



z/TPFDF V1.1

# TPF Users Group Spring 2008

## Title: 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.

© 2008 IBM Corporation

# Agenda

- **Available z/TPFDF Enhancements**
  - Always open z/TPFDF files
  - Residual links after incorrect ZUDFM commands
  - Explicit LREC size using ZUDFM ADD
  - Log CPU-ID for ZUDFM DEF INIT
  - ZUDFM special character translation
- **Future Enhancements**
  - CRUISE support for multiple subsystem users
  - Miscellaneous CRUISE enhancements
  - Other *accepted* User Group requirements

## Always Open z/TPFDF Files – APAR PK59946

- **With complex applications, it is sometimes desirable to keep a file open, even if a specific program requests all files to be closed (DBCLS REF=ALL)**
- **A new DBDEF parameter is now available to specify that a file is to be “always open” (ALWAYS=YES)**
  - This does not imply that the file is automatically opened, only that it will not be closed by DBCLS including DBCLS REF=ALL
  - Such files will also not cause an OPR-DB0112 system error when the ECB exits if the file is still open
  - A new DBCLS parameter (ALWAYS=YES) is also now available which will explicitly force an “always open” file to be closed

## Residual Links After Incorrect ZUDFM – APAR PK59331

- **When a subfile is linked using ZUDFM ACCESS, FAD or FCH, the link persists possibly for a long time until it is replaced by another link**
  - If an syntactically incorrect “linking” command is entered the old link remains exposing data in the “old” subfile to errant automation or an inattentive operator
- **Several approaches are needed to close these exposures**

## Residual Links After Incorrect ZUDFM – APAR PK59331

- 1. When a ZUDFM “linking” command is entered, the previous link is now deleted even if the command is syntactically incorrect**
  - The “old” link is no longer exposed
  - Any ZUDFM modification commands will be rejected until a new, valid link is established
- 2. Any existing links are now deleted when a user logs into the file system using ZPVFS**

## Residual Links After Incorrect ZUDFM – APAR PK59331

### **3. A new general purpose API has been introduced allowing customers to clear links during their sign-in process**

- ZUDFMC CLEARLINK in assembler
- df\_clearlink in C/C++
  - Links can be cleared for the current or all subsystem users
  - Links can be cleared for a specific terminal address, or the current terminal address specified in EBROUT

## Explicit LREC Size Using ZUDFM ADD – APAR PK59331

- **The size of an LREC that can be added using ZUDFM ADD is limited by the length of the command line**
- **A new parameter has been added to ZUDFM ADD allowing an explicit LREC size to be specified**
  - The LREC will be padded with zeroes to ensure its length is equal to the specified value
- **Macro label support (MLS) can then replace specific fields in the LREC using ZUDFM REPLACE**

## Explicit LREC Size Using ZUDFM ADD – APAR PK59331

- **The following example adds an LREC with an explicit length of 13 bytes**
  - Five bytes of padding result because the specified LREC would normally be eight bytes in length consisting of:
    - Two byte size field
    - One byte primary key
    - Five bytes of data (“TEXAS”)

```
zudfm add //13/80|texas
```

```
UDFM0001I      TPFDF LREC DISPLAY
```

```
F.A. 00000000435801AE
```

```
000D80 T E X A S 00000000 00
```



## Log CPU-ID for ZUDFM DEF INIT – APAR PK59331

- **ZUDFM commands that modify data are issued against processor-shared files**
  - Minimizes the need to log the CPU ID of the processor from which the command was entered
- **However, ZUDFM DEF INIT rebuilds in-core DBDEF tables which are processor unique**
  - Logging the affected processor becomes more important
- **ZUDFM LOG will now display the CPU ID of the affected processor**

# Log CPU-ID for ZUDFM DEF INIT – APAR PK59331

zudfm def init

UDFM0303I 07.05.31 UFB-CENTRAL DESCRIPTOR INITIALIZED +

zudfm log

UDFM0421I        OA MODIFICATION LOG DISPLAY

DATE	TIME	CRT	SI/DU	FILEADDR	ID	ENTRY
08APR	07.05	010000	/	00000000	CPUA	DEF INIT
18MAR	22.47	010000	/	00000000	BDE4	INIT BDE4
07MAR	16.25	010000	/	00000000	B245	INIT B245 _
07MAR	16.25	010000	/	00000000	B241	INIT B241
07MAR	16.25	010000	/	00000000	B243	INIT B243 +

## ZUDFM Special Character Translation – APAR PK59331

- **By default, ZUDFM only translates uppercase letters and numbers to displayable characters**
  - Other values are displayed in their hexadecimal form:

```
zudfm display
```

```
UDFM00011      TPFDF LREC DISPLAY
```

```
F.A. 00000000CC04F83D
```

```
000001 003680 A 0 0 0 1 B 0 0 0 1 C 0 0 0 1 D 0 0 0 1  
E
```

```
0 0 0 1 J T H I S I S A T E S T L R E C 0 4 3 6 7 2
```

- **User Exit DFUEX (case 10) now contains a translation table that can be customized**
  - An example is provided which additionally translates lower case letters and most special characters

## 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 Miscellaneous CRUISE Enhancements

- **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**

## Other Accepted TPFUG Requirements

- **DF00177 – mechanism for detecting long record chains**
  - DBDEF will define a threshold that determines when a “long chain” has been detected
  - A new ZRECP parameter will allow information about long chains to be displayed
- **DF00079 – Support memory based z/TPFDF files**
  - DBDEF will specify if a file is memory based
  - DBOPN will allow either:
    - a new instance of the file to be created in memory, or
    - an existing instance of the file to be accessed
  - DBCLS will allow either the instance to be deleted, or simply released allowing other ECBs access



# Questions and Answers