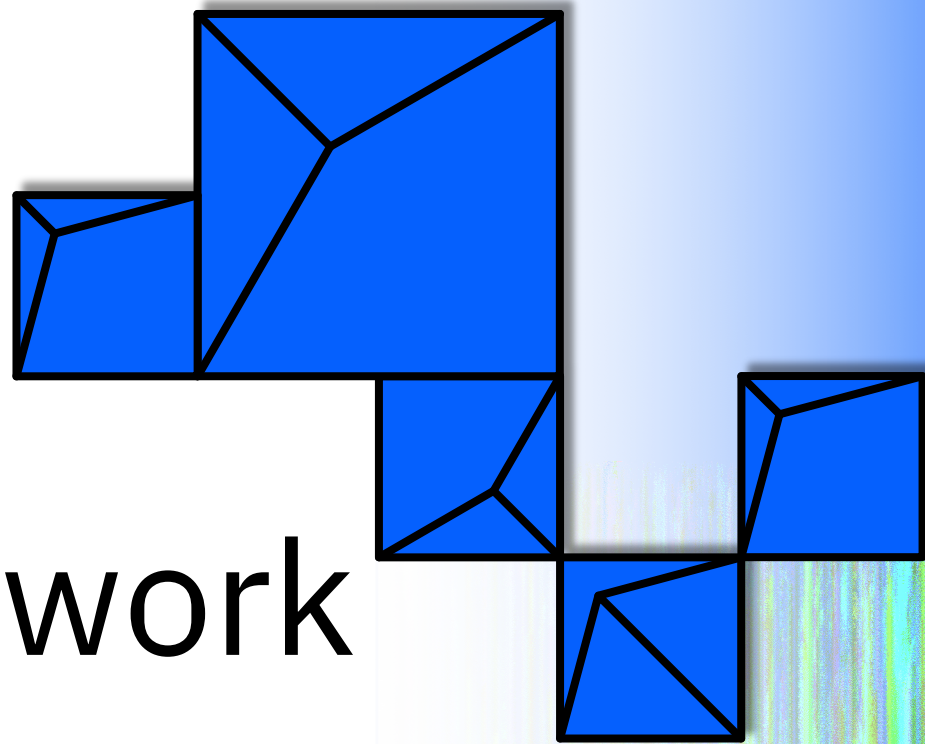


z/TPF Test Framework Enhancements

Daniel Gritter



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.

Agenda

Background

Problem Statement

Pain Points

Value Statement

Technical Details

Conclusion

What's next?

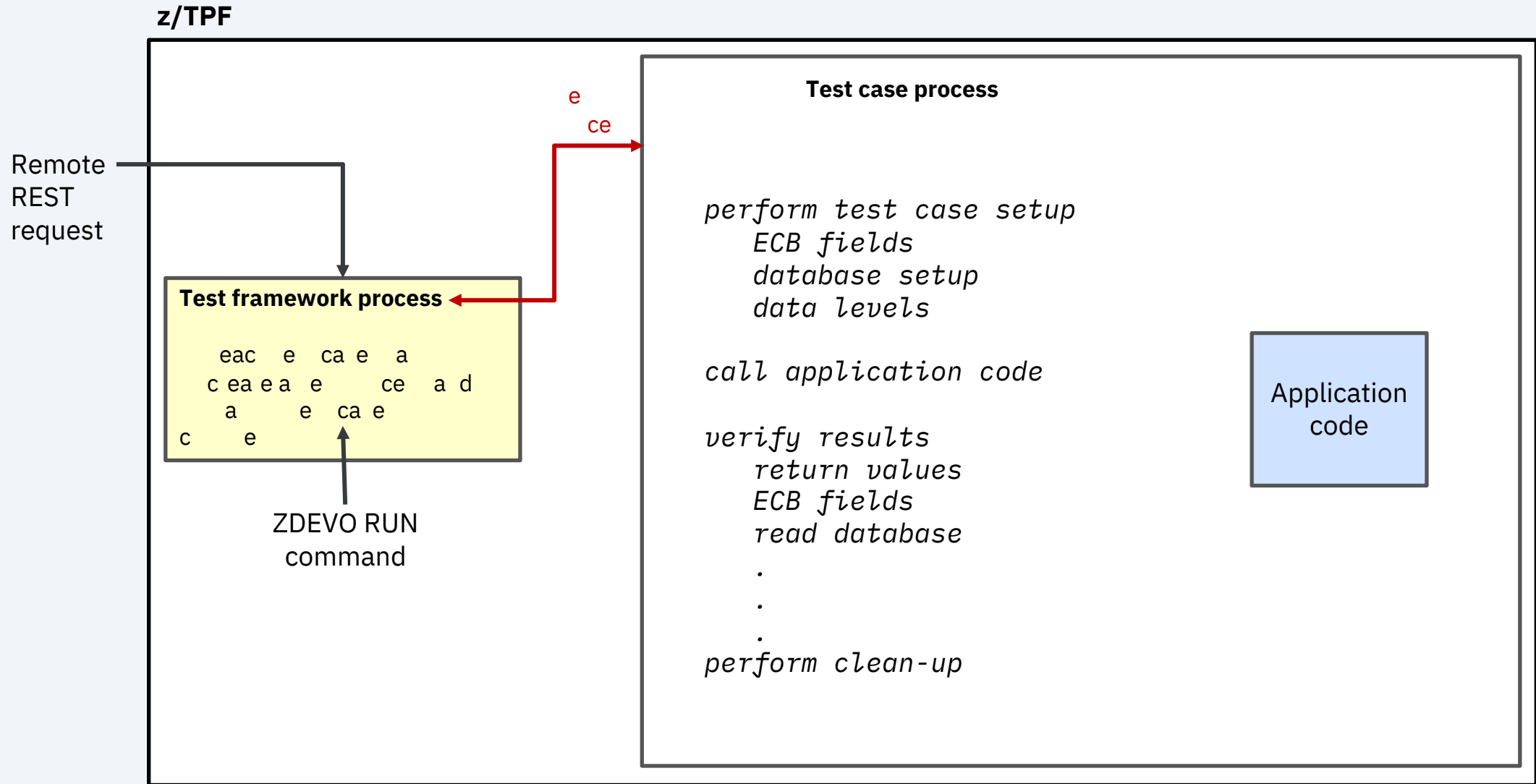
Background

- z/TPF Test Framework Phase 1 (PJ45217 + PJ43782, Aug 2018)
 - Provided a method create new (or convert old), self-validating programmatic test cases similar to other testing frameworks, like Google Test
 - Provided a set of C/C++ unique macros to assist in testing
 - Provided the ability to organize tests by namespace
 - Provided the ability to query and run test cases defined in the z/TPF Test Framework with an operator command (ZDEVO)

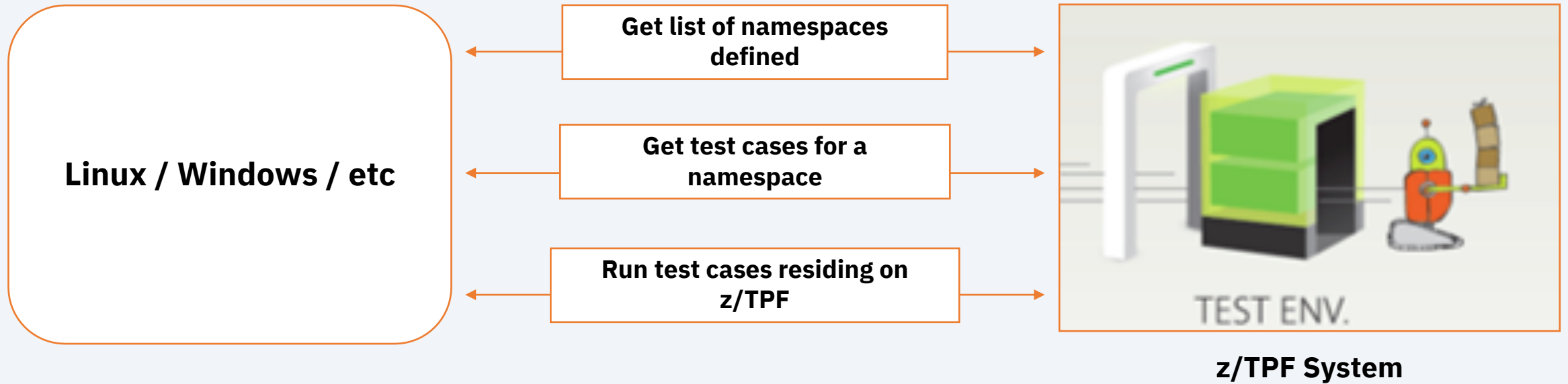
Background

- z/TPF Test Framework Phase 2 (PJ45488, Dec 2018)
 - Provided the ability to run z/TPF tests from a Java application (for example, JUnit) on a remote platform
 - Provided the ability to test code that requires remote invocation (REST, Mongo, etc.) along with local framework tests
 - Provided the ability to integrate into Open Tooling packages like Jenkins to facilitate testing in a DevOps environment

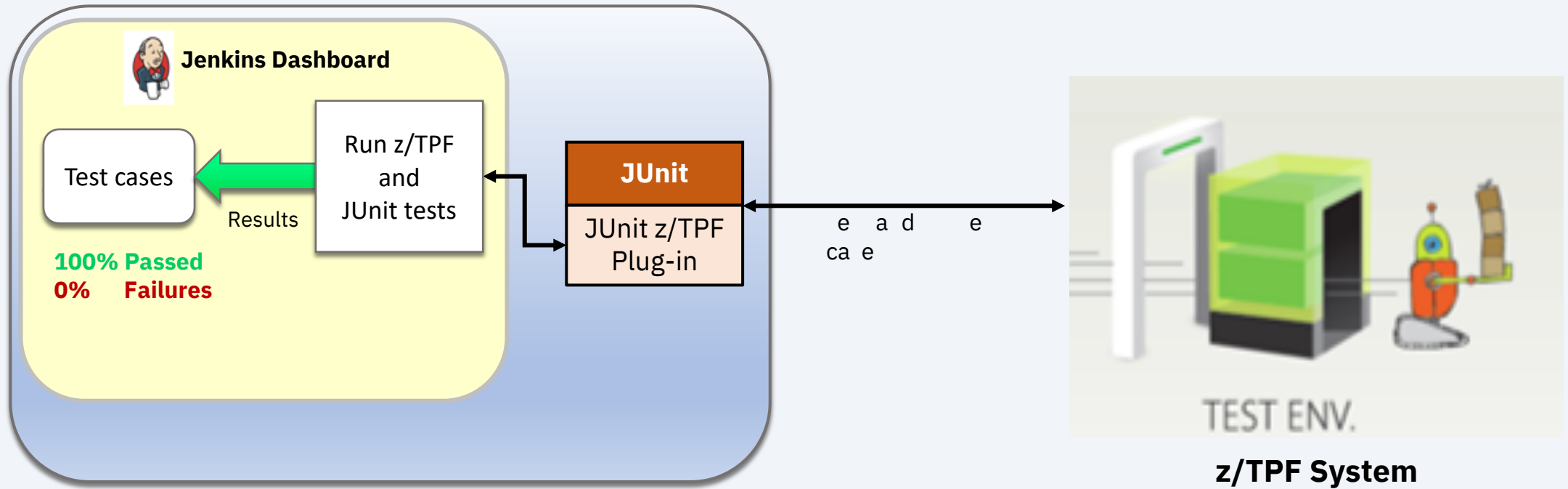
Background

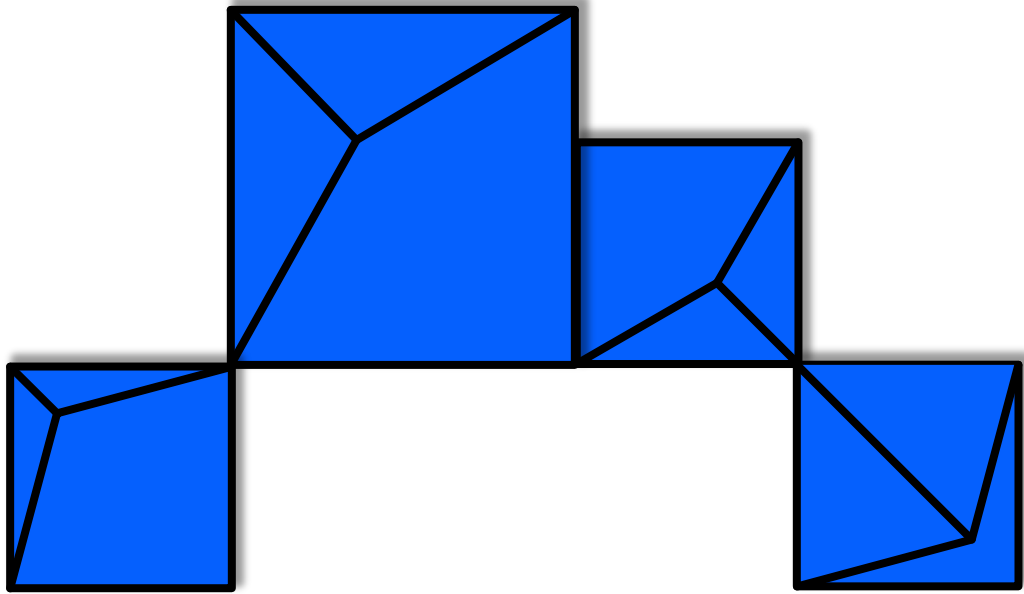


REST interface



Integrating into DevOps platforms (JUnit interface)





z/TPF Test Framework Scaffolding support

Problem Statement

As part of creating a test case, it is useful to customize system / application behavior to simulate error scenarios or unique code paths

Pain Points

- Test of application code dependent on access to external systems
- Tests of application code dependent on specific output from z/TPF system or user functions
- Cannot easily test z/TPF application code that does not follow the call-return model

Value Statement

z/TPF Test Framework Scaffolding support (PJ45801) provides a method to automate more test cases and test more complex application code

Technical Details

- **Override functions**
 - Provides a test framework facility to customize the behavior of an application invoked routine
- **Intercept functions**
 - Provides a way to define boundaries or configure behavior of test framework based on scope

Technical Details

- Using Override Functions
 - TPF_TC_OVERRIDE() api
 - Use function name string to determine overridden function
 - Use scope to limit invocation
 - Function or 4-character program name
 - Custom data pointer / length to maintain state or pass custom data to override function

Technical Details

- Using Intercept Functions
 - TPF_TC_INTERCEPT() api
 - Use function name string to determine overridden function or 4-character program name
 - Use scope to limit invocation
 - Function or 4-character program name
 - Custom data pointer / length to maintain state or pass custom data to override function
 - Function call, Function return, or Function leaving options
 - Leaving takes affect on call out or return from

Technical Details

- `TPF_TC_COMPLETE()`
 - Used to tell the test framework test has finished without return
 - Combined with intercept function can be used to support call / exit or call / drop type of applications

Conclusion

- What's New: PJ45801
 - Intercept Functions
 - Override Functions
 - Eliminate test case code in production build
 - Additional application programming models

What's next?

- The initial framework deliverable was created by the z/TPF lab for use by the lab.
- Due to low/slow adoption, we need to understand whether the strategy used for system testing is still applicable and the right approach for application testing
- The sponsor users process has identified some key area concerns around the approach that we want to explore with the larger TPF community.
- What else is keeping you from using / finding value from the z/TPF Test framework? Join us as a sponsor user!

What's next?

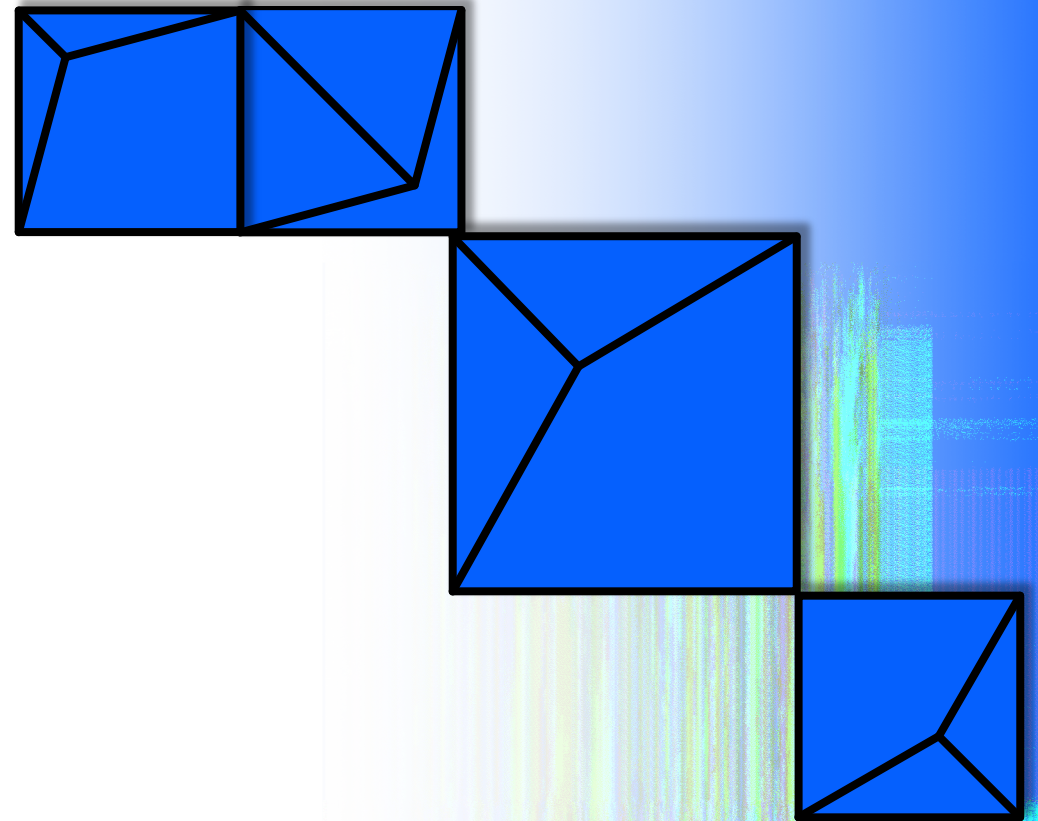
- Test cases written in source files have to go through source management / update process
 - All code loaded to z/TPF system have very strict controls in some shops
- Test cases in dedicated shared objects cannot access private shared object functions, requiring them to link into the shared object, subject to runtime / production overhead
 - Currently provided a compile time switch to exclude test case code as part of a “production” build.
- Need enhancements to handle assertions more cleanly
 - Currently only have TPF_TC_ERROR to manually perform assertion testing
- Override routines have to be manually created – would prefer to “record” current behavior vs write a function

What's next?

- Investigating feasibility for “recording” a test case, instead of coding a test case.
- Focus on identifying and simplifying the scaffolding process to avoid the costly setup
- Intention to help better understand / identify interfaces
 - One of the big pain points we've heard is how to identify the “units” for unit testing
- More examples / sample test cases

Thank You

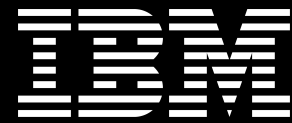
Questions? Comments?



Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

Question	Answer
Q: Assume test cases can be built to run that depend on previous test cases? i.e.: build on one another.	A: We do not recommend creating order dependencies for test cases – it is possible to call testcases or create setup functions that are reused between testcases instead.
Q: Does the T{F Test framework work with Assembler programs as well or it's best suited for TPF C/C++ programs only ?	A: The test automation framework is designed to work with BAL programs. The presentation references a way to handle older programs that do not use (a more recent) API call-return interface. This would allow testing of BAL programs that do NOT return but exit



Trademarks

IBM, the IBM logo, ibm.com and Rational 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.