



z/TPFDF V1.1

# TPF Users Group Fall 2007

## Title: 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 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.

© 2007 IBM Corporation

# Agenda

- **z/TPFDF PUT 4**
- **Data Collection Counters**
- **Future Enhancements**
- **How can I learn more about z/TPFDF?**

## z/TPFDF PUT 4

- **Scheduled to be available in November 2007**
- **58 APARs including two enhancements:**
  - PK33480: reuse pools when replacing a large logical record (LLR)
  - PK37474: allow empty blocks to be automatically deleted from work files (W-type and T-type)
- **Also includes APAR PK45124, which affects certain data collection counters**

## z/TPFDF PUT 4 – Reuse Pools When Replacing LLRs

- **Previously, when a large logical record (LLR) was replaced, all associated pool records would be released, and new pools obtained**
  - A potentially large and unnecessary use of pools
- **z/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”
- **Requires APARs:**
  - PK33480 and PK42403 (z/TPFDF)
  - PJ31724 and PJ31945 (z/TPF)

## z/TPFDF PUT 4 – Delete Empty Blocks from Work Files

- **Detached blocks are managed using the Enhanced DETAC table**
  - If necessary, blocks are written to short-term pool to avoid exhausting the ECB private area (EPA)
- **Depending on usage, work files (W-type and T-type) can include a large number of empty blocks**
  - Short-term pool buffering may occur unnecessarily

## z/TPFDF PUT 4 – Delete Empty Blocks from Work Files

- **To solve this problem, DBDEF parameter *DELEEMPTY* is now permitted on W-type file definitions**
  - Empty blocks will be deleted automatically, preventing unnecessary use of EPA core blocks
  - IBM provided W-type file GW01SR, which is used to store T-type files, has been updated to specify **DELEEMPTY=YES**
  - You may elect to use the DELEEMPTY parameter on your W-type file definitions; the default value is **DELEEMPTY=NO**
- **APAR PK37474 on z/TPFDF PUT 4**

## Data Collection Counters

- **The UNFRC and FILUC data collection counters are now obsolete**
  - UNFRC counts now accumulated in counter RELCC
  - FILUC counts now accumulated in counted CFILU
- **These changes ensure that all counters are accurate**
- **CFILU and PFILU counters have been renamed to CFILE and PFILE**
- **Affects both TPFDF 1.1.3 and z/TPFDF**
  - APAR PK45116 for TPFDF 1.1.3
  - APAR PK45124 planned for z/TPFDF PUT 4

## Data Collection Counters

- **Customers using these counters for statistical analysis should closely examine the migration considerations associated with these APARs**
  - ZUDFC DISPLAY will no longer display data in the UNFRC and FILUC counters, even if the data was collected prior to installing the new support



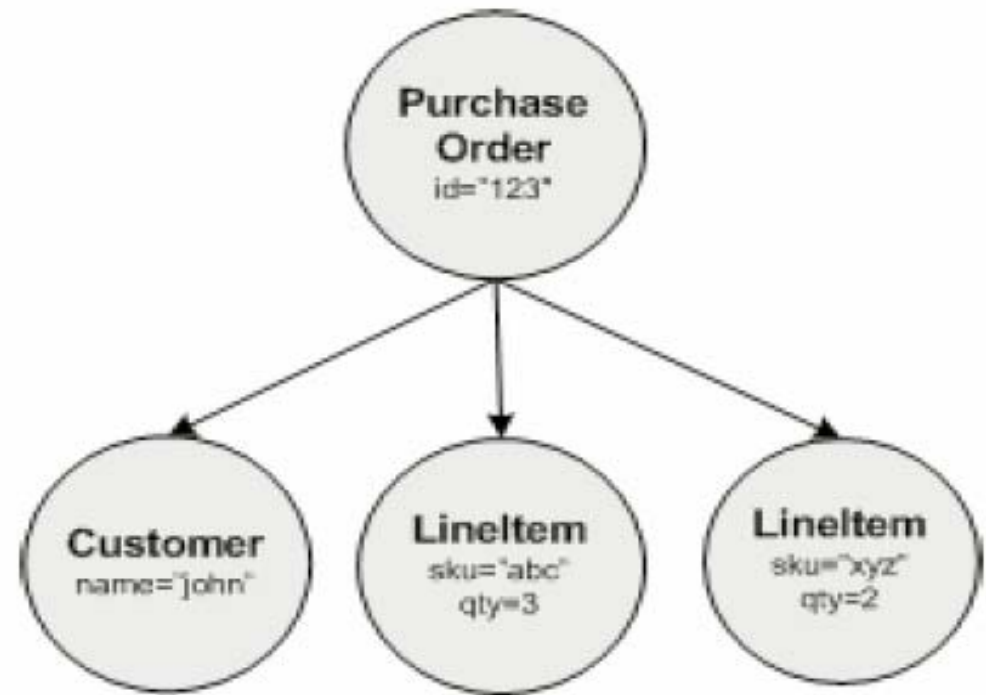
## Future Enhancements – Service Data Objects (SDO)

- **Service Data Objects (SDO) is a new model of data access**
  - Complementary technology to Service Oriented Architecture (SOA)
- **Convenient and generic way to access data**
  - Common unifying format for exchanging data between services
  - Not tied to the data organization, like SQL to relational databases
  - Object-oriented, thus maintenance is easier

# Future Enhancements – Service Data Objects (SDO)

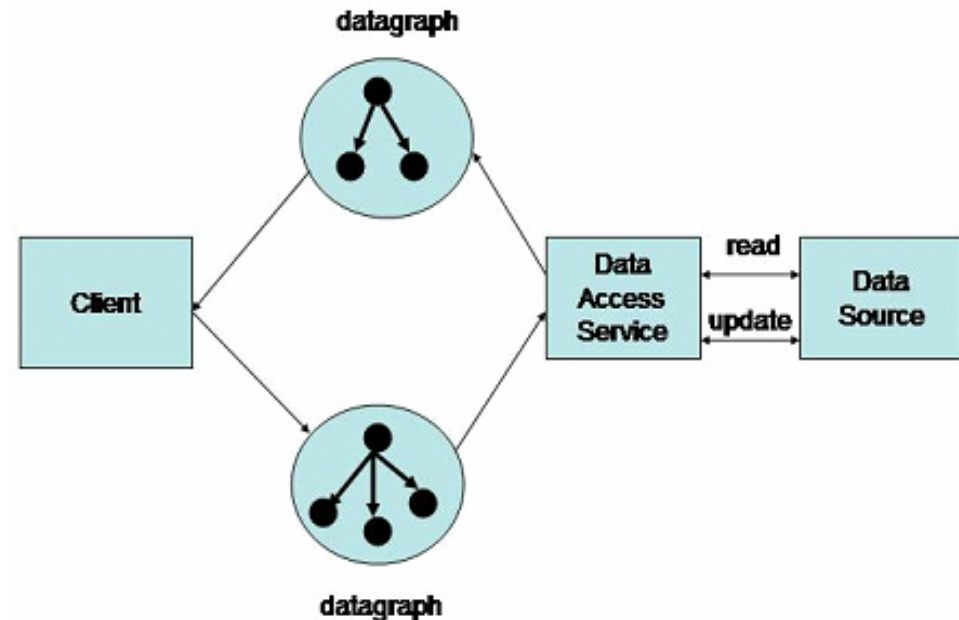
- **Terminology**

- Data object – represents a fragment of data
- Data graph – a hierarchy of data objects connected by references



# Future Enhancements – Service Data Objects (SDO)

- **Data Access Service (DAS)**
  - Loads data graphs from a data source or service
  - Propagates changes back into the data source
  - Disconnected model
  - A single client can access many data sources using different DAS APIs



## Future Enhancements – Service Data Objects (SDO)

- **SDO access to z/TPFDF will provide remote client applications access to z/TPFDF databases**
- **IBM is planning to provide a series of Java APIs for remote client applications**
  - These APIs will form a Data Access Service (DAS) for z/TPFDF databases
- **Customer applications can then use the z/TPFDF DAS APIs together with SDO APIs to access and update data in z/TPFDF databases**

## Future Enhancements – CRUISE Support for SSUs

- **CRUISE does not allow multiple subsystem users (SSUs) to be processed during a single run of a parameter table.**
- **Leads to operational complexity since customers need:**
  - At least one capture tape per SSU, and
  - To execute the parameter table in each SSU
- **Current plans are to enhance CRUISE to support multiple SSUs within the same subsystem**

## Future Enhancements – CRUISE Support for SSUs

- **Three new options will be added to the CRUISE parameters table:**
  - A list of SSUs to be processed, or “ALL” SSUs
    - If not specified, the default is to process only the SSU in which the ZFCRU START command is entered.
  - A parameter indicating if SSU common files are to be processed. The default is “yes”.

## Future Enhancements – CRUISE Support for SSUs

- For restore only, a parameter indicating if data is to be restored to the SSU from which the data was captured (the default), or to the SSU in which the ZFCRU START command was entered
  - Allows data to be captured in one SSU, and restored to another
- **Existing parameter tables will be unaffected**
  - For capture, verify and pack functions, processing will not be affected
  - For restore, processing will not be affected except that the default will be to restore data to the SSU from which it was captured

## Future Enhancements – Other Changes

- **The following customer requested enhancements are also planned:**
  - The ability to define a file as “always open”
    - Will not be closed by DBCLS REF=ALL
    - ECB can exit without the file being closed
  - Allow DBTRD/dftrd to P-type files if standard chaining is used
  - Allow LRECs to be added using ZUDFM ADD that exceed the maximum length of text that can be entered
  - Prevent accidental database corruption from ZUDFM commands due to residual subfile links

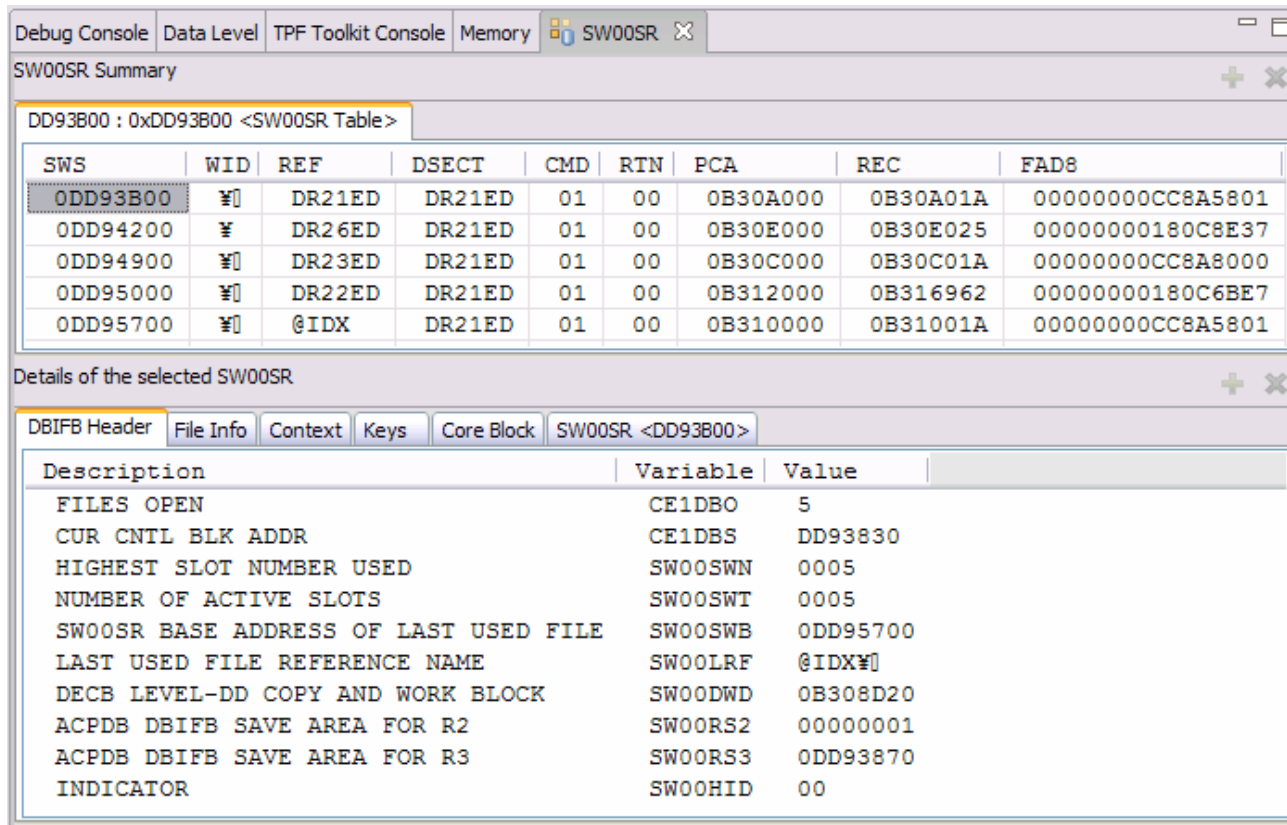


## Future Enhancements – Other Changes

- Enhance ZUDFM LOG to include the CPU ID when ZUDFM DEF INIT commands are entered
- Improve ZFCRU DISPLAY information regarding ordinals, partitions and interleaves
- Allow CRUISE to issue a message as each file ID is completed
- Allow CRUISE to pause automatically at certain points during processing, allowing customers to adjust ECB levels

# How Can I Learn More About z/TPFDF?

- Today, 11:30 am, Development Tools Subcommittee
  - XML generation from z/TPFDF file DSECTs
  - Display of SW00SR information from the Debugger



Debug Console | Data Level | TPF Toolkit Console | Memory | SW00SR

SW00SR Summary

DD93B00 : 0xDD93B00 <SW00SR Table>

SWS	WID	REF	DSECT	CMD	RTN	PCA	REC	FAD8
DD93B00	¥]	DR21ED	DR21ED	01	00	0B30A000	0B30A01A	00000000CC8A5801
DD94200	¥	DR26ED	DR21ED	01	00	0B30E000	0B30E025	00000000180C8E37
DD94900	¥]	DR23ED	DR21ED	01	00	0B30C000	0B30C01A	00000000CC8A8000
DD95000	¥]	DR22ED	DR21ED	01	00	0B312000	0B316962	00000000180C6BE7
DD95700	¥]	@IDX	DR21ED	01	00	0B310000	0B31001A	00000000CC8A5801

Details of the selected SW00SR

DBIFB Header | File Info | Context | Keys | Core Block | SW00SR <DD93B00>

Description	Variable	Value
FILES OPEN	CE1DBO	5
CUR CNTL BLK ADDR	CE1DBS	DD93830
HIGHEST SLOT NUMBER USED	SW00SWN	0005
NUMBER OF ACTIVE SLOTS	SW00SWT	0005
SW00SR BASE ADDRESS OF LAST USED FILE	SW00SWB	DD95700
LAST USED FILE REFERENCE NAME	SW00LRF	@IDX¥]
DECB LEVEL-DD COPY AND WORK BLOCK	SW00DWD	0B308D20
ACPDB DBIFB SAVE AREA FOR R2	SW00RS2	00000001
ACPDB DBIFB SAVE AREA FOR R3	SW00RS3	DD93870
INDICATOR	SW00HID	00

## How Can I Learn More About z/TPFDF?

- **Tomorrow, 9:00 am, Ongoing TPF Education**
  - A review of z/TPFDF
    - The value of z/TPFDF
    - Concepts and Terminology
      - LRECs
      - Subfiles
      - Keys
      - etc.
    - APIs
    - Utilities
    - and more.....

# Questions and Answers