z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM

IBM Software Group

# *TPF Users Group Fall 2005*

# TPFDF and z/TPFDF Status Update

Name : Kevin Jones
Venue : Database Subcommittee

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

Any references 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

- TPFDF 1.1.3
  - PUT 20
  - Users Group Requirements
- z/TPFDF 1.1
  - Packaging and Build
  - z/TPFDF Support for z/TPF Features
  - Internal Changes
  - Enhancements

# TPFDF 1.1.3 PUT 20

- Generally Available in June 2005
- Includes APARs using new "PK" prefix (for example, PK00945)
- Total of 59 APARs (56 with code changes, 3 documentation-only), including:
  - Single Source Enhancement - PQ91889
    - provides wrappers that allow C/C++ language headers to be called using their z/TPF names
      - for example, C$SW00SR is now also called C_SW00SR
    - tools are provided to assist with these application updates

# TPFDF 1.1.3 PUT 20 *(continued)*

- FARF6 support - PQ94935
- Co-requisite support for TPF Continuous Data Collection (CDC) - PQ95747
- TPFDF compatibility with the z/OS 1.5 compiler - PQ97762

# TPFDF 1.1.3 Users Group Requirements

- DF00159 - use a "key block" only if more than 6 keys are defined
  - APAR is PK06268, which is now available electronically (planned for PUT 21)
  - Object code representing the keys will be placed as follows:
    - KEYn parameters: SW00SR work area SW00KL1
    - Keylists:
      - 5 or less keys: SW00SR work area SW00KL1
      - 6 or more keys: "key block" attached to SW00SR
        - exactly 6 keys are placed in the "key block" since they may not fit in SW00KL1
        - z/TPFDF places exactly 6 keys in SW00KL1 as specified in the requirement

# TPFDF 1.1.3 Users Group Requirements *(continued)*

- DF00172 - ZUDFM should provide a summary of DBDEF option bits
  - APAR is PK10080, which is now available electronically (planned for PUT 21)
  - new ZUDFM parameters will implement this support
    - for example, *ZUDFM DEF OPT/OP4* will display OP4 settings for all files
    - optional mask can be used to display only files meeting specified attributes
      - for example, *ZUDFM DEF OPT/OP3/*****Y*** will display all files that use B+TREE indexing (#BIT5 of OP3)

# TPFDF 1.1.3 Users Group Requirements *(continued)*

- DF00175 - status display for TPFDF Data Collection
  - APAR is PK10079, which is now available electronically (planned for PUT 21)
  - new command is ZUDFC STATUS

# z/TPF Database Facility Enterprise Edition Version 1 Release 1

- Commonly called z/TPFDF
- Product is separate from z/TPF 1.1
  - Separate product number
  - Separate PUTs
- z/TPF 1.1 requires z/TPFDF 1.1
  - More z/TPF utilities will use z/TPFDF
    - Recoup
    - Debugger dump manager (ZDDMP)
    - Continuous data collection (CDC)

# z/TPFDF Packaging and Build

- Code will ship as full source
  - part of the z/TPF HFS hierarchy
  - no sequence numbers
- Documentation is part of the z/TPF Product Information Center (InfoCenter)
- Uses the same build tools as z/TPF

# z/TPFDF Support for z/TPF Features

- 64-bit and baseless applications
  - z/TPFDF APIs (macros and functions) can be issued from 64-bit and baseless applications
  - any data passed to z/TPFDF must be below the 2GB line
    - an exception is the use of global or static variables in C
  - new AMODE parameter is provided on most assembler APIs
    - overrides the addressing mode specified by the BEGIN macro
  - Structured Programming Macros (SPMs) are not supported in 64-bit or baseless applications

- Baseless environments include:
  - base register other than R8
  - multiple base registers
  - no base registers

# z/TPFDF Support for z/TPF Features *(continued)*

- I-stream scheduler
  - z/TPFDF central database routines have no I-stream affinity (an ECB's I-stream can change following a loss of control)
  - return is always to the original I-stream if the application caller has an I-stream affinity
- Application time-out value
  - z/TPFDF central database routines will honor the timeout of the application caller
- Dump trace groups
  - z/TPFDF central database routines will be included in a trace group called IZTPFDF
- z/TPFDF utilities can be used in 1052 state when GFS is active

# z/TPFDF Internal Changes

*Note: none of the following internal changes require application updates*

- The TPFDF "fastlink" mechanism is obsolete in z/TPFDF
  - provided program linkage:
    - between applications and central database programs
    - within central database programs

- Standard enter/back calls are now used between applications and central database programs

- Program linkage within central database programs now uses:
  - ENTRC TYPE=TPFDF *and*
  - BACKC TYPE=TPFDF

# z/TPFDF Internal Changes *(continued)*

- Use of standard enter/back:
  - allows z/TPFDF to exploit z/TPF current and future abilities such as the I-stream scheduler
  - no special considerations for the TPF Debugger
  - removes E-type loader restrictions on the loading of z/TPFDF programs
  - simplifies the maintenance and complexity of z/TPFDF
- Functions in the TPFDF Common Entry Point (CEP) are now:
  - handled by z/TPF enter/back linkage, *<u>or</u>*
  - incorporated into the central database programs

# z/TPFDF Internal Changes *(continued)*

- z/TPFDF central database programs have been renamed:
  - Programs previously named UWA*x* are now UA*x*0
  - Programs previously named UWB*x* are now UB*x*0
  - Allows the use of transfer vectors to replace "fastlink cases"

- Central database programs are built and loaded as part of a single shared-object called UTDF

- Example
  - Program UWBB contained four fastlink cases (0 through 3)
    - replaced in z/TPFDF with program UBB0
    - includes transfer vectors UBB0, UBB1, UBB2 and UBB3
    - UBB0 is built and loaded as part of shared-object UTDF

# z/TPFDF Internal Changes *(continued)*

- SW00SR redesigned
  - No longer uses chained core blocks
  - Uses ECB heap storage
  - Fields reorganized
  - Improves maintainability

- Key Processing updates
  - SW00SR area used to processes 6 or less keys
    - using keylists or KEYn parameters
  - "Key block" used to process more than 6 keys
  - Satisfies TPFUG requirement DF00159

# z/TPFDF Enhancements

- New user exit for configuration values in ACPDBE
  - Allows customers to add equates, such as for user-defined algorithms, for use by z/TPFDF
  - Satisfies TPFUG requirement DF00169

- ZUDFM MLS enhancements
  - Offline process eliminated
  - Debug Data loaded to online system
    - can be obtained through DBDEF segments
  - All other ZUDFM MLS externals are unchanged
  - No migration considerations for existing MLS data

# z/TPFDF Enhancements *(continued)*

- Eliminate automatic display of the entire subfile for ZUDFM commands:
  - FAD
  - ADD
  - REPLACE
  - DELETE
    - Addresses TPFUG requirement DF00171

- Data Collection status now available using ZUDFC STATUS
  - satisfies TPF Users Group requirement DF00175

- New ZUDFM parameters available to display DBDEF option bits
  - satisfies TPF Users Group requirement DF00172

# z/TPFDF Enhancements *(continued)*

- Recoup will display a specific message at the beginning and end of chainchase for each DBDEF
- ZUDFM RESTRICT will be easier to use
  - only display information for the command whose restrictions were modified
  - allow all restrictions to be displayed, or only restrictions for one command
- DBDEF segments can exceed 4K in size
- Formatted SW00SR in dumps

# Question and Answer

# Trademarks

IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

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 cost 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 yo 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.