z/TPF V1.1

2013 TPF Users Group

z/TPFDF Status Update

Chris Filachek
Database/TPFDF Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1

Agenda

- **z/TPFDF Enhancements**
  - z/TPFDF PUT 10 Enhancements
  - Future z/TPFDF Enhancements
- **z/TPFDF TPFUG Requirements Update**
  - Requirements with Changed Status
  - Accepted Requirements
  - Top z/TPFDF TPFUG Requirements

z/TPFDF PUT 10 Enhancements

- **PM67148: Additional SPM error checking**
- **PM72950: Support 9-digit ordinal displays**
- **PM77843: Improve data collection accuracy**
- **PM81536: Validate ID upon Release**
- **PM83026: C API Improvements**

PM67148: Additional SPM error checking

- **Provide additional assembler time structured program macro (SPM) error checking**
  - Flag nested subroutines as warnings
  - Missing end for “case”, “do”, and “if” structures flagged with appropriate message
- **Improves application quality**
- **Aides in debugging applications**

PM72950: Support 9-digit ordinal displays

- **Expand z/TPFDF utility displays to accommodate databases with up to 1 billion subfiles**
  - CRUISE
  - ZUDFM
- **Other minor adjustments made to displays to improve readability / usability**
  - For example, eliminate leading zeros in ordinal numbers
- **z/TPF APAR PJ40632 addresses similar issues with the z/TPF recoup status display**

PM77843: Improve data collection accuracy

- **Some counts incremented even when corresponding z/TPF macro wasn't called**
  - Counts updated when *case* containing macro was called
- **Some z/TPF macro counts not incremented enough**
  - RELFC count might be incremented by one when entire chain is released
- **All data collection rates reported about 5% higher than actual**

PM77843: Improve data collection accuracy

- **All counts are now incremented based on actual z/TPF macro calls**
- **Some z/TPF macro counts may have (potentially significantly) higher or lower values with APAR applied**
  - This must be considered when comparing data collection results with and without this APAR applied

PM81536: Validate ID upon Release

- **z/TPFDF should verify file IDs before releasing pool records**
  - Much of z/TPFDF already did this, even before the option was available on the RELFC/relfc APIs
  - B+Tree and LLR routines have been updated to validate the file ID before releasing nodes, LLIBs, and LLDBs
- **DB0100 system error online message enhanced to indicate which I/O error occurred**
  - May eliminate need to process dump
- **Addresses TPFUG requirement DF08189S**

PM83026: C API Improvements

- **Support T-types in C**
  - New APIs to manage T-type LRECs directly instead of manipulating the underlying W-type file
  - Improves application readability
  - Makes applications less prone to error
- **Expand key list facility in C**
  - New API to save search argument values internally
  - Eliminates problem with search argument variable scope
  - Improves application reliability
- **Addresses TPFUG requirement DF00152 (in part)**

PM83026: C API Improvements – T-types in C

- **New APIs**
  - dfadd_ttype()
  - dfdel_ttype()
  - dfred_ttype()
- **Programming considerations**
  - When providing W-type refname, underlying W-type file must be:
    - Opened in advance, or
    - Included in list of files in uak0.asm to automatically open
- **No migration considerations**
  - Existing applications can still use dfadd() with DFADD_NULLREC to create a work area for a T-type LREC in the underlying W-type file

PM83026: C API Improvements – T-types in C

- **Examples**
  - Add T-type LREC to common underlying W-type file GW01SR

```
t_rec_ptr = dfadd_ttype(“IT00DF  “, sizeof(struct IT00DF),
NULL);
```

  - Read T-type LREC contained in underlying W-type file IW00DF

```
t_rec_ptr = dfred_ttype(“IT00DF  “,“IW00DF  “);
```

PM83026: C API Improvements – Key list handling

- **New APIs**
  - dfkey_save()
  - dfkey_nbr_save()
- **Programming considerations**
  - All dfkey() functions store key instructions in SW00SR
  - New APIs also save contents of search fields in SW00SR
    - Allows applications to release search arg storage locations
    - Can’t change search argument without issuing dfkey() again
- **No migration considerations**
  - Existing applications can still use existing dfkey() functions

PM83026: C API Improvements – Key list handling

- **Examples**
  - Activate keys and save contents of search arguments
    ```
    dfkey_save(file_ptr, &keys);
    ```
  - Activate keys, set number of keys and save contents of search arguments
    ```
    dfkey_nbr_save(file_ptr, &keys, 7);
    ```

Future z/TPFDF Enhancements*

- **DBDSP to file system (TPFUG req't DF08187S)**
  - Write z/TPFDF file contents to z/TPF file system
- **Service Data Object (SDO) Usability Improvements**
  - Generate multiple messages when validating metadata
  - Allow longer z/TPFDF DSECT names to be used
- **Enhanced z/TPFDF Structured Programming Macro (SPM) support**
  - 64-bit support
  - Baseless support

* All plans subject to change

Future z/TPFDF Enhancements*

- **Enhanced Multiple LREC Buffer Support**
  - Read/traverse individual LRECs in a buffer?
  - Add/delete individual LRECs to/from buffer?
  - Use keys?
- **z/TPFDF files in core**
- **Data Analytics and Offload**

* All plans subject to change

Agenda

- **z/TPFDF Enhancements**
  - z/TPFDF PUT 10 Enhancements
  - Future z/TPFDF Enhancements
- **z/TPFDF TPFUG Requirements Update**
  - Requirements with Changed Status
  - Accepted Requirements
  - Top z/TPFDF TPFUG Requirements

Requirements with Changed Status

- **#3: Multi-LREC Buffer Delete Enhancement (DF12197)**
  - Was: New
  - Now: Not Likely
- **#5: Multi-LREC Allow Record Access (DF12198)**
  - Was: New
  - Now: Likely

* All plans subject to change

Requirements with Changed Status

- **#8: Allow Optimistic Read Ahead (DF12195)**
  - Was: New
  - Now: Likely
- **#10: Allow Parallel I/O with LRECs (DF12196)**
  - Was: New
  - Now: Not Likely

* All plans subject to change

Requirements with Changed Status

- **Validate ID upon Release (DF08189S)**
  - Also described as “RELFC API for TPFDF”
  - Was: Accepted
  - Now: Available
- **TPFDF C API Improvements (DF00152)**
  - Was: Accepted
  - Now: Available

* All plans subject to change

Accepted Requirements

- **TPFDF Display Command Directed to the TPF File Systems (DF08187S)**

Top z/TPFDF TPFUG Requirements

Rank	Req Num	Description	Was	Now
1	DF05182F	Export LRECs to XML	Likely	Likely
2	DF00079	In-core records	Not Likely	Not Likely
3	DF12197	Multi-buffer delete	**New**	**Not Likely**
4	DF08188S	New CRUISE targets	Likely	Likely
5	DF12198	Multi-buffer rec. access	**New**	**Likely**
6	DF00153	C++ APIs	Likely	Likely
6	DF08191F	Error checking	Likely	Likely
8	DF12195	Optimistic read ahead	**New**	**Likely**
9	DF08186S	Processor unique struct	Likely	Likely
10	DF12196	Parallel I/O with LRECs	**New**	**Not Likely**

Trademarks

- **IBM, the IBM logo, and ibm.com 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.**
- ***(Include any special attribution statements as required – see Trademark guidelines on https://w3-03.ibm.com/chq/legal/lis.nsf/lawdoc/5A84050DEC58FE31852576850074BB32?OpenDocument#Developing%20the%20Special%20Non-IBM%20Tr)***

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.**