



IBM Software Group

TPFDF Status Update

Database/TPFDF Subcommittee

Daniel Jacobs
October 2004

AIM Core and Enterprise Solutions

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 FARF6 Support
- z/TPFDF 1.1
- Future Database Candidates

TPFDF FARF6 Support

- Allows use of FARF6 and 8-byte file addresses in TPFDF databases
- Customers are not required to update existing applications or databases.
- First deliverable is TPFDF APAR PQ74483 / TPF APAR PJ29213
 - ▶ Shipped on TPFDF PUT 19, TPF PUT 18
 - ▶ Provided "infrastructure" support
- Second deliverable will be TPFDF APAR PQ94935 / TPF APAR PJ29582
 - ▶ **Now available! Contact your customer service representative (CSR)!**
 - ▶ Enables the use of FARF6 addresses in TPFDF databases
- Two ways to use FARF6 in your databases
 - ▶ Migrate existing databases / applications
 - ▶ Define new databases / applications
- New TPFDF Migration Guide contains details

TPFDF FARF6 Support (*continued*)

- Migrating an existing database
 - ▶ Enter ZMODE T to enable the TPFDF FARF6 migration switch
 - Requires ZMODE 6 to enable TPF FARF6 support
 - ▶ Migrate applications
 - Update file address fields to be 8-bytes in size
 - Use new 8-byte file address parameters on TPFDF APIs
 - ▶ Migration database definitions
 - Update DBDEF with appropriate FARF6 parameter settings
 - Includes DBDEF parameter updates for recoup/CRUISE
 - ▶ Migrate the database
 - Set up CRUISE table to pack database
 - Specify subfiles to be migrated
 - Run CRUISE

TPFDF FARF6 Support (*continued*)

- Staged migration supported
 - ▶ Use 8-byte file address files in headers and trailers
 - F6HDR DBDEF parameter
 - ▶ Use 8-byte file addresses in TPFDF-managed index references
 - F6MIG and F6MIGIDX DBDEF parameters
 - Only required if using FARF6 addresses in prime records
 - ▶ Use FARF6 addresses in overflow records
 - F6OFP DBDEF parameter
 - ▶ Use FARF6 addresses in prime (and overflow) records
 - F6PRP DBDEF parameter
- Database migration automatically expand headers or index references, as specified
- Subfiles can contain mix of FARF3/4/5 and FARF6 addresses
- Record formats can be fallen back using same migration mechanism

TPFDF FARF6 Support (*continued*)

- Using FARF6 in a new database
 - ▶ Enter ZMODE T to enable the TPFDF FARF6 migration switch
 - Requires ZMODE 6 to enable TPF FARF6 support
 - ▶ Write new DSECTs
 - Specify 32 byte standard header
 - Specify 16 byte TPFDF header
 - Specify 10-byte index references
 - ▶ Write new DBDEFs
 - Specify appropriate FARF6 parameters
 - ▶ Write new applications
 - Use new 8-byte file address parameters on TPFDF APIs
 - ▶ Run your application!

z/TPF Database Facility Enterprise Edition Version 1 Release 1

- Common name: z/TPFDF 1.1, or 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 services will use z/TPFDF
 - Recoup
 - Debugger dump manager (ZDDMP)
 - Continuous data collection (CDC)
- Code will ship as full source
 - ▶ Part of z/TPF HFS hierarchy
 - ▶ **No** sequence numbers
- Documentation is part of z/TPF Product Information Center (InfoCenter)
- Uses same build tools as z/TPF

z/TPFDF 1.1 (*continued*)

- Exploits new z/TPF features
 - ▶ I-stream scheduler
 - ▶ Application time-out value
 - ▶ Dump trace groups
 - ▶ 64-bit applications
 - ▶ Baseless environments
 - Base register other than R8
 - Multiple base registers
 - No base registers

z/TPFDF 1.1 (*continued*)

- SW00SR redesigned
 - ▶ No longer uses chained core blocks
 - Uses ECB heap storage
 - ▶ Fields reorganized
 - Improves performance
 - Improves maintainability
 - ▶ Formatted SW00SR in dumps
- Key Processing updates
 - ▶ SW00SR area used to processes 6 or less keys
 - ▶ Core block used to process more than 6 keys
 - ▶ Satisfies TPFUG requirement DF00159
- New user-defined equates user exit in DFUEX
 - ▶ Allows customers to add equates, such as for user-defined algorithms, for use by z/TPFDF
 - ▶ Satisfies TPFUG requirement DF00169

z/TPFDF 1.1 (*continued*)

- ZUDFM MLS changes
 - ▶ Offline process eliminated
 - ▶ DSECT online symbolic data (ADATA) loaded to online system
 - List DSECTs in new DFUEX user exit
 - Load DSECTs by using DBDEFs
 - Use "ADATA" option when assembling to generate symbolic data
 - ▶ All other ZUDFM MLS externals are unchanged
 - ▶ No migration considerations for existing MLS data
- Eliminate automatic display of entire subfile
 - ▶ ZUDFM FAD
 - ▶ ZUDFM ADD
 - ▶ ZUDFM REPLACE
 - ▶ Addresses TPFUG requirement DF00171
- TPFDF utilities can be used in 1052 state when GFS is active
- Fully compatible with z/TPF Debugger and z/TPF Performance Analyzer

Future Database Candidates

- IBM is actively investigating new projects to standardize database access in TPF
- Two major areas need to be considered:
 - ▶ Existing TPFDF data needs to be represented by industry standard models, **for example**:
 - provide a framework for mapping TPFDF data to C++ objects
 - LRECs may be viewed as objects
 - C++ application may read an LREC as an object, manipulate it
 - Standard APIs would be used to make object persistent
 - model TPFDF data as relational tables that can be accessed through SQL or a JDBC driver
 - provide a new set of APIs conforming to SQL or JDBC standards to access the data
 - Offline translator would convert application to use TPFDF APIs

Future Database Candidates

- ▶ Implement a native relational database manager for new databases
 - New database manager
 - Could be UDB, MYSQL, or other solutions
 - Underlying backing database would be transparent to application

IBM needs significant customer feedback in this area. Please contact us if you have any input.

TPFDF Status Update

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.

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

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