

# TPF Toolkit Enhancements

2022 TPF Users Group Conference  
March 27-30, Dallas, TX  
Subcommittee

Development Tools Subcommittee

—

Kishore Nagareddy

# 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.

# Problem Statement

For a subset of customers, absence of some features from TPF Toolkit 4.2 is delaying the migration to TPF Toolkit 4.6. Support for TPF Toolkit 4.2 was extended.

We have delivered features to make it much easier to adopt TPF Toolkit 4.6. As a result, support for TPF Toolkit 4.2 will end in June this year.

We are continuing to deliver new enhancements to ease the migration process, provide a better user experience and enhance the productivity of several personas such as an application developer, system programmer and a test engineer.

# Users



Zach

*application developer*



Sophie

*system programmer*



Andrew

*new hire application developer*



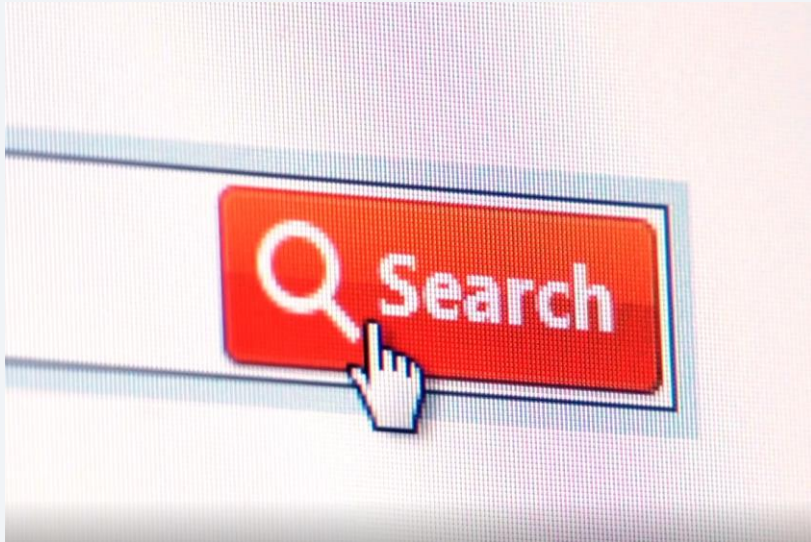
Lawrence

*test engineer*

# Problems

1. Difficult to search for files and text on the remote build system
2. Unable to use modern C/C++ editor capabilities (Code highlighting, navigation, content assist and static analysis)
3. Eclipse compare/merge does not have the same function as the TPF Merge in TPF Toolkit 4.2
4. Copying a file from the Remote File Search view into a TPF project is not straight forward
5. TPF Toolkit does not provide a way to work with z/TPF Automated Test Framework test cases
6. Debugger in TPF Toolkit does not help easily identify null pointer and null address references in code
7. Use of unsecure protocols is becoming increasingly difficult if not impossible to justify
8. DFDL validation does not identify all errors before loading a DFDL schema file to the z/TPF system

# Pain Points



Difficult to search for files and text on the remote build system

- Eclipse Remote Search requires a Remote System Explorer (RSE) connection and a RSE server
- Eclipse Remote Search is unreliable when a SMB (Samba) mount is used
- Terminal view requires Linux knowledge
  - find and grep utilities
  - PTP command

# Prior to fix pack 12 - User Story

Zach is working on a project. He needs to update a structure but does not know which header includes the structure definition.

He uses the “**Remote Search**” menu item to search for the header file on the remote Linux on IBM Z system. The remote Linux on IBM Z system does not have a RSE server. So, Zach uses a SMB mount for the Remote Search. The search progress bar indicates it is performing the search but fails to complete.

He knows that another option to search for remote files is to use the **Terminal** view. He is not comfortable with the Terminal view since it requires him to be knowledgeable with Linux utilities such as find and grep and the ptp command.

# Value Statement

A “**Remote File...**” menu item was added to the TPF Toolkit Search menu that allows the user to search for files or text within files on Remote Connections.

- Search completes in just a few seconds
- Does not depend on the Remote System Explorer (RSE)
- Similar UI as the standard Eclipse Remote Search
- Ability to open a remote file in the editor by a simple double-click
- No Linux or PTP knowledge required

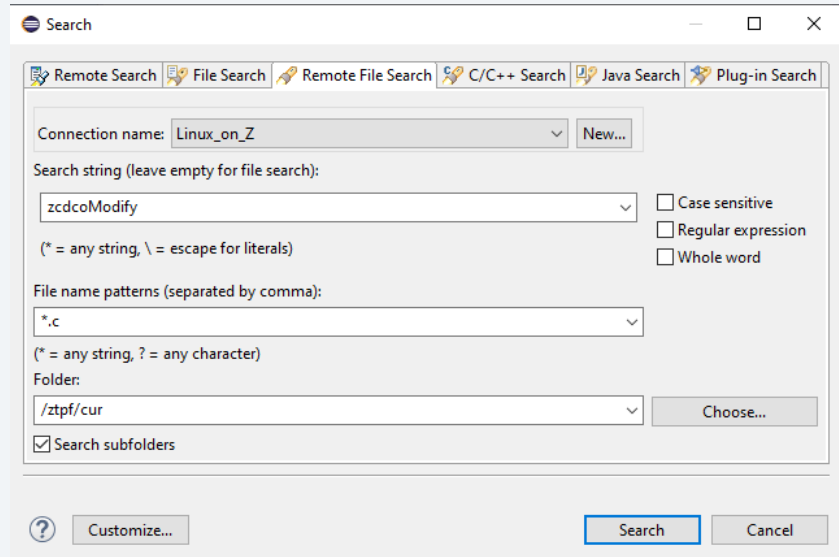
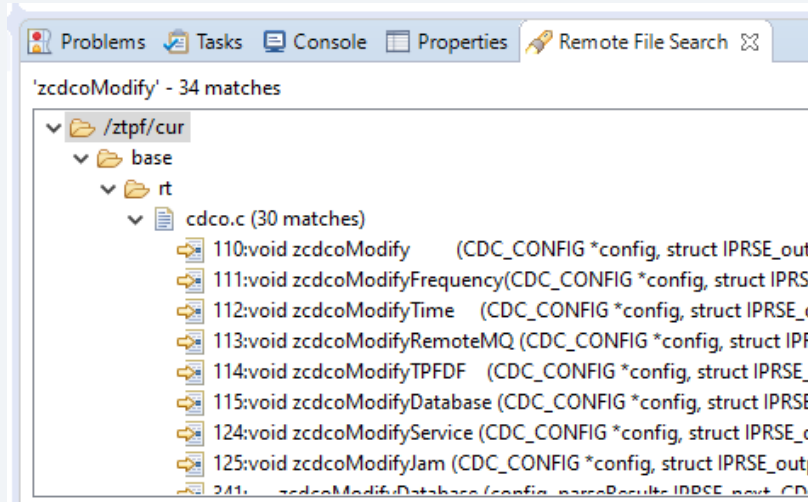
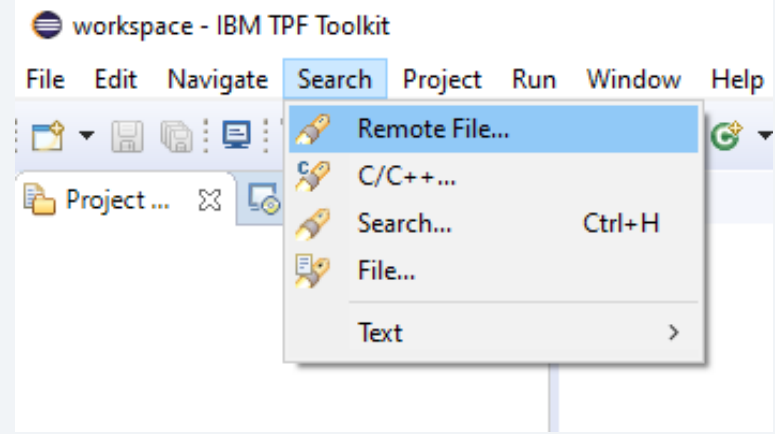
**Delivered in fix pack 12 on September 30, 2021**



# With fix pack 12 - User Story

Zach uses the Remote File Search dialog via the Search > Remote File... menu item. He chooses the Connection name and enters the search criteria and clicks on the Search button.

The results are displayed in the Remote File Search view in just a few seconds. He double-clicks on a file or line to open the file in the editor.



# Pain Points



Unable to use modern C/C++ editor capabilities

Processing does not always complete successfully when the “**Generate Include Paths**” action is selected on a project.

**Generate Include Paths** action creates a list of paths to folders that include header files. It also calls the CDT indexer to create a database of source information that provides the basis for editor features such as code highlighting, navigation, content assist and static analysis.

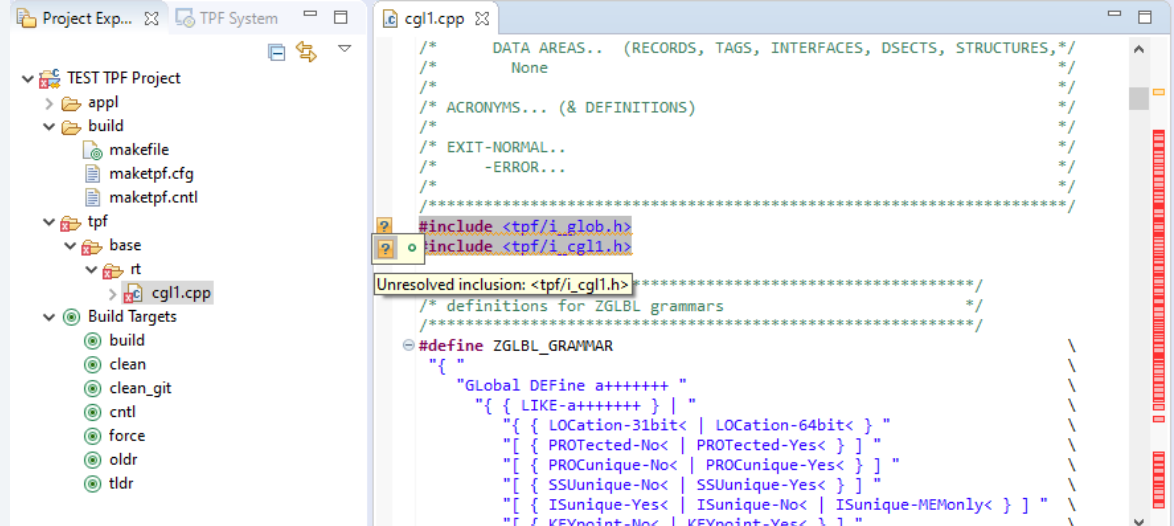
# Prior to fix pack 12 - User Story

Zach would like to use modern editor features. He right-clicks on a TPF project and selects **“MakeTPF actions > Generate Include Paths”** action.

Processing does not always complete successfully due to issues with CDT indexing based on paths to header files on the remote Linux on IBM Z system.

As a result, red error markers are displayed on the source file in the editor and the TPF project.

Hovering over the include statement displays **“unresolved inclusion”** message.

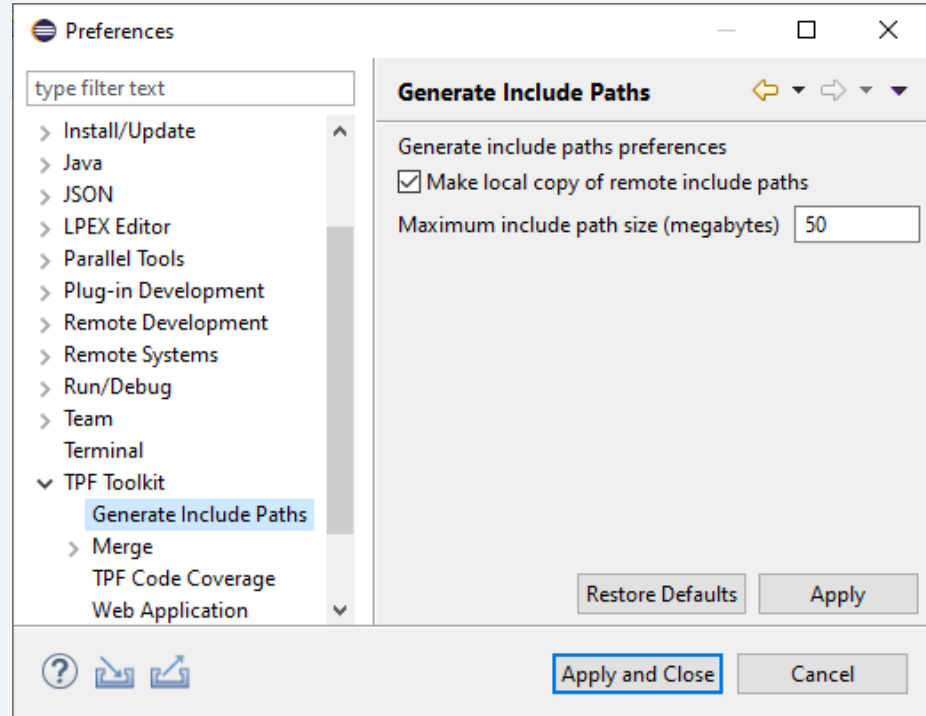


# Value Statement

A new preference setting was added which instructs the "**Generate Include Paths**" action to make temporary local **read-only** copies of TPF Project include files. Making local copies resolves potential reliability issues when using Generate Include Paths.

A successful completion of Generate Include Paths action enables editor capabilities such as code highlighting, navigation, content assist, and static analysis in C/C++ source files.

**Delivered in fix pack 12 on September 30, 2021**



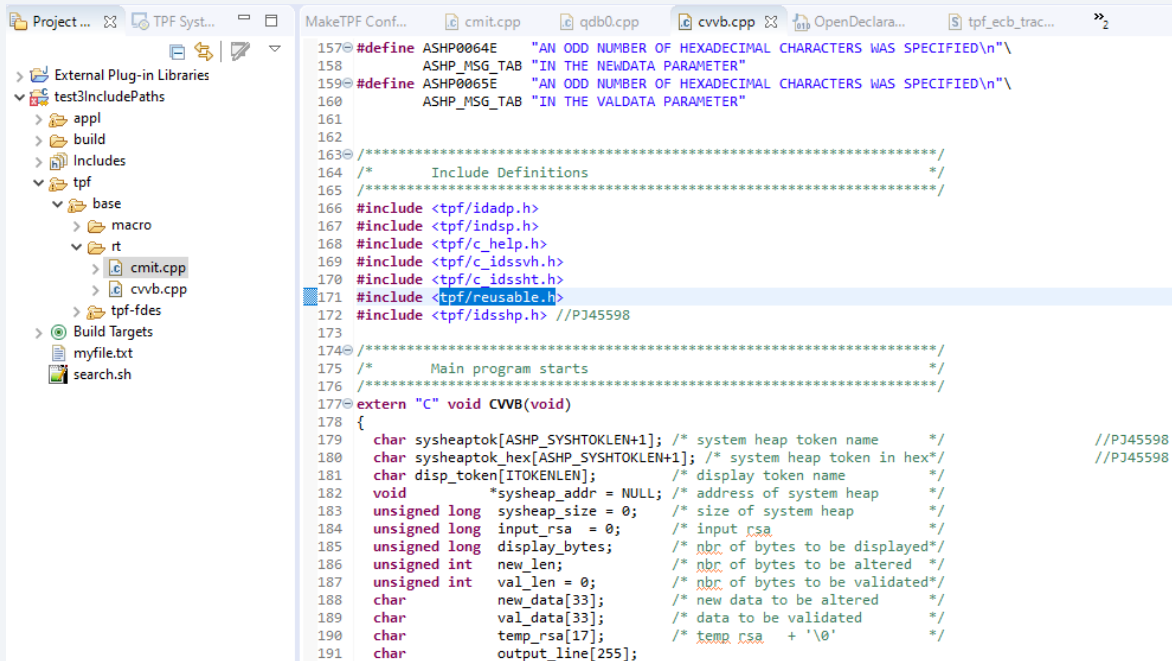
# With fix pack 12 - User Story

Zach selects the preference setting “**Make local copy of remote include paths**” on “**Generate Include Paths**” preference page.

He right-clicks on the TPF project and selects “**Make TPF actions > Generate Include Paths**” action.

Processing completes successfully and the error markers on the source file in the editor and the TPF project disappear.

He can now Ctrl+click on an include statement to open the include file in the editor.



# Pain Points



Eclipse compare/merge does not have the same function as the TPF Merge in TPF Toolkit 4.2

- Files to compare are required to be in a TPF project
- Does not allow merging differences from three files into a fourth output file
- Lacks features such as color highlighted differences
- Does not have a quick way to copy differences into the fourth file
- Does not have the ability to save a merge session
- Time consuming

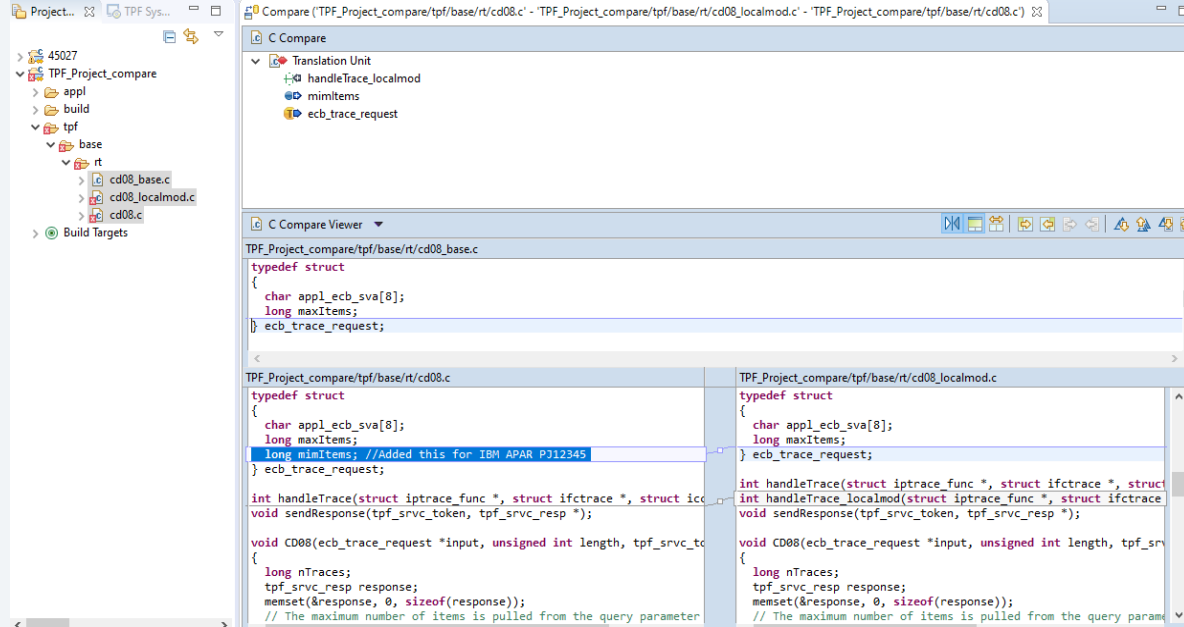
# Prior to fix pack 13 - User Story

Sophie, the system programmer just downloaded an APAR that includes CP changes. She needs to compare the changes to the base version from a prior apar and the local mod version and merge the changes into the local mod version.

She uses the standard Eclipse compare. She copies the three files to compare into her TPF project.

Selects the three files, right clicks and selects "**Compare With > Each Other**". She chooses the common ancestor (base version).

She also needs to select "Show Ancestor Pane" from the toolbar of the Compare viewer to display the base version.

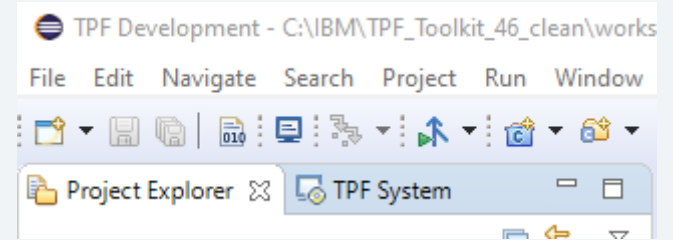


# Value Statement

TPF Toolkit provides the capability to compare and merge three input files into a fourth output file.

Includes the popular features that existed in TPF Toolkit 4.2.

Greatly improves productivity and user-experience.



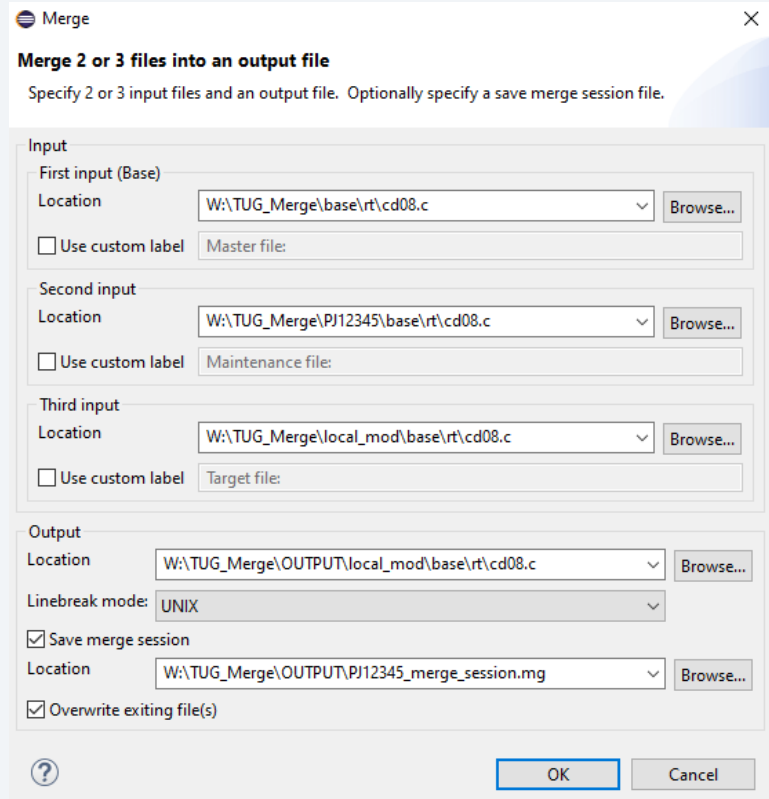
**Delivered in fix pack 13 on December 16, 2021**



# With fix pack 13 - User Story

Sophie clicks on Merge icon from the toolbar to open the Merge dialog.

She enters the paths for the input and output files or uses the Browse button to select them. She provides the Save merge session details.



The screenshot shows a 'Merge' dialog box with the following fields and options:

- Input**
  - First input (Base)**
    - Location: W:\TUG\_Merge\base\rt\cd08.c (with a 'Browse...' button)
    - Use custom label: Master file: (text field)
  - Second input**
    - Location: W:\TUG\_Merge\PJ12345\base\rt\cd08.c (with a 'Browse...' button)
    - Use custom label: Maintenance file: (text field)
  - Third input**
    - Location: W:\TUG\_Merge\local\_mod\base\rt\cd08.c (with a 'Browse...' button)
    - Use custom label: Target file: (text field)
- Output**
  - Location: W:\TUG\_Merge\OUTPUT\local\_mod\base\rt\cd08.c (with a 'Browse...' button)
  - Linebreak mode: UNIX (dropdown menu)
  - Save merge session
    - Location: W:\TUG\_Merge\OUTPUT\PJ12345\_merge\_session.mg (with a 'Browse...' button)
  - Overwrite exiting file(s)

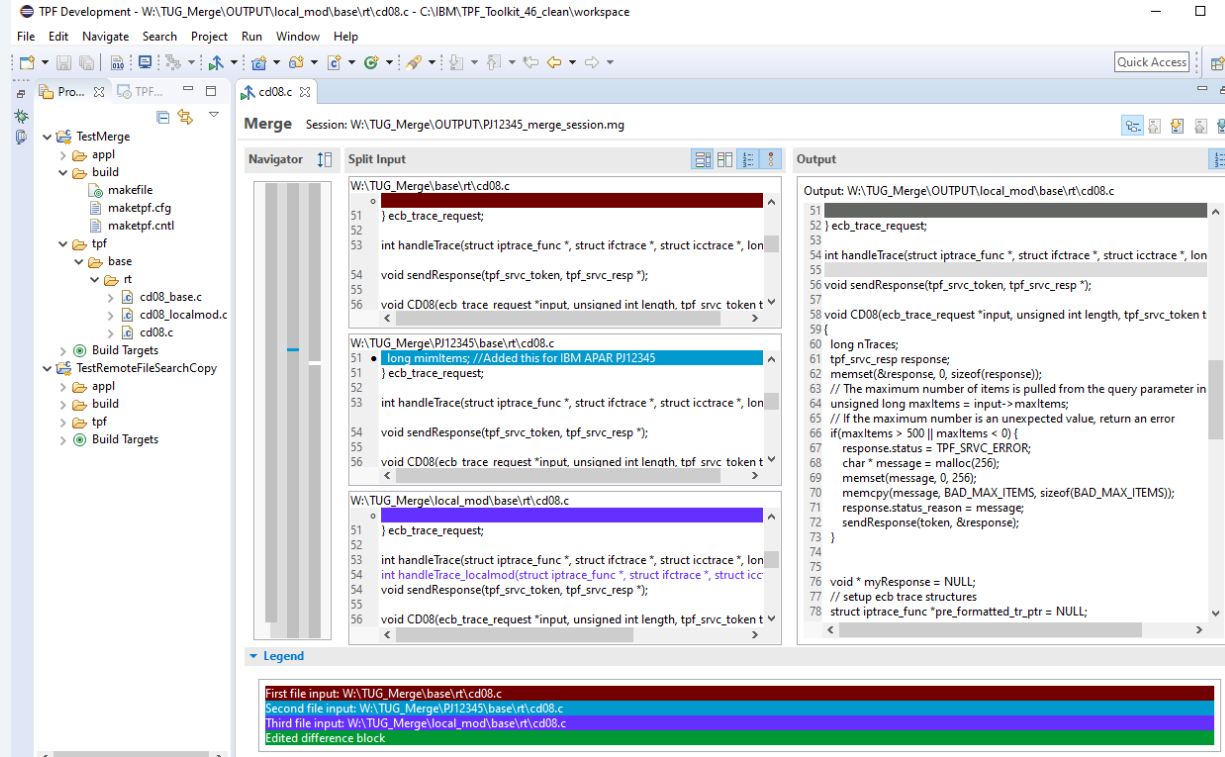
At the bottom, there is a help icon (question mark), an 'OK' button, and a 'Cancel' button.

# With fix pack 13 - User Story (continued)

The differences are shown in the Merge editor. She merges the changes and saves the changes into a fourth file.

She can also pause working on the merge by saving the merge session to finish it later.

She does her job much faster due to the user-friendly interface of the Merge editor.



# Pain Points



Copying a file from the Remote File Search view into a TPF project is not straight forward

# As-Is User Story

Zach uses the “Remote File Search” option to search for a header file on the remote Linux on IBM Z system.

The results are displayed in the Remote File Search view. He double-clicks on the header file name to open in the editor.

He wishes to copy this file into his TPF project to make a quick change and test it. He uses the “File > Save As” option and chooses a folder path outside of the workspace location to save a copy.

From the Windows File Explorer, he browses to the file location, right-clicks on the file and selects “Copy”.

He switches back to TPF Toolkit, right-clicks on the TPF project folder and selects “Paste”.

# Value Statement

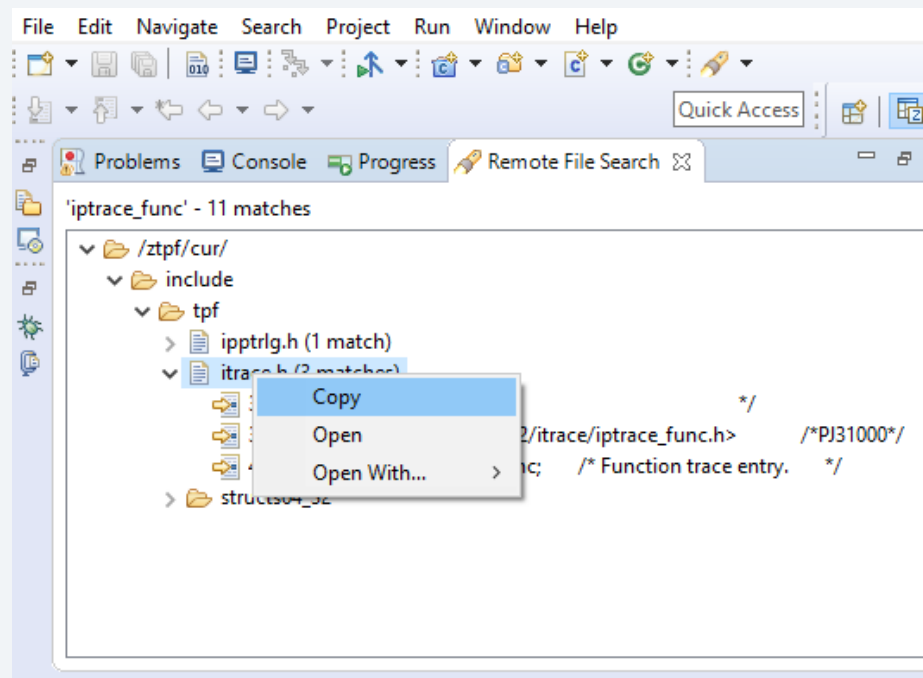
A new right-click context menu action “Copy” is provided in the Remote File Search view, to allow users to copy one or more files into the TPF project .

**Plan to deliver in fix pack 14 (Target April 2022)**

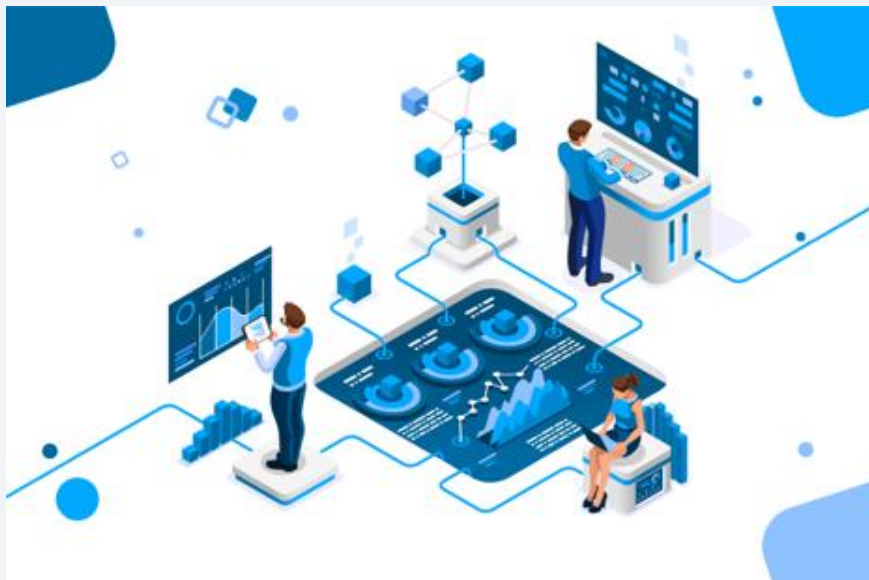
# To-Be User Story

Zach uses the Remote File Search dialog to search for the header file. The results are displayed in the Remote File Search view. He right-clicks on the header file name and selects “Copy”.

He right-clicks on the TPF project folder and selects “Paste”.



# Pain Points



TPF Toolkit does not provide a way to work with z/TPF Automated Test Framework test cases

- Cannot run test cases automatically after loading a test program to the z/TPF system
- Cannot view test cases available on a z/TPF system, run the test cases and view the results in TPF Toolkit

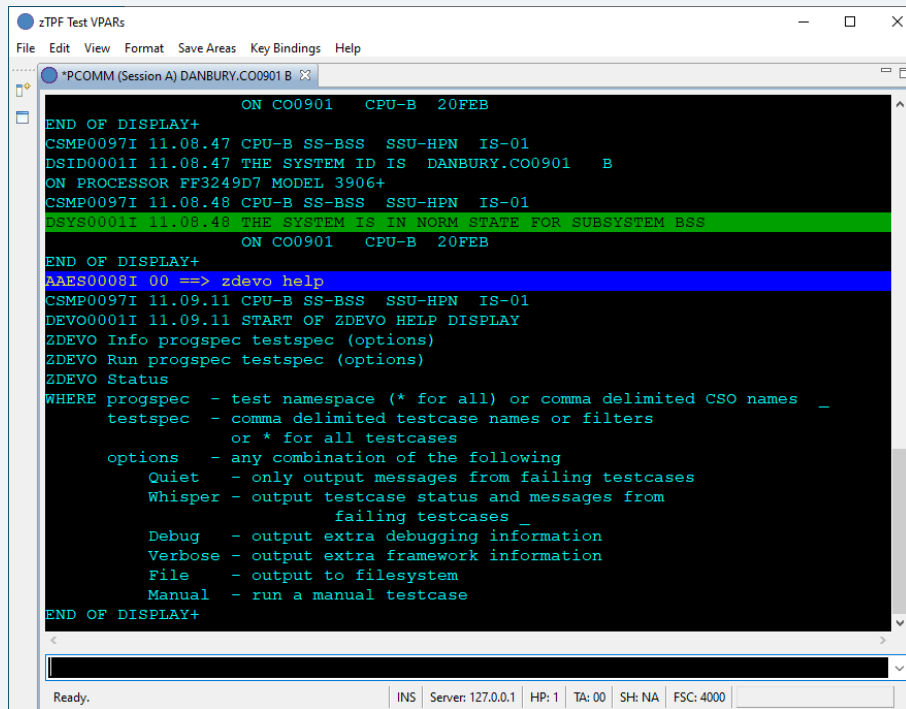
# As-Is User Story

Zach is working on maintenance and updates an application program. He realizes that he needs to add a new test case to a test program to test the updates.

He adds the new test case to the test program and loads and activates the application and test program on to the z/TPF system via the “**Load and Activate**” action on the loader input file.

Zach logs into the z/TPF system, uses the **ZDEVO** command to run the new test case. He also looks up existing test cases for the application program and runs them to make sure he did not break existing function.

Lawrence, the test engineer, is tasked to produce a test plan to run function/regression tests for the changes Zach delivered. Lawrence logs in to the z/TPF system, uses the **ZDEVO** command to lookup the test cases. Once the plan is finalized, Lawrence logs into the z/TPF system and uses the **ZDEVO** to run the tests.



```
z/TPF Test VPARs
File Edit View Format Save Areas Key Bindings Help
*PCOMM (Session A) DANBURY.CO0901 B
ON CO0901 CPU-B 20FEB
END OF DISPLAY+
CSMP0097I 11.08.47 CPU-B SS-BSS SSU-HPN IS-01
DSID0001I 11.08.47 THE SYSTEM ID IS DANBURY.CO0901 B
ON PROCESSOR FF3249D7 MODEL 3906+
CSMP0097I 11.08.48 CPU-B SS-BSS SSU-HPN IS-01
DSYS0001I 11.08.48 THE SYSTEM IS IN NORM STATE FOR SUBSYSTEM BSS
ON CO0901 CPU-B 20FEB
END OF DISPLAY+
AAES0008I 00 ==> zdevo help
CSMP0097I 11.09.11 CPU-B SS-BSS SSU-HPN IS-01
DEV00001I 11.09.11 START OF ZDEVO HELP DISPLAY
ZDEVO Info progspec testspec (options)
ZDEVO Run progspec testspec (options)
ZDEVO Status
WHERE progspec - test namespace (* for all) or comma delimited CSO names _
testspect - comma delimited testcase names or filters
or * for all testcases
options - any combination of the following
Quiet - only output messages from failing testcases
Whisper - output testcase status and messages from
failing testcases _
Debug - output extra debugging information
Verbose - output extra framework information
File - output to filesystem
Manual - run a manual testcase
END OF DISPLAY+
Ready. | INS | Server: 127.0.0.1 | HP: 1 | TA: 00 | SH: NA | FSC: 4000
```



# Value Statement

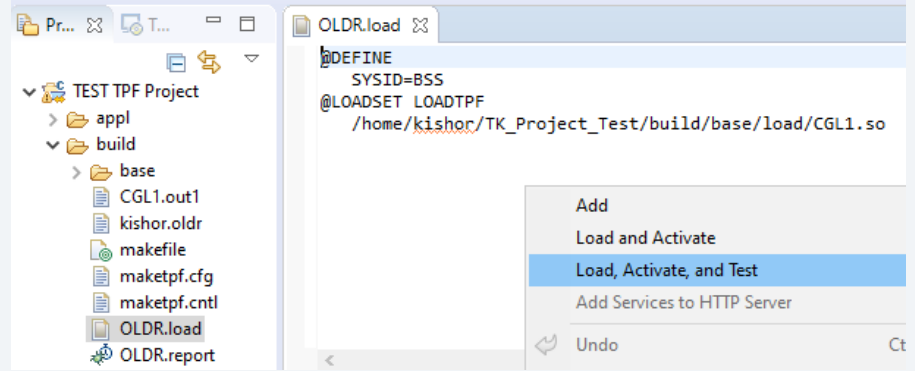
TPF Toolkit will provide the following features to integrate z/TPF Automated Test Framework:

- **“Load, Activate and Test”** action  
Ability to automatically run z/TPF Automated Test Framework test cases included in test programs after load and activate. Results of the test case(s) are displayed in the TPF Toolkit console.  
**Delivered in fix pack 13 on December 16, 2021**
- **Automated Test Framework** view  
New view to display z/TPF Automated Test Framework test cases on a z/TPF system, run and view the results.  
**Plan to deliver in fix pack 14 (Target April 2022)**
- **More Integration of z/TPF Automated Test Framework**  
Ability for the user to select which test cases to automatically run when an application program is loaded and activated on the z/TPF system.  
**Plan to deliver in fix pack 15 (Target 3Q2022)**

# To-Be User Story

Zach is working on maintenance and updates an application program. He realizes that he needs to add a new test case to a test program to test the updates.

He adds the new test case to the test program and selects the **“Load, Activate, and Test”** action to automatically run the new and existing test cases after load and activate is complete.



# To-Be User Story (continued)

Lawrence uses the Automated Test Framework view to lookup the list of test cases. Once the plan is finalized, Lawrence selects the tests and runs them.

Both Zach and Lawrence can perform their work from within TPF Toolkit and have the choice to lookup and run tests on multiple systems with the click of a button.

The screenshot displays the Automated Test Framework (ATF) interface. The top toolbar includes 'Problems', 'Console', 'Remote File Search', 'REST Services', and 'Automated Test Framework'. The main area is divided into three panes:

- Left Pane:** Shows a list of systems. 'VPARS\_Kishore - 9.57.13.214' is selected. Below this, it displays 'Selected system - VPARS\_Kishore', 'Processor - B', 'Subsystem name - BSS', and 'Complex - CO0901'.
- Middle Pane:** Shows a tree view of test cases under the 'ibm' directory. The 'test\_aor' test case is selected, and a context menu is open with the 'Run' button highlighted.
- Right Pane:** Shows the test output for 'test\_aor'. It includes a progress bar indicating 'Runs: 2/2', 'Failed: 0', 'Ignored: 0', and 'Time: 5.738'. Below the progress bar, the test output is displayed, including steps like 'Create sockets for the tests', 'Bad file descriptor tests for AOR', and 'Wait two seconds for timeout to occur'.

# Pain Points



Debugger in TPF Toolkit does not help easily identify null pointer and null address references in code.

# As-Is User Story

## C/C++ Source

```
//  
// structure pointer is set to zero  
//  
void customerInformationStructPointer() {  
    printf("customerInformationStructPointer\n");  
    char *customerName = (char*)malloc(100);  
    memset(customerName,0,100);  
    memcpy(customerName,"JOHN",4);  
    customerInformationStruct * customerDataPtr =  
getCustomerInformation(customerName);  
    int balance = customerDataPtr->balance;  
    if (balance != 0) {  
        printf("mybalancePtr =%d\n",balance);  
    }  
    free(customerDataPtr);  
    free(customerName);  
}
```

Zach sees that his application is returning incorrect results.

He notices that the pointer to a structure `customerInformationStruct` does not have the expected data.

Zach uses the debugger to determine the cause of the problem. While stepping through his application, and looking at `customerDataPtr` in the Variables view, he sees that the function returns null and sets `customerDataPtr` to zero. The application does not check for null return from the function. The application then generates incorrect results because members of `customerDataPtr` are pointing to low core.

Zach was able to resolve the issue, but it was time consuming. It took Zach an hour to resolve the issue.

# As-Is User Story

## Assembly Source

```
*-----*  
* Test zero address detection.  
* Base register has value of zero.  
*-----*  
    LGHI R8,0  
QDBW_LBL L    R5,12(,R8)  
    LTGR R5,R5
```

Zach sees that his application is returning incorrect results.

He notices that the base register R8 does not have the expected data.

Zach uses the debugger to determine the cause of the problem. While stepping through his application and looking at the values of registers in the Registers view, he sees that base register R8 is zero.

Zach was able to resolve the issue, but it was time consuming. It took Zach an hour to resolve the issue.

# Value Statement

TPF Toolkit provides a way to stop at null pointer and null address detection (zero-address-detection) events during a debug session and log these events to a file.

- A new breakpoint “Stop at null pointers and addresses” is provided to stop at null pointer and null address detection events during a debug session.
- A preference setting is provided to enable logging of null pointer and null address detection events that occur during a debug session to a log file on the z/TPF system.
- Much faster identification of the problem code path.

**Target is 2Q2022 (requires PJ46595)**

# To-Be User Story

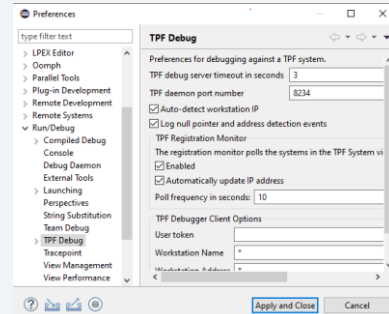
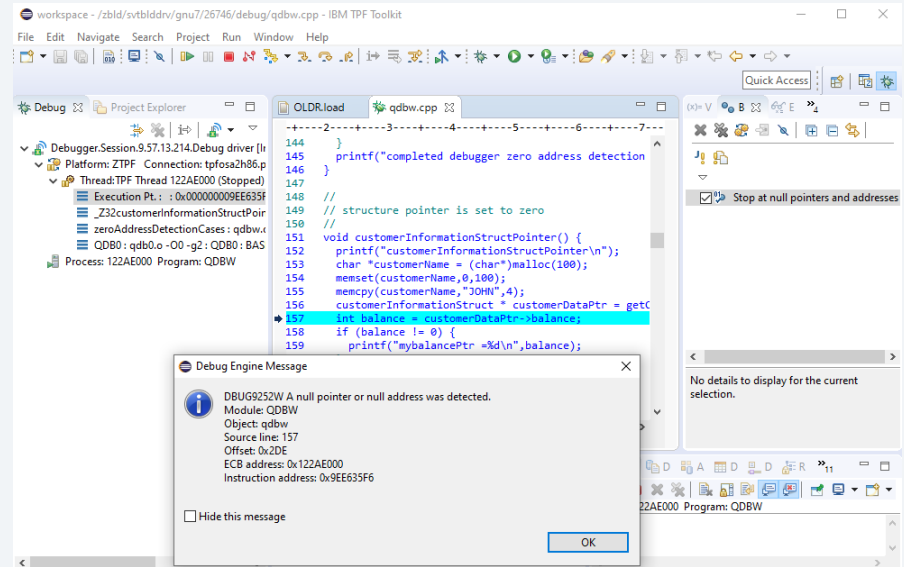
While debugging his application, Zach selects the new breakpoint “Stop at null pointers and addresses”.

He then selects the Resume button, and the debugger stops where a field in customerInformationStruct is accessed.

A pop-up box displays information, indicating that a null pointer or null address event was detected. He selects OK, and processing continues.

Zach updates the code to check for null return from the function and his application now generates correct results. Zach was able to resolve the issue in just 10 minutes.

Zack enables logging and registers the debug session. He selects the Resume button and lets the debug session complete. Zach examines the log and fixes issues identified in the log.





# Pain Points



- The use of unsecure protocols is becoming increasingly difficult if not impossible to justify
- Security auditors are raising concerns over the use of HTTP instead of HTTPS for REST service calls from TPF Toolkit to z/TPF

# Value Statement

TPF Toolkit will provide a preference page to allow a user to choose between a HTTP or HTTPS connection and set the Secure HTTP port and KeyStore information (file name and password).

TPF System definition will be able to use the global preference settings and allow the user to override for each TPF System.

**Target is 3Q2022**

# Pain Points



TPF Toolkit DFDL validation does not identify all errors before loading a DFDL schema file to the z/TPF system

# As-Is User Story

Zach is using TPF Toolkit to create/update a DFDL. He runs a Validate in the DFDL schema editor and fixes any errors before loading it to the z/TPF system.

Processing stops after the first error is encountered during activation on TPF.

He fixes the error and does the load and activate again. Processing stops after the first error is encountered.

He repeats the process of fixing each error, loading and activating until all errors are resolved.

# Value Statement

TPF Toolkit will provide a way to perform offline validation (tpfdatamap) that will report all errors and provide the location of each error.

An application developer can use TPF Toolkit to debug and fix DFDL validation errors in a matter of minutes instead of hours/days.

**Target is 3Q2022**

# To-Be User Story

Zach is using TPF Toolkit to create/update a DFDL. He validates the file in the DFDL schema editor.

He fixes any errors identified by the offline validation, loads and activates the changes.

He is happy that he does not have to repeat the process of fixing one error at a time and then loading.

# Conclusion

For more information:

- TPF Toolkit documentation  
<https://www.ibm.com/docs/en/ztpf/2021?topic=tpf-toolkit>
- TPF Toolkit demo videos  
[https://mediacenter.ibm.com/channel/z\\_TPF/192490123](https://mediacenter.ibm.com/channel/z_TPF/192490123)

Questions?

# 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

We expect to begin engaging with the sponsor users in 2<sup>nd</sup> Quarter 2022.

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

**Kishore Nagareddy** (kishoren@us.ibm.com)



# Thank you

© Copyright IBM Corporation 2022. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM 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 at [Copyright and trademark information](#).

