



IBM Software Group

*25th ALCS User Group Meeting, Dubai, 12-14 June 2007*

# TPFDF Status Update

Mike Hannaford  
*alcs@uk.ibm.com*



# Legal Notices

## Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

RACF*	DB2*	On Demand Business logo*
HiperSockets	Hiperspace	IBM*
IBM logo*	eServer	Multiprise*
MVS	OS/390*	System/390*
System/360	Websphere*	VTAM*
z/Architecture	z/OS*	zSeries*

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

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

## Notes

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 disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.



# Agenda

- TPFDF PUT 21 - enhancements not presented at the last ALCSUG
  - ▶ L3 Support
  - ▶ Empty Subfile Checks
  - ▶ Duplicate Reference Name Detection
  
- TPFDF PUT 22 Review
  
- Maintenance Stream Changes
  
- Questions



## L3 Support

- TPFDF 1.1.3 has always allowed *customer* databases to be defined using an L3 (4000 byte) block size
- However, TPFDF 1.1.3 *system* databases have always used L4 (4095 byte) block sizes
  - ▶ If all customer databases are L3, this requires customers to:
    - generate L4 block sizes for use only by the TPFDF product, or
    - modify TPFDF to use L3 blocks for system databases
- TPFDF has been updated to allow customers to select an L3 or L4 block size for use by system databases
  - ▶ APAR PK20320 on TPFDF PUT 21



## L3 Support

- New configuration options have been provided in ACPDBE, C\$ACPDBE and DBLCL to allow customers:
  - ▶ to define TPFDF system databases with an L3 size, and
  - ▶ to set the size of the DBIFB block (which contains SW00SR areas) to L3
  
- No migration path is provided for databases or usermods
  - ▶ Configuration options default to "no change" - continue to use L4 for system databases and DBIFB blocks
  - ▶ Customers electing to change the default values should carefully examine the *migration considerations* associated with PK20320
  - ▶ C/C++ object code provided by IBM will continue to use L4 blocks



# Empty Subfile Checks

- APAR PK03317 on TPFDF PUT 21
- New parameters enable checking for empty subfiles when LRECs are deleted
- Settings are available at the:
  - ▶ File level using DBDEF EMPTYCHECK, or
  - ▶ API level using DBDEL EMPTYCHECK in assembler, or the DFDEL\_EMPTYCHECK option in C/C++



# Empty Subfile Checks

- ▶ An empty subfile is indicated by field SW00RT2 (#BIT5)
  - Not a new bit setting, but was previously not reliable
  - EMTPYCHECK ensures SW00RT2 #BIT5 is reliable
  - Existing applications testing this bit should be examined and EMPTYCHECK added as appropriate
    - Also examine parameters DBEMPTY (assembler SPMs) and DF\_EMPTY (C/C++) which result in the checking of SW00RT2 #BIT5



# Duplicate Reference Name Detection

- APAR PQ59198 on TPFDF PUT 21
- If subfiles are opened using duplicate reference names, it is possible for TPFDF to modify a subfile unintentionally
- TPFDF has been updated to issue a DB0170 system error when duplicate reference names are used in such a way that the subfile to be accessed is ambiguous
  - ▶ Only one DB0170 system error will be issued per ECB
  - ▶ Duplicate reference names are valid (thus avoiding the DB0170) if the open-close sequence of the second reference is not interrupted by other TPFDF macro calls



# TPFDF PUT 22

- Scheduled to be available in June 2007
- 53 APARs including 4 enhancements:
  - ▶ PK19745 – MLS support for the High Level Assembler Release 5 (HLASM R5)
  - ▶ PK26312 – Automatic de-indexing
  - ▶ PK34634 – Support for z/OS 1.8 compiler
  - ▶ PK33468 – Reuse pools when replacing a Large Logical Record (LLR)



## Automatic De-indexing Enhancement

- Automatic de-indexing is an enhancement to basic indexing
- Prior to this support, when a detail subfile became empty, it was the application's responsibility to de-index on each path and release the pool records
  - ▶ De-indexing requires the algorithm string that was used when the subfile was indexed. TPFDF did not maintain these strings, making any type of automatic de-indexing impossible.
- Automatic de-indexing introduces an optional *Algorithm Information File* to maintain algorithm strings



## Automatic De-indexing Enhancement

- New DBDEF parameters are available to associate a detail file with an *Algorithm Information File*, and to enable automatic de-indexing
- It is possible for an *Algorithm Information File* to be maintained without enabling automatic de-indexing, allowing other applications to access the algorithm strings
- IBM is not providing a means of populating the *Algorithm Information File* for existing databases
  - ▶ recommended only for new databases and applications



## Automatic De-indexing Enhancement

- Also included is a series of new DBDEF parameters that allow DSECT options to be more easily overridden
  - ▶ For example, FULLBCH overrides the SW02OP1 #BIT0 setting in the application DSECT, or the OP1 parameter on the DBDEF
- Support is now available electronically
  - ▶ APAR PK26312 planned for TPFDF PUT 22



## Reuse Pools When Replacing LLRs

- Previously, when a Large Logical Record (LLR) was replaced (DBREP/dfrep), all associated pool records would be released, and new pools obtained
  - ▶ a potentially large and unnecessary use of long-term pools
- TPFDF has been enhanced to reuse pool records when replacing LLRs
  - ▶ If the LLR is smaller, excess pools are still released
  - ▶ If the LLR is larger, additional pools are obtained
- Does not apply to DETAC mode
  - ▶ reusing pools would prevent updates from being rolled back using “close abort”



## Reuse Pools When Replacing LLRs

- A sequence number in each block is used to ensure that any ECB reading the LLR receives consistent data
  - ▶ If a sequence number mismatch is detected while reading an LLR, then the record is being replaced
  - ▶ TPFDF will reattempt the “read” up to 5 times before returning to the application with an error (SW00RT3, #BIT6)
  - ▶ TPFDF APIs that flag LLR errors will detect this new condition upon application reassembly or recompilation



# Reuse Pools When Replacing LLRs

- Support is now available electronically
  - APAR PK33468 planned for TPFDF PUT 22
  - ALCS APAR PK42550 (which appears as AK42550 on the ALCS website) is a pre-requisite



## Previous PUT tapes

- Non-current TPFDF PUT tapes are no longer available from IBM distribution centres
- Note sent to users in February asking if previous tapes received
- Tape distribution is prone to error
  - ▶ Several users do not receive the tapes
  - ▶ Technology is moving away from 3480 tapes
- For those missing back level tapes we propose not to recreate tapes
  - ▶ We have available the consolidated APARFIX files for PUT07 – PUT21
  - ▶ One binary file for each PUT containing all the SMP/E updates
  - ▶ Can (and has) been sent to customers via FTP or CD



# Maintenance Stream Change

- PUT 22 will be last physical PUT tape produced
- All subsequent APARs will be issued on the TPFDF maintenance web pages for electronic download only.
- PUT numbers will continue to be incremented on a yearly basis:
  - ▶ This maintains consistency in the maintenance download site, prerequisite lists, closure sequence lists, TPF Information Center publications.
- PUT number changes will be synchronized with z/TPFDF dates:
  - Nov 2007 - TPFDF 1.1.3 PUT 23 (short PUT cycle)
  - Nov 2008 - TPFDF 1.1.3 PUT 24



## APARs on the TPFDF maintenance web site

- TPFDF APARs for TPF users are in hfs format only
  - ▶ Customers use USS to unpack and install the APARs.
  - ▶ Source shippable source files
    - <http://testcase-yellow.boulder.ibm.com/PK03317.source.ascii.tar.Z>
  - ▶ OCO objects compiled at the current compiler level
    - <http://testcase-yellow.boulder.ibm.com/PK03317.ascii.tar.Z>
  
- TPFDF APARs for ALCS users are in binary format
  - <http://testcase-yellow.boulder.ibm.com/PK03317.bin>
  - ▶ Download APARfix file from TPFDF maintenance website to your PC
  - ▶ Transfer APAR to host in binary format

Example overleaf of web page for first APAR in PUT22 ...



United States [change] | [Terms of use](#)

[Search](#)

[Home](#) | [Products](#) | [Services & solutions](#) | [Support & downloads](#) | [My account](#)

← Host Transaction Processing

- TPF
- More information
- Family of products
- Support
- Services
- Education
- Library
- Business relationships
- Contacts
- Site map

SUBJECT:            APAR    NUMBER: PK03317

REFERENCE:    AREA:            TPFDF CENTRAL DATABASE ROUTINES

SEGMENT: C\$CDFEQ13	- RELEASE: TPFDF (C-Header)
SEGMENT: CDFEQ13	- RELEASE: TPFDF (Macro)
SEGMENT: DBDEF13	- RELEASE: TPFDF (Macro)
SEGMENT: DBDEL13	- RELEASE: TPFDF (Macro)
SEGMENT: SW00SR13	- RELEASE: TPFDF (Macro)
SEGMENT: SW02SR13	- RELEASE: TPFDF (Macro)
SEGMENT: TPFDB13	- RELEASE: TPFDF (Macro)
SEGMENT: UFBW13	- RELEASE: TPFDF (Assembler)
SEGMENT: UFCN13	- RELEASE: TPFDF (Assembler)
SEGMENT: UFGHSO13	- RELEASE: TPFDF (Copy)
SEGMENT: UWAG13	- RELEASE: TPFDF (Assembler)

..... Scroll down to end - after the APAR description ....

Download file(s):

[Login once](#) to access server, leave window open, then click on link(s) below.

ALCS Package
Package
Source



## FTP to the host

```
> FTP myMVS.myCompany
```

```
ftp> logon user/pw
```

```
230 user is logged on. Working directory is "user."
```

```
ftp> cd BDF.V1R1M3.SERV
```

```
250 The working directory "BDF.V1R1M3.SERV" is a partitioned data set
```

```
ftp> binary
```

```
200 Representation type is Image
```

```
ftp> put PK03317.bin PK03317
```

```
200 Port request OK.
```

```
125 Storing data set BDF.V1R1M3.SERV(PK03317)
```

```
250 Transfer completed successfully
```



# Maintenance Stream Change

- PUT 22 will be last physical PUT tape produced
- All subsequent APARs will be issued on the TPFDF maintenance web pages for electronic download **only**
- When a 'virtual' PUT tape cycle completes
  - ▶ We can create a consolidated APARFIX file for the ALCS users
  - ▶ For those who would rather upgrade annually rather than APAR by APAR
  - ▶ Can be sent to customers via FTP or CD



# Wrap Up

- Compiling TPFDF C/C++ code
  - ▶ TPFDF shipped as object (compiler not pre-requisite)
  - ▶ Users who have re-compiled have run into errors
  - ▶ If you have a problem please tell us
    - PQ65588 (PUT21) provides headers C\$GR0YSR and C\$IUDFM
    - PK13711 (PUT21) required to link shipped objects
  
- Support for TPFDF through [alcs@uk.ibm.com](mailto:alcs@uk.ibm.com):
  - Problems
  - Requirements
  - Questions

