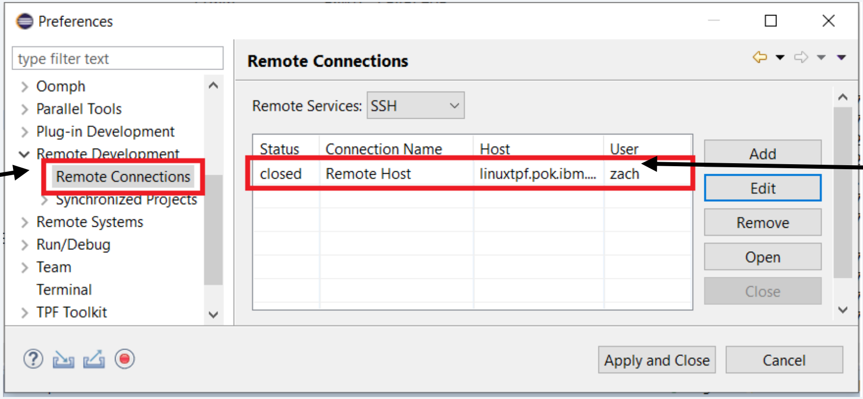
Example 1 – Remote Connection

Push the Remote Connection preference to your linux build machine that will be used by all your users.

Value: All users will by default have a common Remote Connection definition.
The administrator completes Advanced settings that may be required depending on your linux system.
Users will only have to supply their login credentials.

Learn more… TPF Toolkit Knowledge Center
[Configuring a remote connection to Linux]()





Remote Services: SSH

Status	Connection Name	Host	User
closed	Remote Host	linuxtpf.pok.ibm....	zach

Zach
Toolkit User
Enters login credentials

Pushing Preferences to Users

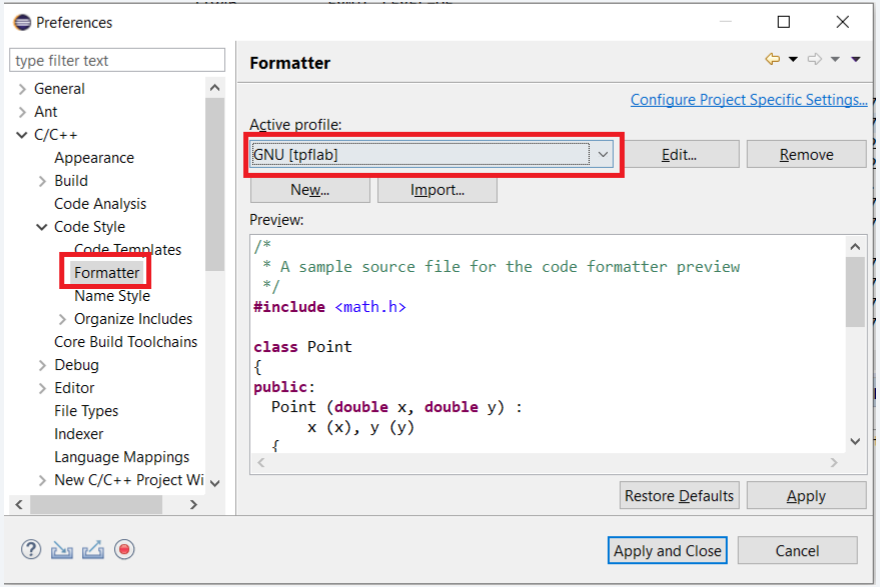
Example 2 – C/C++ Editor Formatting

Push C/C++ formatting preferences such as the use of spaces instead of tabs as formatting characters.

Value: By using spaces for formatting, C/C++ source files are more likely to exhibit consistent formatting in editors inside and outside Toolkit.

Delivered in Fix Pack 4.6.16 (documentation only)

Learn more... TPF Toolkit Knowledge Center
[Configuring C/C++ editor formatting]()



Active profile: GNU [tpflab]

Preview:

```
/*
 * A sample source file for the code formatter preview
 */
#include <math.h>

class Point
{
public:
  Point (double x, double y) :
      x (x), y (y)
  {
```

Security

Securing HTTP Connections – Between Toolkit and TPF

Problem: The use of unsecure protocols is becoming increasingly difficult, if not impossible, to justify.

Solution: New **TPF Connections** settings in the Toolkit preference page.
Secure HTTP using Transport Layer Security (TLS).

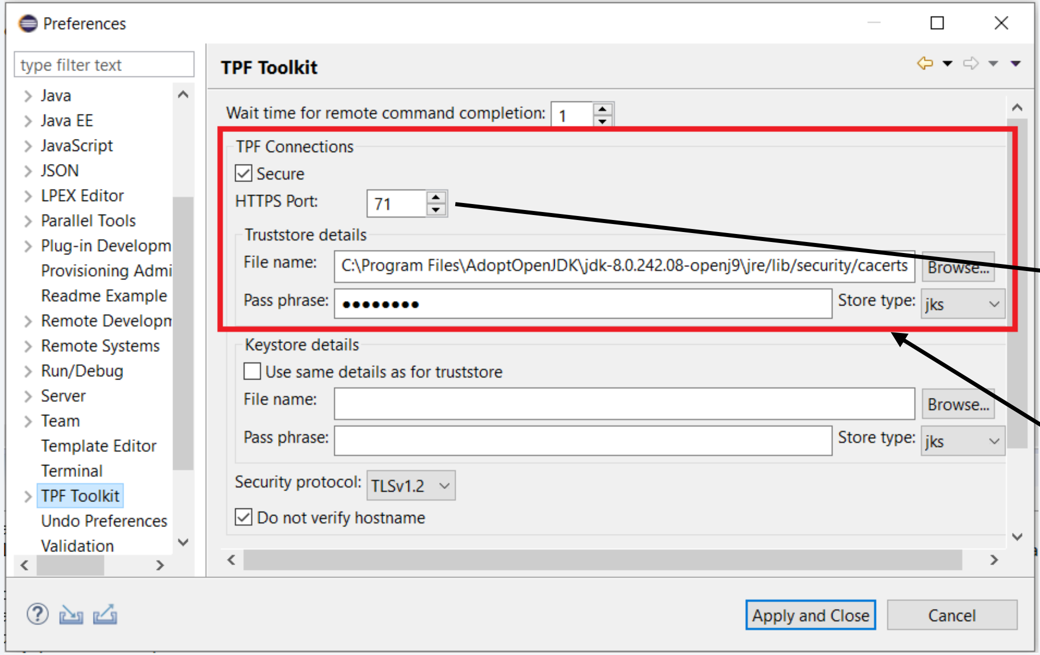
Value: Secure HTTP connections of course!

Delivered in Fix Pack 4.6.15

Learn more… Toolkit Knowledge Center
Enabling TLS security for z/TPF REST connections
TPF System view

Securing HTTP Connections – Between Toolkit and TPF

Complete the TPF Connections fields on the TPF Toolkit preference page.



TPF Toolkit

Wait time for remote command completion: 1

TPF Connections

☑ Secure

HTTPS Port: 71

Truststore details

File name: C:\Program Files\AdoptOpenJDK\jdk-8.0.242.08-openj9\jre/lib/security/cacerts

Pass phrase: ●●●●●●●● Store type: jks

Keystore details

□ Use same details as for truststore

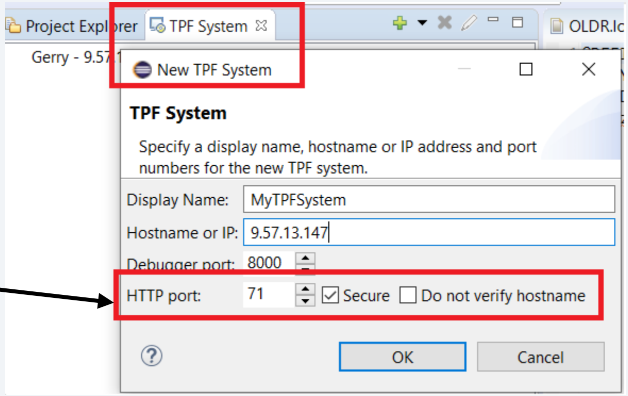
File name:

Pass phrase: Store type: jks

Security protocol: TLSv1.2

☑ Do not verify hostname

Secure HTTP becomes default when defining a TPF system.



TPF System

Specify a display name, hostname or IP address and port numbers for the new TPF system.

Display Name: MyTPFSystem

Hostname or IP: 9.57.13.147

Debugger port: 8000

HTTP port: 71 ☑ Secure □ Do not verify hostname

A potential candidate for pushing of preferences. The Toolkit administrator can dictate that all are using secure HTTP!

Arthur
Toolkit Administrator

Securing HTTP Connections – Between TPF and Linux

Problem: The use of unsecure protocols is becoming increasingly difficult, if not impossible, to justify.

Solution: Secure HTTP using Transport Layer Security (TLS)
Secure z/TPF REST connections for:

- Remote formatting of code coverage results
- Debugger:
  - Automatic source lookup
  - Remote debug information

Value: Secure HTTP connections of course!

Delivered in Fix Pack 4.6.15 (TPFToolkitServices.war uploaded 2/22/23)
APAR PJ46903 Available 02/20/23

Learn more... Blog

Support for Automated Test Framework on TPF

Automated Test Framework View

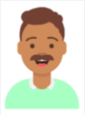
The z/TPF automated test framework provides a way for you to test new or changed software automatically. Compared to manual testing, automated testing improves code quality and reduces the testing cost and time.

Paint Point: I want to run automated test framework testcases on TPF from Toolkit.

Solution: Automated Test Framework View

Value: Better productivity
- No need to leave Toolkit to run testcases and review results
- Easy to find, select and run testcases relevant to my application
  - More control over what testcases are run
  - Easy to re-run a particular set of testcases
- Develop a "test plan" file
  - Import/Export selected testcase in JSON format
  - Potentially used in DevOps pipeline
  - Save for reuse later or as a historical document
- Short term history of testcase runs

Zach
Toolkit User
Uses Automated Testing on TPF



Jeremy
QA Manager
Code has been tested before release

Delivered in Fix Packs 4.6.14 and 4.6.15

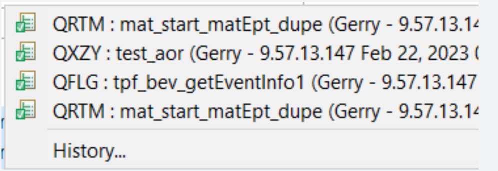
Learn more... TPF Toolkit Knowledge Center [Automated Test Framework view]

Automated Test Framework View

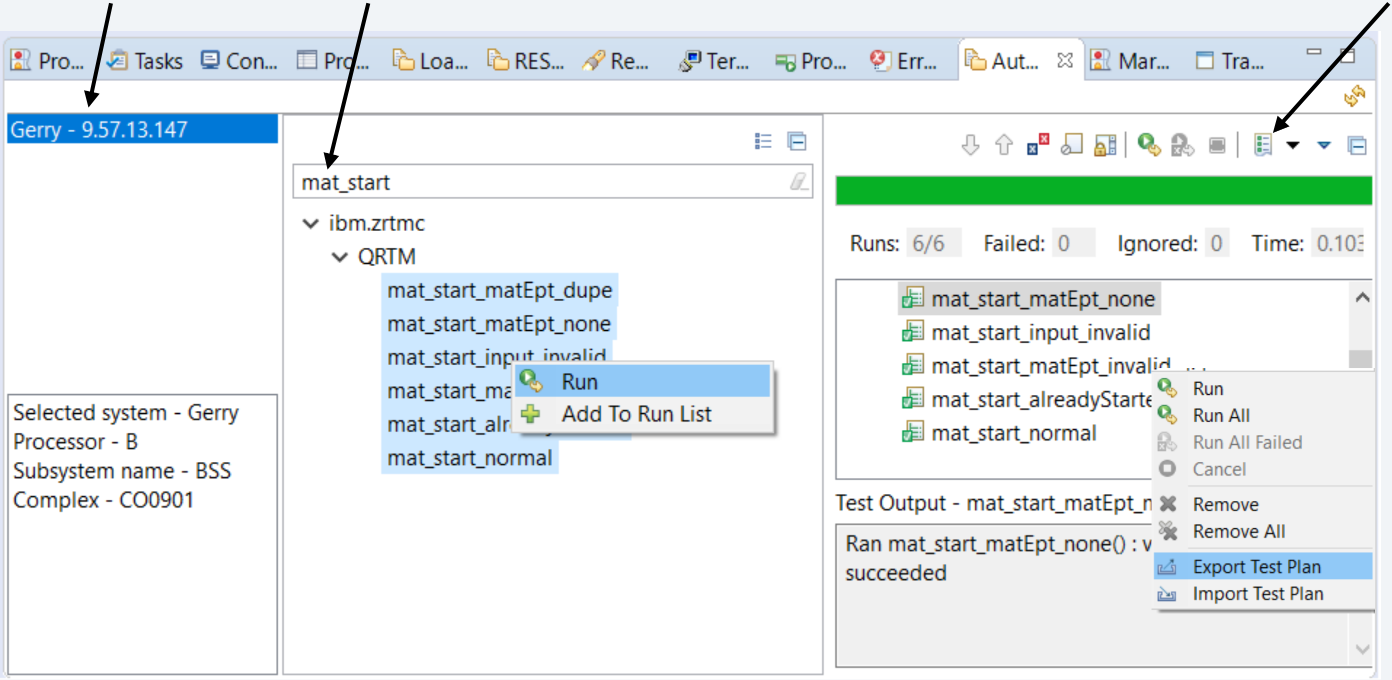
TPF System

Filter

Short Term History



QRTM : mat_start_matEpt_dupe (Gerry - 9.57.13.14
QXZY : test_aor (Gerry - 9.57.13.147 Feb 22, 2023 (
QFLG : tpf_bev_getEventInfo1 (Gerry - 9.57.13.147
QRTM : mat_start_matEpt_dupe (Gerry - 9.57.13.14



Gerry - 9.57.13.147

Selected system - Gerry
Processor - B
Subsystem name - BSS
Complex - CO0901

mat_start

- ibm.zrtmc
  - QRTM
    - mat_start_matEpt_dupe
    - mat_start_matEpt_none
    - mat_start_input_invalid
    - mat_start_ma
    - mat_start_alr
    - mat_start_normal

Runs: 6/6 Failed: 0 Ignored: 0 Time: 0.103

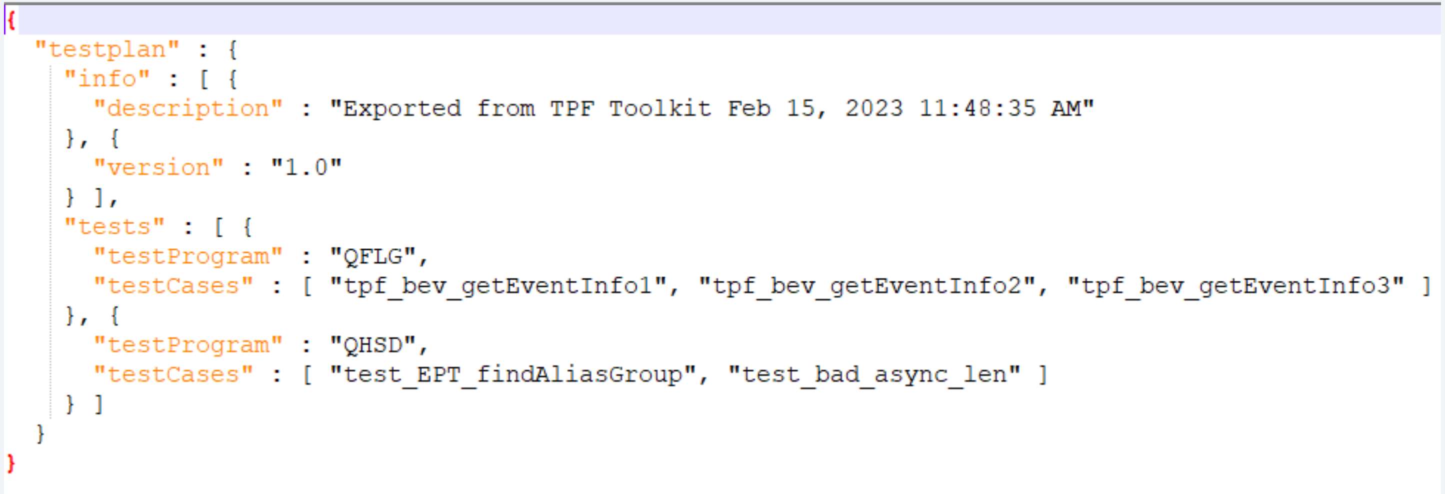
- mat_start_matEpt_none
- mat_start_input_invalid
- mat_start_matEpt_invalid
- mat_start_alreadyStarte
- mat_start_normal

Test Output - mat_start_matEpt_n

Ran mat_start_matEpt_none() : v
succeeded

Automated Test Framework View - Exported Test Plan File Sample

JSON format
Contains programs and testcases within programs



```
{
  "testplan" : {
    "info" : [ {
      "description" : "Exported from TPF Toolkit Feb 15, 2023 11:48:35 AM"
    }, {
      "version" : "1.0"
    } ],
    "tests" : [ {
      "testProgram" : "QFLG",
      "testCases" : [ "tpf_bev_getEventInfo1", "tpf_bev_getEventInfo2", "tpf_bev_getEventInfo3" ]
    }, {
      "testProgram" : "QHSD",
      "testCases" : [ "test_EPT_findAliasGroup", "test_bad_async_len" ]
    } ]
  }
}
```

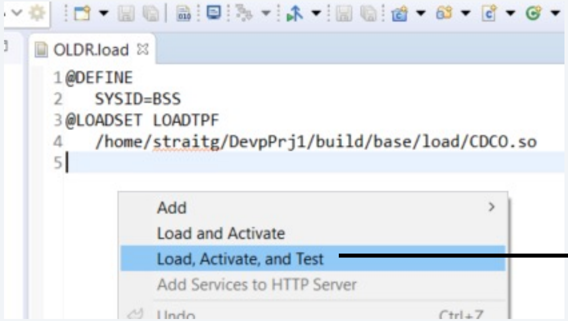
Load, Activate and Test

Toolkit provides a means to load and active programs in a single action.

Pain Point: I want to automatically run testcases when I load and activate programs on TPF.

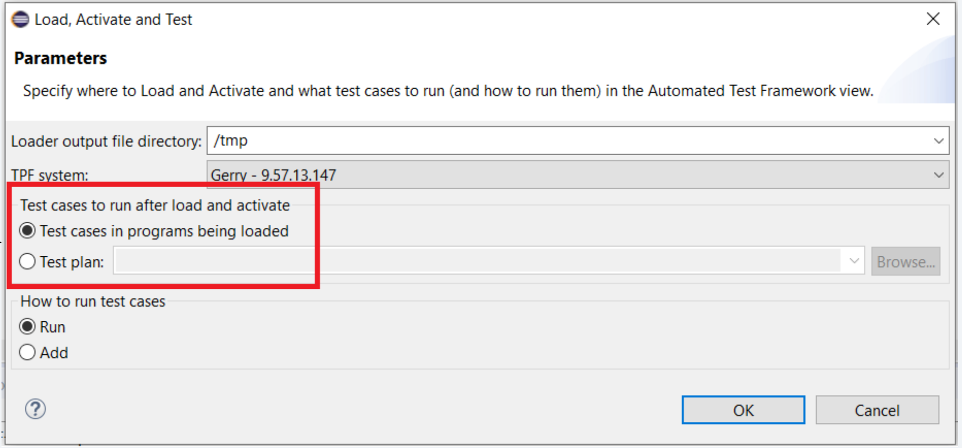
Solution: Add right-click **Load, Activate and Test** menu item on loader input file files. E.g. **.load** files.

Value: Fewer clicks to run and review testcase results.
Automatically run testcases when loading your application updates.
Run testcases for programs being loaded or run a “test plan” of select testcases.
Automatically switches to the Automated Test Framework view.



OLDR.load

```
@DEFINE
   SYSID=BSS
@LOADSET LOADTPF
   /home/straitg/DevpPrj1/build/base/load/CDCO.so
```



Load, Activate and Test

Parameters

Specify where to Load and Activate and what test cases to run (and how to run them) in the Automated Test Framework view.

Loader output file directory: /tmp

TPF system: Gerry - 9.57.13.147

Test cases to run after load and activate
- ● Test cases in programs being loaded
- ○ Test plan:

How to run test cases
- ● Run
- ○ Add

Delivered in Fix Pack 4.6.15

For the Developer

Debugger Null pointer and null address detection

Null pointers in C/C++ code and null address references in assembler code can cause run-time errors that can be difficult to understand and time consuming to fix.

Paint Point: Toolkit debugger does not help identify null pointer and null address references in code.

Solutions:
- New **Stop at null pointers and addresses** breakpoint.
- New **Log null pointer address detection events** preference.

Value: Much faster identification of the problem code and problem resolution.

Delivered in Fix Pack 4.6.14

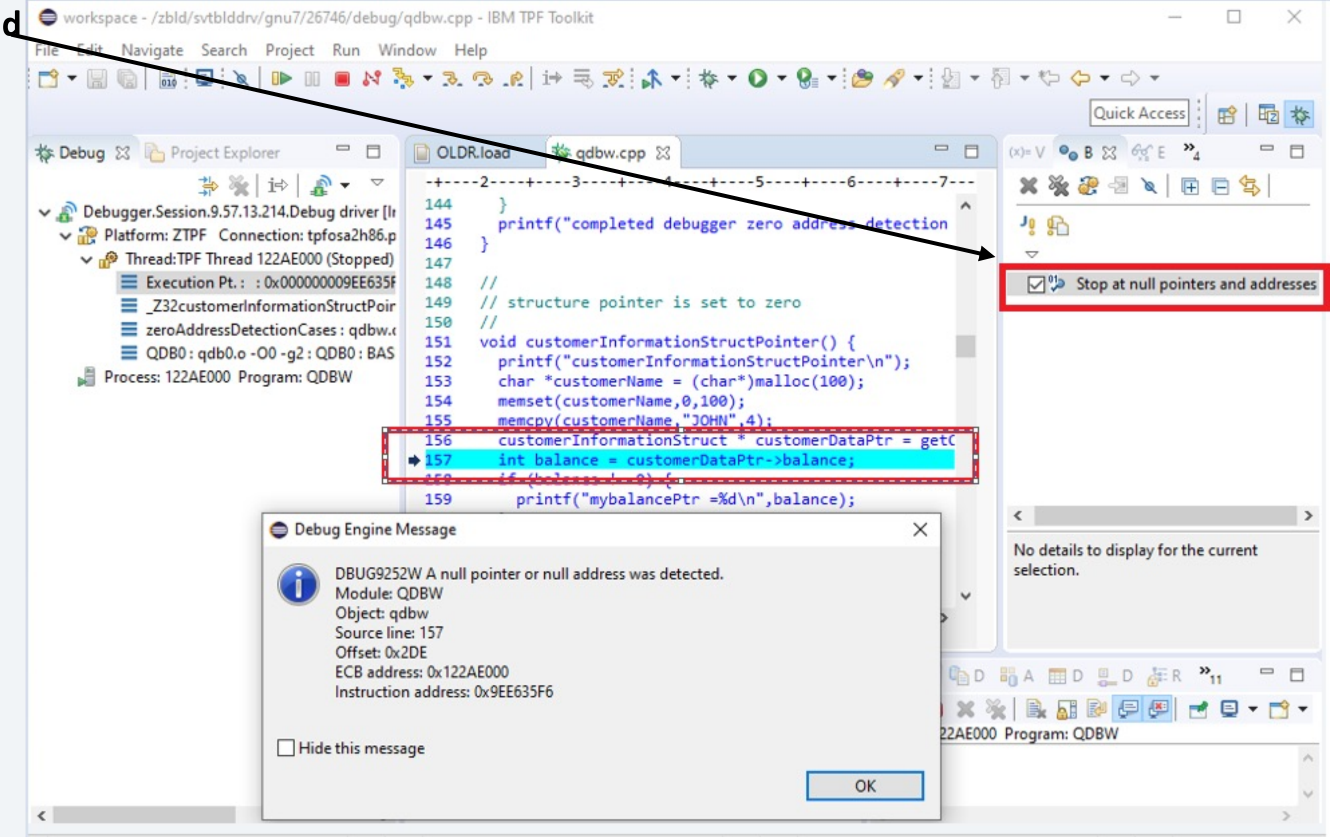
Learn more... Toolkit Knowledge Center
Null pointer and null address detection

Null pointer and null address detection - Example

Zach set a **Stop at null pointers and address** breakpoint.

If the debugger detects a null, then a pop-up box displays information about the event.

Zach can then update source code to take corrective action.



workspace - /zbld/svtblddrv/gnu7/26746/debug/qdbw.cpp - IBM TPF Toolkit
File Edit Navigate Search Project Run Window Help
Quick Access
Debug
Project Explorer
OLDR.load
qdbw.cpp
(x)= V
B
E

```
Debugger.Session.9.57.13.214.Debug driver [Ir
  Platform: ZTPF   Connection: tpfosa2h86.p
    Thread:TPF Thread 122AE000 (Stopped)
      Execution Pt. :   : 0x000000009EE635F
      _Z32customerInformationStructPoir
      zeroAddressDetectionCases : qdbw.c
      QDB0 : qdb0.o -O0 -g2 : QDB0 : BAS
  Process: 122AE000  Program: QDBW
```

```
-+----2----+----3----+----4----+----5----+----6----+----7---
144     }
145     printf("completed debugger zero address detection
146   }
147
148   //
149   // structure pointer is set to zero
150   //
151   void customerInformationStructPointer() {
152     printf("customerInformationStructPointer\n");
153     char *customerName = (char*)malloc(100);
154     memset(customerName,0,100);
155     memcpy(customerName,"JOHN",4);
156     customerInformationStruct * customerDataPtr = getC
157     int balance = customerDataPtr->balance;
158     if (balance != 0) {
159       printf("mybalancePtr =%d\n",balance);
```

☑ Stop at null pointers and addresses

No details to display for the current selection.

Debug Engine Message

DBUG9252W A null pointer or null address was detected.
Module: QDBW
Object: qdbw
Source line: 157
Offset: 0x2DE
ECB address: 0x122AE000
Instruction address: 0x9EE635F6

☐ Hide this message

OK

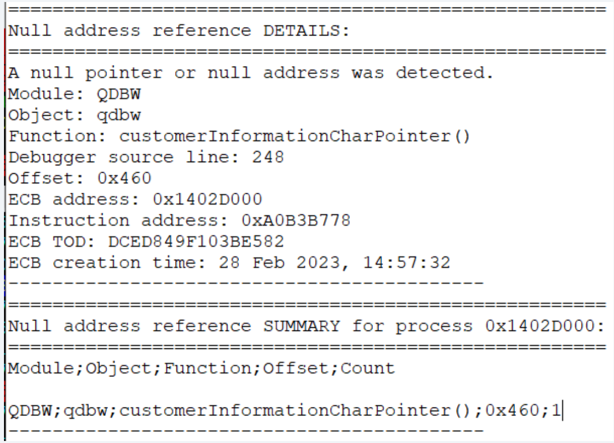
22AE000 Program: QDBW

Null pointer and null address detection – logging

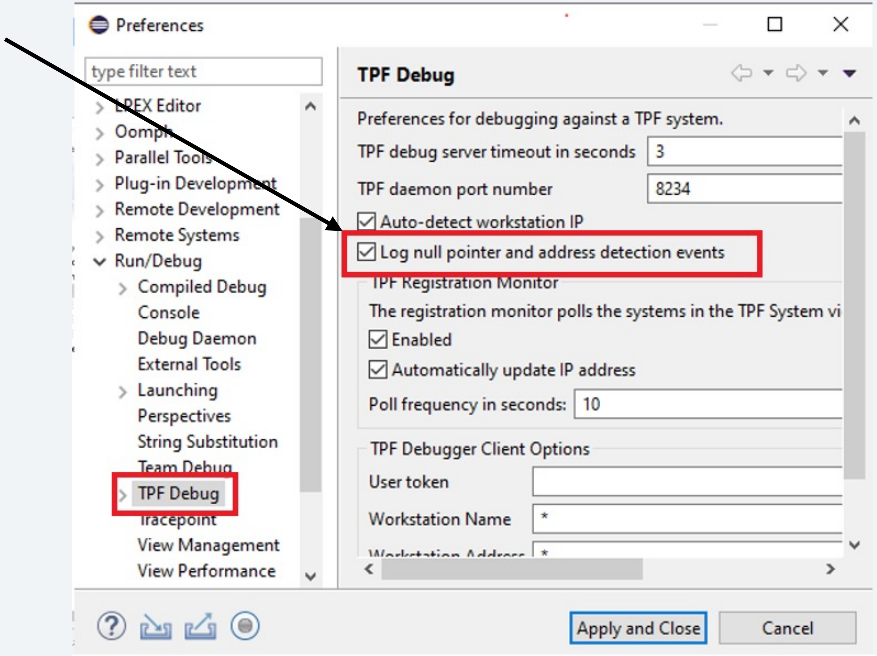
Zach sets the **Log null pointer address detection events** preference.

Zach can then debug his application, review the log file and update code to avoid or handle null pointers.

Sample log file entry



```
======================================================
Null address reference DETAILS:
======================================================
A null pointer or null address was detected.
Module: QDBW
Object: qdbw
Function: customerInformationCharPointer()
Debugger source line: 248
Offset: 0x460
ECB address: 0x1402D000
Instruction address: 0xA0B3B778
ECB TOD: DCED849F103BE582
ECB creation time: 28 Feb 2023, 14:57:32
-------------------------------------------

======================================================
Null address reference SUMMARY for process 0x1402D000:
======================================================
Module;Object;Function;Offset;Count

QDBW;qdbw;customerInformationCharPointer();0x460;1
-------------------------------------------
```



TPF Debug

Preferences for debugging against a TPF system.

TPF debug server timeout in seconds: 3

TPF daemon port number: 8234

☑ Auto-detect workstation IP

☑ Log null pointer and address detection events

TPF Registration Monitor

The registration monitor polls the systems in the TPF System vi

☑ Enabled

☑ Automatically update IP address

Poll frequency in seconds: 10

TPF Debugger Client Options

User token:

Workstation Name: *

Workstation Address: *

Run C/C++ Code Analysis (previously known as Generate Include Paths)

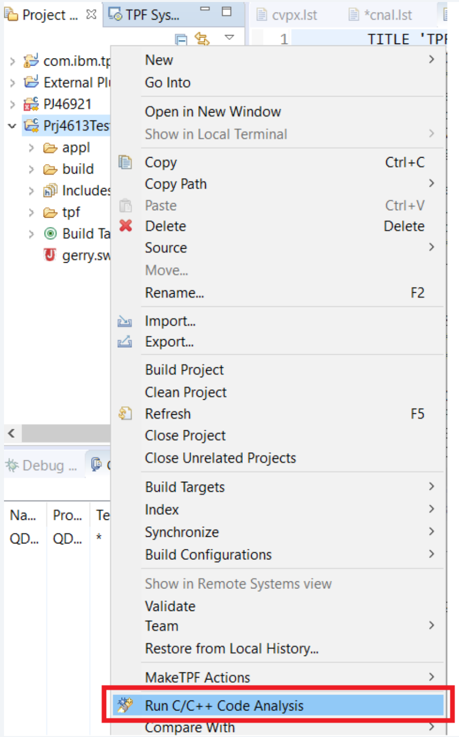
Pain Points:
- Not valid error markers when I edit C/C++ files in Toolkit.
- Toolkit doesn’t provide C/C++ editor capabilities like navigation and static analysis.

Solution: Override the right-click **Run C/C++ Code Analysis** menu item.
** Run C/C++ Code Analysis fixes some reliability issues with the former Maketpf Actions>Generate Include Paths*

Value: By running Run C/C++ Code Analysis, only valid markers are displayed and the following capabilities are enabled:
- **navigation** – jump to locations in code, header files, ect.
- **content assist** – suggestions to complete your typing
- **static analysis** – show errors as you type

Delivered in Fix Pack 4.6.15

Learn more... IBM Media Center Videos
[Run C/C++ Code Analysis]
[Resource Markers]

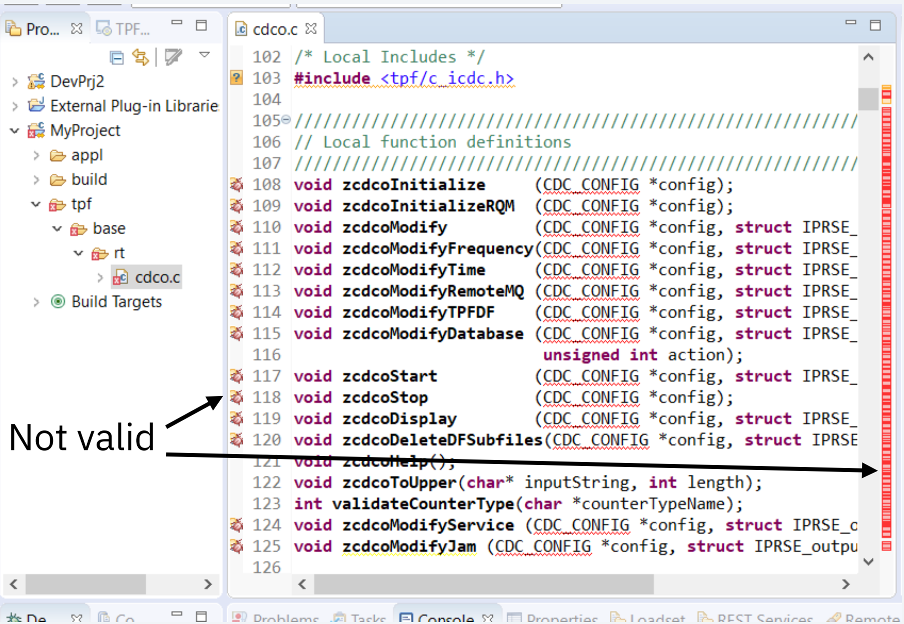


Run C/C++ Code Analysis (previously known as Generate Include Paths)

Before

Not valid markers

No CDT capabilities functional



```
/* Local Includes */
#include <tpf/c_icdc.h>

////////////////////////////////////////////////////////////////
// Local function definitions
////////////////////////////////////////////////////////////////
void zcdcoInitialize      (CDC_CONFIG *config);
void zcdcoInitializeRQM   (CDC_CONFIG *config);
void zcdcoModify          (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyFrequency(CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyTime      (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyRemoteMQ  (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyTPFDF     (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyDatabase  (CDC_CONFIG *config, struct IPRSE_
                           unsigned int action);
void zcdcoStart           (CDC_CONFIG *config, struct IPRSE_
void zcdcoStop            (CDC_CONFIG *config);
void zcdcoDisplay         (CDC_CONFIG *config, struct IPRSE_
void zcdcoDeleteDFSubfiles(CDC_CONFIG *config, struct IPRSE
void zcdcoHelp();
void zcdcoToUpper(char* inputString, int length);
int validateCounterType(char *counterTypeName);
void zcdcoModifyService (CDC_CONFIG *config, struct IPRSE_c
void zcdcoModifyJam (CDC_CONFIG *config, struct IPRSE_outpu
```

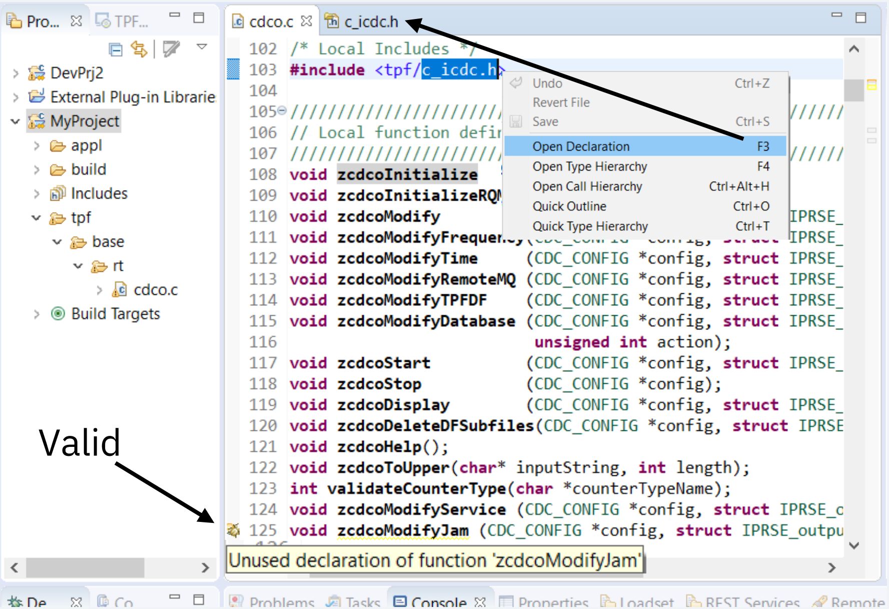
Not valid

Note: CDT is Eclipse C Development Tools

After

Only valid markers

CDT capabilities functional



```
/* Local Includes */
#include <tpf/c_icdc.h>

//////////////////////
// Local function defir
//////////////////////
void zcdcoInitialize
void zcdcoInitializeRQM
void zcdcoModify                                       IPRSE_
void zcdcoModifyFrequen                                IPRSE_
void zcdcoModifyTime      (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyRemoteMQ (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyTPFDF     (CDC_CONFIG *config, struct IPRSE_
void zcdcoModifyDatabase (CDC_CONFIG *config, struct IPRSE_
                           unsigned int action);
void zcdcoStart           (CDC_CONFIG *config, struct IPRSE_
void zcdcoStop            (CDC_CONFIG *config);
void zcdcoDisplay         (CDC_CONFIG *config, struct IPRSE_
void zcdcoDeleteDFSubfiles(CDC_CONFIG *config, struct IPRSE
void zcdcoHelp();
void zcdcoToUpper(char* inputString, int length);
int validateCounterType(char *counterTypeName);
void zcdcoModifyService (CDC_CONFIG *config, struct IPRSE_c
void zcdcoModifyJam (CDC_CONFIG *config, struct IPRSE_outpu
```

Unused declaration of function 'zcdcoModifyJam'

Valid

Note: Open Declaration will open a local read-only cache copy if the file is not in the project.

DFDL Validation

The z/TPF DFDL offline utility is an offline program that you can use to validate DFDL schema files. Previously, DFDL validation occurred when the DFDL schema was loaded to the z/TPF system. Processing stopped after the first error was encountered, and subsequent errors were not identified until the file was loaded to the system again.

Pain Point: Finding out about errors in my DFDL schema files after I load from Toolkit to TPF wastes time.

Solution: Add right-click **Validate** on DFDL schema files in a TPF Project.
Calls the DFDL offline utility on linux and display errors and warnings in an editor.

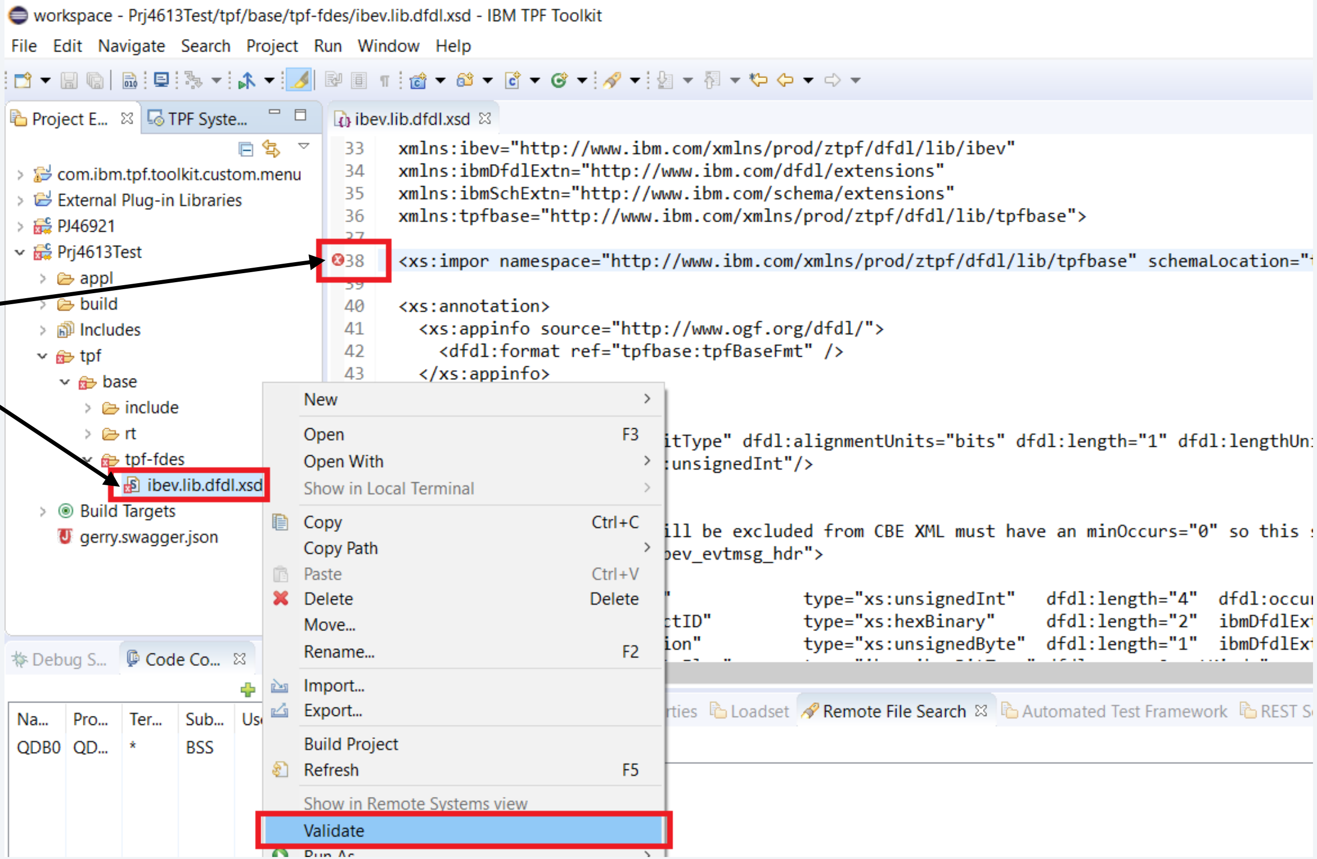
Value: Time savings.
Fixing errors before loading to TPF saves the developer time.

Delivered in Fix Pack 4.6.15

Learn more… TPF Toolkit Knowledge Center
Creating deployment descriptors in XML format
Requires: z/TPF DFDL offline utility (APAR PJ46509)

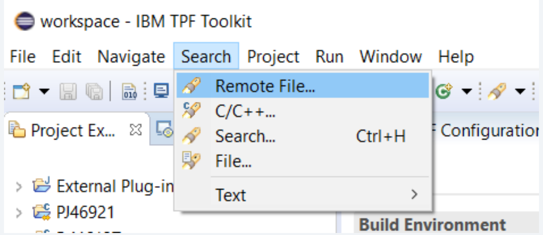
DFDL Validation

Error icons



```
xmlns:ibev="http://www.ibm.com/xmlns/prod/ztpf/dfdl/lib/ibev"
xmlns:ibmDfdlExtn="http://www.ibm.com/dfdl/extensions"
xmlns:ibmSchExtn="http://www.ibm.com/schema/extensions"
xmlns:tpfbase="http://www.ibm.com/xmlns/prod/ztpf/dfdl/lib/tpfbase">

<xs:impor namespace="http://www.ibm.com/xmlns/prod/ztpf/dfdl/lib/tpfbase" schemaLocation="

<xs:annotation>
  <xs:appinfo source="http://www.ogf.org/dfdl/">
    <dfdl:format ref="tpfbase:tpfBaseFmt" />
  </xs:appinfo>

itType" dfdl:alignmentUnits="bits" dfdl:length="1" dfdl:lengthUn
:unsignedInt"/>

ill be excluded from CBE XML must have an minOccurs="0" so this
bev_evtmsg_hdr">

"                type="xs:unsignedInt"    dfdl:length="4"   dfdl:occu
ctID"            type="xs:hexBinary"      dfdl:length="2"   ibmDfdlEx
ion"             type="xs:unsignedByte"   dfdl:length="1"   ibmDfdlEx
```

Remote File Search



Remote File Search is used to search for files or strings within files on your linux build machine.

Pain Point: Search can be slow <u>when search includes many subdirectories</u>.

Solution: We sped up search by excluding binary files and made search of listings optional.

Value: Better user experience.
Roughly an order of magnitude improvement. A search that might take ~30 seconds now takes ~3 seconds.

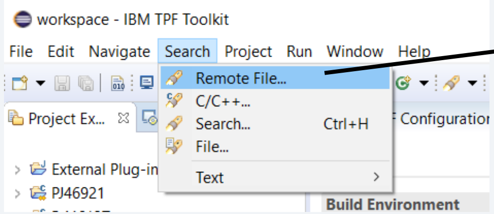
Pain Point: I have to manually select the Remote Connection to search after every Toolkit restart.

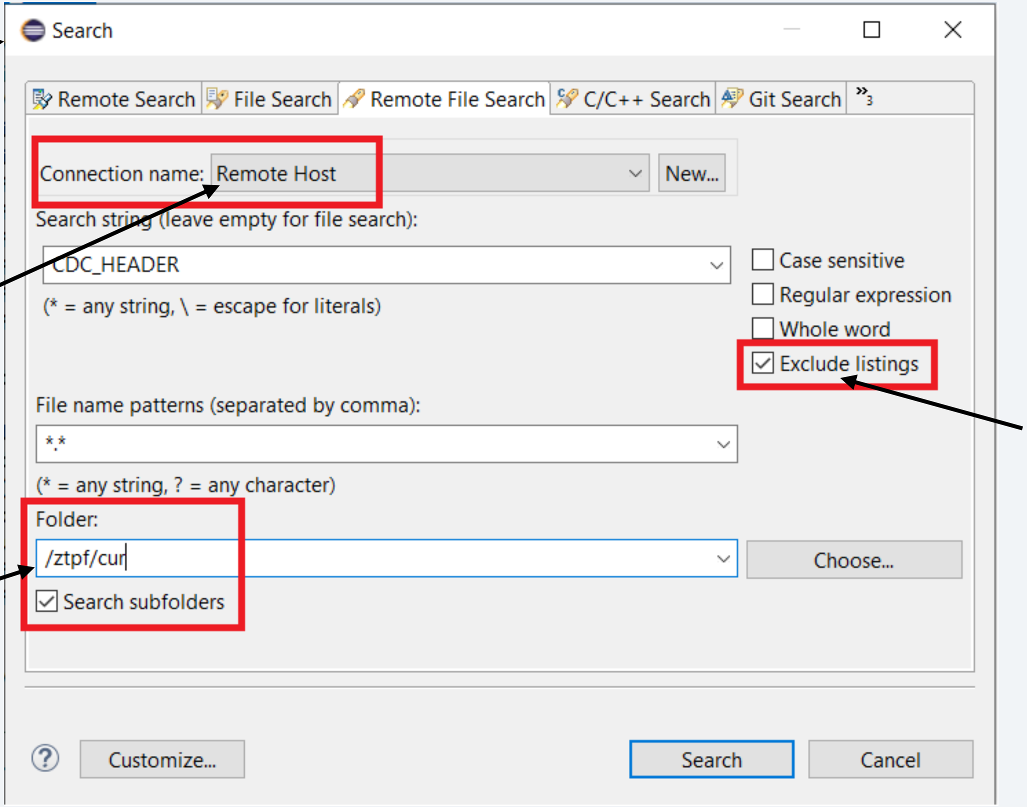
Solution: Autofill the Remote Connection to search.

Value: Better user experience. Fewer clicks.
In most cases the user will never have to select the Remote Connection to search.
The Remote Connection selection is restored across Toolkit sessions.

Delivered in Fix Pack 4.6.16

Remote File Search





Connection name: Remote Host

Search string (leave empty for file search):

CDC_HEADER

(* = any string, \ = escape for literals)

☐ Case sensitive

☐ Regular expression

☐ Whole word

☑ Exclude listings

File name patterns (separated by comma):

.

(* = any string, ? = any character)

Folder:

/ztpf/cur

☑ Search subfolders

Auto-selected

Default

The Folder to search contains a lot of subdirectories, some of which contain listings and binaries.

Trace Log Collection View

This view allows you to collect and analyze macro and function calls by an ECB being debugged. Data for the view is obtained from trace logging on TPF.

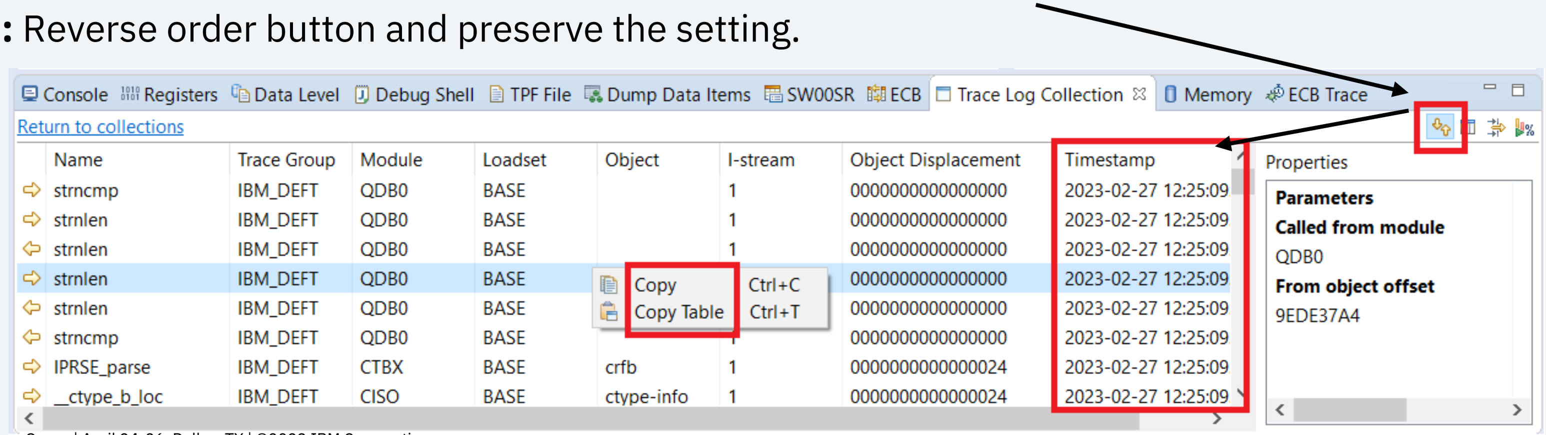
Pain Point: I want to work with the trace log table data outside Toolkit. But there is no way to do so.

Solution: Add right-click **Copy** and **Copy Table** menu items.   Copies to a CSV text file.

Value: Users can easily share and work with trace log data outside of Toolkit. For example, in a spreadsheet.

Pain Point: The table is in the opposite Timestamp order than I want.

Solution: Reverse order button and preserve the setting.



	Name	Trace Group	Module	Loadset	Object	I-stream	Object Displacement	Timestamp
⇨	strncmp	IBM_DEFT	QDB0	BASE		1	0000000000000000	2023-02-27 12:25:09
⇨	strnlen	IBM_DEFT	QDB0	BASE		1	0000000000000000	2023-02-27 12:25:09
⇦	strnlen	IBM_DEFT	QDB0	BASE		1	0000000000000000	2023-02-27 12:25:09
⇨	strnlen	IBM_DEFT	QDB0	BASE			0000000000000000	2023-02-27 12:25:09
⇦	strnlen	IBM_DEFT	QDB0	BASE			0000000000000000	2023-02-27 12:25:09
⇦	strncmp	IBM_DEFT	QDB0	BASE			0000000000000000	2023-02-27 12:25:09
⇨	IPRSE_parse	IBM_DEFT	CTBX	BASE	crfb	1	0000000000000024	2023-02-27 12:25:09
⇨	__ctype_b_loc	IBM_DEFT	CISO	BASE	ctype-info	1	0000000000000024	2023-02-27 12:25:09

Parameters

Called from module

QDB0

From object offset

9EDE37A4

Trace Log Collection View – Resource Counts

When a Toolkit user starts a trace log collection for an ECB during a debug session, counts of resources that the ECB uses are collected at various points in the trace log collection.

Pain Point: The view does not provide resource usage information included in trace logging.

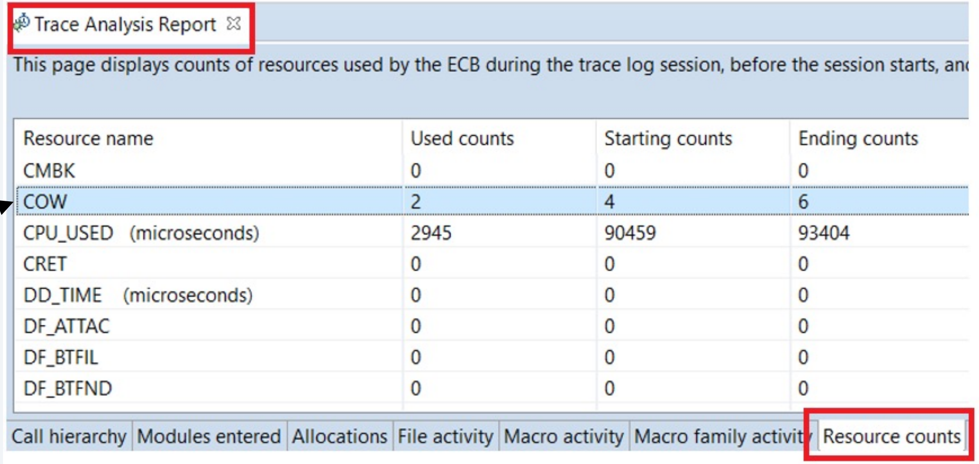
Solution: Add a new **Resource counts** tab in the existing Trace Analysis Report.
Counts values are at the start, at the end, and during the collection period.
Resource counts include the same information as runtime metrics collection database metrics.

Value: A more complete set of information is available to help understand your application.

Delivered in Fix Pack 4.6.16

Learn more… TPF Toolkit Knowledge Center
Trace log resource counts
Runtime metrics collection database metrics

For example, COW, is the number of copy-on-writes that have happened for the ECB.



This page displays counts of resources used by the ECB during the trace log session, before the session starts, an

Resource name	Used counts	Starting counts	Ending counts
CMBK	0	0	0
COW	2	4	6
CPU_USED (microseconds)	2945	90459	93404
CRET	0	0	0
DD_TIME (microseconds)	0	0	0
DF_ATTAC	0	0	0
DF_BTFIL	0	0	0
DF_BTFND	0	0	0

TPF LPEX Editor

The TPF LPEX editor is used to edit assembler files.

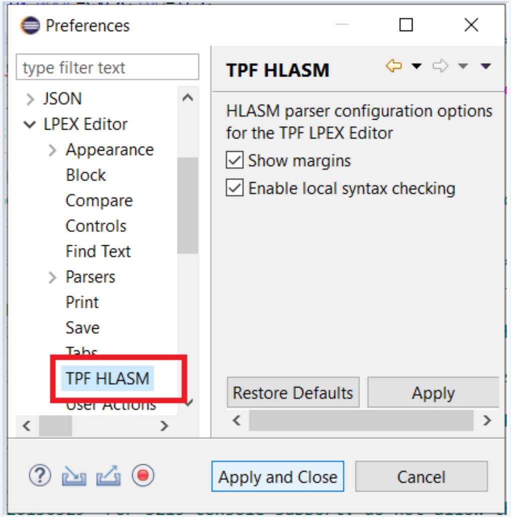
Pain Point: TPF LPEX editor features that I had in Toolkit 4.2 are not in Toolkit 4.6

Solutions: We brought forward TPF LPEX editor preferences into Toolkit 4.6
- Show margins
- HLASM parser enable/disable local syntax checking

Value: Better user experience

Delivered in Fix Pack 4.6.16

TPF LPEX Editor



TPF HLASM

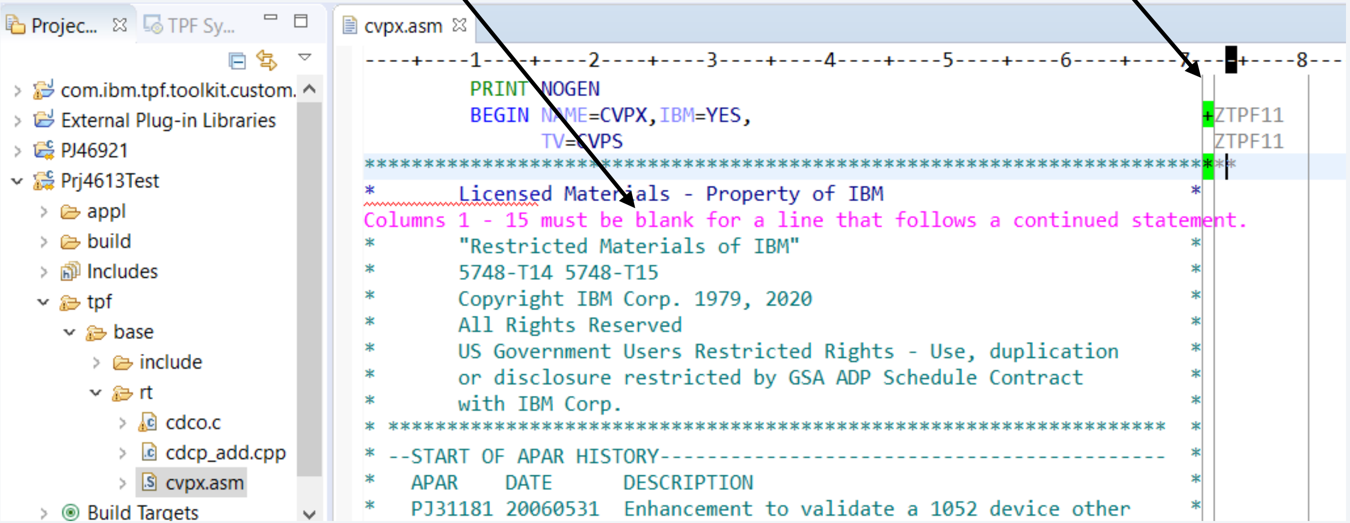
HLASM parser configuration options for the TPF LPEX Editor

☑ Show margins

☑ Enable local syntax checking

Syntax error shown in pink

Margins at columns 71, 72, 81



```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7-+-+----8---
         PRINT NOGEN
         BEGIN NAME=CVPX,IBM=YES,                                      +ZTPF11
               TV=CVPS                                                  ZTPF11
*************************************************************************
*        Licensed Materials - Property of IBM                          *
Columns 1 - 15 must be blank for a line that follows a continued statement.
*        "Restricted Materials of IBM"                                 *
*        5748-T14 5748-T15                                             *
*        Copyright IBM Corp. 1979, 2020                                *
*        All Rights Reserved                                           *
*        US Government Users Restricted Rights - Use, duplication      *
*        or disclosure restricted by GSA ADP Schedule Contract         *
*        with IBM Corp.                                                *
* ******************************************************************** *
* --START OF APAR HISTORY--------------------------------------------- *
*   APAR     DATE      DESCRIPTION                                     *
*   PJ31181 20060531   Enhancement to validate a 1052 device other     *
```

What’s Next In 2023?

- Upgrade Toolkit to Eclipse 4.23 and Java 17 – Target 3Q2023
- Secure All Toolkit Connections:
  Secure HTTP connections – Delivered 1Q2023
  Secure FTP connections to TPF – Target 3Q2023
  Secure Debugger connections to TPF – Target 4Q2023

Conclusion

- TPF Toolkit documentation
- IBM Media Center: TPF Toolkit demo videos
- Blog
- Ideas – We’re always open to ideas to improve Toolkit!

We want sponsor users!

Our development cycle is driven by your feedback.

We are looking for sponsor users to assist in design and implementation of new features, targeting the following personas:

- Application developer
- System programmer
- Test engineer

If you are interested in participating as a sponsor user, please contact:

Kishore Nagareddy (kishoren@us.ibm.com)

Thank you