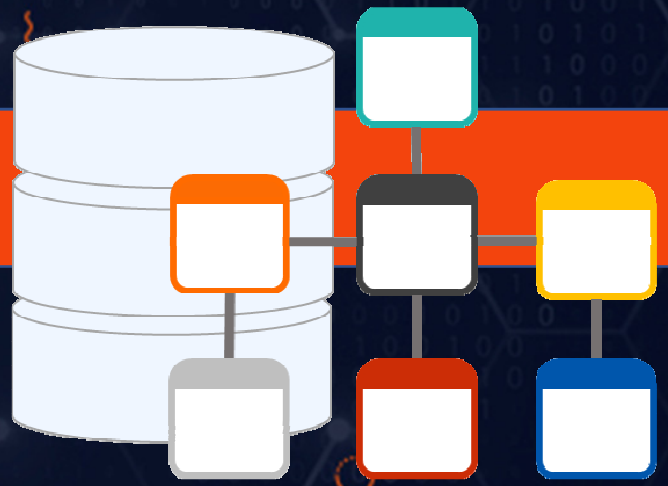
TSI Product Updates

TPF USERS GROUP • 2018 • AUSTIN, TEXAS



DFDL Support

- OVERVIEW
- SCHEMA GENERATION
- SCHEMA EDITING
- SCHEMA TESTING
- SCHEMA LOADING

File Merge

- OVERVIEW
- USING FILE MERGE

Other Updates

- C/C++ CODE FORMATTING
- C/C++ CODE SENSE
- PRODUCTION DUMP VIEWER
- MORE UPDATES

DFDL Overview

DFDL (Data Format Description Language) is a language for describing data formats

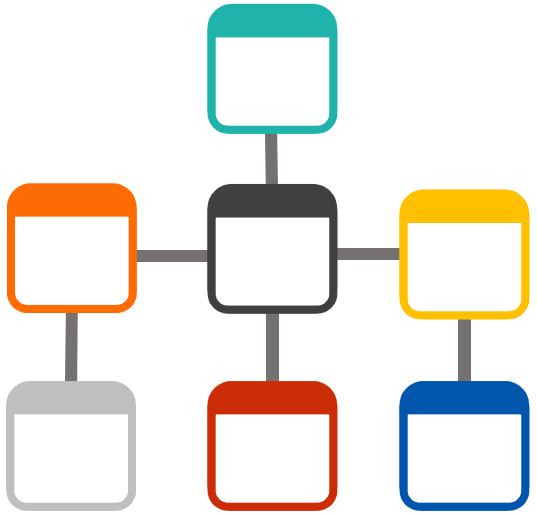
- Open Source standard
- An extension of XML schema (.xsd)
- Allows conversion of data from a native format to an XML “infoset”
- Also allows conversion of an XML “infoset” to a native format

A key piece in implementing IBM-recommended solutions for Data interchange

- MongoDB for z/TPF,
- Business Eventing, and
- Rest API/Swagger utilization

DFDL features in zTPFGI allow users to

- Generate,
- Edit,
- Test, and
- Load DFDL Schemas



Schema Generation

From TPFDF

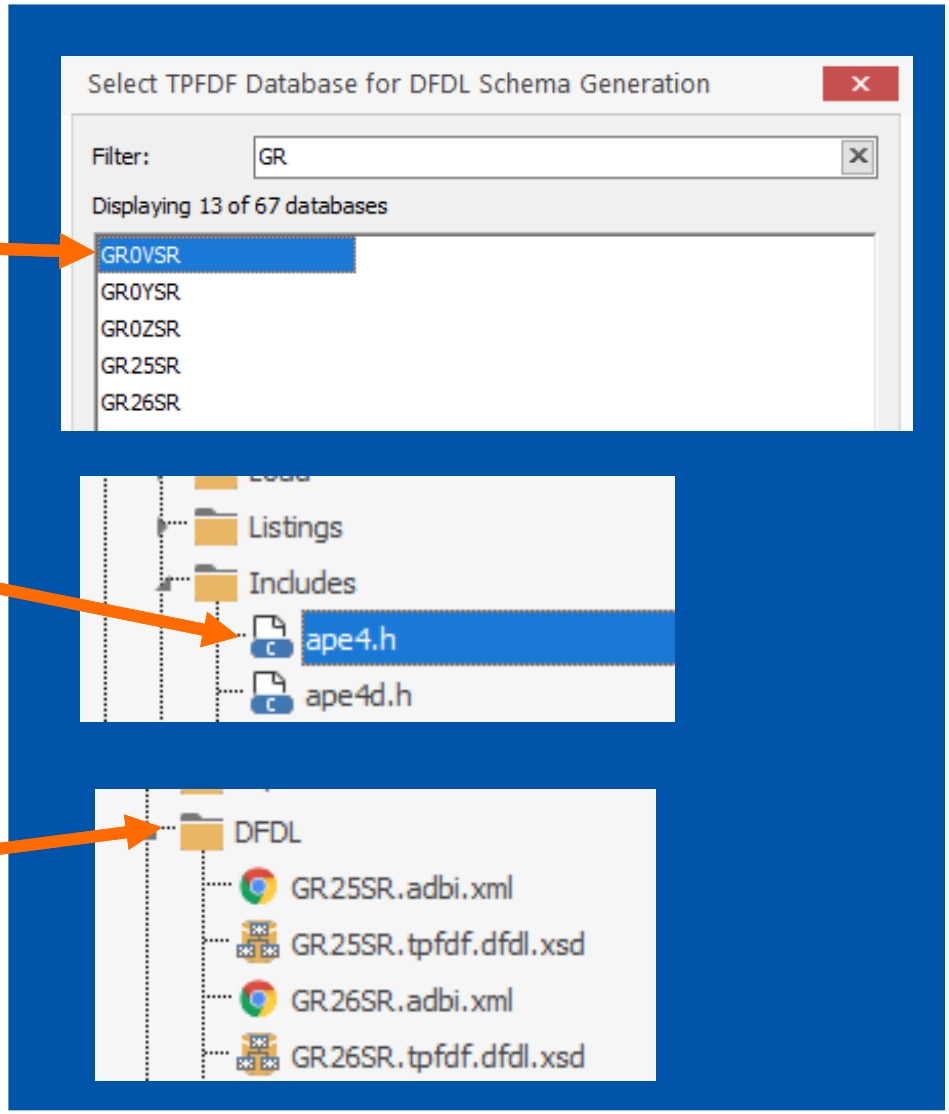
- Connect to a system
- Click button to select DF file
- Select which project to associate schema with

From header files

- Right click a header file in Projects View
- Select Generate DFDL Schema

In both cases

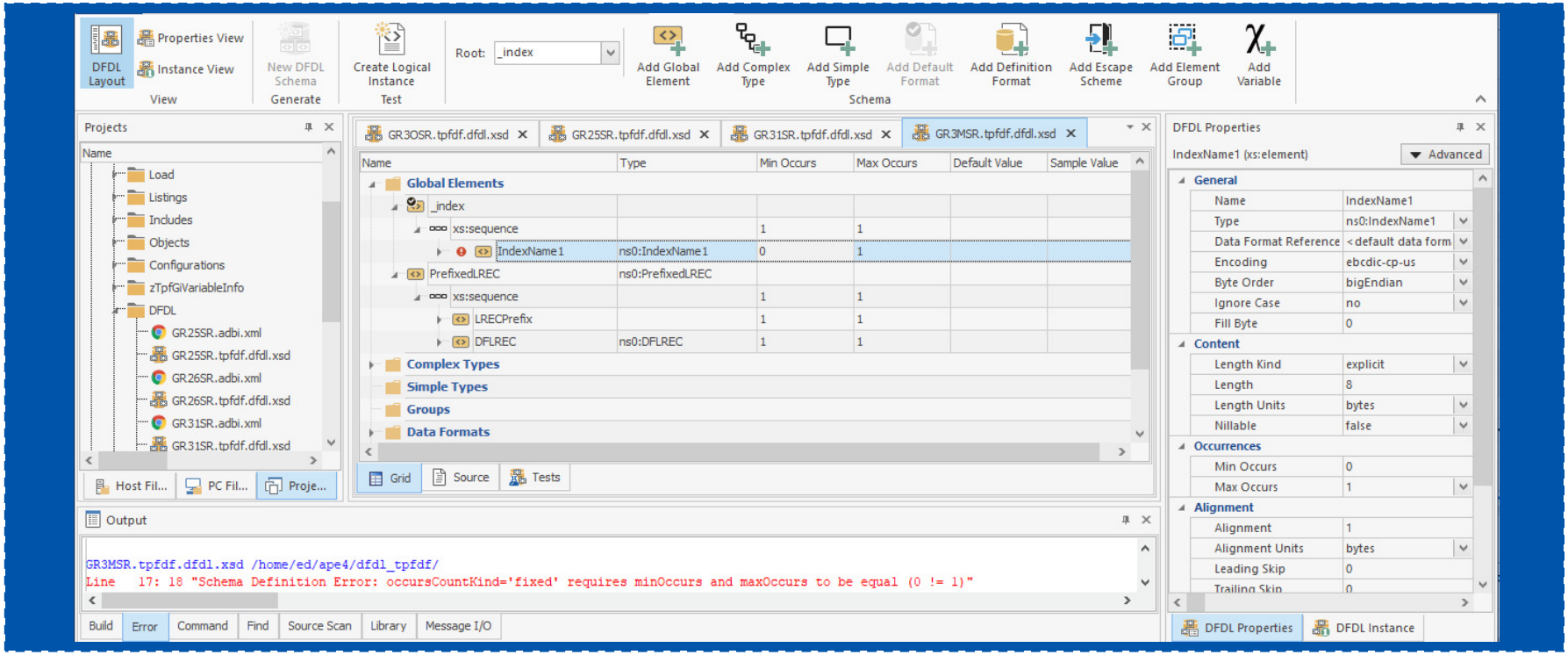
- Schema is generated and placed in the DFDL subfolder of the folder



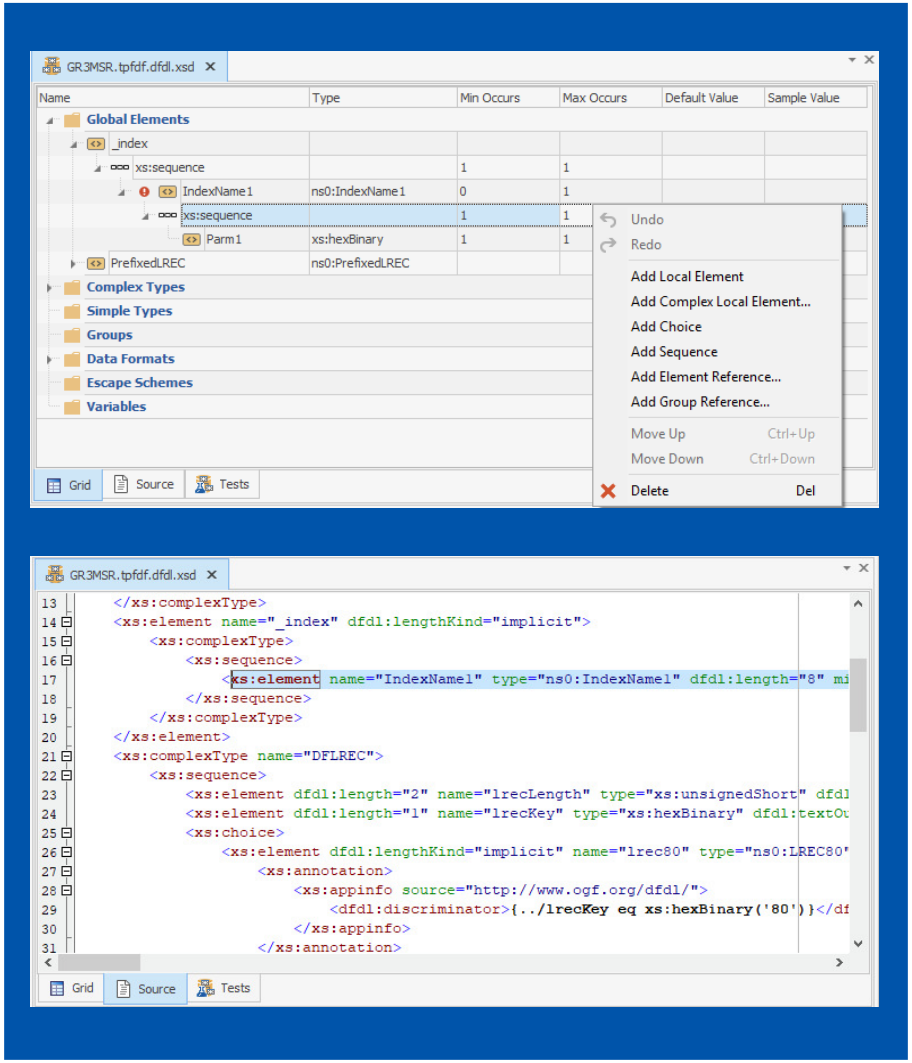
Schema Editing

Views/windows to facilitate editing:

- DFDL Schema Editor
- DFDL Properties View
- Output View



Schema Editing – Schema Editor



Grid tab

- Add/edit/delete Elements, Simple types, Complex types, DFDL variables, Groups, Formats, etc.
- Reorder elements and fields
- Icons/tooltips for warnings and errors
- Full undo and redo

Source tab

- View and edit .xsd source
- Selection syncs to current row in Grid

Tests tab

- Shows saved tests for schema

Schema Editing – Properties View

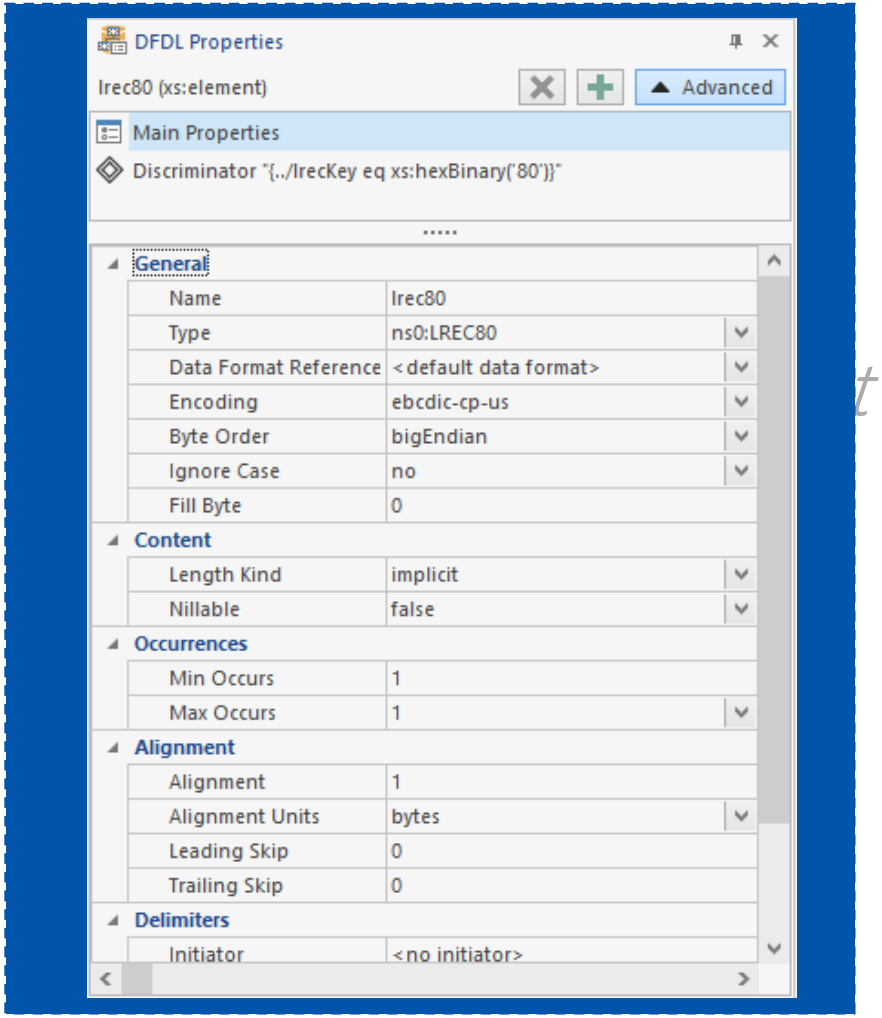
Syncs to focused row in editor Grid

Allows editing of attributes for elements, simple types, complex types, formats, variables, etc.

Drop-down choices for attributes

Error and warning icons with tooltips

"Advanced" area shows discriminators, asserts, variables

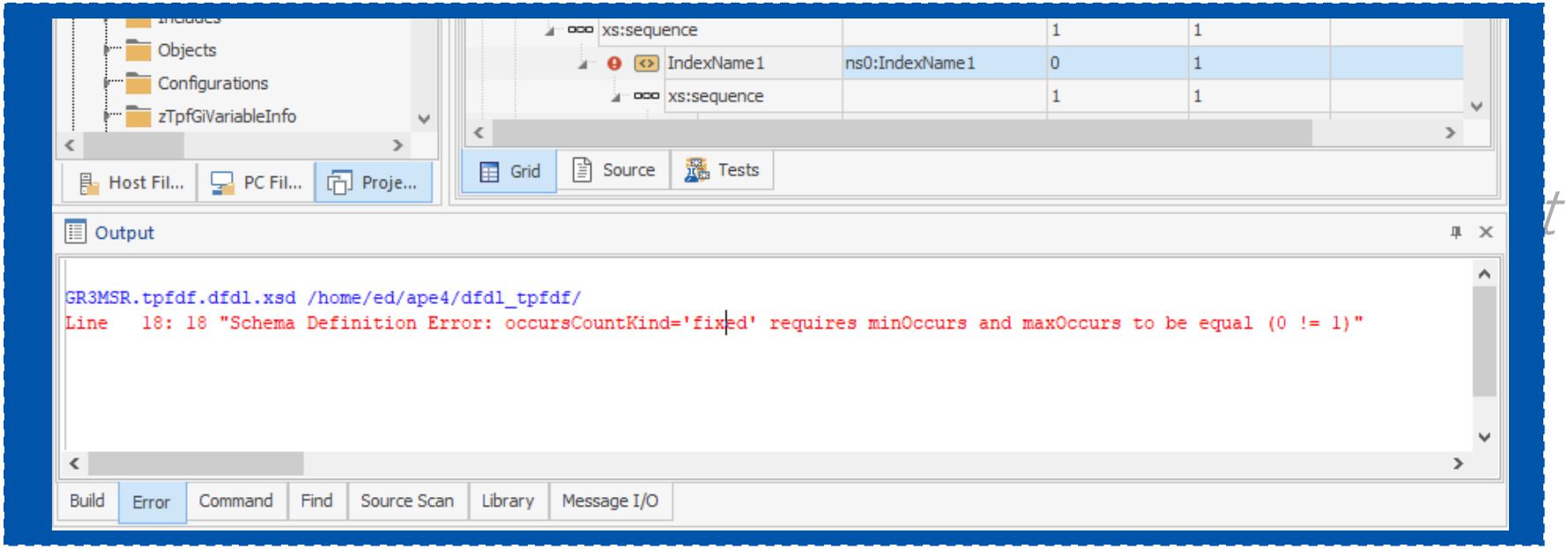


Schema Editing – Output View

Displays output from parser/serializer

Displays warnings and errors

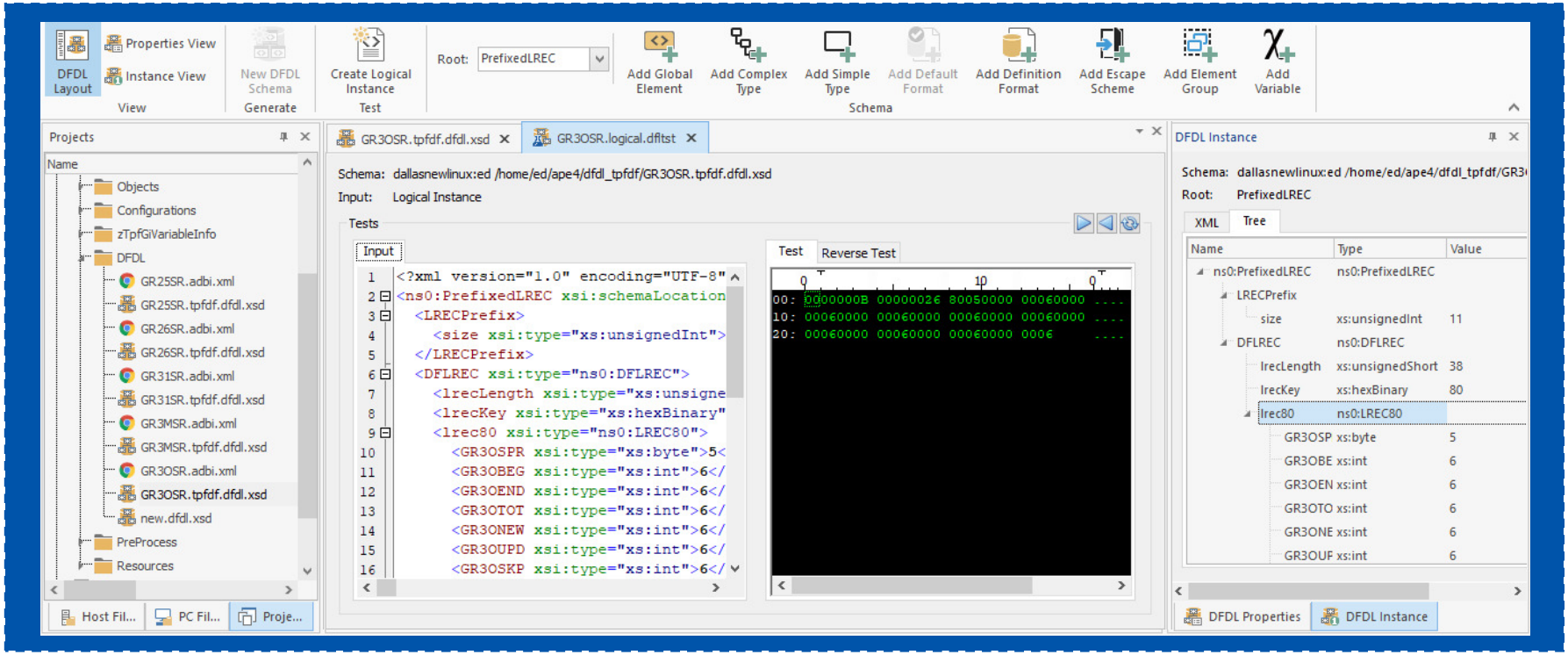
Double clicking warnings/errors navigates to problem row in editor Grid



Schema Testing

Views/windows to facilitate testing:

- DFDL Instance View
- DFDL Test Editor
- Tests tab in DFDL Schema Editor



Schema Testing – Logical Instance

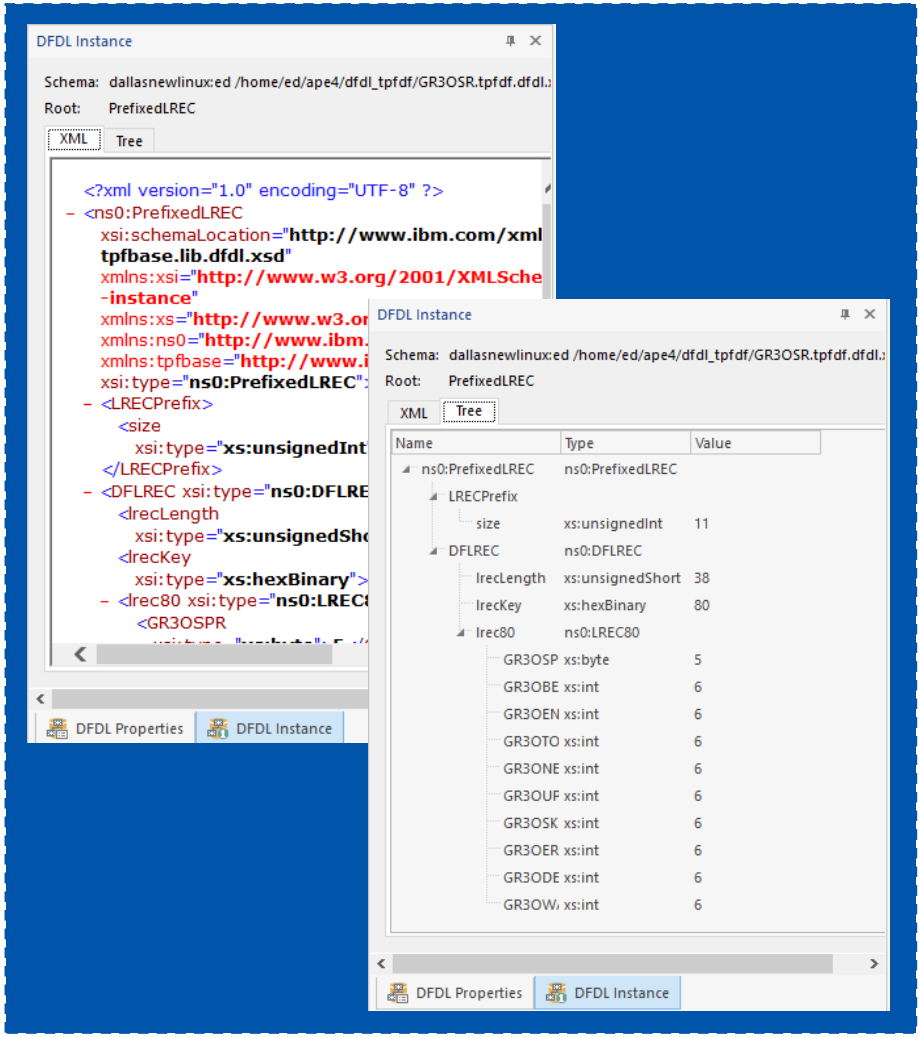
Click ribbon button to create “Logical Instance” for current schema editor

XML infoset created for current schema, appears in DFDL Instance View

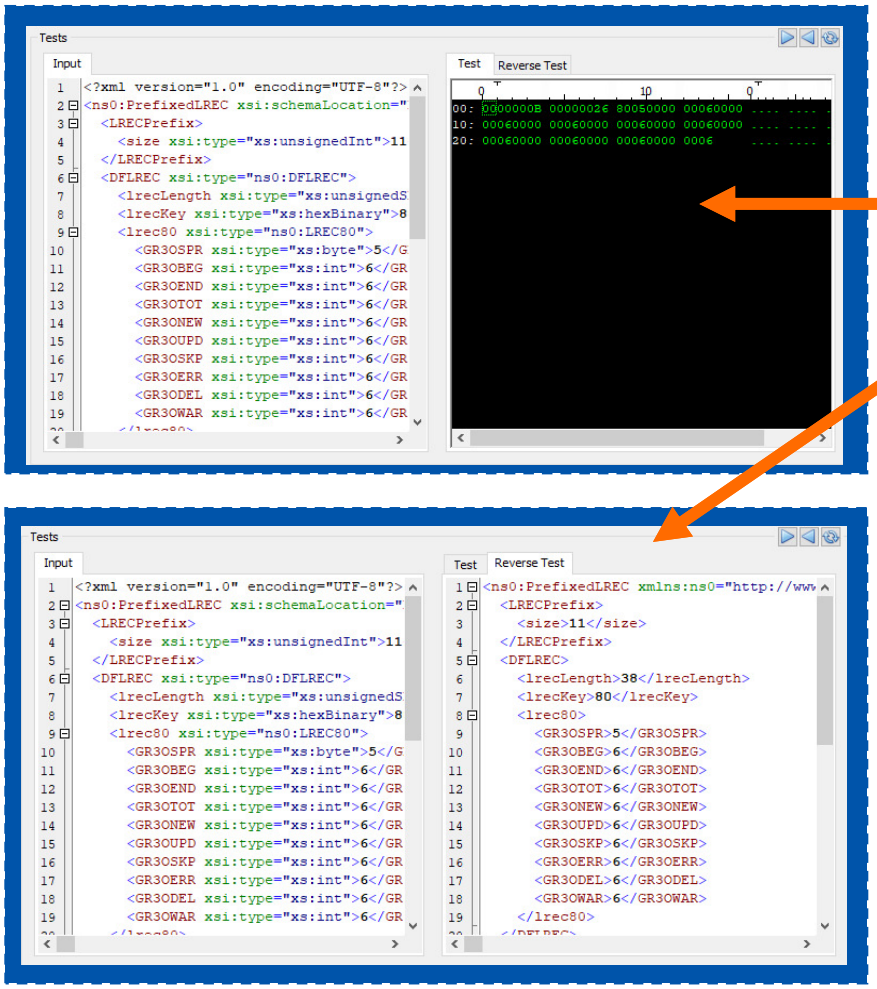
Two ways of viewing

- XML View
- Tree View

Right click DFDL Instane to launch test



Schema Testing – Test Editor



Two possible tests:

- Serialize (from XML infoset data)
- Parse (from data to XML)

"Forward" test:

- Click arrow to parse or serialize from left to right

"Reverse" test:

- Click left arrow to serialize or parse result back (appears on Reverse tab)

Compare:

- Compare left to right visually or through built in or external compare tool

Edit in place:

- Both XML and hex/char data can be edited for new test in place

Save As:

- Save test for future use

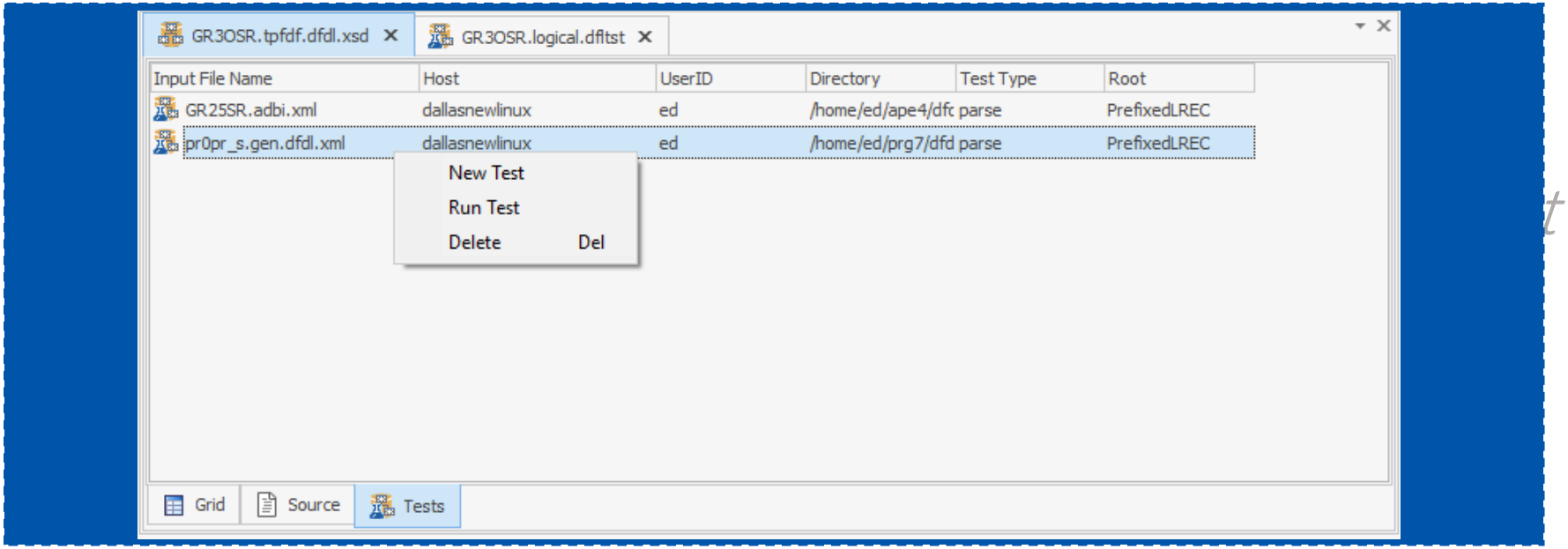
Schema Testing – Test Tab in main Schema editor

Drag and drop files here to create new tests

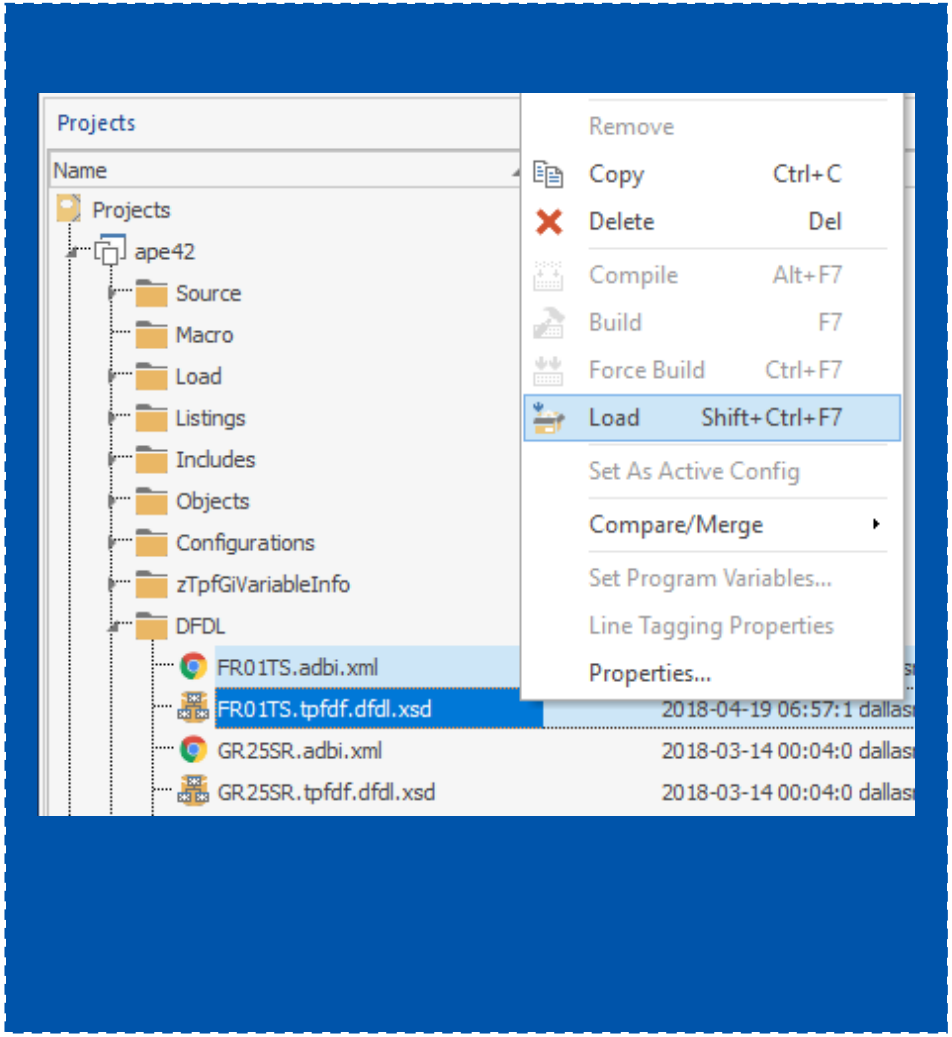
Double click to open Test editor for that xml/data

Serialize and parse tests can be created

Tests are remembered across sessions



Schema Loading



Schema and supporting files can be loaded to the z/TPF system

- Select the files
- Right-click and select "Load"

DFDL Support

- OVERVIEW
- SCHEMA GENERATION
- SCHEMA EDITING
- SCHEMA TESTING
- SCHEMA LOADING

File Merge

- OVERVIEW
- USING FILE MERGE

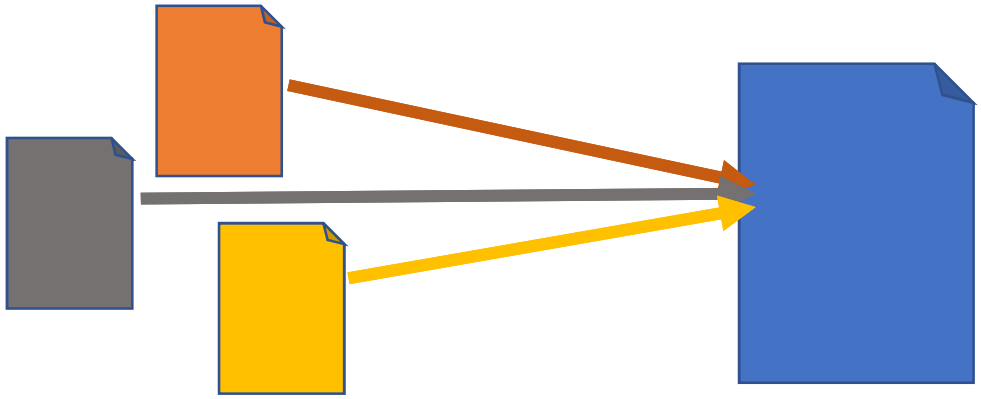
Other Updates

- C/C++ CODE FORMATTING
- C/C++ CODE SENSE
- PRODUCTION DUMP VIEWER
- MORE UPDATES

File Merge Overview

What is File Merge in zTPFGI?

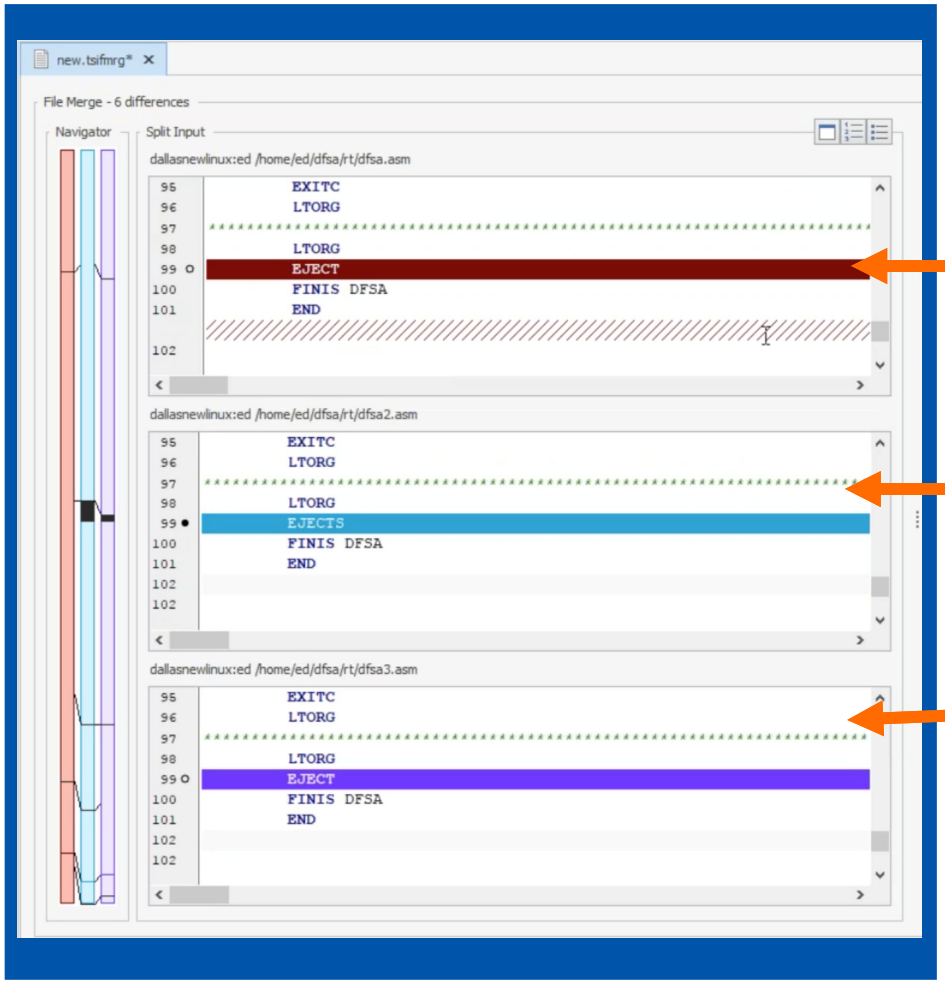
- File Merge allows the differences between two or three files to be merged together to produce a new version of the file



Why use File Merge?

- To incorporate custom changes that were made to a previous version of a source into a new version of that source

File Merge – The Input Files



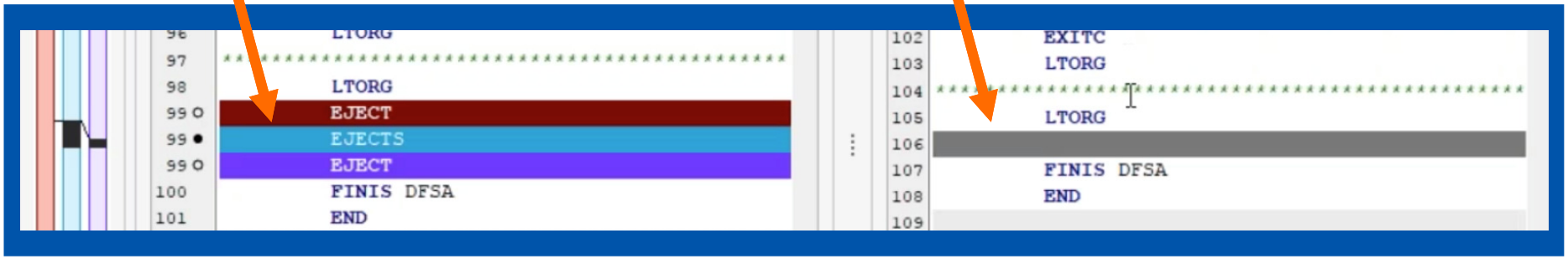
Up to three different versions of a source can be merged:

- File 1 (i.e. "base" – could be the most immediate parent of the other versions)
- File 2 (e.g. the version of the source with your previous custom changes)
- File 3 (e.g. the new version of the source released by the operating system/vendor)

File Merge – The Output

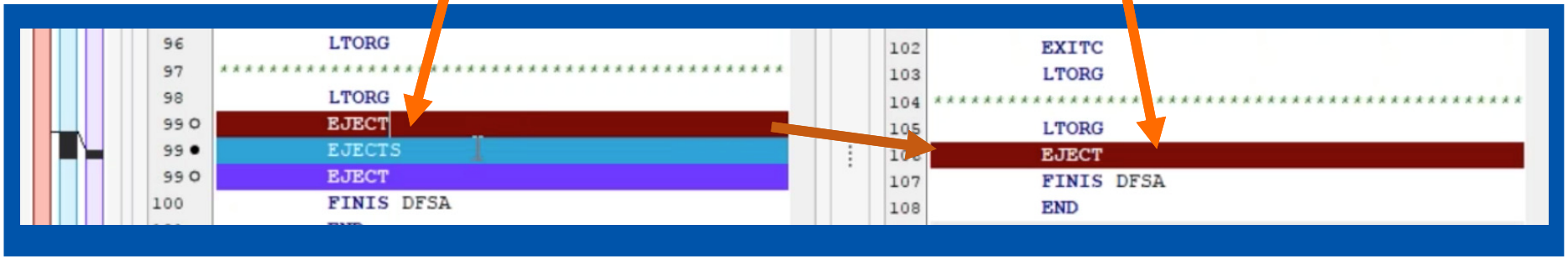
The differences between the three input files are grouped and color coded

The output file has insertion points for each of the groups of differences

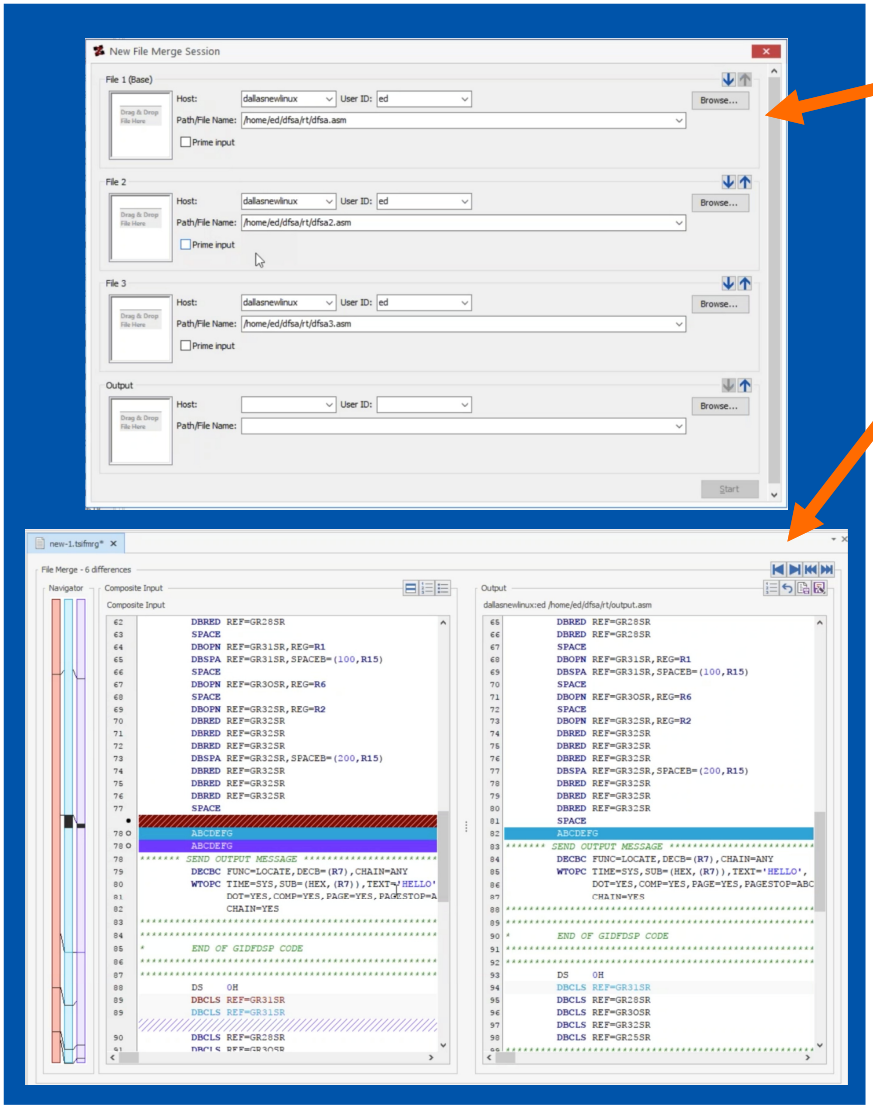


Users can click the difference that they want to move into that insertion point in the output

Once moved, the difference retains its color coding



File Merge – The Merge



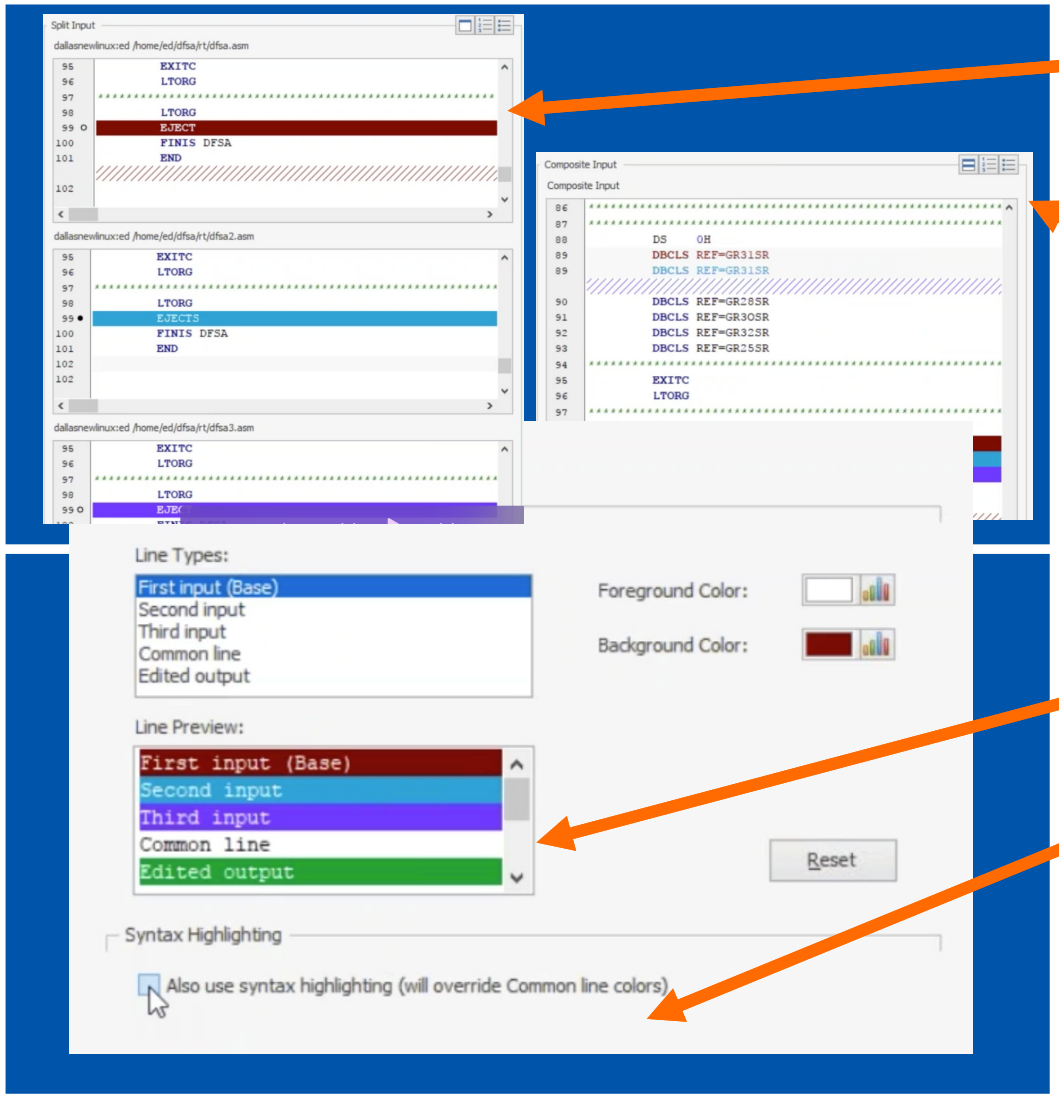
The user selects input files and an output location to start the file merge "session"

The file merge work area then opens as an editor in zTPFGI

Choices that the user makes are saved in a control file

A file merge can be continued across sessions of zTPFGI, and even passed to another user to continue the work

File Merge – Other Features



Input files may be shown in

- Split view or
- Composite view

The output may also be manually edited

The output file may start with blank insertion points or the insertion points may be prefilled with differences from one of the input files

A user may work on multiple file merges at the same time

Coloring is configurable

The user may choose to have syntax highlighting or not

DFDL Support

- OVERVIEW
- SCHEMA GENERATION
- SCHEMA EDITING
- SCHEMA TESTING
- SCHEMA LOADING

File Merge

- OVERVIEW
- USING FILE MERGE

Other Updates

- C/C++ CODE FORMATTING
- C/C++ CODE SENSE
- PRODUCTION DUMP VIEWER
- MORE UPDATES

C/C++ Code Formatting

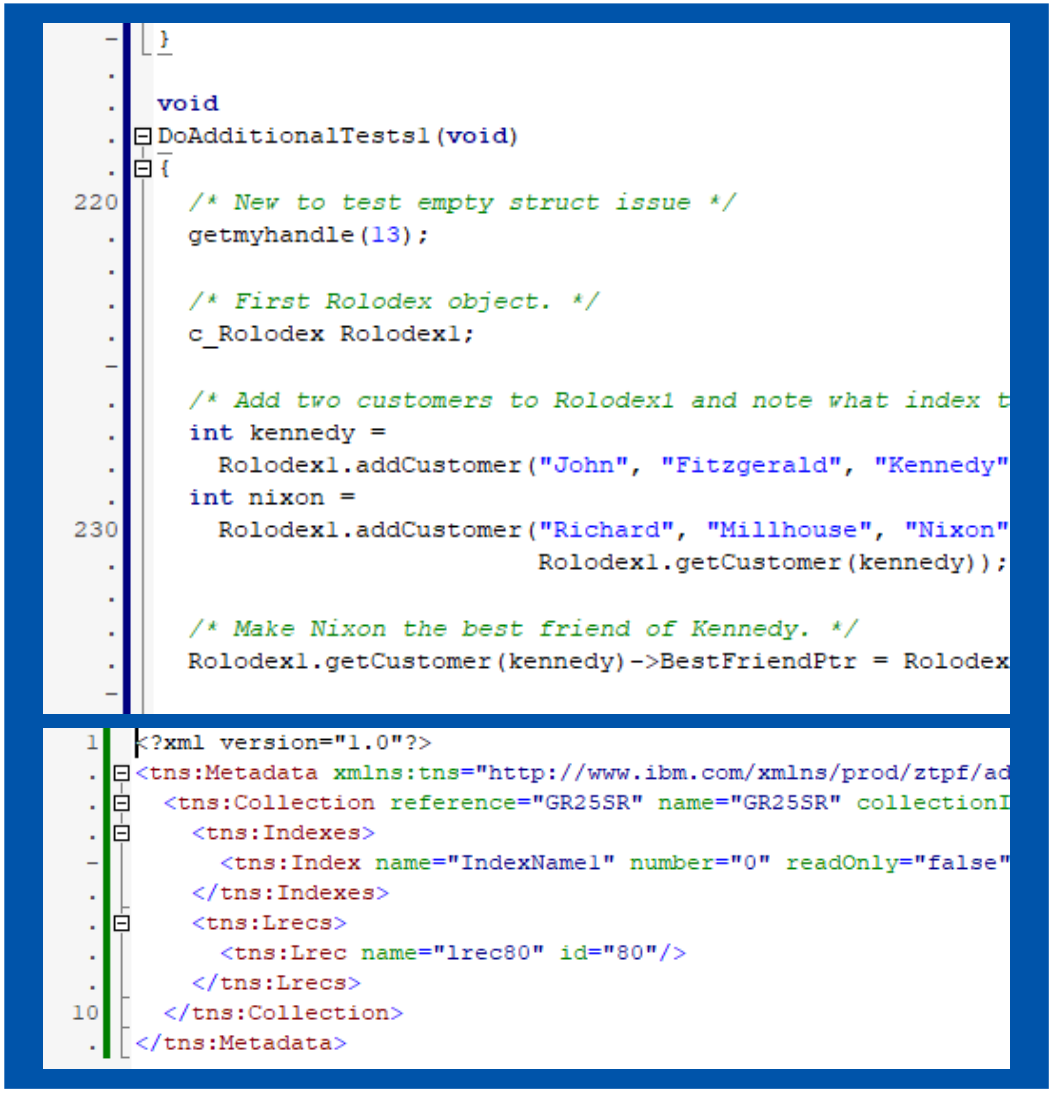
Quickly and easily format C/C++ layout so that it conforms to a specified standard

- Administrator-defined default format
- Pre-defined formats
- Developer-defined custom format

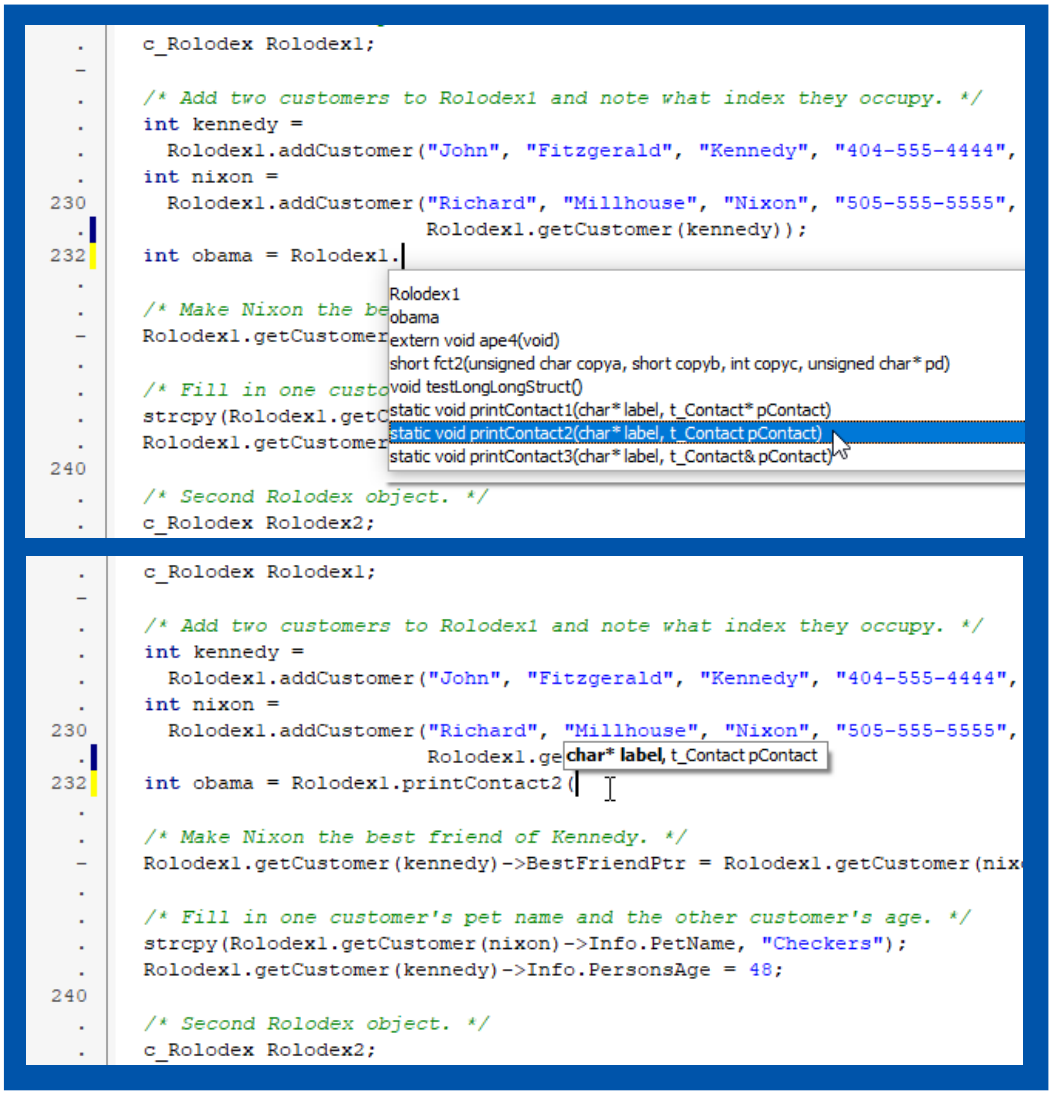
Why do we need Code Formatting?

- Aids in writing well-formatted code
- Aids in adherence to uniform coding standards across the team/organization
- Improves code readability

Can also format XML code



C/C++ Code Sense



Code completion assistance

Displays functions and variables that are in scope relative to cursor position

Simply press CTRL + SPACEBAR to show code completion suggestions

Select a suggestion to apply it

Includes Parameter Tooltips

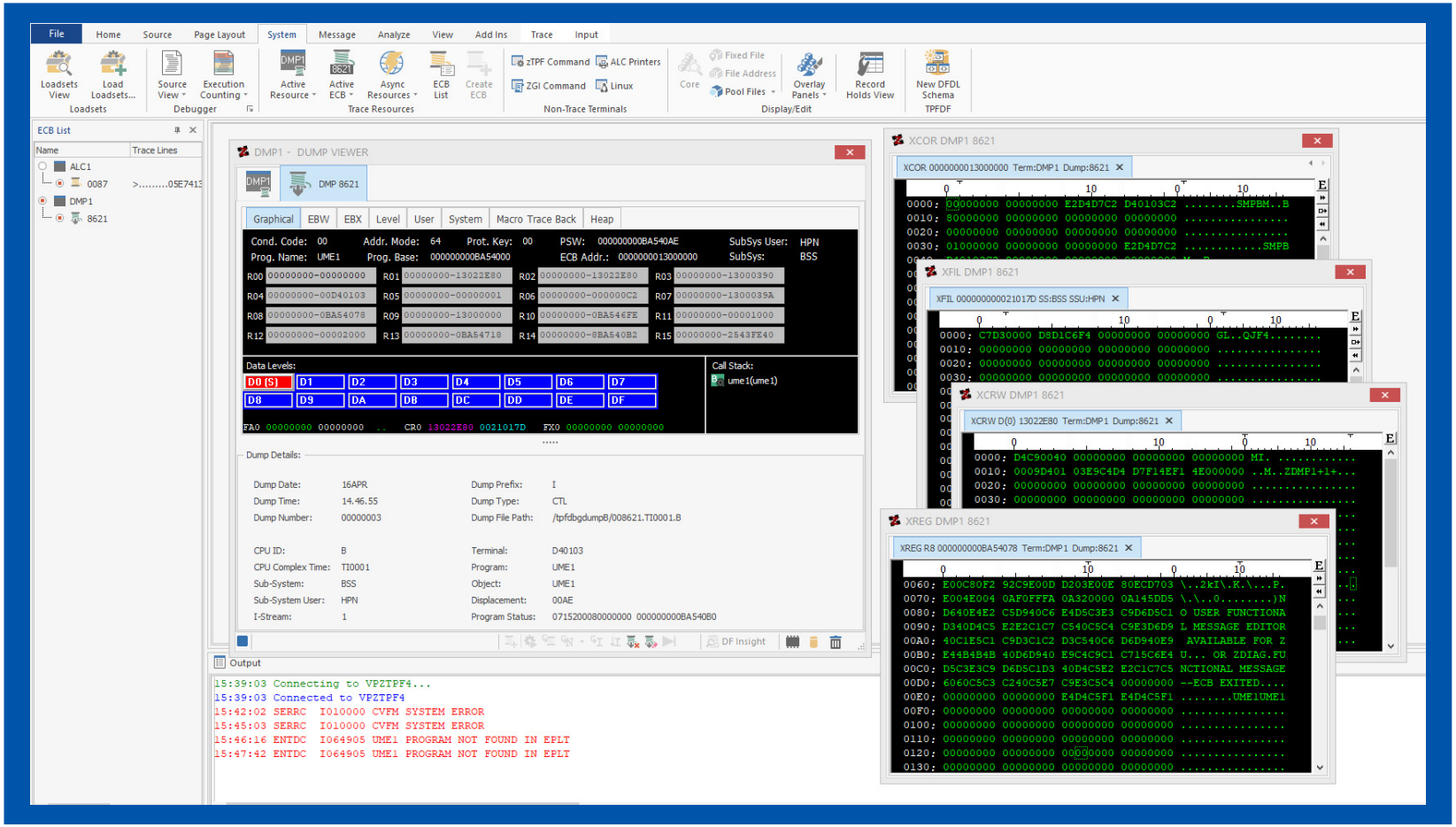
- Simply press CTRL + SHIFT + SPACE to display parameter suggestions when calling a function
- Option to turn on or off the automatic display of tool tips

Both code completion and parameter tooltips work with local & global variables, and with functions within the current file scope

Production Dump Viewer

View production system dumps

Same graphical interface as test system dumps



More in zTPFGI

#	Feature	Description
1	New Entry: ZTSOF ECB ALL	New entry to get Active ECB information for all products that are currently being used
2	New z/TPF instructions display in ECB Trace	When tracing a new ECB Instruction, the instruction details are now displayed in the ECB Window
3	Editor Modification Tag	A new zTPFGI editor feature to add a comment tag to every line that the user modifies in their source files
4	Project Refresh	Right-click a project in the Projects View and select this option to refresh the view of that project
5	Multiple-line comments in Assembler source code	Editor supports multiple-line comments in Assembler source files
6	SabreTalk Compile	Compile SabreTalk from within zTPFGI

More in zTPFGI

#	Feature	Description
7	Pool Usage Display	Shows the details of the pool file addresses that the z/TPF applications used and released using the GETFC and RELFC operations for a given user session
8	Macro Name for SVCs displayed in ECB Window	Source View Trace enhancement to update the appropriate macro name for SVC call macros such as DETAC, GETCC etc., in the trace lines of the ECB Window's Graphical tab
9	Asynchronous Trace Call Stack Enhancement	Updates call stack with previous caller information for assembler sources in the call stack section of the ECB Window when tracing an external ECB
10	Monthly report on Source Scan	Source Scan reports the code violations as error or warning messages based on the company preference, and logs them in the directory *pdwdata/ideserve/SourceScan*

More in zTPFGI

#	Feature	Description
11	Custom Loaddeck utility	The loaddeck utility allows the end users to right-click on one or more .SOs and create an OLDR file (load file) for multiple SOs at a time.
12	Load Support	Support automatic activation of loaddecks containing XML files
13	Java Syntax Highlighting	Syntax highlighting for Java files
14	Multiple level debug option for Source View tracing C/C++ macros	Enhanced Easy way to compile for debugging C/C++ macros
15	Object Insight	Shows header content from .o or .so files
16	Notify 2nd Time Compile/Build//Load	This feature prevents a second Compile, Build, Force Build, or Load command being executed on source or make files, if any of these actions are already underway on a file
17	Execution counting without stopping	Collect Execution Count statistics when tracing in source view without stopping in the code

More in zTPFGI

#	Feature	Description
18	zTREX enhancement	DF function call information for C segments has been added to the zTREX data collection
19	Save file to multiple locations	Save of library files can be propagated to multiple locations
20	Asynchronous Trace Support for C/C++ Functions	Asynchronous Trace - Option to monitor CPP functions
21	Catch System area overwrites	Modify debugger to catch overwrites of System area of ECB
22	Performance improvements for Asynchronous Trace	Support trace for RR & DIR entries
23	zTPFGI to share password for Systems that are sharing a password	With Linux LDAP, multiple Linux System share a password. zTPFGI now recognizes this

Updates in zRTF

#	Feature	Description
1	TM Record Logging in zRTF	New zRTF config setup allows user to capture TTY messages
2	TX, PJ Record Comparison in zRTF	The zRTF Compare functionality is extended to support the Database (DB) and Tape (TP) record comparison

THANK YOU