TPF Database Facility

Utilities

Release 1

SH31-0185-08

TPF Database Facility

Utilities

Release 1

SH31-0185-08

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page v.

Ninth Edition (October 2001)

This is a major revision of, and obsoletes, SH31-0185-07.

This edition applies to Version 1 Release 1 Modification Level 3 of IBM Transaction Processing Facility Database Facility, program number 5706-196, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

IBM welcomes your comments. Address your comments to:

IBM Corporation
TPF Systems Information Development
Mail Station P923
2455 South Road
Poughkeepsie, NY 12601-5400
USA

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

References in this book to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service in this book is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 830A
Mail Drop P131
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Programming Interface Information

This book is intended to help you manage and optimize the performance of the IBM TPF Database Facility (TPFDF) product. This book documents General-use Programming Interface and Associated Guidance Information provided by the TPFDF product.

General-use programming interfaces allow the customer to write programs that obtain the services of the TPFDF product.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ESA/390
IBM

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

About This Book

This book is both a user's guide and reference manual for all of the commands available with the IBM Transaction Processing Facility Database Facility (TPFDF) product.

In this book, abbreviations are often used instead of spelled-out terms. Every term is spelled out at first mention followed by the all-caps abbreviation enclosed in parentheses; for example, structured programming macro (SPM). Abbreviations are defined again at various intervals throughout the book. In addition, the majority of abbreviations and their definitions are listed in the master glossary in *Messages (System Error, Online, Offline) and Master Glossary*.

Before You Begin

Before using this book, see *TPFDF General Information* for an overall understanding of the TPFDF product.

Who Should Read This Book

This book is intended for operators, application programmers, system programmers, database administrators, and any other data processing professionals responsible for database management.

How This Book Is Organized

This book is organized as follows:

- Capture/Restore Utility, Information and Statistics Environment (CRUISE) provides information about the CRUISE commands and how to manage database resources.
- Recoup provides information about the recoup utility commands and how to maintain pool space.
- Data Collection provides information about the data collection utility commands and how to display data collection statistics.
- Maintenance and Support provides information about the utility commands used for database maintenance and support.
- "CRUISE Examples" provides conceptual information about the use of CRUISE.

Conventions Used in the TPFDF Library

The TPFDF library uses the following conventions:

Typography	Examples of Usage
italic	Used for important words and phrases. For example: A *database* is a collection of data. Used to represent variable information. For example: Enter **ZUDFC DISPLAY ID-***fileid*, where *fileid* is the file identifier (ID) of the file for which you want statistics.
bold	Used to represent keywords. For example: Enter **ZUDFC HELP** to obtain help information for the ZUDFC command.

Typography	Examples of Usage

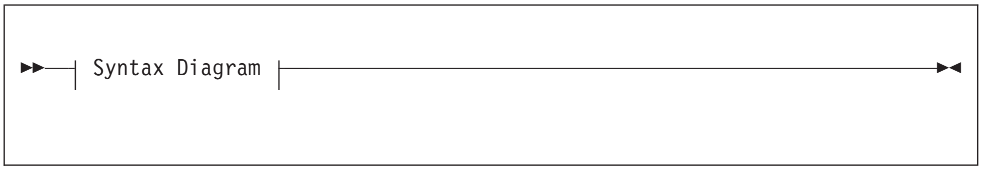
`monospaced`	Used for messages and information that displays on a screen. For example:
	`PROCESSING COMPLETED`
	Used for C language functions. For example:
	`dfcls`
	Used for examples. For example:
	`ZUDFC DISPLAY ID-J5`

bold italic	Used for emphasis. For example: You ***must*** type this command exactly as shown.
CAPital LETters	Used to indicate valid abbreviations for keywords. For example: KEYWord=*option*

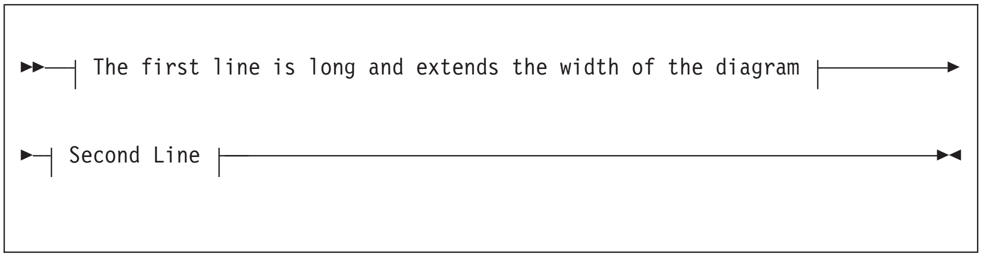
How to Read the Syntax Diagrams

This section describes how to read the syntax diagrams (informally called *railroad tracks*) used in this book.

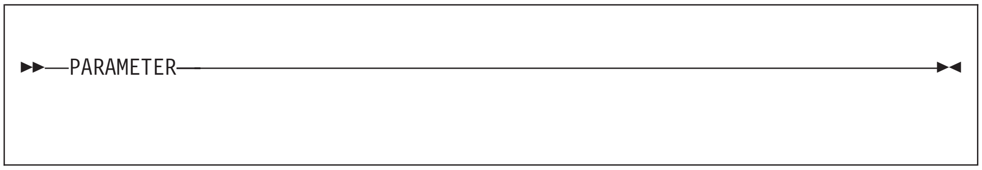
- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads and ends on the right with 2 arrowheads facing each other.



- If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.



- A word in all uppercase is a parameter that you must spell ***exactly*** as shown.



- If you can abbreviate a parameter, the optional part of the parameter is shown in lowercase. (You must type the text that is shown in uppercase. You can type none, one, or more of the letters that are shown in lowercase.)

Note: Some TPF commands are case-sensitive and contain parameters that must be entered exactly as shown. This information is noted in the description of the appropriate commands.

```
►►──PARAMeter────────────────────────────────────────────────────────────►◄
```

- A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

```
►►──variable─────────────────────────────────────────────────────────────►◄
```

- Required parameters and variables are shown on the main path line. You must code required parameters and variables.

```
►►──REQUIRED_PARAMETER──required_variable────────────────────────────────►◄
```

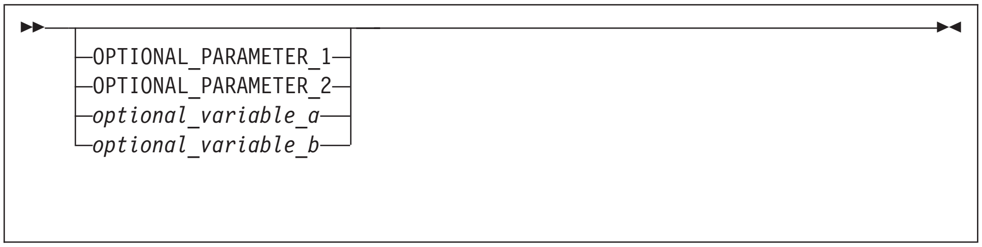
- If there is more than one mutually exclusive required parameter or variable to choose from, they are stacked vertically.

```
►►──┬─REQUIRED_PARAMETER_1─┬─────────────────────────────────────────────►◄
    ├─REQUIRED_PARAMETER_2─┤
    ├─required_variable_a──┤
    └─required_variable_b──┘
```

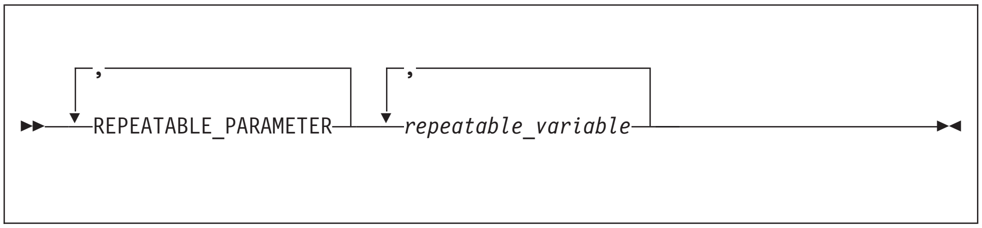
- Optional parameters and variables are shown below the main path line. You can choose not to code optional parameters and variables.

```
►►──┬───────────────────────────────────────┬────────────────────────────►◄
    └─OPTIONAL_PARAMETER──optional_variable─┘
```

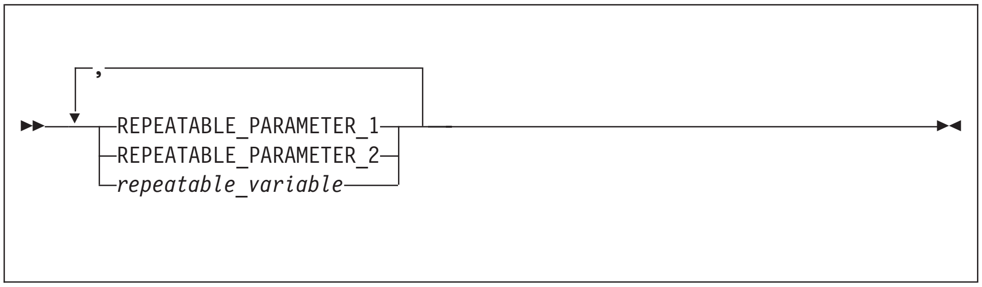
- If there is more than one mutually exclusive optional parameter or variable to choose from, they are stacked vertically below the main path line.



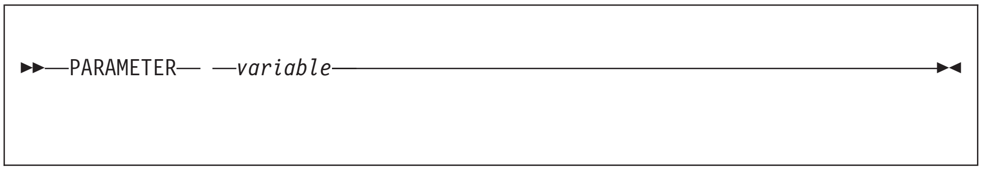
- An arrow returning to the left above a parameter or variable on the main path line means that the parameter or variable can be repeated. The comma (,) means that each parameter or variable must be separated from the next parameter or variable by a comma.



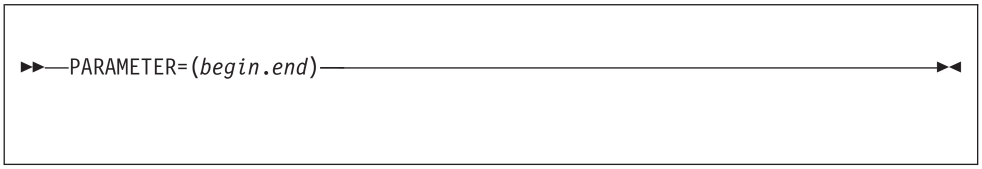
- An arrow returning to the left above a group of parameters or variables means that more than one can be selected, or a single one can be repeated.



- If a diagram shows a blank space, you must code the blank space as part of the syntax. In the following example, you must code **PARAMETER** *variable*.



- If a diagram shows a character that is not alphanumeric (such as commas, parentheses, periods, and equal signs), you must code the character as part of the syntax. In the following example, you must code **PARAMETER=(***begin***.***end***)**.



- Default parameters and values are shown above the main path line. The TPF system uses the default if you omit the parameter or value entirely.

```
    ┌─DEFAULT───┐  ┌─0────────┐
►►──┼───────────┼──┼──────────┼───────────────────────────────►◄
    └─PARAMETER─┘  └─variable─┘
```

- References to syntax notes are shown as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

```
              (1)
►►──PARAMETER──────────────────────────────────────────────────►◄
```

Notes:

1 An example of a syntax note.

- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

```
►►──┤ Reference to Syntax Fragment ├──────────────────────────►◄
```

Syntax Fragment:

```
├──1ST_PARAMETER,2ND_PARAMETER,3RD_PARAMETER──────────────────┤
```

Related Information

A list of related books follows. For information on how to order or access any of this information, call your IBM representative.

IBM TPF Database Facility (TPFDF) Books

- *TPFDF Database Administration*, SH31-0175
- *TPFDF General Information*, GH31-0177
- *TPFDF Installation and Customization*, GH31-0178
- *TPFDF Programming Concepts and Reference*, SH31-0179.

IBM Transaction Processing Facility (TPF) 4.1 Books

- *TPF General Macros*, SH31-0152
- *TPF Operations*, SH31-0162.

IBM Airline Control System (ALCS) Books

- *ALCS Application Programming Reference — Assembler Language*, SH19-6949.

Miscellaneous IBM Books

- *ESA/390 Principles of Operation*, SA22-7201.

Online Information

- *Messages (System Error, Online, Offline) and Master Glossary*.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this book or any other TPF information, use one of the methods that follow. Make sure you include the title and number of the book, the version of your product and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send your comments electronically, do either of the following:
  - Go to http://www.ibm.com/tpf/pubs/tpfpubs.htm.

    There you will find a link to a feedback page where you can enter and submit comments.
  - Send your comments by e-mail to tpfid@us.ibm.com
- If you prefer to send your comments by mail, address your comments to:

  IBM Corporation
  TPF Systems Information Development
  Mail Station P923
  2455 South Road
  Poughkeepsie, NY 12601-5400
  USA
- If you prefer to send your comments by FAX, use this number:
  - United States and Canada: 1 + 845 + 432 + 9788
  - Other countries: (international code) + 845 + 432 +9788

Capture/Restore Utility, Information and Statistics Environment (CRUISE)

This chapter describes the TPFDF capture/restore utility, information and statistics environment (CRUISE) commands. CRUISE is a validation, capture, and recovery tool for database administrators and system programmers and is based on file information that is stored in the TPFDF database definition (DBDEF).

See “CRUISE Examples” on page 123 for information (with examples) for creating a parameter table, setting up parameter table default values, and running a CRUISE function.

CRUISE Concepts

CRUISE is based on a parameter table, which contains all the necessary information required to process a CRUISE function. A parameter table must be created, named, set up, and maintained by the database administrator. Parameter table maintenance allows the setup of default values and prepares for application use. A CRUISE function processes a defined parameter table when you use a ZFCRU command.

Help is available by using the ZFCRU HELP command.

CRUISE Benefits

CRUISE is a versatile and easy to use method of managing your database. Based on file information stored in the TPFDF database definition (DBDEF), CRUISE is a process of recovering and validating data, controlling pool space, and creating statistics. The following are some benefits of using CRUISE:

- Friendly user interface
- Status and statistical displays
- Parameter tables that simplify your maintenance efforts
- Supports running different parameter tables simultaneously on different loosely coupled processors. Additionally, this activity shortens the processing time for CRUISE functions.

CRUISE Terminology

When you use CRUISE, it is important to understand the following terms:

CRUISE function	The process that performs the steps specified by a parameter table. The following are CRUISE functions: • Capture • Pack • Restore • Verify.
CRUISE commands	The ZFCRU commands that perform CRUISE functions.
Parameter table	The source of parameter values that define how a CRUISE function is processed.

CRUISE States

The following list describes the different parameter table states of the capture/restore utility, information and statistics environment (CRUISE) as displayed by the ZFCRU DISPLAY command with the ALL parameter.

Active	A parameter table that is verified after the initial ZFCRU START command is entered.
Running	A parameter table that is processing a specified CRUISE function after a second ZFCRU START command is entered. **Note:** A CRUISE function in running state is in use.
Paused	A parameter table in running state that has paused.
Stopped	A parameter table processing a CRUISE function that is suspended when you use the ZFCRU STOP command.

ZFCRU ABORT–Abort Running CRUISE

Use this command to immediately end a capture/restore utility, information and statistics environment (CRUISE) function.

Requirements and Restrictions

You can only use the ZFCRU ABORT command while a CRUISE function is processing existing parameter tables; that is, the CRUISE function is active, running (in use), paused, or stopped state.

Format

```
►►──ZFCRU ABORT────────────────────────────────────────────────►◄
```

Additional Information

You can enter the ZFCRU ABORT command once or twice depending on how quickly you want a CRUISE function to abort. If CRUISE processing does not abort, you can enter the ZFCRU ABORT command more than twice.

If you enter the ZFCRU ABORT command once, the CRUISE function will process all entry control blocks (ECBs), and then exit.

If you enter the ZFCRU ABORT command twice, the CRUISE function will flush all ECBs immediately.

Examples

The following example shows a CRUISE function ended with a single ZFCRU ABORT command.

```
User:   ZFCRU ABORT

System: FCRU0142I 10.29.08 CRUISE IS ABORTING
```

References

None.

ZFCRU COPY–Create and Copy a Parameter Table

Use this command to create a parameter table by copying the contents of an existing parameter table.

Requirements and Restrictions

- You can copy only a source parameter table to a new target parameter table.
- You can use a free format to name a new parameter table. A parameter table name can contain as many as 16 alphanumeric characters, including special characters such as a comma (,), slash (/), period (.), and underscore (_). Space characters (blanks) are not allowed, and a source parameter table name cannot have a hyphen (-).

Format

```
►►──ZFCRU COPY-──source-target──────────────────────────────────────►◄
```

source
is the source parameter table name.

target
is the name of the parameter table to be created.

Using any of the following names to create a new parameter table can conflict with the ZFCRU DISPLAY command parameter table values because the names are the same.
- ALL
- DEF
- DEFAULT
- STA
- STATISTIC
- LOG.

Additional Information

You can use the ZFCRU COPY command while the capture/restore utility, information and statistics environment (CRUISE) function is processing existing parameter tables; that is, the CRUISE function is in active, running (in use), paused, or stopped state.

Examples

In the following example, a parameter table (TABLE1) is copied to a new parameter table (TABLE2).

```
User:   ZFCRU COPY-TABLE1-TABLE2

System: FCRU0009I 10.31.21 OK TABLE - TABLE1 - COPIED
```

References

None.

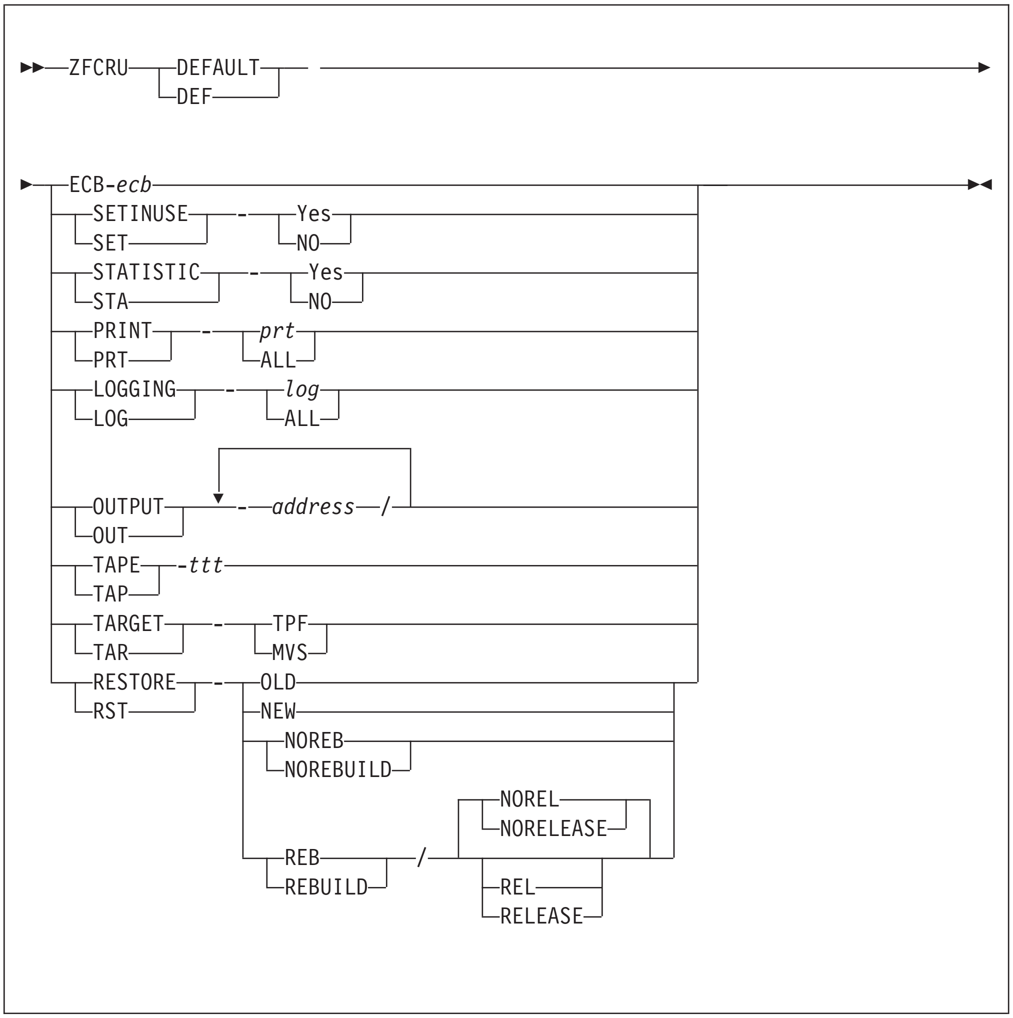
ZFCRU DEFAULT–Set Up and Change Parameter Values

Use this command to set up and change parameter table default values.

Requirements and Restrictions

None.

Format



Note: You can also specify the optional parameter names in their abbreviated form as shown in the syntax diagram.

ECB-*ecb*
specifies the percent of total entry control blocks (ECBs) to be distributed, where *ecb* is a value in the range 1–100%. One hundred percent is the total number of ECBs allocated on the TPF system.

SETINUSE-
specifies pool file use, where:

YES
calls case 3 of user exit macro DFUEX during the processing of the

RESTORE and VERIFY parameters when pool files are used. The RESTORE parameter must have the REF and FAD parameters specified when you use this option.

Note: You can use this parameter only in NORM state.

NO
does not call case 3 of user exit macro DFUEX during the processing of the RESTORE and VERIFY parameters when pool files are used.

STATISTIC
specifies statistics from the parameter table, where:

YES
is the file statistics that are built following successful parameter table processing.

NO
is the that file statistics that are not built following successful parameter table processing.

PRINT
specifies the number of messages to be printed to the output device selected with the ZFCRU OUTPUT command, where:

prt is a number in the range 0–100.

ALL
specifies all the messages are to be printed.

Notes:

1. Printing will stop when the number you specified is reached.
2. Print will stop if the number of messages you specify is less than the number of messages printed.
3. You specify the output device for printed messages with the ZFCRU OUTPUT command, or with the output parameter settings specified with the ZFCRU SETUP command.
4. The number of messages you specified is in effect only when the capture/restore utility, information and statistics environment (CRUISE) is in running state.

LOGGING
specifies messages to be logged to the parameter table, where:

log
is a number in the range 0–100.

ALL
specifies that all the messages are to be logged to the parameter table logging file.

Notes:

1. Logging ends when the specified number of logged messages is reached.
2. Logging will run until available pool space is used if you use the ZFCRU LOG command with the ALL parameter.
3. If the number of logged messages you specify is less than the number of messages logged, logging will complete and no messages are deleted.
4. The number of messages you specified to be logged is in effect only when CRUISE is in running state.

OUTPUT-*address*
specifies the output device, where *address* is a 6-digit hexadecimal printer or terminal address. You can specify as many as three output addresses.

Note: If you start a CRUISE function from a terminal that is not specified as an output terminal, only start and end messages are displayed. However, a prime CRAS receives all messages regardless of where a CRUISE function is entered.

TAPE
specifies the requested tape or general data set (GDS).

ttt specifies any user-defined tape. This must be defined in the tape label mask record (TLMR).

TARGET
specifies the target system where the data will be restored. This parameter is used for CRUISE capture only. The TPF system is the default.

TPF
specifies the data that will be restored on a TPF system or ALCS environment.

MVS
specifies the data that will be restored on an MVS system.

RESTORE
specifies the method in which CRUISE restores data, where:

OLD
specifies that the database is restored to the same fixed and pool file addresses that were captured.

NEW
specifies that the captured fixed and pool files are restored to new pool file addresses where the data structure is rebuilt.

Note: You can use this parameter only in NORM state.

NOREB
specifies that the captured fixed and pool files are restored to new pool file addresses and the data structure is not rebuilt.

Note: You can use this parameter only in NORM state.

REB
specifies that the captured fixed files are restored to the same ordinals that were captured. The captured pool files are restored to new pool file addresses and the database structure is rebuilt.

Note: You can use this parameter only in NORM state.

REL
specifies that old pool file addresses are released when the data structure is rebuilt.

Note: You can this parameter only in NORM state.

NOREL
specifies that old pool file addresses are not released when the structure is rebuilt.

Note: You can use this parameter only in NORM state.

Additional Information

- During the installation of CRUISE, the initial parameter values are set according to the values in the C$CRUUSR header.
- You can use the ZFCRU DEFAULT command while the CRUISE function is processing existing parameter tables; that is, the status of the CRUISE function is active, running (in use), paused, or stopped. However, changes that you make to the parameter table by using the ZFCRU DEFAULT command occur only when new parameter tables are created.
- You must pack a B^+Tree and block index file to validate the file references after CRUISE capture and restore processing because CRUISE capture and restore processing nullifies the validation of block index technical LRECs (TLRs) and releases B^+Tree files.

Examples

In the following example, the default value for ECBs allocated to CRUISE is 50% of the ECBs allocated to the system.

```
User:   ZFCRU DEFAULT ECB-50

System: FCRU0000I 10.31.44 MODIFIED
        FCRU0000I 10.31.44 DEFAULT PARAMETER VALUES
          USABLE ECB START VALUE     50 PCT
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   100
          PRINTER ADDRESSES          306919  000000  000000
          NUMBER OF LOG MESSAGES     ALL
          SET POOL ADDRESSES IN USE  NO
          TAPE NAME                  BFA
          TARGET SYSTEM NAME         TPF
          RESTORE OPTION             REBUILD/NORELEASE
          WID/ADR IMBED REFERENCES   YES
        FCRU0000I 10.31.44 END OF DISPLAY
```

References

- See “ZFCRU SETUP–Parameter Table Maintenance and Setup” on page 24 for information about overriding parameter table default values while CRUISE is in running state.
- See *TPFDF Installation and Customization* more information about the CRUISE C$CRUUSR default header.

ZFCRU DELETE–Delete a Parameter Item

Use this command to selectively delete parameters or to delete a parameter table.

Requirements and Restrictions

- A parameter table must be locked before the parameters can be deleted.
- When parameter table processing is completed successfully, required parameters are verified by the capture/restore utility, information and statistics environment (CRUISE). If an error occurs, you will not be able to start a CRUISE function.

Format

```
►►──ZFCRU── ──┬─DELETE─┬──┬─-name─────────────────┬──────────────────────►◄
              └─DEL────┘  └─  ──┬─ADR───────────┤
                                ├─APL───────────┤
                                ├─FAD───────────┤
                                ├─REF-──┬─id──┬─┤
                                │       └─ALL─┘ │
                                ├─RST───────────┤
                                ├─TAP───────────┤
                                ├─UEX───────────┤
                                └─WID-──┬─id──┬─┘
                                        └─ALL─┘
```

Note: You can also specify optional parameter name DEL in the abbreviated form as shown in the syntax diagram.

name
specifies the parameter table to be deleted.

Note: You cannot delete a parameter table that is locked to another terminal address.

ADR
deletes file addresses specified by the ADR parameter of the ZFCRU SETUP command.

APL
specifies the application code to be deleted.

FAD
deletes file addresses specified by the FAD parameter of the ZFCRU SETUP command.

REF
specifies the file identifiers (IDs) to be deleted that were set up with the ZFCRU SETUP command and the REF parameter.

Note: To delete a range of file IDs, you must specify the first file ID in the range.

id is a 2-character alphanumeric or 4-character hexadecimal file ID.

ALL
specifies all file IDs to be deleted.

RST
specifies all restore parameters to be deleted.

TAP
specifies the tape name to be deleted.

UEX
specifies the user exit to be deleted.

WID
specifies the IDs to be deleted that were set up with the WID parameter of the ZFCRU SETUP command.

Additional Information

You can use the ZFCRU DELETE command while the CRUISE function is processing other parameter tables; that is, the CRUISE function is in active, running (in use), paused, or stopped state.

Examples

The following example shows a parameter table (TABLE2) being deleted.

```
User:   ZFCRU DELETE-TABLE2

System: FCRU0009I 10.32.38 OK TABLE - TABLE2 - DELETED
```

In the following example, file ID B075 is deleted from the parameter table (TABLE1).

```
User:   ZFCRU DELETE REF-B075

System: FCRU0000I 10.36.49 MODIFIED
        FCRU0000I 10.36.49 PARAMETER TABLE DISPLAY FOR TABLE1
           FUNCTION                   VER
           USABLE ECB START VALUE     33 PCT
           SET POOL ADDRESSES IN USE  NO
           BUILD STATISTICS           YES
           NUMBER OF PRINT MESSAGES   100
           OUTPUT ADDRESSES           306919  000000  000000
           NUMBER OF LOG MESSAGES     ALL
        FCRU0000I 10.36.49 END OF DISPLAY
```

References

- See “ZFCRU SETUP–Parameter Table Maintenance and Setup” on page 24 for additional parameter table information.
- See “ZFCRU LOCK–Lock a Parameter Table” on page 17 for additional information about locking a parameter table.
- See “ZFCRU UNLOCK–Unlock a Parameter Table” on page 36 for additional information about unlocking a parameter table.

ZFCRU DISPLAY–Display Information

Use this command to display:

- The capture/restore utility, information and statistics environment (CRUISE) work block
- Existing parameter table names
- Contents of a specific parameter table
- Default parameter table values
- CRUISE entry logging tables
- Parameter tables specific to a logging table
- File statistics.

Requirements and Restrictions

None.

Format

```
►►──ZFCRU DISPLAY──┬──────────────────────────────────┬──────────────►◄
                   ├─-ALL─────────────────────────────┤
                   ├─-name────────────────────────────┤
                   ├─┬─-DEFAULT─┬─────────────────────┤
                   │ └─-DEF─────┘                     │
                   ├─-LOG──┬──────────┬───────────────┤
                   │       └─-logname─┘               │
                   └─┬─-STATISTIC─┬─┬──────┬─-id──────┘
                     └─-STA───────┘ └─-ALL─┘
```

ALL
displays the names of all existing parameter tables.

Note: You can use this parameter only in NORM state.

name
specifies the parameter table to be displayed, where *name* is the parameter table name.

DEFAULT
displays the default parameter table settings.

LOG
specifies the CRUISE command log to be displayed. The CRUISE command log contains all CRUISE command entries and completion messages.

LOG-*logname*
specifies the parameter logging table to be displayed, where *logname* is the logging table name. The logging table contains a log for the specified parameter table.

STATISTIC-*id*
specifies the statistics of a file to be displayed, where *id* is a 2-character alphanumeric or 4-character hexadecimal file ID.

STATISTIC-ALL-*id*
specifies the cumulative statistics of a file to be displayed, where *id* is a 2-character alphanumeric or 4-character hexadecimal file ID.

Additional Information

When you enter the ZFCRU DISPLAY command without any parameters, the parameter table that is locked by your terminal is displayed. If you do not have a parameter table that is locked by your terminal, CRUISE performs a search for the next parameter table in running state to display.

If there are no parameter tables in running state, all parameter tables are displayed.

Examples

The following example displays the contents of a parameter table (TABLE1).

```
User:   ZFCRU DISPLAY-TABLE1

System: FCRU0000I 10.37.27 PARAMETER TABLE DISPLAY FOR TABLE1
          FUNCTION                   VER
          USABLE ECB START VALUE     33 PCT
          SET POOL ADDRESSES IN USE  NO
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   100
          OUTPUT ADDRESSES           306919  000000  000000
          NUMBER OF LOG MESSAGES     ALL
          REFERENCE ID               FDFD
        FCRU0000I 10.37.27 END OF DISPLAY
```

References

None.

ZFCRU ECB–Change ECB Percentage

Use this command to change the percentage of all the entry control blocks (ECBs) allocated on the system that will be used to perform a capture/restore utility, information and statistics environment (CRUISE) function.

Requirements and Restrictions

- The ZFCRU ECB command affects only a CRUISE function in which you have successfully entered the ZFCRU START command twice.
- When you use the ZFCRU ECB command, the parameter table is changed only during the processing of the parameter table. This is not a permanent change.

Format

```
►►──ZFCRU ECB-ecb──────────────────────────────────────────────────►◄
```

ecb
specifies the percent of ECBs used for a CRUISE function, where *ecb* is a value in the range 1–100%.

Additional Information

You can use the ZFCRU ECB command while the CRUISE function is processing existing parameter tables; that is, the CRUISE function is in active state.

Examples

In the following example, the percent of ECBs allocated to CRUISE has been changed from 33% to 50% of the ECBs allocated to the system.

```
User:   ZFCRU ECB-50

System: FCRU0104I 10.38.33 ECB PERCENTAGE CHANGED FROM 33 TO 50
```

References

None.

ZFCRU EXPAND–Create New Index Items

Use this command to create a parameter table index.

Requirements and Restrictions

You can enter this command only in NORM state.

```
►►──ZFCRU EXPAND──────────────────────────────────────────────────────►◄
```

Additional Information

- You use the ZFCRU EXPAND command to create the initial parameter table indexes.
- To expand the number of parameter tables, you must increase the number of ordinals of file address compute program (FACE) type #IRCBDF and enter the ZFCRU EXPAND command.

  The number of ordinals of FACE type #IRCBDF determines the maximum number of tables that you can create.

  Note: When you expand the number of parameter tables, you must increase equally FACE type #IRCKDF to increase the number of log files.
- When you use the ZFCRU EXPAND command, the existing parameter tables will not be changed.
- You can use the ZFCRU EXPAND command while the capture/restore utility, information and statistics environment (CRUISE) function is processing existing parameter tables; that is, the CRUISE function is in active, running (in use), paused, or stopped state.

Examples

In the following example, parameter table indexes are created.

```
User:   ZFCRU EXPAND

System: FCRU0009I 10.38.55 OK TABLE - EXPANSION - COMPLETED
```

References

None.

ZFCRU HELP–Help Information

Use this command to display a help index and detailed information for capture/restore utility, information and statistics environment (CRUISE) topics.

Requirements and Restrictions

None.

Format

```
►►──ZFCRU──┬─HELP─┬──-──┬─┬─CONTROL─┬─┬──────────────────────────────────►◄
           └─?────┘     │ ├─CON─────┤ │
                        │ └─CTL─────┘ │
                        ├─┬─DEFAULT─┬─┤
                        │ └─DEF─────┘ │
                        ├─┬─DISPLAY─┬─┤
                        │ └─DIS─────┘ │
                        ├─┬─REPORT─┬──┤
                        │ └─REP────┘  │
                        ├─┬─SETUP─┬───┤
                        │ └─SET───┘   │
                        └─┬─TABLE─┬───┘
                          └─TAB───┘
```

HELP
displays help information for the ZFCRU commands.

? displays help information for the ZFCRU commands.

CONTROL
displays the ZFCRU commands that control CRUISE functions.

DEFAULT
displays help for the ZFCRU DEFAULT command.

DISPLAY
displays help for the ZFCRU DISPLAY command.

REPORT
displays help for the following commands:
- ZFCRU LOGGING
- ZFCRU OUTPUT
- ZFCRU PRINT.

SETUP
displays help for the ZFCRU SETUP command.

TABLE
displays help for the following commands:
- ZFCRU COPY
- ZFCRU DELETE
- ZFCRU EXPAND
- ZFCRU LOCK
- ZFCRU UNLOCK.

Notes:

1. You can enter the optional parameter names in an abbreviated form, which includes ***at least*** the first three letters of the parameter name.
2. You can enter the optional parameter name CTL for the parameter control.

Additional Information

None.

Examples

The following example displays the ZFCRU commands that control CRUISE function processing.

```
User:   ZFCRU HELP-CONTROL

System: FCRU0000I 10.39.27 CRUISE HELP FOR CONTROL ENTRIES:
        ZFCRU START-CCCCC
              RESTART{-NN}
              PAUSE
              STOP
              ABORT
              ECB-NN
              PRINT-{NN!ALL}
              OUTPUT-{HHHHHH!HHHHHH/HHHHHH}
              LOGGING-{NN!ALL}

        CCCCC  = TABLE NAME (UP TO 16 CHARACTERS)
        NN     = ECB PERCENTAGE: 1-100 / NUMBER OF MESSAGES: 0-100
        HHHHHH = UP TO 3 PRINTER/TERMINAL ADDRESSES IN HEX FORMAT
         OR 0 TO REMOVE ALL ADDRESSES
        PAUSE  = STOP AFTER FINISHING CURRENT ORDINALS
        STOP   = STOP IMMEDIATELY
        FCRU0000I 10.39.27 END OF DISPLAY
```

References

None.

ZFCRU LOCK–Lock a Parameter Table

Use this command to lock a parameter table to the existing terminal address and create a new parameter table.

Requirements and Restrictions

- You cannot lock a parameter table that is processing; that is, the capture/restore utility, information and statistics environment (CRUISE) function is in active, running (in use), paused, or stopped state.
- You cannot lock a parameter table to more than one terminal address.

```
►►──ZFCRU LOCK-name──────────────────────────────────────────────►◄
```

name

specifies the parameter table to be locked.

Note: The parameter table name is in free format; that is, it can contain as many as 16 alphanumeric characters, including special characters such as a comma (,), slash (/), period (.), underscore (_), and hyphen (-). Space characters (blanks) are not allowed. There is one exception to free format.

Using any of the following names to create a new parameter table can conflict with the ZFCRU DISPLAY command parameter table values because the names are the same.

- ALL
- DEF
- DEFAULT
- STA
- STATISTIC
- LOG.

Additional Information

- When you use this command with a parameter table name that already exists, the parameter table will be locked and displayed.
- When you use this command with a parameter table name that does not exist, a parameter table will be created using the default parameters.
- The following CRUISE commands are affected by the ZFCRU LOCK command:

  ZFCRU SETUP

  You can only modify a parameter table that is locked.

  ZFCRU DELETE

  You can selectively delete parameters from a parameter table that is locked to your terminal.

  ZFCRU DISPLAY

  You can display a parameter table locked by your terminal.

Examples

In the following example, a parameter table (TABLE3) is created and locked.

```
User:    ZFCRU LOCK-TABLE3

System:  FCRU0000I 10.39.56 TABLE LOCKED
         FCRU0000I 10.39.56 PARAMETER TABLE DISPLAY FOR TABLE3
           FUNCTION
           USABLE ECB START VALUE     50 PCT
           TAPE NAME                  BFA
           TARGET SYSTEM NAME         TPF
           WID/ADR IMBED REFERENCES   YES
           BUILD STATISTICS           YES
           NUMBER OF PRINT MESSAGES   100
           OUTPUT ADDRESSES           306919  000000  000000
           NUMBER OF LOG MESSAGES     ALL
           RESTORE OPTION             REBUILD/NORELEASE
         FCRU0000I 10.39.56 END OF DISPLAY
```

References

- See “ZFCRU DEFAULT–Set Up and Change Parameter Values” on page 5 for more information about default parameter values.
- See “ZFCRU UNLOCK–Unlock a Parameter Table” on page 36 for more information about how to release a lock.
- See “ZFCRU DISPLAY–Display Information” on page 11 for more information about the parameters used to display tables.
- See “ZFCRU SETUP–Parameter Table Maintenance and Setup” on page 24 for more information about modifying a parameter table.
- See “ZFCRU DELETE–Delete a Parameter Item” on page 9 for more information about deleting parameters from a parameter table.

ZFCRU LOGGING–Log Messages

Use this command to specify the number of messages to be logged in the parameter table logging file.

Requirements and Restrictions

You can enter the ZFCRU LOGGING command only when the CRUISE function is in running state.

Format

```
►►──ZFCRU LOGGING-──┬─log─┬──────────────────────────────────────────────────►◄
                    └─ALL─┘
```

log
: specifies the number of messages to be logged to the parameter table logging file for a capture/restore utility, information and statistics environment (CRUISE) function, where *log* is a number in the range 0–100.

ALL
: specifies that all messages are to be logged to the parameter table logging file.

Additional Information

- Logging ends when the specified number of logged messages is reached.
- Logging will run until available pool space is used if you use the ZFCRU LOGGING command with the ALL parameter.
- If the number of logged messages you specify is less than the number of messages logged, logging will complete and no messages are deleted.
- The number of messages you specified to be logged is in effect only when the CRUISE is in running state.

Examples

The following example shows that all messages are to be logged to the parameter table log table

```
User:     ZFCRU LOGGING-ALL

System:   FCRU0111I 10.43.46 NUMBER OF LOGGING MESSAGES CHANGED FROM
                             100 TO ALL
```

References

None.

ZFCRU OUTPUT–Specify Printer and Terminal Address

Use this command to route capture/restore utility, information and statistics environment (CRUISE) messages to a specified printer address, terminal address, or both.

Requirements and Restrictions

You can enter the ZFCRU LOGGING command only when the CRUISE function is in running (in use) state.

Format

```
                    ┌─/───────┐
                    ▼         │
►►──ZFCRU OUTPUT-───address───┴──────────────────────────────────►◄
```

address
is a 6-digit hexadecimal printer or terminal address. You can specify as many as three addresses.

Additional Information

- The specified terminal address is in effect only when the CRUISE function is in running (in use) state.
- If you start a CRUISE function from a terminal that is not specified as an output terminal, only start and end messages are displayed. However, a prime CRAS receives all messages regardless of where a CRUISE function is started.

Examples

In the following example, output for CRUISE messages is sent to printer address FE2273.

```
User:     ZFCRU OUTPUT-FE2273

System:   FCRU0115I 10.47.31 PRINTER ADDRESSES CHANGED FROM
                   1:010000 2:000000 3:000000   TO
                   1:FE2273 2:000000 3:000000
```

References

None.

ZFCRU PAUSE–Suspend CRUISE

Use this command to suspend a capture/restore utility, information and statistics environment (CRUISE) function after its activity on the current fixed file ordinal is completed successfully. If applicable, the current tape will remain open.

Requirements and Restrictions

You can enter the ZFCRU PAUSE command only when the CRUISE function is in running (in use) state.

Format

```
►►──ZFCRU PAUSE──────────────────────────────────────────────────────────►◄
```

Additional Information

You must enter the ZFCRU RESTART command to restart a parameter table in pause state.

Examples

In the following example, a CRUISE function is paused.

```
User:    ZFCRU PAUSE

System:  FCRU0142I 10.47.55 CRUISE IS PAUSING
         FCRU0126I 10.47.56 00147758 RECORDS FOUND
         FCRU0103I 10.47.56 TABLE -TABLE1- PAUSED
```

References

See “ZFCRU RESTART–RESTART CRUISE” on page 23 for more information about restarting CRUISE after you use the ZFCRU PAUSE command.

ZFCRU PRINT–Report Parameters

Use this command to specify the number of messages to be sent to the printer, terminal, or both.

Requirements and Restrictions

You can enter the ZFCRU PRINT command only when the capture/restore utility, information and statistics environment (CRUISE) is in running state.

Format

```
►►──ZFCRU PRINT-──┬─prt─┬─────────────────────────────────────────────────►◄
                  └─ALL─┘
```

prt specifies the number of messages to be printed in the range 0–100.

ALL
specifies all messages to be printed.

Additional Information

- Printing stops when the number of messages that you specified is reached.
- If the number of messages you specify is less than the number of messages printed, printing will end.
- You specify the output device for printed messages with the ZFCRU OUTPUT command, or with the output parameter settings specified with the ZFCRU SETUP command.
- The number of messages you specified is in effect only when the CRUISE function is in running state.

Examples

In the following example, the number of CRUISE messages to be printed has been changed from 100 to 50.

```
User:    ZFCRU PRINT-50

System:  FCRU0108I 10.48.51 NUMBER OF PRINTER MESSAGES CHANGED FROM
                            100 TO 50
```

References

- See “ZFCRU OUTPUT–Specify Printer and Terminal Address” on page 20 for more information about CRUISE messages.
- See“ZFCRU SETUP–Parameter Table Maintenance and Setup” on page 24 for more information about the output parameter settings.

ZFCRU RESTART–RESTART CRUISE

Use this command to restart a capture/restore utility, information and statistics environment (CRUISE) function that is in a paused or stopped state, or has experienced an unexpected interruption.

Requirements and Restrictions

- You must enter the ZFCRU RESTART command on the same processor and subsystem user (SSU) from which CRUISE was originally started.
- You can enter the ZFCRU RESTART command only when CRUISE is in a paused or stopped state, or a system interruption has occurred.

Format

```
►►──ZFCRU RESTART──┬──────┬──────────────────────────────────────────►◄
                   └-ecb──┘
```

ecb
specifies the entry control block (ECB) value in which a CRUISE function will be restarted, where *ecb* is the percent of ECBs in the range 1–100%.

Additional Information

If you use this command without any parameters, the CRUISE function will be restarted by using the ECB value specified with the ZFCRU ECB command followed by the ECB value specified with the ZFCRU SETUP command.

Examples

The following example shows a CRUISE function being restarted.

```
User:    ZFCRU RESTART

System:  FCRU0102I 10.48.15 TABLE -TABLE1- RESTARTED
```

References

See “ZFCRU ECB–Change ECB Percentage” on page 13 for more information about the allocation of ECBs.

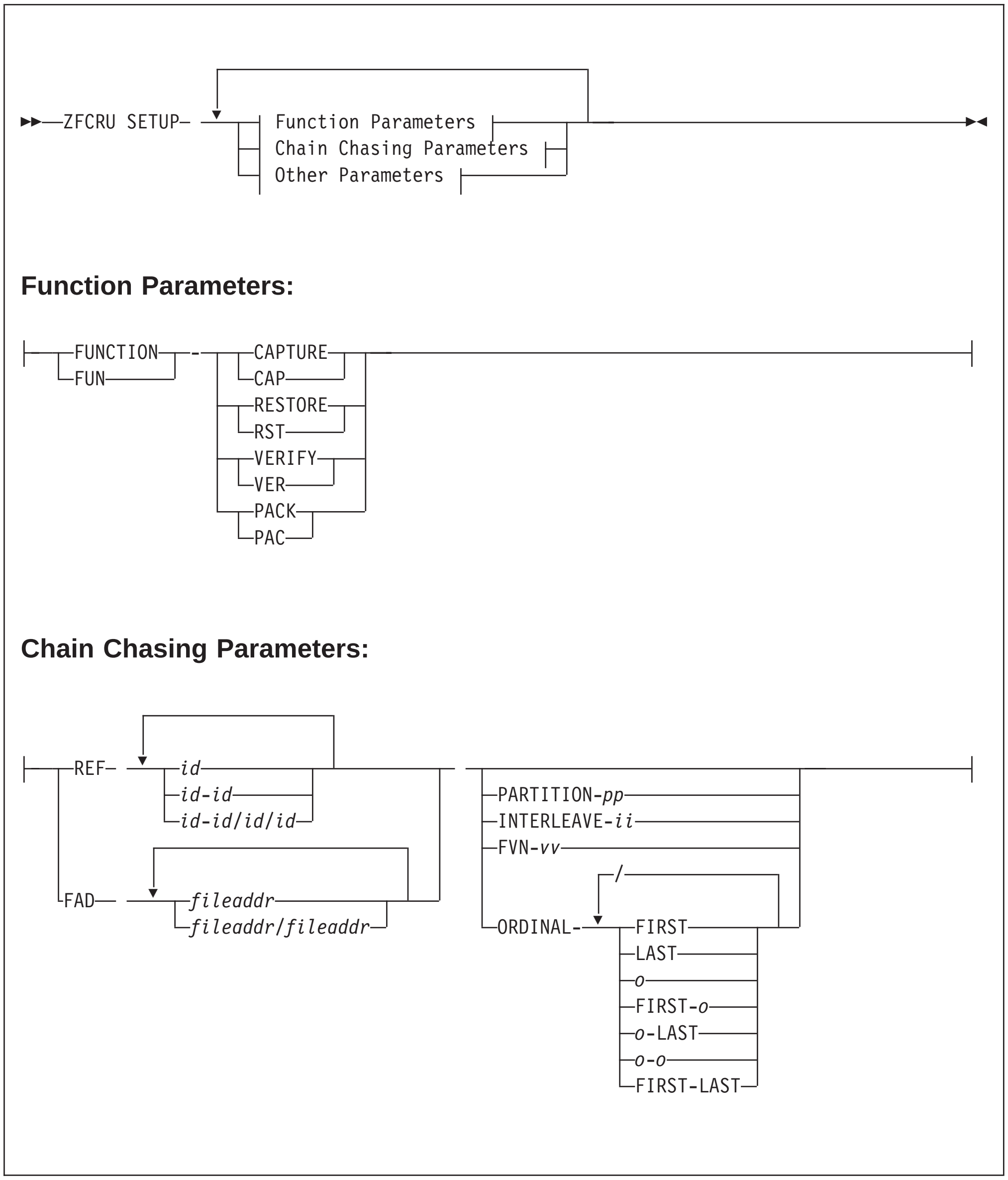
ZFCRU SETUP–Parameter Table Maintenance and Setup

Use this command to update an existing parameter table that has been locked.

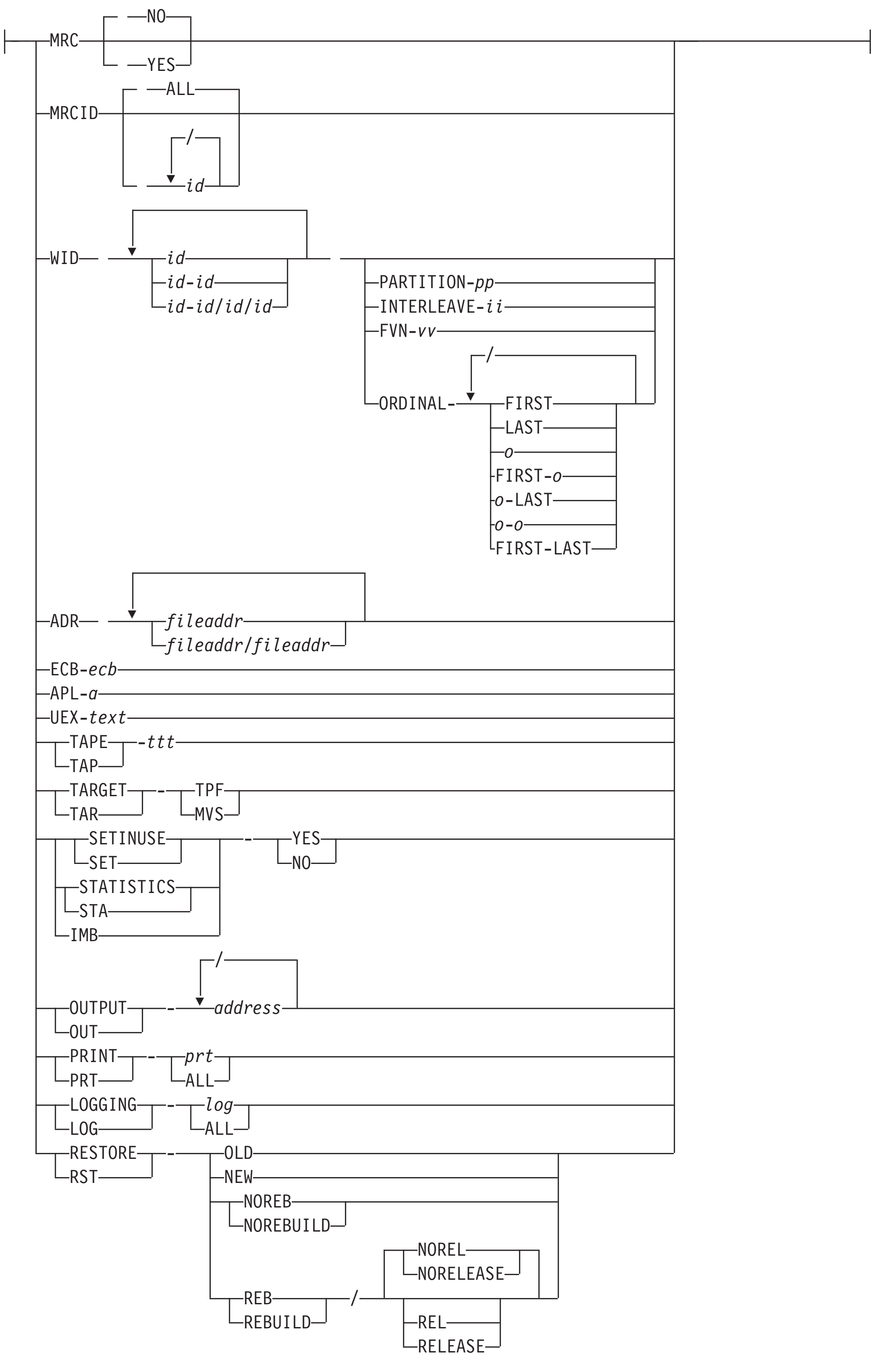
Requirements and Restrictions

- You must lock the parameter table before you can use the ZFCRU SETUP command.
- The ADR and WID parameters are not supported with CRUISE restore processing when the database is captured using multiple reference check (MRC) support. Using the STATISTICS parameter with MRC support can result in incomplete statistics because not all paths are chain chased during MRC processing.

Format



Other Parameters:



FUNCTION
specifies the function to be performed.

CAPTURE
specifies the data to be captured to tape.

RESTORE
specifies the data to be restored from tape.

VERIFY
verifies data structures, and reports broken chains and references that are not valid.

PACK
specifies data structures to pack according to the database definition (DBDEF) macro.

Note: Because packing large database structures requires a significant amount of system resources, commit scope processing is not available with the pack function, regardless of what the DBDEF macro has set as the default.

Note: You must specify a chain chasing parameter when you use the capture, verify, or pack functions.

REF
specifies as many as 30 hexadecimal fixed file identifiers (IDs) to chain chase.

Note: The subparameters associated with the REF parameter (PARTITION, INTERLEAVE, FVN, and ORDINAL) are related to a single REF ID or wanted file ID (WID). You cannot specify them using an ID range.

id is a record ID in 2-character alphanumeric or 4-character hexadecimal format.

id-id
is a range of IDs in 2-character or 4-character hexadecimal format.

id-id/*id*/*id*
is a list of IDs. You can specify single IDs and ID ranges in mixed format.

PARTITION-*pp*
specifies a fixed file partition to be processed, where *pp* is the partition for TPFDF files.

Note: This parameter works for TPFDF files only.

INTERLEAVE-*ii*
specifies a fixed file interleave to be processed, where *ii* is the interleave number for TPFDF files.

Note: This parameter works for TPFDF files only.

FVN-*vv*
specifies the file version (if applicable), where *vv* is the file version number.

ORDINAL
specifies an ordinal number or range of ordinal numbers to be processed.

FIRST
specifies the first relative ordinal number.

LAST
specifies the last relative ordinal number.

o is a relative ordinal number. You can specify a relative ordinal number with a maximum size of 8 digits.

FIRST-*o*
specifies a range of relative ordinal numbers starting with the first ordinal number.

*o***-LAST**
specifies a range of relative ordinal numbers ending with the last ordinal number.

*o***-***o*
is a range of relative ordinal numbers.

FIRST/LAST
specifies a list of relative ordinal numbers beginning with the first and ending with the last ordinal number.

Note: You can specify a maximum of 10 relative ordinal number ranges.

FAD
specifies as many as fifty 8-character hexadecimal file addresses to be chain chased.

Note: The file addresses specified with this parameter are started when you use the RESTORE parameter with the REBUILD option.

fileaddr
is an 8-character hexadecimal file address.

fileaddr/fileaddr
is a list of 8-character hexadecimal file addresses.

MRC
specifies multiple reference check (MRC) support during CRUISE capture processing, where:

NO
specifies that MRC support is not used during CRUISE capture processing.

YES
specifies that MRC support is used during CRUISE capture processing.

MRC support is a method that modifies CRUISE capture processing so that detailed files referenced from more than one index file are chain chased only once.

MRCID
specifies the file IDs to be included with MRC processing, where:

ALL
specifies that all file IDs are to be included with MRC processing.

id is the file ID to be included with MRC processing. You can specify as many as 10 file IDs.

Notes:

1. Because the MRC parameter with the NO value specified ignores the MRCID parameter, they can be used together.
2. If you use the MRCID parameter, the MRC parameter with the YES value is automatically specified by the TPFDF product.

WID
sets up as many as 30 wanted file IDs (WIDs) to be chain chased.

Notes:

1. The WID parameter can be a fixed or pool file.
2. The WID parameter limits the functions to the specified file IDs while chain chasing the structure referenced by the REF or FAD parameter. For example, if the file ID specified with the WID parameter is equal to the fixed file ID specified with the REF parameter, CRUISE will only process the fixed file and disregard any existing references to another file.
3. When you use the RESTORE parameter, the WID parameter specifies the first file ID to be restored. Additionally, all forward chain blocks (if any) will be restored. If you use the IMB parameter, all referenced subfiles will also be processed.
4. The subparameters associated with the WID parameter (PARTITION, INTERLEAVE, FVN, and ORDINAL) are related to a single REF ID or wanted file ID. You cannot specify them using an ID range.
5. The WID parameter is not supported with CRUISE restore processing when the database is captured using MRC support.

ADR
specifies a search of as many as fifty 8-character hexadecimal file addresses. CRUISE will report all references of the specified file addresses during chain chasing as specified by the REF or FAD parameter.

The restore parameter will process this file address and all of its forward chains. When you use the IMB parameter, it will also process all referenced subfiles.

Note: The ADR parameter is not supported with CRUISE restore processing when the database is captured using MRC support.

ECB-*ecb*
specifies the percent of total entry control blocks (ECBs) to be distributed, where *ecb* is a value in the range 1–100%. One hundred percent is the total number of ECBs allocated on the TPF system.

APL-*a*
specifies the application ID of the files that the CRUISE function will process, where *a* is the application code.

Note: This parameter applies to the fixed files specified with the REF parameter.

UEX-*text*
specifies a text string, where text is user-specific text up to 103 bytes that will be available to the user exit segment. The text is contained in the CRUISE workblock (IRCHDF) at assembler label IRCHUTX. The CRUISE workblock is contained in global CINFC, CMMCRU label. You can use the vertical bar (|) to switch between hexadecimal and character input.

TAPE-*ttt*
specifies the requested tape or general data set (GDS), where *ttt* is any user-defined tape. This must be defined in the tape label mask record (TLMR).

TARGET
specifies the target system where the data will be restored. This parameter is used for a CRUISE capture only. TPF system is the default.

TPF
specifies the data that will be restored on a TPF system or an ALCS environment.

MVS
specifies the data that will be restored on an MVS system.

SETINUSE
specifies pool file use, where:

YES
calls case 3 of user exit macro DFUEX during the processing of the RESTORE and VERIFY parameters when pool files are used. The RESTORE parameter must have the REF and FAD parameters specified when you use this option.

Note: You can use this parameter only in NORM state.

NO
does not call case 3 of user exit macro DFUEX during the processing of the RESTORE and VERIFY parameters when pool files are used.

STATISTICS
specifies statistics from the parameter table, where:

YES
specifies that file statistics that are built following successful parameter table processing.

NO
specifies that file statistics are not built following successful parameter table processing.

Note: Using this parameter with MRC support can result in incomplete statistics because not all paths are chain chased during MRC processing.

IMB
specifies embedded references, where:

YES
specifies that references embedded in files indicated with the WID parameter are included in CRUISE capture, restore, or pack processing.

NO
specifies that references embedded in files indicated with the WID parameter are ***not*** included in CRUISE capture, restore, or pack processing. When you use this parameter, only files indicated with the WID parameter are included in capture, restore, or pack processing.

OUTPUT-*address*
specifies the output device, where *address* is a 6-digit hexadecimal printer or terminal address. You can specify as many as three output addresses.

Note: If you start a CRUISE function from a terminal that is not specified as an output terminal, only start and end messages are displayed. However, a prime CRAS receives messages regardless of where a CRUISE function was started.

PRINT
specifies the number of messages to be printed to the output device selected with the ZFCRU OUTPUT command, where

prt is a number in the range 0-100.

ALL
specifies all the messages are to be printed.

Notes:

1. Printing will stop when the number you specified is reached.
2. If the number of messages you specify is less than the number of messages printed, printing will end.
3. The specified number of messages is in effect only when the CRUISE function is in running state.

LOGGING
specifies messages to be logged to the parameter table, where:

log
is a number in the range 0-100.

ALL
specifies that all the messages are to be logged to the parameter table logging file.

Notes:

1. Logging ends when the specified number of logged messages is reached.
2. Logging will run until available pool space is used if you use the ALL parameter with the ZFCRU LOG command.
3. If the number of logged messages you specify is less than the number of messages logged, logging will complete and no messages are deleted.
4. The number of messages you specified to be logged is in effect only when the CRUISE function is in running state.

RESTORE
specifies the method in which CRUISE restores data, where:

OLD
specifies that the database is restored to the same fixed and pool file addresses that were captured.

NEW
specifies that the captured fixed and pool files are restored to new pool file addresses where the data structure is rebuilt.

Note: You can use this parameter only in NORM state.

NOREBUILD
specifies that the captured fixed and pool files are restored to new pool file addresses and the data structure is not rebuilt.

Note: You can use this parameter only in NORM state.

REBUILD
specifies that the captured fixed files are restored to the same ordinals that were captured. The captured pool files are restored to new pool file addresses and the database structure is rebuilt.

Note: You can use this parameter only in NORM state.

NORELEASE
specifies that old pool file addresses are not released when the structure is rebuilt.

Note: You can use this parameter only in NORM state.

RELEASE
specifies that old pool file addresses are released when the data structure is rebuilt.

Note: You can use this parameter only in NORM state.

Note: To improve system performance, pack a subfile using the pack operation after restore processing on B$^+$Tree and block index files.

Note: You can also specify the optional parameter names in their abbreviated form as shown in the syntax diagram.

Additional Information

- If your database contains a large amount of data with subfiles referenced by index files, MRC support improves the performance of CRUISE capture processing by chain chasing each subfile once. By eliminating redundant chain chasing, MRC support reduces the amount of tape required and increases the performance of CRUISE capture processing. The performance of CRUISE restore and verify processing is also improved. Additionally, MRC support enhances the management of your system resources with improved CPU and input/output (I/O) time.
- The ADR and WID parameters are not supported with CRUISE restore processing when the database is captured using MRC support. Using the STATISTICS parameter with MRC support can result in incomplete statistics because not all paths are chain chased during MRC processing.
- You must pack a B$^+$Tree and block index file to validate the file references after CRUISE capture and restore processing because CRUISE capture and restore processing nullifies the validation of block index technical LRECs (TLRs) and releases B$^+$Tree files.

Examples

In the following example, a parameter table (TABLE1) is updated to verify reference ID range 0000–FFFF.

```
User:    ZFCRU SETUP FUN-VERIFY REF-0000-FFFF

System:  FCRU0000I 10.51.11 MODIFIED
         FCRU0000I 10.51.11 PARAMETER TABLE DISPLAY FOR TABLE1
         FUNCTION                   VER
         USABLE ECB START VALUE     33 PCT
         SET POOL ADDRESSES IN USE  NO
         BUILD STATISTICS           YES
         NUMBER OF PRINT MESSAGES   100
         OUTPUT ADDRESSES           306919  000000  000000
         NUMBER OF LOG MESSAGES     ALL
         REFERENCE ID               0000 - FFFF
         FCRU0000I 10.51.11 END OF DISPLAY
```

In the following example, a parameter table (TABLE1) is updated to verify six ranges of relative ordinal numbers for reference ID B075.

```
User:    ZFCRU SETUP FUN-VER REF-B075 ORD-1-2/4-5/669-771/1115330-0206380/1234567
         -12345670/12345678-12345679

System:  CSMP0097I 14.55.28 CPU-B SS-BSS  SSU-HPN  IS-01
         FCRU0042I 14.55.28 MODIFIED
         FCRU0000I 14.55.28 PARAMETER TABLE DISPLAY FOR TABLE1
         FUNCTION                   VER
         USABLE ECB START VALUE     50 PCT
         SET POOL ADDRESSES IN USE  NO
         BUILD STATISTICS           NO
         NUMBER OF PRINT MESSAGES   ALL
         NUMBER OF LOG MESSAGES     ALL
         REFERENCE ID               B075
         ORDINALS                   0000001-0000002 0000004-0000005
                                    0000669-0000771 0206380-1115330
                                    01234567-12345670
                                    12345678-12345679
         FCRU0000I 14.55.28 END OF DISPLAY
```

References

- See the following commands for information about overriding parameter table default values while the CRUISE function is in running state.
  - “ZFCRU ECB–Change ECB Percentage” on page 13
  - “ZFCRU LOGGING–Log Messages” on page 19
  - “ZFCRU OUTPUT–Specify Printer and Terminal Address” on page 20
  - “ZFCRU PRINT–Report Parameters” on page 22.
- See *TPFDF Programming Concepts and Reference* for more information about commit scopes.

ZFCRU START–Start CRUISE

Use this command to prepare and then start a capture/restore utility, information and statistics environment (CRUISE) function.

Requirements and Restrictions

- You must enter this command twice to start a CRUISE function.

  When you enter this command for the first time, the specified parameter table is displayed. Additionally, CRUISE performs a verification check on the specified parameter table. If a verification check error is detected, information that you need to correct the error is displayed. You will not be able to start a CRUISE function with a verification check error. If no verification check errors are detected, the CRUISE function is in active state.

  When you enter this command for the second time, parameter table processing will start. The CRUISE function is in running state.
- You cannot use this command when the parameter table is locked.
- When you use the ZFCRU START command, only 1 parameter table per processor can be in active, running, paused, stopped state.

Format

```
►►──ZFCRU START-name──────────────────────────────────────────────►◄
```

name
specifies a parameter table, where *name* is a valid name.

Additional Information

- You can use the ZFCRU START command while the CRUISE function is processing existing parameter tables; that is, the CRUISE function is active, running, paused, or stopped.
- CRUISE will not time out if you do not enter a second ZFCRU START command. You must enter the ZFCRU ABORT command if you are not going to enter the ZFCRU START command a second time.

Examples

In the following example, the ZFCRU START command is entered for the first time. A verification check has been completed successfully on the parameter table (TABLE1) and the CRUISE function is in active state.

ZFCRU START

```
User:    ZFCRU START-TABLE1

System:  FCRU0149I 10.42.37 TABLE -TABLE1- ACTIVATED
         FCRU0000I 10.42.37 PARAMETER TABLE DISPLAY FOR TABLE1
           FUNCTION                   VER
           USABLE ECB START VALUE     33 PCT
           SET POOL ADDRESSES IN USE  NO
           BUILD STATISTICS           YES
           NUMBER OF PRINT MESSAGES   100
           OUTPUT ADDRESSES           306919  000000  000000
           NUMBER OF LOG MESSAGES     ALL
           REFERENCE ID               E E
         FCRU0000I 10.42.37 REPEAT ENTRY TO START CRUISE
```

In the following example, the ZFCRU START command is entered a second time. The CRUISE function is in running state.

```
User:    ZFCRU START-TABLE1

System:  FCRU0100I 10.43.09 TABLE -TABLE1- FUNCTION-VER STARTED
```

References

None.

ZFCRU STOP–STOP CRUISE

Use this command to immediately stop the capture/restore utility, information and statistics environment (CRUISE) function. This command stops the processing of the current ordinal and, if applicable, keeps the current tape open.

Requirements and Restrictions

You can use the ZFCRU STOP command while the CRUISE function is processing existing parameter tables; that is, the CRUISE function is in running state.

Format

```
►►──ZFCRU STOP──────────────────────────────────────────────────────────►◄
```

Additional Information

You must enter the ZFCRU RESTART command to restart a parameter table in stopped state.

Examples

In the following example, the parameter table (TABLE1) is stopped.

```
User:    ZFCRU STOP

System:  FCRU0126I 10.49.32 00185426 RECORDS FOUND
         FCRU0142I 10.49.32 CRUISE IS STOPPING
         FCRU0143I 10.49.32 TABLE -TABLE1- STOPPED
```

References

See “ZFCRU RESTART–RESTART CRUISE” on page 23 for more information about restarting CRUISE after you use ZFCRU STOP command.

ZFCRU UNLOCK–Unlock a Parameter Table

Use this command to unlock a parameter table.

Requirements and Restrictions

You must have a locked parameter table to use the ZFCRU UNLOCK command.

Format

```
►►──ZFCRU UNLOCK-──┬──────┬───────────────────────────────────────────────►◄
                   └─name─┘
```

name
specifies the parameter table name to be unlocked.

If you specify this parameter, the parameter table will be unlocked regardless of where the terminal lock was entered.

Additional Information

If you do not specify a parameter table name with the ZFCRU UNLOCK command, the locked table for the current terminal address is unlocked.

Examples

In the following example, the parameter table (TABLE1) is unlocked.

```
User:    ZFCRU UNLOCK-TABLE1

System:  FCRU0009I 10.52.12 OK TABLE - TABLE1 - UNLOCKED
```

References

None.

Recoup

This chapter describes the TPFDF recoup commands.

TPFDF recoup is a real-time database validation utility that runs as an extension to TPF or ALCS recoup phase 1. Its primary purpose is to provide a database administrator or system programmer with support to maintain long-term pool records. Long-term records that are not in use can be identified and returned to the ALCS or TPF system.

TPFDF recoup uses information contained in database definition (DBDEF) tables to chain chase long-term pool file records that are defined using the DBDEF macro. See *TPFDF Database Administration* for more information about the DBDEF macro.

The following is an example of the message that is displayed after each file ID is chain chased during TPFDF recoup. You can suppress this message by setting symbol &DFREP to 1 in the DBLCL macro. See *TPFDF Installation and Customization* for more information about the DBLCL macro.

```
RECP9011I 12.45.53 TPFDF RECOUP - ACTIVITY REPORT
** ID-X'B075',FVN 00,DBDEF: UF97,FILE: IR75DF
** CPU-B,SS-BSS,SSU-HPN,LEVEL-8,ECBS-0
** TIME: 4S,I/O: 213
** ERRORS : ID: 0,TOTAL: 0,FIXED: 0
** RECORDS: ID: 855,TOTAL: 935
```

TPFDF Recoup on a TPF System

When TPFDF recoup is installed on a TPF system, the following ZRECP commands are available in addition to the TPF ZRECP commands:

- ZRECP LOG
- ZRECP PRT
- ZRECP REPORT
- ZRECP STA.

In addition, TPF commands ZRECP SEL and ZRECP RECALL are enhanced to include a file version number as an additional parameter. See *TPF Operations* for more information about the ZRECP SEL and ZRECP RECALL commands.

The following syntax diagrams show how the TPFDF file version parameter can be used with TPF commands ZRECP SEL and ZRECP RECALL.

```
                                   ┌─0───────┐
►►──ZRECP SEL── ──recid── ──addr── ─┴─filever─┴──────────────────────────►◄
```

```
►►──ZRECP RECALL──┬──────────────────────────┬──────────────────────►◄
                  └─ SEL recid──┬─0───────┬──┘
                                └─filever─┘
```

SEL
starts recoup phase 1 selective chain-chasing for the specified record type.

recid
is the 2- or 4-character record ID of a fixed file or pool record.

addr
is the file address of a fixed file or pool record. If the record is a global pool record, specify the file address offset for the pool record.

filever
is a file version number given in hexadecimal format from X'00' to X'FF'.

RECALL
specifies a recall to recoup.

TPFDF Recoup in an ALCS Environment

In an ALCS environment, TPFDF recoup is installed using the TPFDF sample recoup user exit code. See *TPFDF Installation and Customization* for more information about the installation of the recoup user exit code.

ZRECP LOG–Display a Recoup Log

Use this command to display the contents of the recoup log file.

TPFDF recoup produces a log, of the primary processor, containing information about files processed and the number of errors detected from the last recoup.

Requirements and Restrictions

None.

Format

```
►►──ZRECP LOG──┬───┬──────────────────────────────────────────────────────►◄
               └─*─┘
```

***** displays only the general information messages, which are those containing two asterisks(**).

Additional Information

If you enter the ZRECP LOG command while recoup phase 1 is not chain chasing the TPFDF database, no status is displayed.

Examples

The following example displays the contents of the recoup log file (GR0ZSR).

```
User:   ZRECP LOG

System: System: RECP9009I          RECOUP PHASE 1 ACTIVITY LOG

                HPN  14.37.23  ** 28MAY97 DATE OF LAST RECOUP RUN
                HPN  14.37.23  ** 14.37 PHASE I STARTED
                HPN  14.37.23  ** 00000000 TRAD. BLOCKS PROCESSED
                HPN  14.37.23  ** PARAM RECOUP STARTED
                HPN  14.37.23  ** *- NON-FIXED-MODE PROCESSING -*
                HPN  14.37.23  B073 00  F 00000026  T 0000 I/O 00026
                HPN  14.37.24  ** 00000000 TPFDF RECOUP ERRORS DETECTED
                HPN  14.37.24  ** 000000026 TPFDF RECOUP BLOCKS PROCESSED
                HPN  14.37.24  ** PARAMETRIC RECOUP COMPLETED
                HPN  14.37.24  ** 00000000 FIXED ERRORS
```

References

See *TPF Operations* for more information about the TPF ZRECP command.

ZRECP MPLOG–Display TPFDF Recoup Status

Use this command to display the status of TPFDF recoup while it is processing. You can display the TPFDF records chain chased on a processor when TPFDF recoup processing is complete.

Requirements and Restrictions

None.

Format

```
►►──ZRECP MPLOG──────────────────────────────────────────────────────────►◄
```

Additional Information

None.

Examples

The following example displays TPFDF recoup status.

```
User:   ZRECP MPLOG

System: RecID: FD22  COMPLETED   PROC: B  01/11/2001 17:33:13 - 17:33:15
        RecID: FDDD  COMPLETED   PROC: B  01/11/2001 17:33:15 - 17:33:17
        RecID: B071  COMPLETED   PROC: B  01/11/2001 17:33:17 - 17:33:18
        RecID: FD1C  COMPLETED   PROC: B  01/11/2001 17:33:17 - 17:33:18
        RecID: FDE9  COMPLETED   PROC: B  01/11/2001 17:33:18 - 17:33:18
        RecID: FDF3  COMPLETED   PROC: B  01/11/2001 17:33:18 - 17:33:18
        RecID: FD1F  COMPLETED   PROC: B  01/11/2001 17:33:18 - 17:33:18
        RecID: FDED  COMPLETED   PROC: B  01/11/2001 17:33:18 - 17:33:19
        RecID: FDF6  COMPLETED   PROC: B  01/11/2001 17:33:19 - 17:33:19
        RecID: FD17  IN PROGRESS PROC: B  01/11/2001 17:33:19
        RecID: FDDC  MPPRCD-NONE OUTSTANDING
        RecID: FDE5  MPPRCD-NONE OUTSTANDING
        RecID: FD1B  MPPRCD-NONE OUTSTANDING
        RecID: FDDF  MPPRCD-NONE OUTSTANDING
        RecID: FDE8  MPPRCD-NONE OUTSTANDING
        RecID: FDFD  MPPRCD-NONE OUTSTANDING
        RecID: B073  MPPRCD-NONE OUTSTANDING
        RecID: FD1E  MPPRCD-NONE OUTSTANDING
        RecID: FDA5  MPPRCD-NONE OUTSTANDING
        RecID: FDE1  MPPRCD-NONE OUTSTANDING
```

References

See *TPF Operations* for more information about the TPF ZRECP command.

ZRECP PRT–Specify the Print Option

Use this command to write information about broken chains and all specific file ID recoup information (for example, recoup retrieval error messages or TPFDF activity reports) to a specified printer.

Requirements and Restrictions

None.

Format

```
►►──ZRECP PRTprinter_addr──────────────────────────────────────────────►◄
```

printer_addr
is a 6-digit hexadecimal printer address for TPFDF recoup information.

Additional Information

- A TPFDF recoup activity report is printed every 5 minutes on the specified printer.
- If you do not want information about broken chains printed on the specified printer, set symbol &RECERR to 1 in the DBLCL macro. See *TPFDF Installation and Customization* for more information about the DBLCL macro.

Examples

In the following example, the ZRECP PRT command routes information about broken chains and file ID specific recoup information to the specified printer.

```
User:   ZRECP PRT010000

System: RECP9013I 17:23:32 TPFDF RECOUP PRINTER ADDRESS SET UP
```

References

See *TPF Operations* for more information about the TPF ZRECP command.

ZRECP Q RESET–Reset Queue Control Indicator

Use this command to reset the queue control indicator. For example, if you restart recoup following a system outage and receive a `QUE-ENTRIES INHIBITED` message, the ZRECP Q RESET command will reset the queue control indicator.

Requirements and Restrictions

None.

Format

```
►►──ZRECP Q RESET──────────────────────────────────────────────────────────►◄
```

Additional Information

None.

Examples

In the following example, the recoup queue control indicator is reset.

```
User:   ZRECP Q RESET

System: RECP9124I 08.14.31 RECOUP QUE-INDICATOR RESET
```

References

See *TPFDF Database Administration* for more information about the queue control indicator.

ZRECP REPORT–Recoup Activity Report

Use this command to display the activity report during TPFDF recoup.

Requirements and Restrictions

None.

Format

```
►►──ZRECP──┬─────────────┬─ REPORT──────────────────────────────────────────►◄
           └─ PROC cpuid─┘
```

cpuid
specifies the ID of the processor for which you want to display TPFDF recoup status.

Additional Information

The TPFDF recoup activity report can also be printed automatically every 5 minutes to a printer by using the ZRECP PRT command.

Examples

The following example displays the recoup activity report.

```
User:   ZRECP REPORT
System: RECP9011I 17:32:23 TPFDF RECOUP - ACTIVITY REPORT
        ** ID-X'C1C1',FVN 00,DBDEF: UF1A,FILE: IRDIDF
        ** CPU-B,SS-GW,SSU-GW1,LEVEL-99,ECBS-40
        ** TIME: 6S,I/O: 13
        ** ERRORS : ID: 0,TOTAL: 0,FIXED: 0
        ** RECORDS: ID: 81,TOTAL: 81
        ** RCT: 0193,BOR: 0,EOR: 498,CRD: 120
        ** RCP SEQ: 0,RESTART: 0,PRINTER: 010000
        ** LEVEL: 01010100-0028004F 00570060 006A0075 0081008E
        ** ECB ALLOCATION: 00270000 00000000 00000000 00000000
        ** RECALC: 1/500,CHAINS: 1/0-1/0,X'C1C1',FVN 00
        ** PAUSE: OFF,FIXED ERRORS: NO,ABORT: NO
```

The fields in the activity report contain the following information:

ID	The currently processed file ID.
FVN	The file version number of the currently processed data structure.
DBDEF	The database definition (DBDEF) program containing the current data structure.
FILE	The DSECT name.
CPU	The control processing unit (CPU) on which the records are processed.
SS	The subsystem name.
SSU	The subsystem user (SSU) name.
LEVEL	The maximum number of entry control blocks (ECBs) that can be allocated.

ECBS		The currently active ECBs.
TIME		The time used so far to process the data structure.
I/O		The number of inputs and outputs (I/Os) for this data structure per second.
ERRORS		The following information is displayed:
	ID	Indicates the broken chain count for this ID.
	TOTAL	Indicates the total number of broken chains.
	FIXED	Indicates the number of fixed errors remaining since the last full or recall selective recoup was run.
RECORDS		The following information is displayed:
	ID	Indicates the number of recoup references found in the current data structure.
	TOTAL	Indicates the total number of recoup references.
RCT		The record type.
BOR		The beginning ordinal number.
EOR		The ending ordinal number.
CRD		The currently processed ordinal number.
RCP SEQ		The RCP tape sequence numbers.
RESTART		The number of recoup restarts in TPFDF recoup.
PRINTER		The printer address specified with the ZRECP PRT command.
LEVEL		The first 4 bytes (X'01010100' in the example) provide the following information:
	Byte 0	The levels of the data structures calculated from DBDEF.
	Byte 1	The level value of the last recalculation.
	Byte 2	The current level value.
	Byte 3	The level value that is reset every 20th ordinal to detect level values lower than that of the last recalculation (byte 1) in order to shift ECB allocation to upper levels.
		The next 2 bytes (X'0028' in the example) indicate the top-level ECB allocation.
		The remaining bytes indicate the ECB distribution values for the other 7 levels.
ECB ALLOCATION		Indicates the maximum ECBs actually allocated for each level.
RECALC		The first number (1 in the example) indicates the number of recalculations. The second number (500 in the example) indicates the number of ordinals before the next check for shifting the ECB allocation to upper levels.
CHAINS		The first number (1 in the example) indicates the level of the

maximum chain count. The second number (0 in the example) indicates the maximum chain count.

The third and fourth numbers (1/0 in the example) indicate the level and chain count that gets reset every 50 chains to determine the chaining trend.

The ID and file version number of the maximum chaining structure is also shown on this line of the display.

PAUSE, FIXED ERRORS, ABORT
Indicates the status of TPFDF recoup.

References

See *TPF Operations* for more information about the TPF ZRECP command.

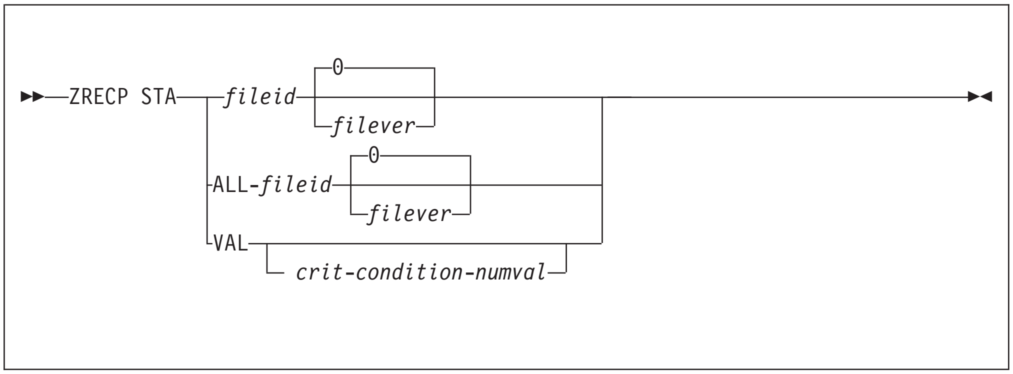
ZRECP STA–Display Statistical Data

Use this command to display statistics for a file with a specified file ID (and file version number).

Requirements and Restrictions

None.

Format



fileid
is a 2-character alphanumeric or 4-character hexadecimal file ID. If you do not specify the ALL parameter, statistics for this file ID are displayed from the last recoup.

filever
is the file version of the file ID, which is a hexadecimal number from X'00' to X'FF'.

ALL
displays all the available statistics for the file ID.

VAL
displays, without any additional parameters, the criteria available to select statistics. With additional parameters, statistics meeting the conditions specified are displayed.

crit
is the criteria used to select recoup statistics. You can display a criteria list by entering ZRECP STA VAL without additional parameters.

condition
is specified by the following:

H Higher (greater) than

E Equal

L Lower (less) than.

numval
is the numerical value specified to compare against the condition.

Additional Information

When you request statistics for files that are referenced with recoup chain-chasing indicator (RCI) IDs, the display will be different depending on whether there was a full recoup or a selective recoup done for those IDs.

Examples

The following example displays the statistics for a B$^+$Tree data file (file ID FDED) and an associated B$^+$Tree index file (file ID BBBB). The B$^+$Tree file has 8000 data blocks and 421 nodes. File version 00 of FDED shows the statistics for the prime block. It shows one block (the prime block). File version 01 of FDED shows that 7999 overflow data blocks have been chain chased. File version 00 of BBBB shows the statistics for the node blocks.

Note: The statistics for a B$^+$Tree file may vary depending on how the database administrator defines chain chasing.

```
User:   ZRECP STA FDED

System: RECP0101I          ZRECP STATISTICS DISPLAY
        RECOUP  28MAY97  14.19.23  -  ID-FDED    TOTAL COUNT: 00000026
        --------------------------------------------------------------
         FDED 00 BLOCKS  P/00000001  F/00000001  E/00000000  X/00000000
                 CHAINS  1/00000000  2/00000000  3/00000000  4/00000000
                         5/00000000  6/00000000  7/00000000  8/00000000
                         9/00000000  M/00000000  MAX/000000  A/00000000
                 USAGE F 0/000000 25/000000 50/000000 75/000001 PCT/096
                 USAGE N 0/000000 25/000000 50/000000 75/000000 PCT/000
                 ITEMS   T/00000076 F/000076 B/000002 ITS/0001 PNB/0016

         BBBB 00 BLOCKS  P/00000001  F/00000421  E/00000000  X/00000000
                 CHAINS  1/00000000  2/00000000  3/00000000  4/00000000
                         5/00000000  6/00000000  7/00000000  8/00000000
                         9/00000000  M/00000001  MAX/000420  A/802B650B
                 USAGE F 0/000000 25/000000 50/000000 75/000420 PCT/095
                 USAGE N 0/000000 25/000000 50/000000 75/000001 PCT/095
                 ITEMS   T/00008420 F/008420 B/000002 ITS/0001 PNB/0016

         FDED 01 BLOCKS  P/00000001  F/00008000  E/00000000  X/00000000
                 CHAINS  1/00000000  2/00000000  3/00000000  4/00000000
                         5/00000000  6/00000000  7/00000000  8/00000000
                         9/00000000  M/00000001  MAX/007999  A/80080330
                 USAGE F 0/000000 25/000000 50/000000 75/007999 PCT/096
                 USAGE N 0/000000 25/000000 50/000000 75/000001 PCT/096
                 ITEMS   T/00608000 F/608000 B/000002 ITS/0001 PNB/0016
```

The fields in the ZRECP STA *fileid* display provide the following information:

BLOCKS	Shows the following:	
	P/	Number of prime blocks.
	F/	Total number of blocks (prime and overflow).
	E/	Error count (the number of prime blocks with broken chains).
	X/	Number of fixed errors.
CHAINS	Shows the following:	
	1/	Prime blocks with one overflow block.
	2/	Prime blocks with two overflow blocks.
	3/	Prime blocks with three overflow blocks.
	4/	Prime blocks with four overflow blocks.
	5/	Prime blocks with five overflow blocks.
	6/	Prime blocks with six overflow blocks.
	7/	Prime blocks with seven overflow blocks.

8/	Prime blocks with eight overflow blocks.
9/	Prime blocks with nine overflow blocks.
M/	Prime blocks with more than nine overflow blocks.
MAX/	Number of overflow blocks in the longest chain.
A/	The file address of the prime block that has the largest number of overflow blocks (specified by **MAX/**).

The following display is only generated for a TPFDF file:

USAGE F	Provides statistics about the use of all blocks (prime and overflow) that have an overflow block attached through the forward chain field.

0/	Number of blocks 0 to 25% filled.
25/	Number of blocks 25 to 50% filled.
50/	Number of blocks 50 to 75% filled.
75/	Number of blocks 75 to 100% filled.
PCT/	The average use (in percentage) of all blocks associated with a file.

Note: USAGE N provides the same type of information as USAGE F, except it applies to the last blocks in the overflow chains.

ITEMS	Provides information about the LRECs in a TPFDF file:

T/	Total number of LRECs.
F/	Average number of LRECs in a subfile (calculated as **T/** divided by **P/**).
B/	Average number of LRECs in each block (calculated as **T/** divided by **F/**).
ITS/	LREC size, as defined in the file’s DSECT. A value of 1 indicates that the LREC is of variable length.
PNB/	Position of next available byte (NAB) pointer.

The following example displays the keywords available to specify criteria for selecting recoup statistics:

```
User:   ZRECP STA VAL

System: RECP9010I          STATISTICS VALIDATION PARAMETERS

        P    NR OF PRIME BLOCKS         F    NR OF FOUND BLOCKS
        E    NUMBER OF ERRORS           X    FIX ERROR COUNT
        1    NR OF FILES WITH 1 CHAIN   2    NR OF FILES WITH 2 CHAINS
        3    NR OF FILES WITH 3 CHAINS  4    NR OF FILES WITH 4 CHAINS
        5    NR OF FILES WITH 5 CHAINS  6    NR OF FILES WITH 6 CHAINS
        7    NR OF FILES WITH 7 CHAINS  8    NR OF FILES WITH 8 CHAINS
        9    NR OF FILES WITH 9 CHAINS  M    NR OF FILES WITH MT 9 CH.
        MAX  MAXIMUM NR OF CHAINS       A    MAX CHAIN FILE ADDRESS
        F0   BLOCKUSAGE 0-25PCT FWCH    F25  BLOCKUSAGE 25-50PCT FWCH
        F50  BLOCKUSAGE 50-75PCT FWCH   F75  BLOCKUSAGE 75-100PCT FWCH
        N0   BLOCKUSAGE 0-25PCT NOFCH   N25  BLOCKUSAGE 25-50PCT NOFCH
        N50  BLOCKUSAGE 50-75PCT NOFCH  N75  BLOCKUSAGE 75-100PCT NFCH
        T    TOTAL NR OF LRECS          ITS  TPFDF LREC SIZE
        PNB  POSITION OF NAB            LB   NR OF LRECS PER BLOCK
        LF   NR OF LRECS PER FILE       NPC  AVERAGE BLOCKUSAGE NOFCH
        FPC  AVERAGE BLOCKUSAGE FCH
```

The following example displays statistics for all recoup runs in the statistics file for version 0 of file B075.

```
User:   ZRECP STA ALL-B0750

System: RECOUP  29MAY97  17.30.57  -  ID-B075    TOTAL COUNT: 00000128
        -------------------------------------------------------------

        B075 00 BLOCKS  P/00000026  F/00000128  E/00000000  X/00000000
                CHAINS  1/00000000  2/00000000  3/00000001  4/00000000
                        5/00000000  6/00000000  7/00000000  8/00000000
                        9/00000000  M/00000003  MAX/000033  A/F404F8D0
                USAGE F 0/000000 25/000069 50/000000 75/000033 PCT/049
                USAGE N 0/000025 25/000001 50/000000 75/000000 PCT/008
                ITEMS   T/00000310 F/000011 B/000002 ITS/0001 PNB/0016
```

The following example displays all files processed during TPFDF recoup that have more than four LRECs in the file.

```
User:   ZRECP STA VAL T-H-4

System: THESE RECOUP STATISTICS MATCH
        TOTAL NR OF LRECS HIGHER THAN 4

        ID   FVN         T    ID   FVN         T    ID   FVN         T
        ---- ---  --------    ---- ---  --------    ---- ---  --------
        B075  00       425    FDA5  00        25    FDA4  00        50
        FDF3  00        35    FD03  00      1142    FDF5  00         4
        FDF6  00        22    FD00  00        23
```

References

- See *TPF Operations* for more information about the TPF ZRECP command.
- See *TPFDF Database Administration* for additional information about RCI processing.

Data Collection

This chapter describes TPFDF data collection commands.

These commands display statistics relating to system use and are used by those who maintain the TPFDF product (for example, database administrators or system programmers).

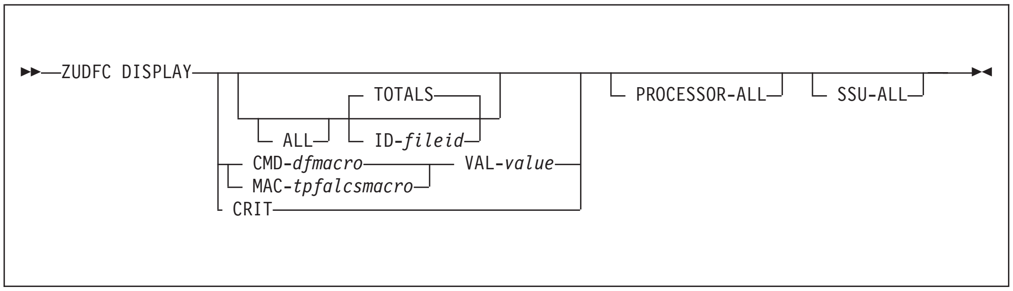
ZUDFC DISPLAY–Display Data Collection

Use this command to display data collection statistics.

Requirements and Restrictions

- You cannot view the statistics for a data collection that is currently running.
- If you are in a TPF environment, you can enter ZUDFC commands from any subsystem user (SSU). The commands can apply to the current subsystem user or all subsystem users in the subsystem.

Format



ALL
displays all historical data collection statistics (for a maximum of 30 data collections) for the specified ID or for TOTALS.

ID-*fileid*
displays data collection statistics for one ID, where *fileid* is a 2-character alphanumeric or 4-character hexadecimal file ID.

TOTALS
displays data collection statistics for all IDs.

CMD-*dfmacro*
displays data collection statistics for all files, where *dfmacro* is a TPFDF macro and its averages are greater than the number of calls per second specified by the VAL parameter.

MAC-*tpfalcsmacro*
displays data collection statistics for all files, where the *tpfalcsmacro* is a TPF or ALCS macro and its averages are greater than the number of calls per second specified by the VAL parameter.

VAL-*value*
displays data collection statistics, where *value* is the number of calls per second. Data collection statistics are displayed for all files where the average for the macro selected is greater than the calls per second value specified.

CRIT
displays data collection statistics, as defined in user exit UFE7, for files that issue more than 10 percent of all DETAC or ATTAC macros, or files in which the ratio of chain record FINDs is at least 1.5 times greater than the number of prime record FINDs.

You can change the definitions of UFE7 to meet whatever critical conditions you want displayed.

PROCESSOR-ALL
displays data collection statistics for all processors. The statistics displayed

correspond to data collections that were initiated with the PROCESSOR-ALL parameter and ended either with a MODE-ALL parameter or when the duration specified with a TIME parameter has been completed. Omit this parameter to display statistics for the current processor only.

PROCESSOR-ALL is valid only on TPF systems.

SSU-ALL

displays data collection statistics for all subsystem users in the subsystem. Omit this parameter to display statistics for the current SSU only.

Use this parameter only if you specify SSU-ALL on the START command. Similarly, omit the parameter if you omit SSU-ALL on the START command.

SSU-ALL is valid only on TPF systems.

Additional Information

- In a TPF environment, you can control data collection displays on the TPF system by using the ZPAGE command. If variable &DFPAGE in macro DBLCL is set to 1, ZPAGE is necessary to scroll through the data collection display output. If &DFPAGE is set to default 0, the data collection statistics will be displayed without using ZPAGE. The ALCS environment is not affected by the value of &DFPAGE. See *TPFDF Installation and Customization* for more information about the DBLCL macro.
- You can use the ZUDFC HELP command for a complete list of macros that can be used with the CMD and MAC parameters.

Examples

The following example displays TPFDF data collection statistics for all IDs.

```
User:   ZUDFC DISPLAY

System: UDFC0000I 14.05.51 TPFDF DATA COLLECTION
        31JUL00  13.00.38-13.04.06  - AVERAGE PER SECOND  - TOTAL

        DBADD:    95.7 DBADR:     0.0 DBCKP:      -- DBCLS:    31.6
        DBCPY:      -- DBCRE:     1.2 DBDEL:    20.4 DBDIX:      --
        DBDSP:     0.0 DBIDX:     4.0 DBKEY:     0.3 DBMOD:    22.5
        DBMRG:      -- DBOPN:    31.6 DBRED:   186.7 DBREP:     0.0
        DBRET:     2.2 DBRST:      -- DBSPA:     6.0 DBSRT:     0.0
        DBTLD:      -- DBTLG:      -- DBTRD:      -- DBUKY:     0.5
        DDA  :      --


        FILNC:     0.0 FILEC:      -- FILUC:      -- CFILU:     3.9
        PFILU:     3.4 TWRTC:      -- GETFC:     1.3 RELFC:     0.0
        DETAC:    23.0 ATTAC:    23.2 UNFRC:    92.5 GETCC:     4.1
        RELCC:      -- CFIND:    69.5 PFIND:    44.7


        CURRENT DATA COLLECTION COUNTS :

        DBADD:   18954 DBADR:       1 DBCKP:      -- DBCLS:    6266
        DBCPY:      -- DBCRE:     245 DBDEL:    4041 DBDIX:      --
        DBDSP:       6 DBIDX:     811 DBKEY:      74 DBMOD:    4458
        DBMRG:      -- DBOPN:    6266 DBRED:   36970 DBREP:       3
        DBRET:     444 DBRST:      -- DBSPA:    1206 DBSRT:       1
        DBTLD:      -- DBTLG:      -- DBTRD:      -- DBUKY:     104
        DDA  :      --

        FILNC:       7 FILEC:      -- FILUC:      -- CFILU:     773
        PFILU:     687 TWRTC:      -- GETFC:     275 RELFC:       9
        DETAC:    4568 ATTAC:    4599 UNFRC:   18323 GETCC:     831
        RELCC:      -- CFIND:   13776 PFIND:    8865


        *** END OF DISPLAY ***
```

This display provides the following information:

- The average number of times per second any TPFDF macros were issued. See *TPFDF Programming Concepts and Reference* for a description of the TPFDF macros.
- The average number of times per second any TPF or ALCS macro was used. The following shows the TPF or ALCS macros for which you may see statistics:

Display	Description
PFIND	Find a prime file address.
CFIND	Find a chain file address.
FILNC	File a record with no release (FILNC) for a prime or chain file address.
PFILU	File and unhold a record (FILUC) for a prime file address.
CFILU	File and unhold a record (FILUC) for a chain file address.
FILEC	File a record (FILEC) for a prime or chain file address.
FILUC	File and unhold a record (FILUC) for a prime or chain file address.
UNFRC	Unhold a file record (UNFRC).

GETFC	Get a file pool address and storage block (GETFC).
TWRTC	Write a general tape record (TWRTC).
ATTAC	Attach a detached working storage block (ATTAC).
DETAC	Detach a working storage block (DETAC).
GETCC	Get a working storage block (GETCC).
RELCC	Release a core storage block (RELCC).
RELFC	Return a file pool address (RELFC).

- Data collection counts; that is, the total number of times a macro was issued during the data collection period.

The following example displays statistics for all file IDs that averaged more than 25 DBADD macros per second. One such file ID, X'FDA2', is found and displayed.

```
User:   ZUDFC DISPLAY CMD-DBADD VAL-25

System: UDFC0000I 13.04.26 TPFDF DATA COLLECTION

        31JUL00  13.00.38-13.04.06  - AVERAGE PER SECOND  - FDA2

        DBADD:    34.3 DBADR:      -- DBCKP:      -- DBCLS:     1.1
        DBCPY:      -- DBCRE:      -- DBDEL:    10.2 DBDIX:      --
        DBDSP:      -- DBIDX:      -- DBKEY:      -- DBMOD:    18.6
        DBMRG:      -- DBOPN:     1.1 DBRED:    74.7 DBREP:      --
        DBRET:      -- DBRST:      -- DBSPA:      -- DBSRT:      --
        DBTLD:      -- DBTLG:      -- DBTRD:      -- DBUKY:      --
        DDA  :      --

        FILNC:      -- FILEC:      -- FILUC:      -- CFILU:      --
        PFILU:      -- TWRTC:      -- GETFC:     0.0 RELFC:      --
        DETAC:     0.7 ATTAC:     0.7 UNFRC:     1.2 GETCC:     1.2
        RELCC:      -- CFIND:      -- PFIND:      --


        CURRENT DATA COLLECTION COUNTS :

        DBADD:    6798 DBADR:      -- DBCKP:      -- DBCLS:     231
        DBCPY:      -- DBCRE:      -- DBDEL:    2024 DBDIX:      --
        DBDSP:      -- DBIDX:      -- DBKEY:      -- DBMOD:    3701
        DBMRG:      -- DBOPN:     231 DBRED:   14806 DBREP:      --
        DBRET:      -- DBRST:      -- DBSPA:      -- DBSRT:      --
        DBTLD:      -- DBTLG:      -- DBTRD:      -- DBUKY:      --
        DDA  :      --

        FILNC:      -- FILEC:      -- FILUC:      -- CFILU:      --
        PFILU:      -- TWRTC:      -- GETFC:      17 RELFC:      --
        DETAC:     144 ATTAC:     144 UNFRC:     248 GETCC:     248
        RELCC:      -- CFIND:      -- PFIND:      --


        *** END OF DISPLAY ***
```

References

- See *TPFDF Programming Concepts and Reference* for additional information about TPFDF macros.
- See *TPF General Macros* for additional information about TPF macros.
- See *ALCS Application Programming Reference — Assembler Language* for additional information about ALCS macros.

ZUDFC HELP–Help Information

Use this command to display help information for the ZUDFC command.

Requirements and Restrictions

None.

Format

```
►►──ZUDFC ──┬─?────┬──┬─────────┬──────────────────────────────────►◄
            └─HELP─┘  ├─ CMD────┤
                      └─ MAC────┘
```

? displays help information for the ZUDFC commands.

HELP
displays help information for the ZUDFC commands.

CMD
displays a list of all TPFDF, TPF, and ALCS macros that can be searched.

MAC
displays a list of all TPFDF, TPF, and ALCS macros that can be searched.

Additional Information

None.

Examples

The following example displays help information for ZUDFC using the ? parameter.

```
User:   ZUDFC ?

System: UDFC0004I 14.26.10 TPFDF DATA COLLECTION HELP
        -----------------------------------------------------------
        DESCRIPTION:  ZUDFC COLLECTS AND DISPLAYS STATISTICS FOR
                      TPFDF FILES. THE DISPLAY SHOWS THE AVERAGE
                      PER SECOND AND TOTAL COUNTS.
        RESTRICTION:  NONE
        ENTRIES:
         ZUDFC START .....   START DATA COLLECTION
         ZUDFC DISPLAY ..... DISPLAY STATISTICAL INFORMATION
         ZUDFC STOP .....    STOP DATA COLLECTION
        OPTIONAL PARAMETERS ZUDFC START:
           TIME-..            COLLECTION TIME 01-60 MINUTES.
           PROCESSOR-ALL      CURRENT OR ALL PROCESSORS.
           SSU-ALL            CURRENT OR ALL SSUS.
        OPTIONAL PARAMETERS ZUDFC DISPLAY:
           ALL                ALL HISTORY ITEMS - VALID FOR ID OR TOT
           ID-..              SPECIFIED FILE ID
           TOTALS             ALL FILE IDS. DEFAULT VALUE.
           CMD-..... VAL-.... ALL IDS WHERE AVERAGE USAGE OF GIVEN
                              DBCMD IS HIGHER THAN GIVEN VALUE
           MAC-..... VAL-.... SAME FOR GIVEN SVCMAC
           CRIT               ALL IDS WITH CRITICAL FILE ACCESS
                              CMD/MAC/CRIT NOT VALID FOR ID OR TOT
           PROCESSOR-ALL      CURRENT OR ALL PROCESSORS.
           SSU-ALL            CURRENT OR ALL SSUS.
        OPTIONAL PARAMETERS ZUDFC STOP:
           MODE-ALL           ALL PROCESSORS, CURRENT MUST BE RUNNING
           MODE-FORCE         ALL PROCESSORS, CURRENT NOT RUNNING.
        EXAMPLES:
         ZUDFC START TIME-10 PROCESSOR-ALL SSU-ALL
         ZUDFC DISPLAY ID-C005 PROCESSOR-ALL SSU-ALL
         ZUDFC DISPLAY CMD-DBRED VAL-20
         ZUDFC STOP MODE-ALL
         *** END OF DISPLAY ***
```

The following example displays a list of the available macros to be used with the CMD and MAC parameters on the ZUDFC DISPLAY command.

```
User:   ZUDFC HELP CMD

System: UDFC0005I 12.58.57 TPFDF DATA COLLECTION STATISTICS HELP
        -----------------------------------------------------------
        VALID DB-COMMANDS TO BE USED IN CMD-.... PARAMETER

         DBADD   DBADR   DBCKP   DBCLS   DBCPY   DBCRE   DBDEL   DBDIX
         DBDSP   DBIDX   DBKEY   DBMOD   DBMRG   DBOPN   DBRED   DBREP
         DBRET   DBRST   DBSPA   DBSRT   DBTLD   DBTLG   DBTRD   DBUKY

         VALID MACROS TO BE USED IN MAC-.... PARAMETER

         FILNC   FILEC   FILUC   CFILU   PFILU   TWRTC   GETFC   RELFC
         DETAC   ATTAC   UNFRC   GETCC   RELCC   CFIND   PFIND

        *** END OF DISPLAY ***
```

References

- See *TPFDF Programming Concepts and Reference* for additional information about TPFDF macros.
- See *TPF General Macros* for additional information about TPF macros.
- See *ALCS Application Programming Reference — Assembler Language* for additional information about ALCS macros.

ZUDFC START–Start Data Collection

Use this command to start data collection.

Requirements and Restrictions

- If you are in a TPF environment and IPL a processor running data collection, the collection is terminated on that processor and an appropriate message is sent to the receive-only computer room agent set (RO CRAS). Any statistics accumulated during the collection session for that processor are discarded and not applied to the cumulative totals.
- If you are in a TPF environment, you can enter ZUDFC commands from any subsystem user (SSU). The commands can apply to the current subsystem user or all subsystem users in the subsystem.

Format

```
►►──ZUDFC START──┬─────────────┬──┬─────────────────┬──┬───────────┬──────────►◄
                 └─TIME-time───┘  └─PROCESSOR-ALL───┘  └─SSU-ALL───┘
```

TIME-*time*
specifies the duration of the measurement period in minutes, where *time* is a 2-digit number from 1–60. If you do not specify this parameter, data collection continues until you enter a ZUDFC STOP command. The ZUDFC STOP command can be used to terminate data collection before the end of the measurement period.

PROCESSOR-ALL
starts data collection on all processors. If you do not specify this parameter, data collection will start only on the current processor.

PROCESSOR-ALL is valid only on TPF systems.

SSU-ALL
starts data collection for all subsystem users in the subsystem. This parameter produces one set of data for all SSUs. If you do not specify this parameter, data collection will start only on the SSU for the ZUDFC command that was entered.

SSU-ALL is valid only on TPF systems.

Additional Information

When you specify the PROCESSOR-ALL and TIME parameters, if the initiating processor is not active when data collection is going to end, data collection does not stop automatically on any other processor. In this occurs, you must stop data collection on the other processors manually with the ZUDFC STOP command. In addition, the PROCESSOR-ALL totals are lost.

Examples

The following example displays data collection started on all processors.

```
User:   ZUFDC START PROCESSOR-ALL

System: UDFC0001I 13.40.03 DATA COLLECTION STARTED ON PROCESSOR B
```

The following example displays data collection that is started on all processors and will run for 30 minutes.

```
User:   ZUFDC START TIME-30 PROCESSOR-ALL

System: UDFC0001I 13.41.13 DATA COLLECTION STARTED ON PROCESSOR B
```

References

None.

ZUDFC STOP–Stop Data Collection

Use this command to stop data collection.

Requirements and Restrictions

None.

Format

```
►►──ZUDFC STOP──┬────────────────────────┬──────────────────────────►◄
                └─ MODE-──┬─ ALL───┬─┘
                          └─ FORCE─┘
```

MODE-
specifies which mode to use when stopping data collection. Omit the MODE parameter to stop data collection for only the current processor.

ALL
stops data collection on all processors in a loosely coupled complex that are running data collection.

FORCE
specifies data collection to stop on all processors even if the current processor is not running data collection. If you use this parameter, the results of the statistics cannot be predicted.

Additional Information

- If you use the ZUDFC STOP command without any parameters, TPFDF data collection stops for the current processor only.
- The MODE parameter has no affect on the ALCS environment.
- You must stop data collection with MODE-ALL to obtain cumulative totals for all processors. If you stop all processors individually, all cumulative totals are lost.
- You can use the ZUDFC STOP command to stop data collection before the end of the measurement period specified by the ZUDFC START command.

Examples

The following example shows data collection being stopped.

```
User:   ZUDFC STOP

System: UDFC0002I 14.23.54 DATA COLLECTION STOPPED ON PROCESSOR B
```

References

None.

Maintenance and Support

This chapter describes how you can use the TPFDF utility commands for database maintenance and support. This chapter includes:

- Commands used during installation to:
  - Initialize files
  - Initialize and set up a help utility.
- Commands used for database and system maintenance to:
  - Restrict command use
  - Display file information, data, and the contents of the database definition (DBDEF) tables
  - Display a log of modifications made with the ZUDFM command entries
  - Load the macro label set (MLS) tape and display the loading information
  - Add, delete, or replace logical records (LRECs)
  - Copy or pack a subfile
  - Convert a time-of-day (TOD) clock value.

Traditional and Standard ZUDFM Commands

To simplify using the TPFDF utility commands for database maintenance and support, you can use either the traditional or standard form of the ZUDFM commands. See Table 1 for a list of these commands.

Table 1. ZUDFM Comand Table

Traditional Command	**Standard Command**	**Command Description**
ZUDFM OA	ZUDFM FAD	Link to a Subfile Using the File Address
ZUDFM OA*	ZUDFM DISPLAY	Display LRECS in a Subfile
ZUDFM OAA	ZUDFM ADD	Add LRECs to a Subfile
ZUDFM OAB	ZUDFM INFO	Display Block Trailer Information
ZUDFM OAC	ZUDFM COPY	Copy and Restore a Subfile
ZUDFM OAE	ZUDFM RESTRICT	Restrict Table Handling
ZUDFM OAF	ZUDFM FCH	Display Forward and Backward Chaining
ZUDFM OAH	ZUDFM HELP, ZUDFM ?	Help for the ZUDFM OA Commands
ZUDFM OAI	ZUDFM ACCESS	File Access Using the ID or DSECT Name
ZUDFM OAI/DBTAB	ZUDFM DEF	Display Information about DBDEF Table Items
ZUDFM OAI/FILE	ZUDFM FILE	Display Information about DSECTs
ZUDFM OAINIT	ZUDFM INIT	Initialize Files
ZUDFM OAL	ZUDFM LOG	Display Recent ZUDFM OA Commands
ZUDFM OAP	ZUDFM PACK	Pack a Subfile
ZUDFM OAR	ZUDFM REPLACE	Replace User Data in an LREC
ZUDFM OAS	ZUDFM TRACE	Display File Information during Tracing
ZUDFM OAT	ZUDFM MLS	Load Macro Label Set
ZUDFM OAX	ZUDFM DELETE	Delete LRECs from a Subfile
ZUDFM OAZ	ZUDFM TOD	Convert Time-of-Day Clock into Standard Format

ZUDFM OA–Link to a Subfile Using the File Address

Use this command to link to a subfile and display the contents of the subfile when you know the file address of the prime block.

Requirements and Restrictions

On a TPF system, you must enter the ZUDFM OA command in the subsystem user (SSU) that owns the file address.

Format

```
►►──ZUDFM── ──┬─OA/──┬─fileaddr─────────────────────────────────────────►◄
              └─FAD──┘
```

Note: You can use either the ZUDFM OA or the ZUDFM FAD command.

fileaddr
specifies an 8-character hexadecimal file address.

Additional Information

- You can also use this command to link to a traditional TPF file.

Examples

The following example displays the subfile linked to the file address specified. The logical records (LRECs) in the linked subfile are fixed-length LRECs.

```
User:   ZUDFM OA/F402D804

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F402D804
        000001 80 D A N
        000002 80 J O H  N
        000003 80 K E V  I N
        000004 80 L A U  R A
        000005 80 R O B
        000006 80 T O M
```

The following example displays the subfile linked to the file address specified. The LRECs in the linked subfile are variable-length LRECs.

```
User:   ZUDFM OA/000805EC

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805EC
        F.A. 000805EC
        000001 003680 A  0 0 0 4  B 0 0 0  4 C 0 0  0 4 D 0  0 0 4 E
                0 0 0 4  J T H I  S  I S   A  T  E S T   L R E C
                        0004
        000002 003680 A  0 0 0 5  B 0 0 0  5 C 0 0  0 5 D 0  0 0 5 E
                0 0 0 5  J T H I  S  I S   A  T  E S T   L R E C
                        0005
```

References

See the following for more information about linking to a subfile:

- “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83
- “ZUDFM OA*–Display LRECS in a Subfile” on page 64.

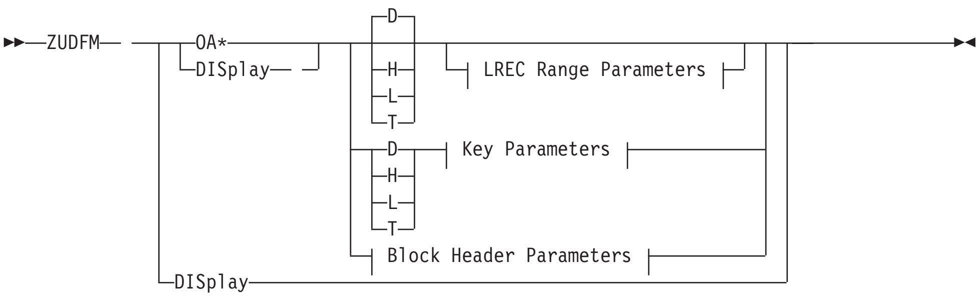
ZUDFM OA*–Display LRECS in a Subfile

Use this command to display the contents of block headers or logical records (LRECs) in various formats using different types of search criteria.

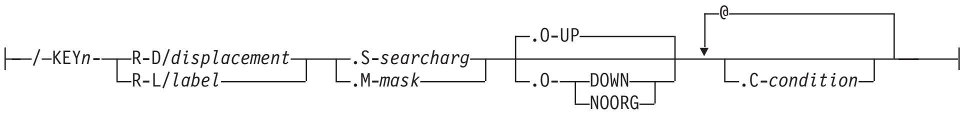
Requirements and Restrictions

Before using this command, you must first link to the file that you want to access.

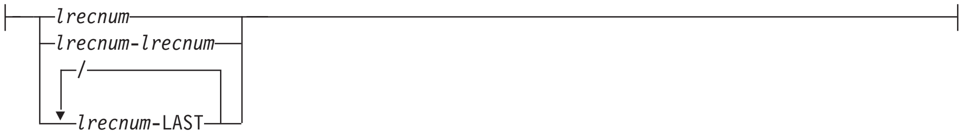
Format



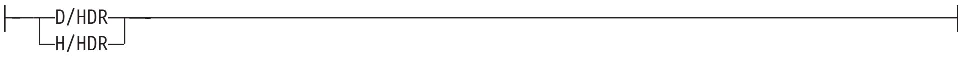
Key Parameters:



LREC Range Parameters:



Block Header Parameters:



Note: You can use either the ZUDFM OA* or the ZUDFM DISPLAY command.

D displays the file address and displacement (in the block) of each LREC and the contents of each LREC in character format.

H displays the file address and displacement (in the block) of each LREC and the contents of each LREC in hexadecimal format.

L displays the contents of each field in each LREC together with their DSECT labels.

Note: When using this parameter, you must build a macro label set (MLS) using MLS support.

T displays technical LRECs (TLRECs).

KEY*n*
specifies a search of LRECs where *n* is the key parameter from 1–6. Separate each set of key parameters from the next by using the @ symbol. Combining keys allows you to make very specific LREC retrievals.

R-D/*displacement*
specifies the displacement of a field in the LREC that is to be used as a key field. You cannot use a displacement of 0 on a variable format LREC.

R-L/*label*
specifies a label, as defined in the DSECT macro, of a field in the LREC that is used as a key field.

Note: When using this parameter, you must build a macro label set (MLS) using MLS support. Do not include the first 4 characters of the label.

.S-*searcharg*
specifies a search key to match against the contents of the key field specified by one of the R subparameters. The search key cannot be longer than 49 bytes.

Note: This variable can contain a mixture of hexadecimal and character data. To switch between these types of data, use the vertical bar (|) as a delimiter. The following example shows how the types of data can be mixed:

```
8013B4|THIS IS CHARACTER DATA|4A5F90
```

The TPFDF product assumes that the variable starts with hexadecimal data. To start with character data, place a vertical bar at the beginning of the variable. For example:

```
|STARTING WITH CHARACTER DATA|1F2E3D4E
```

You can use a different delimiter by setting variable #DBENUFB in the ACPDBE macro. See *TPFDF Installation and Customization* for more information about the ACPDBE macro.

.M-*mask*
specifies a 1-byte hexadecimal mask that is matched against the contents of the key field specified by one of the R subparameters.

.O-*fileorg*
specifies the organization of the subfile. The values for subfile organization are UP, DOWN, and NOORG.

.C-*condition*
specifies the comparison operator between the LREC field and search argument specified. You can use the following values:

Value	**Condition**
GT	Greater than
LT	Less than
NE	Not equal

EQ	Equal (this is the default)
NH	Not high
NL	Not low
GE	Greater than or equal
LE	Less than or equal
Z	Zeros
O	Ones
M	Mixed
NZ	Not zeros
NO	Not ones
NM	Not mixed.

lrecnum
is the sequence number of the LREC. The sequence numbers start at 1.

lrecnum–*lrecnum*
is a range of LRECs in a subfile.

lrecnum–**LAST**
is a range of LRECs from the specified LREC to the last LREC, inclusively, in the subfile.

D/HDR
displays the subfile block header in character format.

H/HDR
displays the subfile block header in hexadecimal format.

Additional Information

- When you use the ZUDFM OA* command without parameters, all the LRECs in the subfile are displayed.
- When selected keys are used, the ZUDFM OA* command may produce a display of LRECs that match the selected keys. The LRECs are numbered sequentially: 0001, 0002, and so on.
- You must specify LREC ranges in ascending order. If one is found out of order that range and all subsequent ranges are ignored.
- You can use the ZUDFM OA command to link to the subfile if you know the file address of the prime block of the subfile. If you only know the DSECT name of the file, or the file ID, you can use the ZUDFM OAI command.

Examples

The following example displays all the LRECs in the subfile. Each line in the display shows the following information:

- The LREC sequence number
- The 2-byte field indicating the LREC length
- The 1-byte primary key
- The data.

```
User:   ZUDFM OA*

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404E801
        000001 001480 E  L M   S  T R E E  T         682186
        000002 001480 M  A I N    S T R E  E T       469089
        000003 001480 P  O S T    R O A D            214968
        000004 001480 C  I R C L  E   L A  N E       326588
```

The following example displays a range of LRECs in character format.

```
User:   ZUDFM OA*D2-3

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404E801
        F.A. F404E801 002E
        000002 001480 M  A I N    S T R E  E T       469089
        F.A. F404E801 0042
        000003 001480 P  O S T    R O A D            214968
```

The following example displays a range of LRECs in hexadecimal format.

```
User:   ZUDFM OA*H1-2

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404E801
        F.A. F404E801 001A
        000001 001480C5 D3D440E2 E3D9C5C5 E3404040 40682186
        F.A. F404E801 002E
        000002 001480D4 C1C9D540 E2E3D9C5 C5E34040 40469089
```

The following example displays the DSECT labels of the LREC fields with the contents of each field for the first LREC.

```
User:   ZUDFM OA*L1

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404F8F4
        F.A. F404F8F4 001A ITMNBR 000001 SIZE 0036  KEY 80
        IR75A80               0003 0019  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
        IR75AK                0003 0005  A 0 0 0 1
        IR75BK                0008 0005  B 0 0 0 1
        IR75CK                000D 0005  C 0 0 0 1
        IR75DK                0012 0005  D 0 0 0 1
        IR75EK                0017 0005  E 0 0 0 1
        IR75AL                001C 0001  J
        IR75TXT               001D 0019  T H I S40 I S40 A40 T E S T40 L
                                         R E C4040404000001
        IR75DAT               001D 0017  T H I S40 I S40 A40 T E S T40 L
                                         R E C40404040
        IR75RCN               0034 0002 0001
```

The following example displays the contents of the header for each prime and overflow block in the subfile in character format.

```
User:   ZUDFM OA*D/HDR

System: UDFM0002I          TPFDF BLOCK HEADER DISPLAY
        F.A.  F404F8F4  BID- B075   CHK- 00    CTL- 00   PGM-  Q X D G
                        FCH- 000806B4   BCH- 000806BC   NAB- 00BC
                        SEQ- 0002   SBA- 00000000   SBC- 00   AUT- 00

        F.A.  000806B4  BID- B075   CHK- 00    CTL- 08   PGM-  Q X D G
                        FCH- 000806B8   BCH- 00000000   NAB- 00BC
                        SEQ- 0002   SBA- 00000000   SBC- 00   AUT- 00

        F.A.  000806B8  BID- B075   CHK- 00    CTL- 08   PGM-  Q X D G
                        FCH- 000806BC   BCH- 00000000   NAB- 00BC
                        SEQ- 0002   SBA- 00000000   SBC- 00   AUT- 00

        F.A.  000806BC  BID- B075   CHK- 00    CTL- 08   PGM-  Q X D G
                        FCH- 00000000   BCH- 00000000   NAB- 0050
                        SEQ- 0002   SBA- 00000000   SBC- 00   AUT- 00
```

BID	File ID
CHK	Record code check
CTL	Control byte
PGM	Last program that updated the block
FCH	Forward chain file address
BCH	Backward chain file address
NAB	Next available byte
SBA	Subfile block address.

Note: For a B^+Tree data file, the SBA of the prime block contains the address of the root node of the B^+Tree index file. If the value is zero, no B^+Tree index exists for this subfile.

The remaining fields are used internally by the TPFDF product.

The following example displays two LRECs that satisfy the search criteria using one key. The display is in hexadecimal format. Byte 12 of each LREC is examined for an equal match with search argument X'E7'.

```
User:   ZUDFM OA*H/KEY1-R-D/12.S-E7.C-EQ

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404E801
        F.A. F404E801 001A
        000001 000E8040 E3C5E2E3 D3D9C5C3 E701
        F.A. F404E801 0044
        000004 000E9040 E3C5E2E3 D3D9C5C3 E701
```

The following example displays two LRECs that satisfy search criteria using two keys. The display is in character format. KEY1 examines byte 2 of each LREC (the primary key) for an equal match with search argument X'80'. KEY2 examines byte 12 of each LREC for a “greater than or equal to” match with search argument character 'Y'.

```
User:   ZUDFM OA*D/KEY1-R-D/2.S-80.C-EQ@KEY2-R-D/12.S-|Y.C-GE

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404E801
        F.A. F404E801 0028
        000002 000E80    T E S T  L R E C  Y02
        F.A. F404E801 0036
        000003 000E80    T E S T  L R E C  Z03
```

References

- See the following for more information about linking to a subfile:
  - “ZUDFM OA–Link to a Subfile Using the File Address” on page 62
  - “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83.
- See “ZUDFM OAT–Load Macro Label Set” on page 117 for more information about MLS.

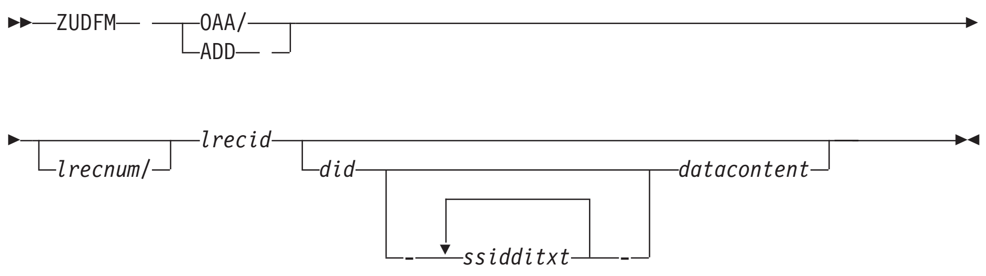
ZUDFM OAA–Add LRECs to a Subfile

Use this command to add logical records (LRECs) to the subfile to which you are currently linked. You can add LRECs to the end of the subfile or at a specific position in the subfile.

Requirements and Restrictions

- Before using this command, you must first link to and display the file that you want to access.
- You cannot use this command with a sequence number when the file has default keys defined.
- You cannot use this command to add technical LRECs (TLRECs).

Format



Note: You can use either the ZUDFM OAA or the ZUDFM ADD command.

lrecnum
is the sequence number that will be used for the added LREC. This parameter is a decimal number between 1 and 999999.

lrecid
is the 1-byte primary key of the LREC.

did
is the 2-byte userLREC data identifier.

ssi is the 2-byte size of the subLREC.

ddi
is the data definition that identifies the type of 2-byte subLREC.

txt is the data content of the subLREC.

datacontent
is EBCDIC data, or hexadecimal data, or both, to define the contents of the userLREC data field.

Note: This variable can contain a mixture of hexadecimal and character data. To switch between these types of data, use the vertical bar (|) as a delimiter. The following example shows how the types of data can be mixed:

```
8013B4|THIS IS CHARACTER DATA|4A5F90
```

The TPFDF product assumes that the variable starts with hexadecimal data. To start with character data, place a vertical bar at the beginning of the variable. For example:

```
|STARTING WITH CHARACTER DATA|1F2E3D4E
```

You can use a different delimiter by setting variable #DBENUFB in the ACPDBE macro. See *TPFDF Installation and Customization* for more information about the ACPDBE macro.

Additional Information

- If there are no keys defined in the database definition (DBDEF) table, and if you do not specify the positional LREC number parameters, the TPFDF product adds the new LREC at the end of the subfile. Similar to the TPFDF DBADD macro, the ZUDFM OAA command uses designated keys to identify the location where an LREC will be added.
- You must to specify a sequence number with a file that uses algorithm #TPFDB0D.
- When the ZUDFM OAA command adds an LREC, the LREC size is dynamically calculated and inserted in the 2-byte field.
- You can ensure that every new LREC is added in the correct position by having the default keys and file organization defined in the DBDEF by the database administrator.
- When you use a ZUDFM OA command to display a subfile that you are going to modify (ZUDFM OAA, ZUDFM OAR, or ZUDFM OAX) there is a time interval during which another entry could modify the same subfile. Consequently, your modification could inadvertently change the wrong LREC in the subfile and damage the database.

  To overcome this problem, the TPFDF product maintains a sequence counter that is used by ZUDFM to ensure file integrity. The sequence counter is updated in the block header whenever the subfile is changed. In subsequent entries, the retained value is compared with the actual value to ensure that the database is updated in the correct sequence.

  Heavily modified databases may become difficult to maintain with ZUDFM entries if the sequence count is constantly being changed. In this case, write your own utilities to locate the file, hold it, and change it.

  Note: If you enter multiple ZUDFM OA commands that modify a subfile, the sequence counter is not checked. In other words, you are allowed to modify the subfile more than once without having to display it before each ZUDFM OA command. You will only receive a sequence error if another user or application program modifies the subfile while you are entering the ZUDFM OA commands.

Examples

The following example shows the specified LREC added to the subfile. The LREC is inserted into the appropriate position of the subfile because the file has default keys defined. The standard TPFDF LREC DISPLAY heading is displayed after the LREC has been added to the file.

```
User:   ZUDFM OAA/80|SMITH|0000000000123456

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404F868
        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 D  U N N00 00000000 00328065
        000003 001080 I  N O U Y  E000000 00690794
        000004 001080 J  A C O B  S000000 00669412
        000005 001080 J  O N E S 00000000 00789441
        000006 001080 S  M I T H 00000000 00123456
        000007 001080 T  A Y L O  R000000 00123964
        000008 001080 T  H A Y E  R000000 00236348
        000009 001080 U  N D E R  H I L L 00894350
        OAA/80|SMITH|0000000000123456
```

The following example adds an extended LREC, where the primary key is X'C0' and the DID is X'2010', as the second LREC in the subfile that is currently linked.

```
User:   ZUDFM OAA2/C02010|LONDON, ENGLAND

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. A076908B
        000001 000EC000 00060008 2000 M A  D R I D  ,   S P  A I N
        000002 0018C000 00060012 2010 L O  N D O N  , E N G  L A N D
        000003 001FC000 00060019 2080 P A  R I S ,    F R A  N C E
                T A   P  A R T
        ZUDFM OAA 2/C02010|LONDON, ENGLAND
```

The following example adds an extended LREC with two subLRECs to the currently linked file, which already contains one extended LREC with no subLRECs.

```
User:   ZUDFM OAA/C02222-0006C1C1F1F10007C2C2F2F2F2-|USERLREC

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 00980013         000001 0012C000 0006000C
        1111 U S  E R L R  E C         000002 001FC000 00130006
        A A 1 1 0007 B B  2 2 200 0C2222 U
        S E R L  R E C           ZUDFM ADD /C022220006C1C1F1F10007C2C2F2F2F2!USERLREC
```

The previous example shows information for LREC 1 and LREC 2.

- Information for LREC 1 includes:

0012	The size of the entire LREC (SIZ)
C0	The primary key (PKY)
00	The reserved byte
0006	The offset to the userLREC (OFS)
000C	The size of the userLREC (SZE)
1111	The DID of the userLREC (DID)
U S E R L R E C	The data of the userLREC (data content).

- Information for LREC 2 includes:

001F	The size of the entire LREC (SIZ)
C0	The primary key (PKY)
00	The reserved byte
0013	The offset to the userLREC (OFS)

0006	The size of the subLREC 1 (SSI)
A A	The DDI of the subLREC 1 (DDI)
1 1	The data of the subLREC 1 (TXT)
0007	The size of the subLREC 2 (SSI)
B B	The DDI of the subLREC 2 (DDI)
2 2 2	The data of the subLREC 2 (TXT)
000C	The size of the userLREC (SZE)
2222	The DID of the userLREC (DID)
U S E R L R E C	The data of the userLREC (data content).

References

- See the following for more information about linking to a subfile:
  - “ZUDFM OA–Link to a Subfile Using the File Address” on page 62
  - “ZUDFM OA*–Display LRECS in a Subfile” on page 64
  - “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83.
- For more information about the DBADD macro, see *TPFDF Programming Concepts and Reference*.

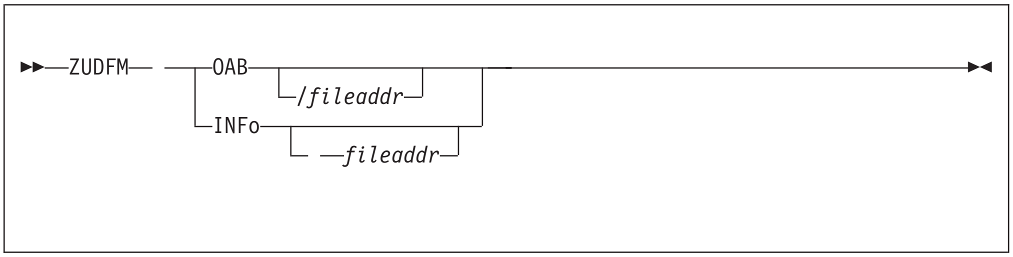
ZUDFM OAB–Display Block Trailer Information

Use this command to display the block trailer information of a single data block.

Requirements and Restrictions

The block trailer information is not available if the &ACPDBFS variable is set to 0 in the DBLCL macro. See *TPFDF Installation and Customization* for more information about the DBLCL macro.

Format



Note: You can use either the ZUDFM OAB or the ZUDFM INFO command.

fileaddr
specifies the 8-character hexadecimal file address of the block trailer to be displayed.

Additional Information

If you enter this command without a parameter, information is displayed from the file address that is currently linked.

Examples

The following example displays the block trailer information of the file that is currently linked.

```
User:   ZUDFM INFO

System: UDFM9999I          TPFDF BLOCK TRAILER INFO F.A. 4404D001
        FILE ID            FDFD
        PRIME ADDRESS      4404D001
        OWN ADDRESS        4404D001
        PROGRAM BASE       FF00  BSS0
        DATA BASE          FF00  BSS0
        SUBSYSTEM USER     FF00  BSS0
        LAST COMMAND       DBCLS
        FILING PROGRAM     BRC1
        FILING CPU ID      C
        FILING I-S         1
        FILING TIME        SU 12SEP1999 23.33.48
```

References

See “ZUDFM OA*–Display LRECS in a Subfile” on page 64 for more information about linking to a subfile.

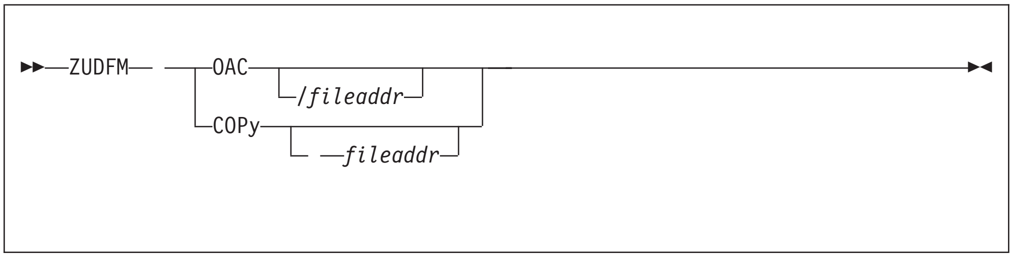
ZUDFM OAC–Copy and Restore a Subfile

Use this command to copy a subfile to a new pool file address or to restore the contents of a subfile to a specific file address.

Requirements and Restrictions

- Before using this command, you must first link to the file that you intend to copy.
- As a precaution against the accidental overwriting of another subfile, you must enter the ZUDFM OAC command twice.

Format



Note: You can use either the ZUDFM OAC or the ZUDFM COPY command.

fileaddr
is an 8-character hexadecimal target file address to which the currently linked source subfile will be copied.

Additional Information

- To restore a copied subfile to the original file address, specify the address of the original subfile with the ZUDFM OAC command. This will copy the modified subfile over the original version.
- If you enter the ZUDFM OAC command without a parameter, the subfile to which you are currently linked will be copied to a new pool file address.
- The TPFDF product leaves you linked to the copy (target) of the subfile.
- When a B$^+$Tree file is copied, ZUDFM OAC builds the B$^+$Tree index for the new file.

Examples

The following examples restore a subfile to the specified address. The current contents of the address are displayed after the command is entered for the first time. The contents are overwritten when the command is entered a second time for confirmation.

```
User:   ZUDFM OAC/000805B0

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805B0
        UDFM0183A          ** ATTENTION ** CONTENT OF COPY-TO-ADDRESS

        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 D  U N N00 00000000 00328065
        000003 001080 I  N O U Y  E000000 00690794
        000004 001080 J  A C O B  S000000 00669412
        000005 001080 J  O N E S 00000000 00789441
        000006 001080 T  A Y L O  R000000 00123964
        000007 001080 T  H A Y E  R000000 00236348
        000008 001080 U  N D E R  H I L L 00894350
```

```
User:   ZUDFM OAC/000805B0

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805B0
        000001 001080 B  R O C K  E R     00345874
        000002 001080 D  U N N            00328065
        000003 001080 I  N O U Y  E       00690794
        000004 001080 J  A C O B  S       00669412
        000005 001080 J  O N E S          00789441
        000006 001080 T  A Y L O  R       00123964
        000007 001080 T  H A Y E  R       00236348
        000008 001080 U  N D E R  H I L L 00894350
```

References

See the following for more information about linking to a subfile:

- “ZUDFM OA–Link to a Subfile Using the File Address” on page 62
- “ZUDFM OA*–Display LRECS in a Subfile” on page 64
- “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83.

ZUDFM OAE–Restrict Table Handling

Use this command to inspect and edit the table of restricted commands. The entry restriction table is used to specify the terminals from which certain ZUDFM commands can be entered.

Requirements and Restrictions

You ***cannot*** use this command to restrict the INIT parameter of the ZUDFM OAI/DBTAB command because the entry restriction table is a TPFDF database. If the DBDEF table is corrupted and the INIT parameter is restricted, any attempt to use the INIT parameter to rebuild the DBDEF table would fail because the entry restriction table would not be able accessible. See “ZUDFM OAI/DBTAB–Display DBDEF Table Items” on page 87 for more information about the ZUDFM OAI/DBTAB command and INIT parameter.

Format

```
►►──ZUDFM── ──┬─OAE──────┬──┬─┬─A/─┬──terminal|command─┬──────────────►◄
              └─REStrict─┘  │ └─X/─┘                   │
                            └─*────────────────────────┘
```

Note: You can use either the ZUDFM OAE or the ZUDFM RESTRICT command.

A specifies additional entries to the restricted command table.

X specifies deleted entries from the restricted command table.

terminal
is a 6-digit hexadecimal terminal address to be added to or deleted from the restricted command table.

command
is the ZUDFM command to be restricted. You do not enter the ZUDFM portion of the command when using this parameter.

Note: Variable #DBENUFB in the ACPDBE macro sets the vertical bar (|) delimiter. You can use a different delimiter by setting variable #DBENUFB in ACPDBE. See *TPFDF Installation and Customization* for more information about the ACPDBE macro.

***** displays the ZUDFM restricted command table.

Additional Information

- A ZUDFM command is unrestricted if it is not listed in the entry restriction table.
- A ZUDFM command that is listed in the entry restriction table can be entered only from the terminals specified.

Examples

The following example displays which ZUDFM commands are restricted to specific terminal addresses.

```
User:   ZUDFM OAE*

System: UDFM0201I          OA - ENTRY RESTRICTION TABLE DISPLAY
        -------------------------------------------------------

        NBR    RESTRICTED ENTRY                       TERMINAL

        0001   OAA.................................... 8406BA
        0002   OAC.................................... 8406BA
        0003   OAR.................................... 8406BA
        0004   OAX.................................... 8406BA
```

The following example displays the addition of the specified terminal address and ZUDFM command to the restriction table.

```
User:   ZUDFM OAEA/8406BA|OAS

System: UDFM0201I          OA - ENTRY RESTRICTION TABLE DISPLAY
        -------------------------------------------------------

        NBR    RESTRICTED ENTRY                       TERMINAL

        0001   OAA.................................... 8406BA
        0002   OAC.................................... 8406BA
        0003   OAR.................................... 8406BA
        0004   OAS.................................... 8406BA
        0005   OAX.................................... 8406BA
```

The following example displays the deletion of the specified terminal address and ZUDFM command from the restriction table.

```
User:   ZUDFM OAEX/8406BA|OAX

System: DFM0201I          OA - ENTRY RESTRICTION TABLE DISPLAY
        -------------------------------------------------------

        NBR    RESTRICTED ENTRY                       TERMINAL

        0001   OAA.................................... 8406BA
        0002   OAC.................................... 8406BA
        0003   OAR.................................... 8406BA
        0004   OAS.................................... 8406BA
```

References

None.

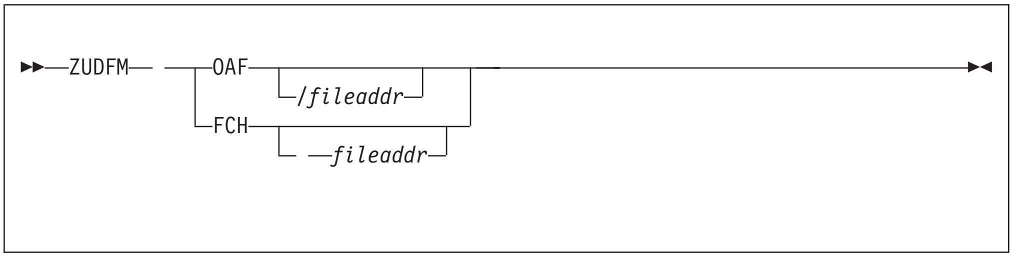
ZUDFM OAF–Display Forward and Backward Chaining

Use this command to display the forward and backward chaining information of a subfile or an 8-character (4-byte) hexadecimal file address.

Requirements and Restrictions

Output from the ZUDFM OAF command stops after 500 chains have been displayed.

Format



Note: You can use either the ZUDFM OAF or the ZUDFM FCH command.

fileaddr
specifies an 8-character hexadecimal file address.

Additional Information

- If you enter the ZUDFM OAF command without a parameter, chaining is displayed beginning with the file address that is currently linked.
- The ZUDFM OAF command displays chaining information beginning with the block in which the file address is linked or specified, and continues for all other blocks along the forward chain.
- If you use the ZUDFM OAF command with a specified file address, that file address becomes the currently linked file.
- Block trailer and command information will not be displayed under the following conditions:
  - The &ACPDBFS variable is set to 0 in the DBLCL macro. See *TPFDF Installation and Customization* for more information about the DBLCL macro.
  - The block trailer size of the file is less than 36 bytes. See *TPFDF Database Administration* for more information about how to set the trailer size.
  - Nodes are being displayed in a B$^+$Tree index file.

Examples

This following example displays the forward chaining, backward chaining, or both, of the currently linked file.

```
User:   ZUDFM OAF

System: UDFM0221I          OAF  F.A. F402D808  ID FDEC  RCC 00

        NBR  FILEADDR PGM  FWDCHAIN BCKCHAIN BLOCK TRAILER     COMMAND
        ---------------------------------------------------------------
        000  F402D808 QT19 000C05A9 000C05A5 25JUN97 11.40.42  CLOSE
        001  000C05A9 QT19 000C05A5          25JUN97 11.39.52  ADD
        002  000C05A5 QT19 00000000          25JUN97 11.40.42  CLOSE
```

The following describes the information provided in the previous example:

FA	The file address of the block.
ID	The file ID of the block.
RCC	A 1-byte field that contains the record code check (RCC). This is a 1-byte value that is common to all the blocks that are being pointed to in the subfile.
NBR	The relative number of the block in the chain.
FILEADDR	The file address of the block.
PGM	The program stamp of the block, which is held in the STDPGM field.
FWDCHAIN	The file address of the block to which this block is forward chained.
BCKCHAIN	The file address of the block to which this block is backward chained. (The BCKCHAIN field in the prime block points to the address of the last block in the chain.) **Note:** The backward chain is not displayed if it is zero. This indicates that the file does not use full backward chaining.
BLOCK TRAILER	The date and time in which the block was written to DASD.
COMMAND	Indicates the TPFDF macro for the function that was used to write the block. In this example, CLOSE indicates the DBCLS macro or `dfcls` functions.

The following example displays the forward chaining, backward chaining, or both, of a specific file address.

```
User:   ZUDFM OAF/000C05A9

System: UDFM0221I          OAF  F.A. 000C05A9  ID FDEC  RCC 00

        NBR  FILEADDR PGM  FWDCHAIN BCKCHAIN BLOCK TRAILER     COMMAND
        ---------------------------------------------------------------
        000  000C05A9 QT19 000C05A5          25JUN97 11.39.52  ADD
        001  000C05A5 QT19 00000000          25JUN97 11.40.42  CLOSE
```

References

See the following for more information about linking to a subfile:

- “ZUDFM OA–Link to a Subfile Using the File Address” on page 62
- “ZUDFM OA*–Display LRECS in a Subfile” on page 64
- “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83.

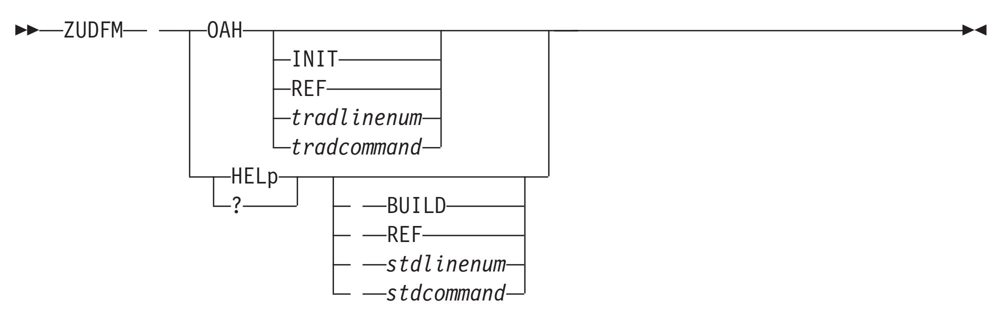
ZUDFM OAH–Help for ZUDFM OA Commands

Use this command to get help information about specified ZUDFM commands.

Requirements and Restrictions

Initialize the help database with the INIT parameter before using this command.

Format



Note: You can use either the ZUDFM OAH, the ZUDFM HELP, or the ZUDFM ? command.

INIT
initalizes the help table. You must initialize the help table before using the ZUDFM OAH command.

REF
displays the segments used to build the help menu.

tradlinenum
is the number of the traditional ZUDFM command that you want to display information about. See Table 1 on page 61 for a list of the traditional and standard commands.

tradcommand
is the traditional ZUDFM command, by name, that you want to display information about. See Table 1 on page 61 for a list of the traditional and standard commands.

BUILD
initalizes the help table. You must initialize the help table before you use the ZUDFM HELP or ZUDFM ? command.

stdlinenum
is the number of the standard ZUDFM command that you want to display information about. See Table 1 on page 61 for a list of the traditional and standard commands.

stdcommand
is the standard ZUDFM command, by name, that you want to display information about. See Table 1 on page 61 for a list of the traditional and standard commands.

Additional Information

- If you use ZUDFM OAH without any parameters, all the ZUDFM commands that are available on your system are displayed.
- You may not be authorized to use every command on which you can receive help information. Command authorizations are determined by a restriction table edited with the ZUDFM OAE command.

Examples

The following example displays all the ZUDFM OA commands that are available on your system. This display is the main index to the help table.

```
User:   ZUDFM OAH

System: UDFM0241I          SYSTEM HELP UTILITY -- OA
        ------------------------------------------------

        0001   OA*            TPFDB FILE DISPLAY
        0002   OA/            BASIC INITIALIZATION
        0003   OAA            ADD LREC TO FILE
        0004   OAC            COPY/RESTORE FUNCTION
        0005   OAE            OA ENTRY RESTRICTION HANDLING
        0006   OAF            FILE CHAINING DISPLAY
        0007   OAH            HELP UTILITY FOR OA-PACKAGE
        0008   OAI            INITIALIZATION OF OA-WORKFILE
        0009   OAI/DBTAB      CENTRAL DESCRIPTOR MAINTENANCE
        0010   OAI/FILE       DSECT LABEL DISPLAY
        0011   OAINIT         INITIALIZE TRADIT AND TPFDB FILES
        0012   OAL            OA MODIFICATION LOG
        0013   OAP            COMPRESS CURRENT FILE
        0014   OAR            REPLACE EXISTING LREC
        0015   OAS            DISPLAY PARAM-/DATA-BLOCK IN SSTR
        0016   OAT            READ AND STORE MACRO LABEL SET
        0017   OAX            DELETE LREC
        0018   OAZ            DISPLAY TOD-CLOCK CONVERSION
```

The following example displays which modules were used to create the tables once the help database is initialized.

```
User:   ZUDFM OAHREF

System: UDFM0242I          HELP UTILITY - PROGRAM REFERENCES
        ----------------------------------------------------

        0001   OA*            UFCB
        0002   OA/            UFCE
        0003   OAA            UFCA
        0004   OAC            UFCG
        0005   OAE            UFCG
        0006   OAF            UFCA
        0007   OAH            UFCD
        0008   OAI            UFCC
        0009   OAI/DBTAB      UFCB
        0010   OAI/FILE       UFCD
        0011   OAINIT         UFCF
        0012   OAL            UFCA
        0013   OAP            UFCA
        0014   OAR            UFCA
        0015   OAS            UFCG
        0016   OAT            UFCE
        0017   OAX            UFCE
        0018   OAZ            UFCD
```

The following example displays information about the specified line number.

ZUDFM OAH

```
User:   ZUDFM OAH11

System: UDFM0243I          TPFDF HELP FACILITY
        O A - H E L P  -  OAINIT      INITIALIZE TRADIT AND TPFDB FILES
        ---------------------------------------------------------------

        DESCRIPTION:      INITIALIZATION OF TRADITIONAL AND TPFDB
                          FILES
        RESTRICTION:      NOT ALLOWED DURING RECOUP OR ZCREC FILE UTI-
                          LITY AND RESTRICTED ONLINE VIA OA-ENTRY TABLE
        REFERENCE:        SEE UFB DOCUMENTATION
        ENTRIES:
        OAINIT ID/..      INITIALIZE SPECIFIC ID
                          AND SPECIFIC FVN WITH ID/.. FVN/..
        OAINIT ID/..!BOR - EOR
                          INITIALIZE SPECIFIC ID FROM BOR TO EOR
                          AND SPEC FVN WITH ID/.. FVN/..
        OAINIT ID/..!BOR - LAST
                          INITIALIZE SPECIFIC ID FROM BOR TO THE LAST
                          ORDINAL AND SPECIFIC FVN WITH ID/.. FVN/..
```

The following example displays help information about the specified ZUDFM command. Depending on the command, you may get several help panels to browse.

```
User:   ZUDFM OAHOAR

System: UDFM0243I          TPFDF HELP FACILITY
        O A - H E L P  -  OAR         REPLACE EXISTING LOGICAL RECORD
        -------------------------------------------------------------

        DESCRIPTION:      REPLACE LREC WITH A NEW ITEM SPECIFIED IN
                          INPUT MESSAGE. THE SIZE OF THE NEW LREC IS
                          AUTOMATICALLY ADJUSTED AND DOES NOT HAVE TO
                          BE SPECIFIED IN THE INPUT. A PREVIOUS OA*
                          DISPLAY IS REQUIRED FOR SEQ-DISCREPANCY-
                          CHECK. THE DELIMITER -!- IS USED TO SWITCH
                          BETWEEN HEX AND CHAR INPUT.
        RESTRICTION:      NONE
        REFERENCE:        SEE UFB DOCUMENTATION

        ENTRIES:
        OARN/XX!CCC!XX!C  REPLACE LREC NBR -N- BY SPECIFIED
                          ITEM. X-HEX/C-CHAR INPUT. N-DEC NBR 1-9999.
        OARN/LBL-.../XX!C!XX/LBL-
                          REPLACE 1 OR MORE FIELDS WITH LABELNAME ...
                          IN EXISTING LREC N WITH GIVEN HEX/CHAR VALUE.
```

References

See “ZUDFM OAE–Restrict Table Handling” on page 76 for more information about the restriction table.

ZUDFM OAI–File Access Using the ID or DSECT Name

Use this command to link to a subfile when you know the file ID or the DSECT name of the file. This command also displays information about the file to which you are linking.

Requirements and Restrictions

None.

Format

```
►►──ZUDFM── ──┬─OAI/────┬─filespec──────────────────────────────────────►
              └─ACCess──┘

►─┬─┬──────────────┬─┬──────────────────┬─┬───────────────────────────►◄
  │ ├─/ALG-algarg──┤ ├─/PARTITNpartn────┤ │
  │ ├─/ORDCordnum──┤ ├─/PTNpartn────────┤ │
  │ └─/ORDXordnum──┘ ├─/INTERLVinterlv──┤ │
  │                  └─/ILVinterlv──────┘ │
  │          ┌─0────────────┐             │
  ├─/CREATE──┴─/PATHpathnum─┴─┬───────────────┬┤
  │                           └─/ALG-algarg───┘│
  └─/RELEASE/─ALG-algarg───────────────────────┘
```

Note: You can use either the ZUDFM OAI or the ZUDFM ACCESS command.

filespec
is the file specification, which can be a 2-character or 4-hexadecimal file ID, or the DSECT name (up to 6 characters).

ALG-
specifies an argument that is used to determine the subfile to be linked.

algarg
is the algorithm argument, which is entered according to the syntax required by the algorithm for this file structure. It uses the algorithm defined in the file DSECT macro definition (in global symbol &SW00RBV). The length of the algorithm argument is taken from the database definition (DBDEF). See *TPFDF Database Administration* for more information about algorithm arguments.

Note: This variable can contain a mixture of hexadecimal and character data. To switch between these types of data, use the vertical bar (|) as a delimiter. The following example shows how the types of data can be mixed:

```
8013B4|THIS IS CHARACTER DATA|4A5F90
```

The TPFDF product that assumes the variable starts with hexadecimal data. To start with character data, place a vertical bar at the beginning of the variable. For example:

```
|STARTING WITH CHARACTER DATA|1F2E3D4E
```

You can use a different delimiter by setting variable #DBENUFB in the ACPDBE macro. See *TPFDF Installation and Customization* for more information about the ACPDBE macro.

ORDC
specifies the ordinal number in decimal format.

ORDX
specifies the ordinal number in hexadecimal format.

ordnum
is the relative record ordinal number of the prime block of the subfile to which you want to link between the beginning ordinal (BOR) and end ordinal (EOR), inclusively. It will be mapped to the appropriate FACE ordinal based on the partition and interleave entered.

PARTITN
specifies a partition (a file where the prime blocks are grouped together into different partitions).

PTN
an abbreviation of the PARTITN parameter that can be used interchangeably with it.

partn
is a partition number that starts at 0. The TPFDF product locates the prime block, and any pool blocks chained to it, at the specified position and links to them.

INTERLV
specifies an interleave (a file where the prime blocks are grouped together into different groups).

ILV
an abbreviation of the INTERLV parameter that can be used interchangeably with it.

interlv
is an interleave number that starts at 0. The TPFDF product locates the prime block, and any pool blocks chained to it, at the specified position and links to them.

CREATE
is used to create an empty pool file, which is not indexed, with the specified DSECT name or file ID of the subfile. You can also create a new indexed subfile by using a CREATE and ALG- parameter together.

PATH
specifies the path value of the index that you specify with the ALG- parameter. The path value is an integer. If you omit it, the command assumes a path of zero.

pathnum
is a numerical path value as defined in the database definition (DBDEF) table.

RELEASE
removes all index LRECs from all index levels. The pool file is not returned to the TPF or ALCS system using the TPF or ALCS RELFC macro.

Additional Information

- The file ID or DSECT only identifies a file (not a particular subfile in the file). ZUDFM OAI links the first subfile in the file unless you use optional parameters to identify a particular subfile.
- You can use the PATH and CREATE parameters to create a new index entry for a pool subfile that already exists. These parameters will also create a new index entry for a pool subfile that is already indexed from another index file, or from several index files.
- You can use the PATH parameter with the RELEASE parameter to restrict the release action to specific paths. Only the index LRECs in the index file that are required for this path are deleted. With a single-level index, the action to one index file is restricted.
- The PARTITN parameter is ignored when you specify it with the ALG parameter.
- If you specify the CREATE parameter and ALG parameter together, the ALG parameter is ignored unless the file uses algorithm #TPFDBFF.

Examples

The following shows an example of linking to a file using the specified file ID.

```
User:   ZUDFM OAI/FDE9

System: UDFM0301I          FILE INFORMATION DISPLAY
        FILE IDENTIFIER               FDE9
        FILE ADDRESS                            X F402E004
        FACE RECORD TYPE            D 00000190  X 000000BE
        FACE START ORDINAL          D 00000001  X 00000001
        FACE END ORDINAL            D 00000001  X 00000001
        CURRENT FACE ORDINAL        D 00000001  X 00000001
        CURRENT RELATIVE ORDINAL    D 00000000  X 00000000
        TPFDF ALGORITHM             D 00000004  X 00000004
        INITIAL NAB VALUE           D 00000026  X 0000001A
        TPFDF END ORDINAL           D 00000001  X 00000001
        VARIABLE LOGICAL RECORDSIZE
```

The following shows an example of linking to a file using the specified DSECT name.

```
User:   ZUDFM OAI/IR20DF

System: UDFM0301I          FILE INFORMATION DISPLAY
        FILE IDENTIFIER               FDE9
        FILE ADDRESS                            X F402E004
        FACE RECORD TYPE            D 00000190  X 000000BE
        FACE START ORDINAL          D 00000001  X 00000001
        FACE END ORDINAL            D 00000001  X 00000001
        CURRENT FACE ORDINAL        D 00000001  X 00000001
        CURRENT RELATIVE ORDINAL    D 00000000  X 00000000
        TPFDF ALGORITHM             D 00000004  X 00000004
        INITIAL NAB VALUE           D 00000026  X 0000001A
        TPFDF END ORDINAL           D 00000001  X 00000001
        VARIABLE LOGICAL RECORDSIZE
```

The following shows an example of linking to a file using the ORDC and PTN2 parameters.

```
User:   ZUDFM OAI/FDE5/ORDC03/PTN2

System: UDFM0301I          FILE INFORMATION DISPLAY
        FILE IDENTIFIER               FDE5
        FILE ADDRESS                            X F40179C5
        FACE RECORD TYPE            D 00000165  X 000000A5
        FACE START ORDINAL          D 00000000  X 00000000
        FACE END ORDINAL            D 00000498  X 000001F2
        CURRENT FACE ORDINAL        D 00000113  X 00000071
        CURRENT RELATIVE ORDINAL    D 00000003  X 00000003
        CURRENT PARTITION           D 00000002  X 00000002
        TPFDF ALGORITHM             D 00000015  X 0000000F
        INITIAL NAB VALUE           D 00000026  X 0000001A
        TPFDF END ORDINAL           D 00000054  X 00000036
        VARIABLE LOGICAL RECORDSIZE
```

The following example displays a new pool file address assigned to the specified file ID using the specified algorithm.

```
User:   ZUDFM OAI/FD01/CREATE/ALG-42

System: UDFM0301I          FILE INFORMATION DISPLAY
        FILE IDENTIFIER               FD01
        FILE ADDRESS                            X 00142909
        FACE RECORD TYPE            D 00000000  X 00000000
        FACE START ORDINAL          D 00000000  X 00000000
        FACE END ORDINAL            D 00000000  X 00000000
        CURRENT FACE ORDINAL        D 00000000  X 00000000
        CURRENT RELATIVE ORDINAL    D 00000000  X 00000000
        TPFDF ALGORITHM             D 00000255  X 000000FF
        INITIAL NAB VALUE           D 00000026  X 0000001A
        TPFDF END ORDINAL           D 00000000  X 00000000
        VARIABLE LOGICAL RECORDSIZE
```

The following example displays the released pool file address associated with the specified algorithm and file ID.

```
User:   ZUDFM OAI/FD01/RELEASE/ALG-42

System: UDFM0301I          FILE INFORMATION DISPLAY
        FILE IDENTIFIER               FD01
        FILE ADDRESS                            X 00142909
        FACE RECORD TYPE            D 00000000  X 00000000
        FACE START ORDINAL          D 00000000  X 00000000
        FACE END ORDINAL            D 00000000  X 00000000
        CURRENT FACE ORDINAL        D 00000000  X 00000000
        CURRENT RELATIVE ORDINAL    D 00000000  X 00000000
        TPFDF ALGORITHM             D 00000255  X 000000FF
        INITIAL NAB VALUE           D 00000026  X 0000001A
        TPFDF END ORDINAL           D 00000000  X 00000000
        VARIABLE LOGICAL RECORDSIZE
```

References

None.

ZUDFM OAI/DBTAB–Display DBDEF Table Items

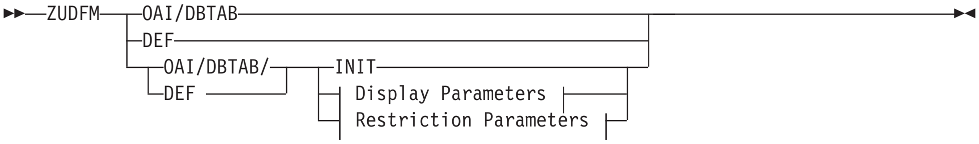
Use this command to maintain and display information from various TPFDF tables.

The DBTAB parameters have three main purposes:
- Rebuild the DBDEF tables
- Display the contents of the DBDEF table
- Maintain the ID restriction table.

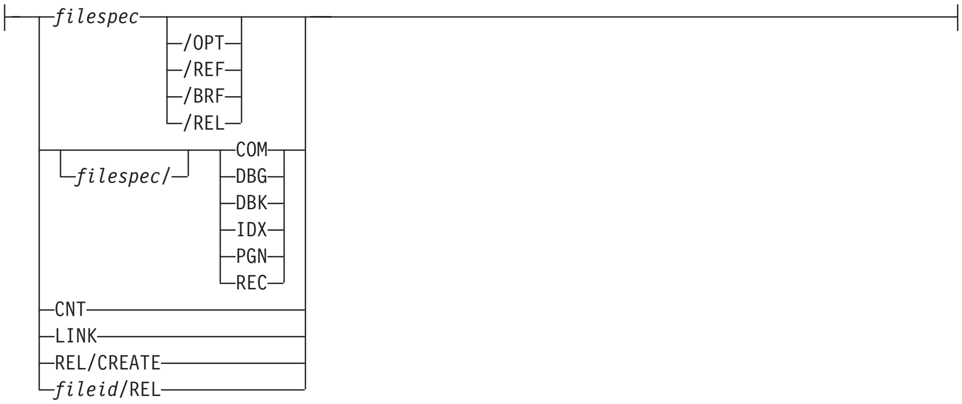
Requirements and Restrictions

None.

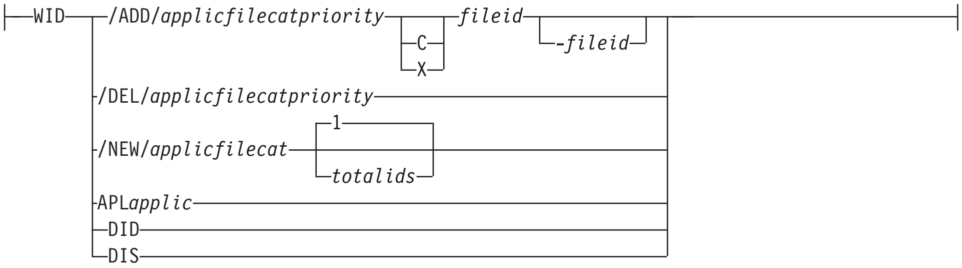
Format



Display Parameters:



Restriction Parameters:



Note: You can use either the ZUDFM OAI/DBTAB or the ZUDFM DEF command.

INIT
rebuilds the DBDEF table. Do this after loading a new DBDEF, or if a DBDEF becomes corrupted.

Note: The INIT parameter ***cannot*** be restricted. See “ZUDFM OAE–Restrict Table Handling” on page 76 for more information about the restriction of commands and parameters.

filespec
is the file specification, which can be a 2-character or 4-hexadecimal file ID, or the DSECT name (up to 6 characters).

OPT
displays the option bits in the DBDEF. The display indicates which, if any, of the bits are set.

REF
displays the file IDs that are forward referenced from a file used in an index structure. The REF parameter displays as many as 6 levels of indexing.

BRF
displays the file IDs that are backward referenced from a file used in an index structure. The BRF parameter displays as many as 6 levels of indexing and will only display path 0 references. For backward references other than path 0, you should use the ZUDFM OAI/DBTAB/REL command.

REL
specifies all ID relationship files defined in the DBDEF table.

This includes multiple path backward references, recoup references, and capture/restore utility, information and statistics environment (CRUISE) references, as specified in the ID2= or ID3= parameters in the DBDEF. Any referenced ID that can only be found in the logical record (LREC) itself will also be displayed.

Note: You must first initialize the ID relationship files before using the ZUDFM OAI/DBTAB/*fileid*/REL command for the first time. See the REL/CREATE parameter for more information about initializing ID relationship files.

COM
displays common DBDEF subtable information on every file.

DBG
displays the DBDEF for each TPFDF file on the system and information for multiple file versions of the same file.

DBK
displays the default key information for all files that use default keys.

IDX
displays index information defined in the DBDEF.

PGN
displays the program that contains the DBDEF for the file.

REC
displays recoup information defined in the DBDEF.

CNT
displays the number of each type of subtable.

LINK
displays the program fast-link area.

REL/CREATE
initializes the ID relationship files using the DBDEF definitions.

fileid
is a 2-character alphanumeric or 4-character hexadecimal file ID.

WID
specifies ID restriction table maintenance.

ADD
specifies IDs to be added to an application.

applic
specifies a single alphabetic character defining an application type.

filecat
specifies the category of a file:

R Real-time file.

W Work file.

priority
specifies the order in which the ID ranges will be used. IDs assigned a priority of 1 are used first, followed by the IDs with a priority of 2, and so on.

C specifies that the IDs are in character format (2 characters).

X specifies that the IDs are in hexadecimal format (4 characters).

DEL
specifies IDs to be removed from an application.

NEW
specifies one or more IDs to be reserved (and locked) for an application. The IDs are released after 11 months if they are not used.

totalids
is the total number of IDs to reserve. If no number is specified, a single ID is reserved.

APL
specifies the application type.

DID
displays IDs that have been reserved with the NEW parameter.

DIS
displays IDs being used by each application.

Additional Information

- To keep data current, initialize the ID relationship files with the ZUDFM OAI/DBTAB/REL/CREATE command periodically, depending on the frequency of DBDEF change activity affecting the indexes. ID relationship files are created from the DBDEF definitions.
- All files (both TPFDF and traditional TPF) must be defined with a DBDEF table when using the ADD, DEL and NEW parameters to prevent the same ID being allocated to more than one application.
- In a TPF environment, if you use the alternate fast-link mechanism, the program fast linkages are displayed differently. The program name displays as DF*xx* instead of UF*xx*.

Examples

The following example displays DBDEF parameter settings for all files.

```
User:   ZUDFM OAI/DBTAB

System: UDFM0302I 17:23:32 CENTRAL DATA - FILE DEFINITIONS

                          F  D R D I   D W  A
                          V  B E B D   D R  P
            COREADR  RCID N  G C K X   A S  L FILE    RCT BEGORD   ENDORD
            ---------------------------------------------------------------
            001CA5C0  B K 00 . . . . . . L4 G BK0RP  0026 00000003 00000003
            001CA5FE  B K 01 . . . . . . L4 G BK0RP  0026 00000004 00000004
            001C9D82  S W 00 . . . . . . L2 G SW00SR
            001C9C70 FD00 00 Y Y . Y . . L4 G GR26SR
            001C99DE FD01 00 Y Y . Y . . L4 G GR29SR
            001CAB00 FD02 00 Y Y . . . . L4 G GR30SR
            001C9F5E FD03 00 Y Y . Y . . L4 G GR3NSR
            001CBC00 FD04 00 Y Y . Y . . L1 G GR94SR
            001CBD6C FD05 00 Y Y . Y . . L1 G GR95SR
            001CB686 FD06 00 Y . . . . . L4 G IW01DF
            001CA0D2 FD0A 00 Y Y . Y . . L4 I IRDCDF
            001C95C0 FD0B 00 Y Y . . . . L2 G ZZZZZZ
            001C964C FD0C 00 Y Y . . . . L2 G IIIIII
            001C9DC0 FD0D 00 Y . . . . . L4 G GW01SR
            001C9764 FD0E 00 Y . . . . . L4 G SW05SR
            001C96D8 FD0F 00 Y Y . . . . L4 G SR05SR
            001CA22E FDC0 00 Y . . . . . L4 I IWDWDF
            001CAAC2 FDE2 00 . . . . . . L4 G BF0WK  0028 00000004 00000004
            001CA1A2 FDE4 00 Y Y . . . . L1 I IRDKDF 0042 00000002 00000002
            001CA02E FDE5 00 Y Y . . . . L4 I IRDIDF 0193 00000000 00000498
            001CC9E0 FDE9 00 Y Y . . . . L4 I IR20DF 0193 00000000 00000498
            001CB5C0 FDEA 00 Y Y . . . Y L4 G IRQ1DF 0028 00000002 00000002
            001CB6EE FDEB 00 Y Y . . . . L4 G IRD1DF 0028 00000003 00000003
            001CBB56 FDEC 00 Y Y . . . . L1 G GR93SR 0186 00000002 00000002
            001CBAAC FDED 00 Y Y . . . . L1 G GR92SR 0187 00000000 00000025
            001CB9E0 FDEE 00 Y Y Y . . . L1 G GR91SR 0186 00000001 00000001
        END OF DISPLAY
```

The previous example contains information about every file defined to the TPFDF product. Use the information to ensure that you do not duplicate file IDs when creating new files. The categories are as follows:

COREADR	The DBDEF core address of the file.
RCID	The ID of the file (character or hexadecimal).
FVN	The file version number.
DBG	Indicates if the file is a TPFDF file with a Y. A period (.) indicates it is not a TPFDF file.
REC	Indicates if there is recoup definition for this file with a Y. A period (.) indicates that there is no recoup definition.
DBK	Indicates if there are default keys defined for this file with a Y. A period (.) indicates that there are no default keys defined.
IDX	Indicates if the file is an indexed file with a Y. A period (.) indicates that it is not an indexed file.
DDA	Indicates if distributed data access (DDA) is defined for this file with a Y. A period (.) indicates that DDA is not defined.
WRS	The primary block size used in file.
APL	A single alphabetic character that identifies the application type associated with this file.

FILE	The DSECT name of the file.
RCT	The record type, according to the FACE equate.
BEGORD	The FACE begin ordinal.
ENDORD	The FACE end ordinal.

The following example displays a single entry from the DBDEF index table corresponding to file FDEA.

```
User:   ZUDFM OAI/DBTAB/FDEA

System: UDFM0302I 17:23:32 CENTRAL DATA - FILE DEFINITIONS
                      F  D R D I   D W  A
                      V  B E B D   D R  P
        COREADR  RCID N  G C K X   A S  L FILE    RCT BEGORD   ENDORD
        --------------------------------------------------------------
        001A75F0 FDEA 00 Y Y . . . Y L4 G IRQ1DF 0170 00000002 00000002
```

The following example initializes the DBDEF table.

```
User:   ZUDFM OAI/DBTAB/INIT

System: UDFM0303I 14.41.09 UFB-CENTRAL DESCRIPTOR INITIALIZED
```

The following example displays the number of each type of subtable defined.

```
User:   ZUDFM OAI/DBTAB/CNT

System: TOTAL COUNTS OF SUBTABLES

         UDFM0312I 17:23:32 DBDEF SUBTABLE COUNT DISPLAY
         COM:   47  FIN:   47  DBG:   42  REC:   35
         DBK:    1  IDX:    7  GDS:    0  DDA:    2
         FILE VERSION 0: TOTAL:   45  TPFDF:   41
         TPFDF FILES: 91,1  PCT
        END OF DISPLAY
```

The values in the previous example represent the number of each type of table. The headings correspond to the following:

COM	The DBDEF common subtable. Because every DBDEF has a common subtable, this displays the total number of files in the system.
FIN	The file information subtable. Because every DBDEF has a common subtable this displays the total number of files in the system.
DBG	The TPFDF subtable.
REC	The recoup subtable.
DBK	The default key subtable.
IDX	The TPFDF index subtable.
GDS	The general data set subtable.
DDA	The distributed data access subtable.
TOTAL	The total number of data definitions with file version 0.
TPFDF	The total number of TPFDF data definitions with file version 0.

TPFDF FILES	The percentage of the total file version 0 definitions that are TPFDF definitions.

The following example displays detailed information for the specified file ID.

```
User:   ZUDFM OAI/DBTAB/SW/COM

System: UDFM0304I          COMMON DESCRIPTION

        WID  FILE ID               S W  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME          SW00SR  APL  APPLICATION INDICATOR    G
        PGN  TABLE PGM NAME       UF1A  SIZ  TABLE SIZE            003E

        RCT  FCTB FILE TYPE       0000  BOR  BEGIN ORDINAL     00000000
        PLT  PF0 ST / PF1 ST / PF2 LT   EOR  FACE END ORDINAL  00000000
        WRS  BLOCK SIZE             L2  GCI  CPU ID                   B
        ARS  ALTERNATE BLOCK SIZE   --  AUT  AUTHORIZATION MASK      00

        MTY  MDBF TYPE               C
```

Note: In the previous display, the multiply database function (MDBF) type indicates the following:

Type	Description
U	The subsystem user unique files
C	The subsystem user common files
N	No MDBF information.

The following example displays recoup information for the specified file ID. This display will vary depending on the file ID in the DBDEF.

```
User:   ZUDFM OAI/DBTAB/FD04/REC

System: UDFM0309I          RECOUP TABLE

        WID  FILE ID              FD04  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME          GR94SR  RCT  FCTB FILE TYPE        0000
        BOR  BEGIN ORDINAL    00000000  EOR  FACE END ORDINAL  00000000

        ID1  80   BIT 0 - 1  NAB TYPE
        ID4  10   BIT 3 - 1  NEW STYLE DBDEF

        STP  MONITOR BRANCH VECTOR  00  LEV  STRUCTURE LEVEL CTRL    00
        PFC  POS. OF FCH ADDRESS  0008  NAB  INITIAL NAB VALUE     001A
        CBV  TPFDF FIX OR VAR ITEM  05  ITS  ITEM SIZE             0001
        PNB  POS. OF DISP POINTER 0010  RDE  DEL. POINTER IN FILE  0000
        FSZ  SIZE OF NAB/COUNT FLD  03  ISZ  LENGTH OF SIZE FLD      03
        DIS  POS. OF DISP POINTER 0000  TIM  TIMEOUT VALUE FLD     0384
        MPR  DEPENDENT RECORD ID  0000  MPP  DEPENDENT CPU ID      NONE
        RCI  RCI DEPENDENT ITEM   FD05  NXT  NEXT ID TO CHASE      0000

        VSZ  SIZE OF TOT ID2 DESCRIPTIONS         0017

        NIT  NO. OF DIFFERENT ITK 0001  DIT  TPFDC DATA IDENTIFIER 0000
        ITK  ITEM KEY               80  .RF  NO. OF REFERENCES       01

        ID2  80   BIT 0 - 1  RCI ITEM                              FD04
        FNR  FILE DEFINITION NO.    00  RID  REFERENCED ID         FD05
        ADR  POS. OF REC ADDRESS  0003  RCP  POS. OF REC CODE CHCK 0007
        FAL  LENGTH OF FILE POINTER 0F  PI2  POS. OF SUBITEM START   00
        CP2  POS. OF CNT FLD SUBITM 00  IT2  ITEM SIZE OF SUBITEM    00
        SSZ  SUBITEM COUNT FLD SIZE 03
```

The following example displays which DBDEF program belongs to a particular file.

```
User:   ZUDFM OAI/DBTAB/PGN

System: UDFM0308I          PROGRAM REFERENCE
        WID  B K   FVN 00   FIL BK0RP   PGN UF1B  APL G
        WID  B K   FVN 01   FIL BK0RP   PGN UF1B  APL G
        WID  S W   FVN 00   FIL SW00SR  PGN UF1A  APL G
        WID FD00   FVN 00   FIL GR26SR  PGN UF1A  APL G
        WID FD01   FVN 00   FIL GR29SR  PGN UF1A  APL G
        WID FD02   FVN 00   FIL GR30SR  PGN UF1B  APL G
        WID FD03   FVN 00   FIL GR3NSR  PGN UF1A  APL G
        WID FD04   FVN 00   FIL GR94SR  PGN UF97  APL G
        WID FD05   FVN 00   FIL GR95SR  PGN UF97  APL G
        WID FD06   FVN 00   FIL IW01DF  PGN UF1D  APL G
        WID FD07   FVN 00   FIL IR24DF  PGN UF98  APL I
        WID FD0A   FVN 00   FIL IRDCDF  PGN UF1A  APL I
        WID FD0B   FVN 00   FIL ZZZZZZ  PGN UF1A  APL G
        WID FD0C   FVN 00   FIL IIIIII  PGN UF1A  APL G
        WID FD0D   FVN 00   FIL GW01SR  PGN UF1A  APL G
        WID FD0E   FVN 00   FIL SW05SR  PGN UF1A  APL G
        WID FD0F   FVN 00   FIL SR05SR  PGN UF1A  APL G
        WID FD11   FVN 00   FIL IR60DF  PGN UF99  APL I
```

The previous example provides the following information:

WID	The ID of the file
FVN	The file version number
FIL	The name of the file DSECT
PGN	The DBDEF program name
APL	The application type.

The following example displays a single DBDEF subtable for the specified file ID.

```
User:   ZUDFM OAI/DBTAB/IRA5DF/DBG

System: UDFM0305I          TPFDF TABLE

        WID  FILE ID              FDA5  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME          IRA5DF  RCT  FCTB FILE TYPE        0170
        BOR  BEGIN ORDINAL    00000005  EOR  FACE END ORDINAL  00000005

        INB  INITIAL NAB VALUE    001A  MNB  MAX NAB VALUE         0FDB
        ILT  LREC SIZE            0001  MIS  MAX LREC SIZE         0FC1
        EOR  END ORD TPFDF    00000005  NOC  NUMBER OF CHAINS        00
        ILV  TOT INTERLEAVES DEF  0000  PTN  NO. OF PARTITIONS     0000
        TYP  FILE TYPE               R  NLR  TPFDB0D LRECS P. BLK  0000
        GF0  OS DATA SET INDICATOR  00  TQK  HIGHEST TECHNICAL KEY   0F
        TRS  TRAILER SIZE         0024  LR   NO. LOG REC IDX REFS  0000
        SKE  EXTRACT SIZE BLK IDX 0000  RBV  ALG NO. FIXED FILE    0004
        PLI  PACKING LIMIT         100  PIN  PACKING VALUE           75
        HLD  GDS GR0HSR HOLD ORD   000
```

The following example displays default keys for a specific file.

```
User:   ZUDFM OAI/DBTAB/B071/DBK

System: UDFM0306I          KEY TABLE

        WID  FILE ID              B071  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME          IR71DF  RCT  FCTB FILE TYPE        0000
        BOR  BEGIN ORDINAL    00000000  EOR  FACE END ORDINAL  00000025

        PKY 80 TDI 0000 DUI 40 KCD
                                   INS  REG   PKY    L1/L2  COMPARAND
                    95804002       CLI        X-80-         NO INFO
                    474E0004       BC   STEP TO NEXT ITEM IN UP ORG
                    477E0018       BC   NOT FOUND - END PROCESSING
                    586030DC       L    R6                  SW00SK1
                    D51840036003   CLC              L-25  NO INFO
                    474E0004       BC   STEP TO NEXT ITEM IN UP ORG
                    477E0018       BC   NOT FOUND - END PROCESSING
                    47FE0030       BC   TOTAL MATCH FOUND
```

The following example displays the DBDEF index table for the specified file ID.

```
User:   ZUDFM OAI/DBTAB/FD00/IDX

System: UDFM0307I          DBDEF INDEX SUBTABLE

        WID  FILE ID              FD00  FVN  FILE DEF VERSION NO.    00
        TSZ  TOT INDEX TABLE SIZE 0080

        PDL  PATH DESCR. LENGTH   0042  PTH  RETRIEVAL PATH NO.    0000
        IMI  IDX METHOD INDICATORS  00  IFB  PKY OF LREC CAUS. IDX   C0
        IDD  DATA IDTFR CAUS. IDX 0000  IID  ID OF INDEX FILE      FDF6
        IKY  PKY OF INDEX LREC      80  ALG  LENGTH OF ALG DEF       06
        IPA  OFFSET OF ALG PART     00  ILA  LENGTH OF ALG PART      00
        IPK  OFFSET OF KEY PART     00  ILK  LENGTH OF KEY PART      06
        IFR  FILE REF NO.IN IDX ITM 00  IKA  START OF KEY AREA       08
        IRL  TOT LGTH OF IDX LREC 000E  DID  TPFDC DATA IDENTIFIER 0000
        LDI  DISP FOR AREA- EXTRACT 00  LLE  LGTH FOR AREA EXTRACT   0A
        ADI  DISP IN AREA- FOR AREA 02  IPE  END ALG OFFSET          00
        SP0  SPARE                  00  IWI  BIT1 - 1  UP ORGANIZ.   40
        IKI  POINTER TO KEY FINAL INITIAL ROUTINE                  0024
        KEY1-95804002474E0004477E0018586030DC
        KEY2-D50540086000474E0004477E001847FE003007F1
```

The following example displays the file IDs that are forward referenced from the specified ID in an index structure.

```
User:   ZUDFM OAI/DBTAB/FD04/REF

System: UDFM0310I          FORWARD REFERENCE OF INDEXED FILE
        FIXED ID  1. LEV   2. LEV   3. LEV   4. LEV   5. LEV   6. LEV

        FD04 00  FD05 00


        WID  FILE ID             FD04  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME         GR94SR  RCT  FCTB FILE TYPE        0000
        BOR  BEGIN ORDINAL   00000000  EOR  FACE END ORDINAL  00000000

        ID1  80   BIT 0 - 1  NAB TYPE
        ID4  10   BIT 3 - 1  NEW STYLE DBDEF

        STP  MONITOR BRANCH VECTOR  00  LEV  STRUCTURE LEVEL CTRL    00
        PFC  POS. OF FCH ADDRESS  0008  NAB  INITIAL NAB VALUE     001A
        CBV  TPFDF FIX OR VAR ITEM  05  ITS  ITEM SIZE             0001
        PNB  POS. OF DISP POINTER 0010  RDE  DEL. POINTER IN FILE  0000
        FSZ  SIZE OF NAB/COUNT FLD  03  ISZ  LENGTH OF SIZE FLD      03
        DIS  POS. OF DISP POINTER 0000  TIM  TIMEOUT VALUE FLD     0384
        MPR  DEPENDENT RECORD ID  0000  MPP  DEPENDENT CPU ID      NONE
        RCI  RCI DEPENDENT ITEM   FD05  NXT  NEXT ID TO CHASE      0000

        VSZ  SIZE OF TOT ID2 DESCRIPTIONS         0017

        NIT  NO. OF DIFFERENT ITK 0001  DIT  TPFDC DATA IDENTIFIER 0000
        ITK  ITEM KEY               80  .RF  NO. OF REFERENCES       01

        ID2  80   BIT 0 - 1  RCI ITEM                              FD04
        FNR  FILE DEFINITION NO.    00  RID  REFERENCED ID         FD05
        ADR  POS. OF REC ADDRESS  0003  RCP  POS. OF REC CODE CHCK 0007
        FAL  LENGTH OF FILE POINTER 0F  PI2  POS. OF SUBITEM START   00
        CP2  POS. OF CNT FLD SUBITM 00  IT2  ITEM SIZE OF SUBITEM    00
        SSZ  SUBITEM COUNT FLD SIZE 03


        WID  FILE ID             FD05  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME         GR95SR  RCT  FCTB FILE TYPE        0000
        BOR  BEGIN ORDINAL   00000000  EOR  FACE END ORDINAL  00000000

        ID1  80   BIT 0 - 1  NAB TYPE
        ID4  10   BIT 3 - 1  NEW STYLE DBDEF

        STP  MONITOR BRANCH VECTOR  00  LEV  STRUCTURE LEVEL CTRL    00
        PFC  POS. OF FCH ADDRESS  0008  NAB  INITIAL NAB VALUE     001A
        CBV  TPFDF FIX OR VAR ITEM  05  ITS  ITEM SIZE             0001
        PNB  POS. OF DISP POINTER 0010  RDE  DEL. POINTER IN FILE  0000
        FSZ  SIZE OF NAB/COUNT FLD  03  ISZ  LENGTH OF SIZE FLD      03
        DIS  POS. OF DISP POINTER 0000  TIM  TIMEOUT VALUE FLD     0384
        MPR  DEPENDENT RECORD ID  0000  MPP  DEPENDENT CPU ID         C
        RCI  RCI DEPENDENT ITEM   FD05  NXT  NEXT ID TO CHASE      0000
```

The following example displays the file IDs that are backward referenced from the specified ID in an index structure.

```
User:   ZUDFM OAI/DBTAB/FD05/BRF

System: UDFM0311I          BACKREFERENCE OF INDEXED FILES
        ---------------------------------------------------------------
        6. LEV   5. LEV   4. LEV   3. LEV   2. LEV   1. LEV   FIXED ID

                                                 FD05 00  FDED 00

        WID  FILE ID              FD05  FVN  FILE DEF VERSION NO.    00
        FIL  FILE NAME          GR95SR  RCT  FCTB FILE TYPE        0000
        BOR  BEGIN ORDINAL    00000000  EOR  FACE END ORDINAL  00000000

        ID1  80   BIT 0 - 1  NAB TYPE
        ID4  10   BIT 3 - 1  NEW STYLE DBDEF

        STP  MONITOR BRANCH VECTOR  00  LEV  STRUCTURE LEVEL CTRL    00
        PFC  POS. OF FCH ADDRESS  0008  NAB  INITIAL NAB VALUE     001A
        CBV  TPFDF FIX OR VAR ITEM  05  ITS  ITEM SIZE             0001
        PNB  POS. OF DISP POINTER 0010  RDE  DEL. POINTER IN FILE  0000
        FSZ  SIZE OF NAB/COUNT FLD  03  ISZ  LENGTH OF SIZE FLD      03
        DIS  POS. OF DISP POINTER 0000  TIM  TIMEOUT VALUE FLD     0384
```

The following example displays the relationship between the file ID specified and other file definitions.

```
User:   ZUDFM OAI/DBTAB/FD05/REL

System: UDFM0324I          RELATIONS FOR ID-FD05 FILE-GR95SR
        R RECOUP/C RCI/Z CRUISE/I INDEX  VAR VARIABLE ID
        FIXED ID   1.LEV      2.LEV      3.LEV      4.LEV      5.LEV

        FDEC00-R-I-FD0400-C-I-FD0500

        FDED00-C-I-FD0500
```

The following describes the parameters in the previous example:

R	The reference is a recoup reference (ID2= or ID3=) that will be used for recoup and CRUISE.
C	The reference is a recoup reference (ID2= or ID3=) that will be used for recoup and CRUISE and is also a recoup chain-chasing indicator (RCI) candidate.
Z	The reference is a ZFCRU reference (ID2= or ID3=) that will be used for ZFCRU only.
I	A backward reference using the TPFDF index mechanism.
VAR	The referenced ID is variable. The referenced ID can only be found in the logical record (LREC) itself and is not determined in the DBDEF.

The following example initializes ID relationship files using the DBDEF definitions.

```
User:   ZUDFM OAI/DBTAB/REL/CREATE

System: CSMP0097I 16.05.46 CPU-B SS-BSS  SSU-HPN  IS-01
        UDFM0322I 16.05.46 TPFDF FILE RELATION TABLE BUILD STARTED
        CSMP0097I 16.05.48 CPU-B SS-BSS  SSU-HPN  IS-01
        UDFM0323I 16.05.48 TPFDF FILE RELATION TABLE BUILD ENDED
```

The following example displays the fast-link directory on a TPF 4.1 system.

```
User:     ZUDFM OAI/DBTAB/LINK

System:   UDFM0314I          TPFDF FAST-LINK AREA DISPLAY
                      FAST LINK TABLE BASE:     032E1000
          PROGRAM FAST LINKAGES:
          CASE  PROGRAM DC DESCRIPTION
             0  UWAA  0  Y DBADD LAST/NBR
             1  UWAA  1  Y DBADD AFTER
             2  UWAA  2  Y DBADD BEFORE
             8  UWAB  0  Y DBADR
            16  UWAC  0  Y DBCKP
            24  UWAD  0  Y DBCLS
            32  UWAE  0  Y DBCPY
            40  UWAF  0  Y DBCRE
            48  UWAG  0  Y DBDEL LAST
            49  UWAG  1  Y DBDEL NEXT/BEGIN/LRECNBR
            50  UWAG  2  Y DBDEL LOCATED
            (...)
           448  UWBK  0  N DBOPN/INTERNAL
           449  UWBK  1  N DBIFB REF/INTERNAL
           450  UWBK  2  N DBIFB FIRST/INTERNAL
           451  UWBK  3  N DBIFB NEXT/INTERNAL
```

The following example displays the fast-link directory in an ALCS environment.

```
User:     ZUDFM OAI/DBTAB/LINK

System:   UDFM0314I          TPFDF FAST-LINK AREA DISPLAY
          UWAA   UWAB   UWAC   UWAD   UWAE   UWAF   UWAG   UWAH
          UWAI   UWAJ   UWAK   UWAL   UWAM   UWAN   UWAO   UWAP
          UWAQ   UWAR   UWAS   UWAT   UWAU   UWAV   UWAW   UWAX
          UWAY   UWA9   UWB0   UWB1   UWB2   UWB3   UWB4   UWB5
          UWB6   UWB7   UWB8   UWB9   UWBA   UWBB   UWBC   UWBD
          UWBE   UWBF   UWBG   UWBI   UWBJ   UWBK
```

The following example displays option bit settings in the DBDEF of the specified file.

```
User:   ZUDFM OAI/DBTAB/GRT1SR/OPT

System: UDFM0313I          DISPLAY OF OPTIONBITS IN SW02SR
        ----------------------------------------------------------------

        FIL GRT1SR    WID C211    FVN 00
          OP1 00
              BIT
              6 - 0  PACK ONLY AFTER -DELETE- COMMAND
          OP2 06
              BIT
              4 - 0  -FIWHC- PRIME BLOCK RETRIEVAL  -FINWC- IN SHORTTERM
              5 - 1  -FINWC- CHAIN BLOCK RETRIEVAL IF -HOLD- IS GIVEN
              6 - 1  -SERRC R DB010C- IF FILE MODIFIED WITHOUT -HOLD-

          OP3 00

          OP4 00

          CIN 00
```

The following example displays the file IDs for an application type.

```
User:   ZUDFM OAI/DBTAB/WID/APLG

System: UDFM0316I          IDENTIFIER USAGE BY APPLICATION
             CURRENTLY USED DATA IDENTIFIERS FOR APPLICATION  -  G

        ID   V  NAME   T DB | ID   V  NAME   T DB | ID   V  NAME   T DB
        --------------------|---------------------|---------------------
         B K 00 BK0RP  F    |  B K 01 BK0RP  F    |  S W 00 SW00SR P
        FD00 00 GR26SR P *  | FD01 00 GR29SR P *  | FD02 00 GR30SR P *
        FD03 00 GR3NSR P *  | FD04 00 GR94SR P *  | FD05 00 GR95SR P *
        FD06 00 IW01DF P *  | FD0B 00 ZZZZZZ P *  | FD0C 00 IIIIII P *
        FD0D 00 GW01SR P *  | FD0E 00 SW05SR P *  | FD0F 00 SR05SR P *
        FDE2 00 BF0WK  F    | FDEA 00 IRQ1DF F *  | FDEB 00 IRD1DF F *
        FDEC 00 GR93SR F *  | FDED 00 GR92SR F *  | FDEE 00 GR91SR F *
        FDF0 00 SR0GSR F *  | FDF1 00 SR01SR F *  | FDF2 00 SR00SR F *
        FDF3 00 GR3MSR F *  | FDF4 00 GR0ZSR F *  | FDF5 00 GR3OSR F *
        FDF6 00 GR25SR F *  | FDF7 00 GR28SR F *  | FDF8 00 GR27SR F *
        FDF9 00 GR0YSR F *  | FDFA 00 GR0WSR F *  | FDFB 00 GR0VSR F *
        FDFC 00 SATAS  F    | FDFD 00 GR32SR F *  | FDFF 00 GR31SR F *
                              TOTAL OF FILES:  35
```

The previous example provides the following information:

ID	The file ID
V	The file version number
NAME	The DSECT name
T	The file type (P=pool file and F=fixed file)
DB	The TPFDF file (*); not a TPFDF file (blank).

The following example specifies a range of IDs that are to be added to application C.

```
User:   ZUDFM OAI/DBTAB/WID/ADD/CR1F700-F7FF

System: UDFM0315I 08.41.20 UFB-WID-RESTRICTION TABLE MODIFIED
        END OF DISPLAY
```

The following example displays the type that each application is using in the ID restriction table.

```
User:   ZUDFM OAI/DBTAB/WID/DIS

System: UDFM0318I          ID RESTRICTION TABLE

        APL TYP PRI   ID RANGE       APL TYP PRI   ID RANGE
        ----------------------------------------------------
         C   G   0    4 A- 9 9  *      C   R   0    4 A- 9 9  *
         C   R   1   F700-F7FF         C   W   0   F000-F0FF
        ----------------------------------------------------
         E   G   0   F400-F4FF         E   R   0    3 A- 3 9  *
         E   R   1   F300-F3FF         E   W   0   F400-F4FF
        ----------------------------------------------------
         F   G   0    F A- F 9  *      F   G   1   C6C1-C6F9
         F   R   0    F A- F 9  *      F   R   1   C6C1-C6F9
         F   R   2   F200-F2FF         F   W   0   F100-F1FF
        ----------------------------------------------------
         G   G   0    1 0- 2 9  *      G   R   0    1 A- 2 9  *
         G   R   1   C000-C0FF         G   W   0   C000-C0FF
         G   X   0   C200-C2FF
        ----------------------------------------------------
         I   G   0   C100-C1FF
```

In the previous example, the following headings correspond to parameters that are used to add IDs to the restriction table:

APL	The application type
TYP	The file category
PRI	The priority
ID RANGE	The range of file IDs.

The asterisk (*) indicates that IDs were added with the **C** option.

The following example requests to reserve six IDs. The specific IDs that were requested are displayed.

```
User:   ZUDFM OAI/DBTAB/WID/NEW/GR6

System: UDFM0319I          NEXT ID-S FOR APPLICATION G AND TYPE R ARE:
        1 A   1 B   1 C   1 D   1 E   1 F
        END OF DISPLAY
```

The following example displays the IDs and ID ranges that have been reserved.

```
User:   ZUDFM OAI/DBTAB/WID/DID

System: UDFM0317I          ALREADY USED ID-S

        27JUN97  GR  1 A  1 B  1 C  1 D  1 E  1 F
```

References

See *TPFDF Database Administration* for more information about application types.

ZUDFM OAI/FILE–Display Information about DSECTs

Use this command to examine all the DSECT macros with macro label support (MLS) entries generated in your installation. You can also see the labels defined in each DSECT.

Requirements and Restrictions

None.

Format

```
►►──ZUDFM── ──┬─OAI/FILE─┬───────────┬─┬────────────────────────────────────►◄
              │          └─/filespec─┘ │
              └─FILe───────────────────┘
                     └─ ──filespec─┘
```

Note: You can use either the ZUDFM OAI/FILE or the ZUDFM FILE command.

filespec
is the file specification, which can be a 2-character or 4-hexadecimal file ID, or the DSECT name (up to 6 characters).

Additional Information

- If you use the ZUDFM OAI/FILE command without any parameters, a list of all DSECTS that have MLS entries generated will be displayed.
- You must load an MLS tape or data set to display DSECT information.

Examples

The following example displays all the DSECT names known to the TPFDF product in your installation that have MLS.

```
User:   ZUDFM OAI/FILE

System: UDFM0320I          DATA LABEL COLLECTION INDEX
        ----------------------------------------------

        GR0VSR    GR0WSR    GR0YSR    GR0ZSR    GR25SR    GR26SR
        GR27SR    GR28SR    GR29SR    GR3MSR    GR3NSR    GR3OSR
        GR30SR    GR31SR    GR32SR    GR91SR    GR92SR    GR93SR
        GR94SR    GR95SR    IRDCDF    IRDIDF    IRDKDF    IRD1DF
        IRQ1DF    IR20DF    IR24DF    IR25DF    IW01DF    SR0GSR
        SR00SR    SR01SR


        TOTAL MACRO LABEL SET ITEMS:   32
```

The following example displays the DSECT macro labels for a particular file. Because the # character is not a legal character for some terminal displays, the period (.) is used to represent it in the label name.

```
User:   ZUDFM OAI/FILE/GR95SR

System: UDFM0321I          DATA MACRO LABEL COLLECTION

        GR95SR01

        TPFDF KEYS: YES   KEYS/EOKS:  OK     NBRKEYS: 03   NBREOKS: 03
        TPFDC DIDS:  NO   DIDS/EODS: NOK     NBRDIDS: 00   NBREODS: 00

                    LABEL                TYP DISP VAL/LENG
                    .GR95SRS              E       00000021
                    .GR95SRI              E       0000FD05
                    GR95VAR               E       0000001A
                    GR95HDL               E       0000001A
                    .GR95K80              E       00000080
                    .GR95K90              E       00000090
                    .GR95KA0              E       000000A0
                    .GR95L80              E       00000021
                    .GR95L90              E       00000023
                    .GR95LA0              E       00000008
                    GR95ORG               E       00000008
                    GR95*0BEG             E       0000000D
                    GR95*0END             E       00000021
                    GR95E80               E       00000021
                    GR95E90               E       00000023
                    GR95EA0               E       00000008

                    LABEL                TYP DISP LENG/KEY DID
                    GR95HDR               C  0000 0010  00 0000
                    GR95REC               C  0000 FFFF  01 0000
                    GR95SIZ               H  0000 0002  01 0000
                    GR95KEY               X  0002 0001  01 0000
                    GR95NUM               C  0003 0004  80 0000
                    GR95SPC               C  0007 0001  80 0000
                    GR95SEX               C  0008 0001  80 0000
                    GR95SP1               C  0009 0001  80 0000
                    GR95AGE               C  000A 0002  80 0000
                    GR95SP2               C  000C 0001  80 0000
                    GR95NAM               C  000D 0014  80 0000
                    GR95NUM               C  0003 0004  90 0000
                    GR95SPC               C  0007 0001  90 0000
                    GR95SAL               C  0008 0006  90 0000
                    GR95SP3               C  000E 0001  90 0000
                    GR95JOB               C  000F 0014  90 0000
                    GR95NUM               C  0003 0004  A0 0000
                    GR95SPC               C  0007 0001  A0 0000
                    GR95CIT               C  0008 FFFF  A0 0000
```

In the previous example, TYP shows how the label is defined to the DSECT:

E	Equate
C	Character
X	Hexadecimal
F	Fullword
H	Halfword
A	Address.

References

See “ZUDFM OAT–Load Macro Label Set” on page 117 for more information about MLS.

ZUDFM OAINIT–Initialize Files

Use this command to initialize an entire file or part of a file.

Requirements and Restrictions

- You cannot enter this command while the capture and restore (ZFCRU) or recoup (ZRECP) utilities are running.
- Any ZUDFM commands that affect a file (for example, packing) hold the file for the duration of the command. The file is held at whatever level (index level) is appropriate to the command by holding the prime block.
- When you first install the TPFDF product, you must initialize the TPFDF control files in order to use ZUDFM OAINIT. See *TPFDF Installation and Customization* for more installation information.

Format

```
►►─ZUDFM─ ──┬─OAINIT─ ─ID/fileid─ ─┬──┬───────────────┬──┬────────────┬────►◄
            └─INIt─ ─fileid─ ──────┘  └─FVN/filever─┘  └─|ordrange─┘
```

Note: You can use either the ZUDFM OAINIT or ZUDFM INIT command.

fileid
: initializes a file, where *fileid* is a 2-character alphanumeric or 4-character hexadecimal file ID.

FVN/*filever*
: specifies the initialization of a file, where *filever* is the file version number given in hexadecimal format from X'00' to X'FF'. It is the same as that used in the ZUDFM OAI/DBTAB/*fileid* command display.

ordrange
: specifies the ordinal range within the file that will be initialized. Use one of the following forms:

  - **|***beginord***-***endord*
  - **|***beginord***-LAST**

  The begin and end ordinals are stated as decimal ordinal numbers that can be as many as 8 digits.

Note: You can use a different delimiter by setting variable #DBENUFB in the ACPDBE macro. See *TPFDF Installation and Customization* for more information about ACPDBE.

Additional Information

- The use of the ZUDFM OAINIT command should be restricted on the live system with the ZUDFM OAE command.

Examples

In the following example, the specified file ID is initialized. Because a file is changed permanently when initialized, you must enter the ZUDFM OAINIT command twice as a precaution. The first entry displays file ID information and a warning. The second entry completes the initialization of the file ID.

```
User:   ZUDFM OAINIT ID/B075

System: UDFM0380I          YOU WILL INITIALIZE THE FOLLOWING FILE
        ID - B075 FVN - 00 BOR - 00000052 EOR - 00000077
        FACET - #BTREE   SSU - WP1

        UDFM0381A          DANGEROUS ENTRY / MAKE SURE:

              1 - YOU KNOW WHAT YOU ARE DOING
              2 - YOU HAVE A RECENT CAPTURE AT HAND
              3 - THE FILE YOU ARE ABOUT TO INITIALIZE
                  IS PROPERLY DEFINED IN DBDEF
              4 - THE SYSTEM IS NOT IN TOO BAD A SHAPE

              IF YOU STILL WANT TO INITIALIZE YOUR FILE
              REPEAT YOUR ENTRY WITHIN 2 MINUTES
```

```
User:   ZUDFM OAINIT ID/B075

System: UDFM0383I          ZUDFM DATABASE INITIALIZATION COMPLETED
        ID - B075 FVN - 00 ORDS PROC - 00000026
```

References

- See *TPFDF Installation and Customization* for more information about initializing TPFDF control files.
- See “ZUDFM OAE–Restrict Table Handling” on page 76 for more information about restricted tables.

ZUDFM OAL–Display Recent ZUDFM OA Commands

Use this command to display the modification log that the TPFDF product maintains to identify ZUDFM OA commands that have been used recently.

Requirements and Restrictions

None.

Format

```
►►──ZUDFM── ──┬─OAL─┬───────────────────────────────────────────────────────────►◄
              └─LOG─┘
```

Note: You can use either the ZUDFM OAL or ZUDFM LOG command.

Additional Information

- All ZUDFM commands that modify a nonsystem file are logged in the ZUDFM modification log file.

Examples

The following example displays a list of the most recent ZUDFM commands used.

```
User:   ZUDFM OAL

System: UDFM0421I          OA MODIFICATION LOG DISPLAY
        DATE  TIME   CRT     SI/DU  FILEADDR   ID   ENTRY
        04JUL 12.03  842277  HN/PR  0B8913BA   IY   OAX7
        04JUL 11.41  84098D  EF/PR  00980BDB   R9   OAC/8004E46B
        04JUL 11.39  84098D  EF/PR  00980BDB   R9   OAX2-51
        04JUL 11.36  84098D  EF/PR  00980BDB   R9   OAA2/000A0032E4
        04JUL 11.34  84098D  EF/PR  00980BDB   R9   OAA/TESTRECORD
        04JUL 11.33  84098D  EF/PR  00980BDB   R9   OAC/8004E46B
        END OF DISPLAY
```

DATE	The date of the modification entry.
TIME	The time of the modification entry.
CRT	The terminal address from which the modification was entered.
SI/DU	The initials of the programmer and duty code. This requires the AAA block.
FILEADDR	The file address of the record that was modified.
ID	The 2-byte file ID of the file that was modified.
ENTRY	The first 15 bytes of the input message.

References

None.

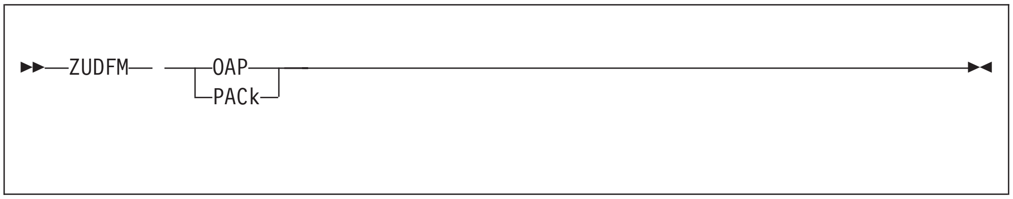
ZUDFM OAP–Pack a Subfile

Use this command to remove unused space in a subfile or to build or rebuild a B^+Tree index.

Requirements and Restrictions

- Before using this command, you must first link to the file that you want to access.
- Any ZUDFM commands that affect a file (for example, packing) hold the file for the duration of the command. The file is held at whatever level (index level) is appropriate to the command by holding the prime block.

Format



Note: You can use either the ZUDFM OAP or the ZUDFM PACK command.

Additional Information

- Blocks are filled to the packing limit percentage specified in the database definition (DBDEF) macro (parameter PLI) of the file.
- You can use this command to pack all the logical records (LRECs) in a subfile to which you are currently linked. For B^+Tree data files, the ZUDFM OAP command also builds or updates the B^+Tree index files.

  By default, the TPFDF product packs a subfile when it closes it if any LRECs have been deleted and if the packing density of one or more blocks is below a limit set by the database administrator in the DBDEF macro.

  Note: TPFDF only packs B^+Tree data files when there are no B^+Tree index nodes and there are forward chains in the data file.
- You can use the ZUDFM OAP command to pack LRECs into the minimum number of blocks regardless of the current packing density in the subfile and whether LRECs have been deleted or not.
- Because packing large database structures requires a significant amount of system resources, commit scope processing is not available when you use this command, regardless of what the DBDEF macro has set as the default.

Examples

The following example packs the file that is currently linked. Because the file is a B+Tree file, the B+Tree index is also re-created.

ZUDFM OAP

```
User:   ZUDFM OAP

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. F404F868
        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 D  U N N00 00000000 00328065
        000003 001080 I  N O U Y  E000000 00690794
        000004 001080 J  A C O B  S000000 00669412
        000005 001080 J  O N E S 00000000 00789441
        000006 001080 T  A Y L O  R000000 00123964
        000007 001080 T  H A Y E  R000000 00236348
        000008 001080 U  N D E R  H I L L 00894350
```

References

- See *TPFDF Database Administration* for more information about packing files.
- See *TPFDF Programming Concepts and Reference* for more information about commit scopes.
- See the following for more information about linking to a subfile:
  - "ZUDFM OA–Link to a Subfile Using the File Address" on page 62
  - "ZUDFM OA*–Display LRECS in a Subfile" on page 64
  - "ZUDFM OAI–File Access Using the ID or DSECT Name" on page 83.

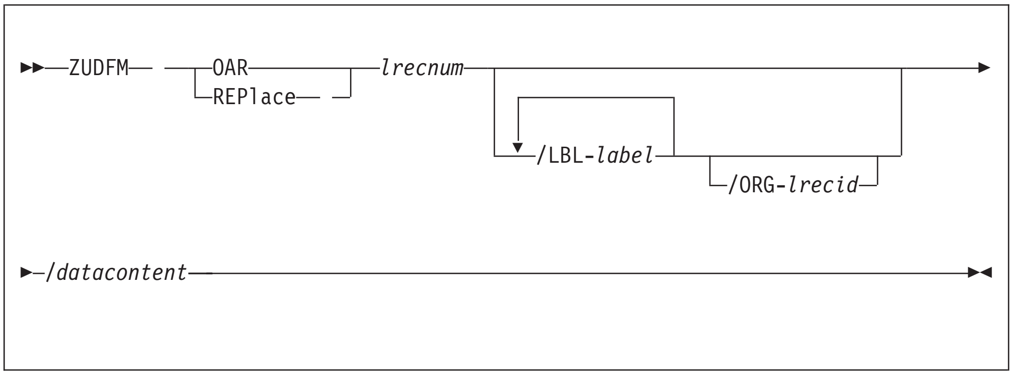
ZUDFM OAR–Replace User Data in an LREC

Use this command to replace the user data portion of a logical record (LREC).

Requirements and Restrictions

- Before using this command, you must first link to and display the file that you want to access.
- If you do not replace an LREC successfully, you must display it again before you can use this command.
- You must specify the LREC to be changed by its sequence number in the subfile. Locate the sequence number by displaying a subfile using the ZUDFM OAI, ZUDFM OA, or ZUDFM OA* command.
- You cannot use this command to replace technical LRECs (TLRECs) or to replace an LREC with a TLREC.

Format



Note: You can use either the ZUDFM OAR or the ZUDFM REPLACE command.

lrecnum
is the sequence number of the LREC.

LBL-*label*
specifies the DSECT field to be replaced. Do not include the first 4 characters of the DSECT name. When using this parameter with fixed length fields, the length of the data content must match the length of the field.

Note: When using this parameter, you must build a macro label set using macro label set support (MLS).

ORG-*lrecid*
specifies the LREC ID, where *lrecid* is a 1-byte field in which the label is defined. Use this parameter when the label specified by the LBL parameter is not defined for the LREC ID of the record being replaced. This parameter indicates the LREC ID for which the label is defined.

datacontent
is the user data portion of the LREC.

Note: This variable can contain a mixture of hexadecimal and character data. To switch between these types of data, use the vertical bar (|) as a delimiter. The following example shows how the types of data can be mixed:

```
8013B4|THIS IS CHARACTER DATA|4A5F90
```

The TPFDF product assumes that the variable starts with hexadecimal data. To start with character data, place a vertical bar at the beginning of the variable. For example:

```
|STARTING WITH CHARACTER DATA|1F2E3D4E
```

You can use a different delimiter by setting variable #DBENUFB in the ACPDBE macro. See *TPFDF Installation and Customization* for more information about ACPDBE.

Additional Information

- Use the ZUDFM OAR command to read, modify, and replace user data in LRECs in a subfile to which you are currently linked.
- If you do not use the label parameter, you must specify the primary key of the LREC (the LREC ID) when you add data to it. Do not enter the LREC length field, if any.
- You cannot change the size field in the LREC directly; the TPFDF product adjusts the LREC size field for you.
- There is a danger of damaging the file organization using this command. The best way to ensure database integrity is for the database administrator to define default keys and file organization in the database definition (DBDEF) table. In this case, if you change data in an LREC that would corrupt the organization of the subfile, the TPFDF product automatically moves the LREC with the replacement data to the correct position in the subfile.
- When you use a ZUDFM OA command to display a subfile that you are going to modify (ZUDFM OAA, ZUDFM OAR, or ZUDFM OAX) there is a time interval during which another entry could modify the same subfile. Consequently, your modification could inadvertently change the wrong LREC in the subfile and damage the database.

  To overcome this problem, the TPFDF product maintains a sequence counter that is used by ZUDFM to ensure file integrity. The sequence counter is updated in the block header whenever the subfile is changed. In subsequent entries the retained value is compared with the actual value to ensure that the database is updated in the correct sequence.

  Heavily modified databases may become difficult to maintain with ZUDFM entries if the sequence count is constantly being changed. In this case, write your own utilities to locate the file, hold it, and change it.

  Attention: If you enter multiple ZUDFM OA commands that modify a subfile, the sequence counter is not checked. In other words, you are allowed to modify the subfile more than once without having to display it before each ZUDFM OA command. You will only receive a sequence error if another user or application program modifies the subfile while you are entering the ZUDFM OA commands.

Examples

The following example shows the text portion of an LREC that is being replaced.

```
User:   ZUDFM OAR3/80|UPDATED TEXT

System: UDFM0001I          TPFDF LREC DISPLAY

        F.A. F404E801
        000002 000E80 L  R E C    T E X T    2
        000003 000F80 U  P D A T  E D   T  E X T
        000004 000E80 L  R E C    T E X T    4
        000005 000E80 L  R E C    T E X T    5
        OAR3/80|UPDATED TEXT
```

References

- See the following for more information about linking to a subfile:
  - “ZUDFM OA–Link to a Subfile Using the File Address” on page 62
  - “ZUDFM OA*–Display LRECS in a Subfile” on page 64
  - “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83.
- See “ZUDFM OAT–Load Macro Label Set” on page 117 for more information about MLS.

ZUDFM OAS–Display File Information during Tracing

Use this command to display information during a trace.

Requirements and Restrictions

- The trace facility that you use and the organization of the database are installation dependent. The format of data supplied to the ZUDFM OAS command must be defined independently.
- When you use the TPF system, the database administrator must update user exit UFDA so that it is compatible with the ZUDFM OAS command. The ZUDFM OAS command is compatible in an ALCS environment.

Format

```
►►──ZUDFM── ─────────────────────────────────────────────────────────►

►──┬─┬─OAS───┬────────────────────────────────────────────────┬────►◄
   │ └─TRAce─┘                                                │
   └─┬─OAS/────┬──┬─W──┬───────────────────────────────────┬─┬─┘
     └─TRAce── ┘  │    └─┬─/refnum──────┬─┬──────────────┬─┘ │
                  │      ├─/R3──────────┤ └─┬─/fieldnum─┬┘   │
                  │      ├─/regdispl────┤   └─/OPS──────┘    │
                  │      ├─/dbifbaddr───┤                    │
                  │      ├─-refname─────┤                    │
                  │      ├─/EBWfieldnum─┤                    │
                  │      └─/EBXfieldnum─┘                    │
                  └─D──┬───────────────────────────────────┬─┘
                       └─┬─/refnum──────┬─┬─────────────┬──┘
                         ├─/R3──────────┤ └─┬─/LBL─┬────┘
                         ├─/regdispl────┤   └─/*───┘
                         ├─/dbifbaddr───┤
                         ├─-refname─────┤
                         ├─/EBWfieldnum─┤
                         └─/EBXfieldnum─┘
```

Note: You can use either the ZUDFM OAS or the ZUDFM TRACE command.

W specifies information in the SW00SR slot. When used with another parameter, W specifies the core address of the SW00SR slot that is currently being processed.

D specifies information in the subfile data block. When used with another parameter, D specifies the core address of the subfile data block that is currently being processed.

refnum
is a subfile using a reference number from 1 to 99 of a SW00SR slot. The reference number is displayed using the ZUDFM OAS command.

Note: Do not specify this parameter if you opened the file by using the DBOPN macro with the FILE and PARAM parameters specified.

R3
specifies register 3, which contains the core address of the SW00SR slot that is currently being processed.

regdispl
is a register value followed by a displacement value. You can specify one of the following register values:

- R1–R7
- RE–RF.

The displacement value is a 3-character numeric value.

dbifbaddr
is the database interface block (DBIFB) address of the SW00SR slot that specifies a file or subfile.

refname
is the file ID or the file name of a subfile that has been opened.

Note: Do not specify this parameter if you opened the file using the DBOPN macro with the FILE and PARAM parameters.

EBW
specifies EBW work area 1, located in the entry contol block (ECB).

EBX
specifies EBX work area 2, located in the ECB.

fieldnum
is a 3-digit ECB application work area number that contains a SW00SR slot address.

OPS
displays the bit settings of option fields.

LBL
displays the subfile with the DSECT label information.

Note: When using this parameter, you must build a macro label set using the macro label set (MLS) support.

***** displays the subfile without DSECT label information.

Additional Information

- If you use the ZUDFM OAS command without any parameters, all subfiles opened during a trace will be displayed.
- You must use the ZUDFM OAS command parameters to display specific subfile information if there is more than one subfile opened during a trace.

Examples

The following example uses the command format.

```
User:   ZUDFM OAS

System: UDFM0521I            CURRENTLY PENDING FILES

        NR.   FILE NAME  SW00-BASE  LEV   ID    REF NAME
        --------------------------------------------------------
        0001  06-GR91SR  038A5D30    A   FDEE   G R 9 1 S R
        0002  06-GR95SR  038A5A18    B   FD05   G R 9 5 S R
```

NR.	The reference number of this subfile in the DBIFB.
FILE NAME	The file name.
SW00-BASE	The storage address of the base of the SW00SR slot.
LEV	The ECB data level of the TPFDF subfile.
ID	The file ID.
REF NAME	The 6- or 8-character reference name.

The following example displays the first subfile in the database interface block and lists the LRECs in the subfile.

```
User:   ZUDFM OAS/D/1

System: SSTR - TPFDF COREADDR 0372E460   FILENAME..06-GR92SR

        RECORD CONTENT AND CURRENT POINTER *---*

        0001  001A  0372E47A   001C8000 6195 053  K E V I  N404040
                               40404040 40404040 40404040
        0002  0036  0372E496   001C8000 6196 0EB  L A U R  A404040
                               40404040 40404040 40404040
        0003  0052  0372E4B2   001C8000 6197 00B  T O M40 40404040
                               40404040 40404040 40404040
        0004  006E  0372E4CE   001C8000 6198 003  J O H N 40404040
                               40404040 40404040 40404040
        0005  008A  0372E4EA   001C8000 6199 09F  D A N40 40404040
                               40404040 40404040 40404040
        0006  00A6  0372E506   001C8000 619A 03B  R O B40 40404040
                               40404040 40404040 40404040
```

The following example displays information for two parameter keys of the first SW00SR slot in the database interface block.

```
User:   ZUDFM OAS/W/1

System: UDFM0520I           SW00SR BASE 0047FCE0    FILENAME..06-GR32SR

        MEANING                   LABEL     CONTENT
        --------------------------------------------------------
        PROGRAM STAMP             SW00PGM   U T N C
        FILE IDENTIFIER           SW00WID   FDFD
        ACTUAL TPFDF COMMAND      SW00CMD   READ
        AT PROGRAM LOCATION       SW00PL2   00C25122   UTNC 0122
        ERROR 1                   SW00RTN   00
        ERROR 2                   SW00RT2   00
        ERR COUNT SINCE OPEN      SW00RT1   00
        CURRENT FILE ADDRESS      SW00CFA   16180003
        CURRENT CORE ADDRESS      SW00CCA   004B2BE0
        CURRENT NAB               SW00NAB   001A
        CURRENT LREC CORE ADDR.   SW00REC   004B2BFA
        NEWLREC ADR FOR ADD/INS   SW00IPT   00000000
        CURRENT DATA LEVEL        SW00LVL   50   LEVEL 10
        CURRENT PARAM LEVEL       SW00SLV   FF   DYNAMIC PARAM ALLOC.
        PRIME FILE ADDRESS        SW00FAD   16180003
        PRIME CORE ADDRESS        SW00PCA   004B2BE0
        LAST CHAIN ADDRESS        SW00LCH   00000000
        LAST MATCH. LREC F.A.     SW00PAD   00000000
        LAST MACH. LREC NAB       SW00PNB   0000
        RETRIEVAL ALGORITHM       SW00ALG   00000000000000000000
                                            00000000000000000000
        SPACE WORKAREA ADDRESS    SW00WKA   00000000
        EXTENDED KEY AREA         SW00KEY   00000000
        KEY INDICATORS            SW00NKY   A0
        KEY INSTRUCTIONS          KEY   1   CLI  95804002
                                            BC   474E0004
                                            BC   477E0018
                                  KEY   2   L    586030E0 *00403390
                                            CLC  D50040036000
                                            BC   474E0004
                                            BC   477E0018
                                            BC   47FE0030
```

The following example displays information for the key list of the first SW00SR slot in the database interface block.

```
User:   ZUDFM OAS/W/1

System: UDFM0520I           SW00SR BASE 0047FCE0    FILENAME..06-GR32SR
        MEANING                   LABEL      CONTENT
        --------------------------------------------------------
        PROGRAM STAMP             SW00PGM   U T N C
        FILE IDENTIFIER           SW00WID   FDFD
        ACTUAL TPFDF COMMAND      SW00CMD   KEY
        AT PROGRAM LOCATION       SW00PL2   00C2515E   UTNC 015E
        ERROR 1                   SW00RTN   00
        ERROR 2                   SW00RT2   00
        ERR COUNT SINCE OPEN      SW00RT1   00
        CURRENT FILE ADDRESS      SW00CFA   16180003
        CURRENT CORE ADDRESS      SW00CCA   004B2BE0
        CURRENT NAB               SW00NAB   001A
        CURRENT LREC CORE ADDR.   SW00REC   004B2BFA
        NEWLREC ADR FOR ADD/INS   SW00IPT   00000000
        CURRENT DATA LEVEL        SW00LVL   50   LEVEL 10
        CURRENT PARAM LEVEL       SW00SLV   FF   DYNAMIC PARAM ALLOC.
        PRIME FILE ADDRESS        SW00FAD   16180003
        PRIME CORE ADDRESS        SW00PCA   004B2BE0
        LAST CHAIN ADDRESS        SW00LCH   00000000
        LAST MATCH. LREC F.A.     SW00PAD   00000000
        LAST MACH. LREC NAB       SW00PNB   0000
        RETRIEVAL ALGORITHM       SW00ALG   00000000000000000000
                                            00000000000000000000
        SPACE WORKAREA ADDRESS    SW00WKA   00000000
        EXTENDED KEY AREA         SW00KEY   004B9000
        KEY INDICATORS            SW00NKY   B0
        KEY INSTRUCTIONS          KEY   1   CLI  95804002
                                            BC   474E0004
                                            BC   477E0018
                                  KEY   2   L    58610CF4 *00403390
                                            CLC  D50040036000
                                            BC   474E0004
                                            BC   477E0018
                                            BC   47F0E030
```

The following example displays the SW00FAD field in the SW00SR slot of the first subfile opened during the trace. The display shows the address of the SW00SR base and the name of the subfile, followed by the label of the field, its displacement from the SW00SR base in bytes, its address in main storage (COREADDR), and its data content.

```
User:   ZUDFM OAS/W/1/FAD

System: UDFM0520I           SW00SR BASE 0372DD30    FILENAME..06-GR91SR

             LABEL        DISP   COREADDR  CONTENT
             --------------------------------------

             SW00FAD      0044   00357EAC  0098065B
        END OF DISPLAY
```

The following example displays the option fields (OPS) for the second subfile opened during the trace and the indicator bit settings of the second SW00SR slot.

```
User:   ZUDFM OAS/W/2/OPS

System: UDFM0472I           SPECIAL OPTIONS DISPLAY FILENAME..06-GR95SR
        LABEL     OPTION *BIT ON*
        -----------------------------------------------------------
        SW00OP1   BIT6 *OFF* PACK ONLY AFTER DELETE
        SW00OP2   BIT4 *OFF* FIWHC FOR PRIME BLOCK
                  BIT5 FINWC IF HOLD/CH.
                  BIT6 DB010C-IF NO HOLD
        SW00ID1   BIT4 NEXT OPTION
        SW00ID3   BIT6 ALG WAS GIVEN
        SW00ID4   BIT7 *OFF* HOLD
        SW00ID6   BIT7 FILE IS OPEN
        SW00IDB   BIT0 READ INDEX
```

The following example displays the first subfile opened during tracing and data in the current LREC, together with DSECT labels identifying each field.

```
User:   ZUDFM OAS/D/1/LBL

System: SSTR - TPFDF COREADDR 00353040   FILENAME..06-GR32SR

        NBR   LABEL   DIS  LEN   COREADDR  CONTENT
        ----  ------- ---- ----  --------  --------------------
        F.A. 0098065B 001A
        0001  GR32SIZ 0000 0002  0035305A  000B
              GR32KEY 0002 0001  0035305C  80
              GR32APL 0003 0001  0035305D   A
              GR32TYP 0004 0001  0035305E   B
              GR32PRI 0005 0001  0035305F   0
              GR32IND 0006 0001  00353060  00
              GR32IDF 0007 0002  00353061  C00A
```

The following example displays the SW00SR slot information of the subfile that is currently being processed.

ZUDFM OAS

```
User:   ZUDFM OAS/W/R3

System: UDFM0520I           SW00SR BASE 0372D3E8    FILENAME..06-GR92SR

        MEANING                   LABEL     CONTENT
        --------------------------------------------------------
        PROGRAM STAMP             SW00PGM   Q T 1 9
        FILE IDENTIFIER           SW00WID   FDED
        ACTUAL TPFDF COMMAND      SW00CMD   READ
        AT PROGRAM LOCATION       SW00PL2   7F46B9E4   UF0C 0894
        ERROR 1                   SW00RTN   00
        ERROR 2                   SW00RT2   00
        ERR COUNT SINCE OPEN      SW00RT1   00
        CURRENT FILE ADDRESS      SW00CFA   00001042
        CURRENT CORE ADDRESS      SW00CCA   0372E460
        CURRENT NAB               SW00NAB   014E
        CURRENT LREC CORE ADDR.   SW00REC   00000000
        NEWLREC ADR FOR ADD/INS   SW00IPT   00000000
        CURRENT DATA LEVEL        SW00LVL   48   LEVEL 09
        CURRENT PARAM LEVEL       SW00SLV   FF   DYNAMIC PARAM ALLOC.
        PRIME FILE ADDRESS        SW00FAD   00001042
        PRIME CORE ADDRESS        SW00PCA   0372E460
        LAST CHAIN ADDRESS        SW00LCH   00000000
        LAST MATCH. LREC F.A.     SW00PAD   00000000
        LAST MACH. LREC NAB       SW00PNB   0000
        RETRIEVAL ALGORITHM       SW00ALG   0372D715000000000000
                                            00000000000000000000
        SPACE WORKAREA ADDRESS    SW00WKA   00000000
        EXTENDED KEY AREA         SW00KEY   00000000
        KEY INDICATORS            SW00NKY   A0
        KEY INSTRUCTIONS          KEY   1   CLI  95804002
                                            BC   474E0004
                                            BC   477E0018
                                  KEY   2   L    586030DC *0372D715
                                            CLC  D51340086000
                                            BC   474E0004
                                            BC   477E0018
                                            BC   47FE0030
```

The following example displays the current LREC (SW00REC) of the subfile that is currently being processed. The display information varies depending on the subfile to which you are linked.

```
User:   ZUDFM OAS/D/R3

System: SSTR - TPFDF COREADDR 0372E460   FILENAME..06-GR92SR

        RECORD CONTENT AND CURRENT POINTER *----*

        0001  001A  0372E47A   001C8000 6195 053  K E V I  N404040
                               40404040 40404040 40404040
        0002  0036  0372E496   001C8000 6196 0EB  L A U R  A404040
                               40404040 40404040 40404040
        0003  0052  0372E4B2   001C8000 6197 00B  T O M40 40404040
                               40404040 40404040 40404040
        0004  006E  0372E4CE   001C8000 6198 003  J O H N 40404040
                               40404040 40404040 40404040
        0005  008A  0372E4EA   001C8000 6199 09F  D A N40 40404040
                               40404040 40404040 40404040
        0006  00A6  0372E506   001C8000 619A 03B  R O B40 40404040
                               40404040 40404040 40404040
```

References

See “ZUDFM OAT–Load Macro Label Set” on page 117 for more information about MLS.

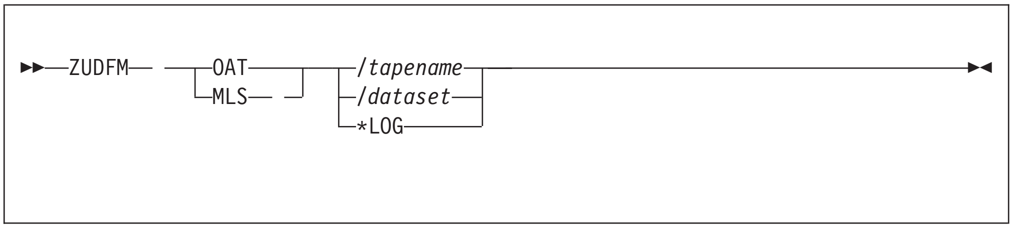
ZUDFM OAT–Load Macro Label Set

Use this command to build the macro label set (MLS) database online.

Requirements and Restrictions

You must create an MLS tape or data set before using this command.

Format



Note: You can use the ZUDFM OAT or the ZUDFM MLS command.

tapename
is the tape name in the job control language (JCL) used to create the macro label set.

dataset
is the data set in the JCL used to create the macro label set.

***LOG**
specifies the tape loading information.

Additional Information

- MLS support consists of an online command and an offline program used to extract information from source libraries. The information comprises labels from the data definition (DSECT) macros.
- Build an MLS in two stages:
  1. Offline
     a. Extract the DSECT label information from the source library using JCL. The DSECTS included must be specified in the JCL.
     b. Use a program to assemble the selected DSECT macros and create a DSECT label print file.
     c. Use segment UFOF to transform, sort, and write the labels to a sequential file.
  2. Online
     a. Use the ZUDFM OAT command to load the data set. The tape name references a sequential data set that is usually a disk (in ALCS) or tape (in TPF).
     b. You can then use the ZUDFM OAI/FILE command to display the DSECT label information.

        Note: If you enter the ZUDFM OAI/FILE command with no parameters, a list of files that have macro label set entries generated is displayed.
- To use the JCL in an environment with IBM High-Level Assembler (TPF or ALCS), change the compile program to ASMA90 as follows:

```
//COMPILE  EXEC PGM=ASMA90,PARM='NODECK,LINECOUNT(32000),XREF(FULL)'
```

- A sample JCL (MLST) for running MLS on a TPF system is provided on the shipment tape.

Examples

The following example shows the MLS data set being loaded from a JOB tape.

```
User:   ZUDFM OAT/JOB

System: UDFM0482I 16.28.16 TAPELOAD STARTED
        UDFM0483I 16.28.17 TAPELOAD ENDED
```

The following example shows the tape loading information (LOG) of macro label sets loaded.

```
User:   ZUDFM OAT*LOG

System: UDFM0481I          MACRO LABEL SET - TAPELOAD STATISTICS

         RUNNING          NEW
        LAST RUN          TIME       TOTAL LOADED SKIPPED UPDATED ERRORS
        ----------------------------------------------------------------
        29JAN97 07.57.09  01.28.18      35                   35
        21NOV96 15.48.59  01.28.18      35                   35
        21NOV96 15.42.19  01.28.18      35                   35
        21NOV96 15.38.43  01.28.18      35   35
```

References

- See “ZUDFM OAI/FILE–Display Information about DSECTs” on page 100 for more information about MLS.
- See *TPFDF Installation and Customization* for additional about the JCL used with this command.

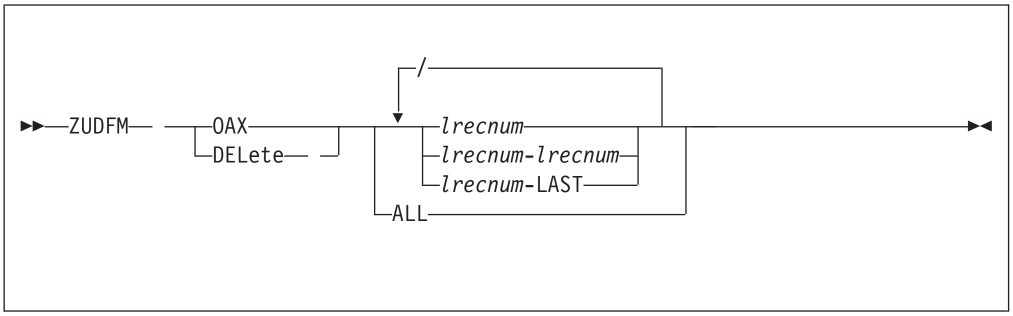
ZUDFM OAX–Delete LRECs from a Subfile

Use this command to delete logical records (LRECs) from a subfile.

Requirements and Restrictions

- Before using this command, you must first link to and display the subfile that you want to access.
- You must specify the LREC to be deleted by its sequence number in the subfile. Locate the sequence number by displaying a subfile using the ZUDFM OAI, ZUDFM OA, or ZUDFM OA* command.
- LREC sequence numbers start at 1.
- LREC ranges must be in ascending order; if one is found out of order, that range and all subsequent ranges are ignored.

Format



Note: You can use either the ZUDFM OAX or the ZUDFM DELETE command.

lrecnum
is an LREC to be deleted.

lrecnum-lrecnum
is a particular range of LRECs to be deleted.

lrecnum-**LAST**
is a range of LRECs to be deleted, beginning from a particular LREC to the last LREC in the subfile.

ALL
specifies all LRECs in a subfile to be deleted.

Additional Information

- When you use a ZUDFM OA command to display a subfile that you are going to modify (ZUDFM OAA, ZUDFM OAR, or ZUDFM OAX) there is a time interval during which another entry could modify the same subfile. Consequently, your modification could inadvertently change the wrong LREC in the subfile and damage the database.

  To overcome this problem, the TPFDF product maintains a sequence counter that is used by ZUDFM to ensure file integrity. The sequence counter is updated in the block header whenever the subfile is changed. In subsequent entries, the retained value is compared with the actual value to ensure that the database is updated in the correct sequence.

Heavily modified databases may become difficult to maintain with ZUDFM entries if the sequence count is constantly being changed. In this case, write your own utilities to locate the file, hold it, and change it.

Note: If you enter multiple ZUDFM OA commands that modify a subfile, the sequence counter is not checked. In other words, you are allowed to modify the subfile more than once without having to display it before each ZUDFM OA command. You will receive a sequence error only if another user or application program modifies the subfile while you are entering the ZUDFM OA commands.

Examples

In the following examples, the entries corresponding to BROCKER, JACOBS, and TAYLOR are deleted from the subfile that is currently linked. A ZUDFM OA* command is necessary before the entries are deleted to determine the correct sequence numbers for the LRECs that are to be deleted.

```
User:   ZUDFM OA*

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805C8
        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 D  U N N00 00000000 00328065
        000003 001080 I  N O U Y  E000000 00690794
        000004 001080 J  A C O B  S000000 00669412
        000005 001080 J  O N E S 00000000 00789441
        000006 001080 S  M I T H 00000000 00123456
        000007 001080 T  A Y L O  R000000 00123964
        000008 001080 T  H A Y E  R000000 00236348
        000009 001080 U  N D E R  H I L L 00894350
```

```
User:   ZUDFM OAX1/4/7

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805C8
        000001 001080 D  U N N00 00000000 00328065
        000002 001080 I  N O U Y  E000000 00690794
        000003 001080 J  O N E S 00000000 00789441
        000004 001080 S  M I T H 00000000 00123456
        000005 001080 T  H A Y E  R000000 00236348
        000006 001080 U  N D E R  H I L L 00894350
```

In the following examples, the LRECs in the second half of the subfile that is currently linked are deleted. A ZUDFM OA* command is necessary before the entries are deleted to determine the correct sequence numbers for the LRECs that are to be deleted.

```
User:   ZUDFM OA*

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805C8
        000001 001080 D  U N N00 00000000 00328065
        000002 001080 I  N O U Y  E000000 00690794
        000003 001080 J  O N E S 00000000 00789441
        000004 001080 S  M I T H 00000000 00123456
        000005 001080 T  H A Y E  R000000 00236348
        000006 001080 U  N D E R  H I L L 00894350
```

```
User:   ZUDFM OAX4/LAST

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805C8
        000001 001080 D  U N N00 00000000 00328065
        000002 001080 I  N O U Y  E000000 00690794
        000003 001080 J  O N E S 00000000 00789441
        000004 001080 T  H A Y E  R000000 00236348
```

In the following examples, two different ranges of LRECs are deleted from the subfile that is currently linked. A ZUDFM OA* command is necessary before the entries are deleted to determine the correct sequence numbers of the LRECs that are to be deleted.

```
User:   ZUDFM OA*

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805B0
        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 D  U N N00 00000000 00328065
        000003 001080 I  N O U Y  E000000 00690794
        000004 001080 J  A C O B  S000000 00669412
        000005 001080 J  O N E S 00000000 00789441
        000006 001080 S  M I T H 00000000 00123456
        000007 001080 T  A Y L O  R000000 00123964
        000008 001080 T  H A Y E  R000000 00236348
        000009 001080 U  N D E R  H I L L 00894350
```

```
User:   ZUDFM OAX2-4/7-LAST

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805B0
        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 J  O N E S 00000000 00789441
        000003 001080 S  M I T H 00000000 00123456
```

In the following examples, all the LRECs are deleted in the subfile that is currently linked. A ZUDFM OA* command is used to display the LREC numbers that are going to be deleted.

```
User:   ZUDFM OA*

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805B0
        000001 001080 B  R O C K  E R0000 00345874
        000002 001080 J  O N E S 00000000 00789441
        000003 001080 S  M I T H 00000000 00123456
```

```
User:   ZUDFM OAXALL

System: UDFM0001I          TPFDF LREC DISPLAY
        F.A. 000805B0
```

References

See the following for more information about linking to a subfile:

- “ZUDFM OA–Link to a Subfile Using the File Address” on page 62
- “ZUDFM OA*–Display LRECS in a Subfile” on page 64
- “ZUDFM OAI–File Access Using the ID or DSECT Name” on page 83.

ZUDFM OAZ–Convert Time-of-Day Clock Time into Standard Format

Use this command to display time and date information based on a 4- to 16-character hexadecimal time stamp.

An internal time-of-day (TOD) system clock records the time since 00:00:00 hours GMT on 1 January 1900. This TOD system clock gives the time as a hexadecimal value from 4 to 16 characters long.

Requirements and Restrictions

None.

Format

```
►►──ZUDFM── ──┬─OAZ/───┬──timevalue──────────────────────────────────────►◄
              └─TOD── ─┘
```

Note: You can use either the ZUDFM OAZ or the ZUDFM TOD command.

timevalue
is the time value, which is a 4-16 hexadecimal character time stamp.

Additional Information

- Trailing zeros on the time value are assumed.

Examples

The following example displays the specified hexadecimal clock value converted to readable form.

```
User:   ZUDFM OAZ/7C8B7654

System: UDFM0501I          TOD CLOCK VALUE DISPLAY
        CONVERSION OF TOD-CLOCK VALUE 7C8B765400000000 :
        ------------------------------------------------

        DATE:  FRI, 06 JUN 69

        TIME:  18.14.00
```

References

See *ESA/390 Principles of Operation* for more information about the internal TOD clock.

Appendix. CRUISE Examples

This appendix provides information (with examples) for creating a parameter table, setting up parameter table default values, and running a CRUISE function.

Setting Up CRUISE

To run a CRUISE function:

1. The following example shows the ZFCRU EXPAND command entered to create a parameter table index.

```
User:   ZFCRU EXPAND

System: CSMP0097I 16.04.13 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0009I 16.04.13 OK TABLE - EXPANSION - COMPLETED+
```

2. The following example shows the ZFCRU DEFAULT command entered to specify parameter table default values. This example shows that 20% of the entry control blocks (ECBs) allocated to the system to be used by CRUISE.

```
User:   ZFCRU DEFAULT ECB-20

System: CSMP0097I 16.04.26 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0000I 16.04.26 MODIFIED
        FCRU0000I 16.04.26 DEFAULT PARAMETER VALUES
          USABLE ECB START VALUE     20 PCT
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   ALL
          PRINTER ADDRESSES          000000  000000  000000 _
          NUMBER OF LOG MESSAGES     ALL
          SET POOL ADDRESSES IN USE  NO
          TAPE NAME                  BFA
          TARGET SYSTEM NAME         TPF
          RESTORE OPTION             REBUILD/NORELEASE
          WID/ADR IMBED REFERENCES   YES
        FCRU0000I 16.04.26 END OF DISPLAY
```

To capture the database referenced by the B075 file identifier (ID) to the BFA capture tape:

1. The following example shows the ZFCRU LOCK command entered to create parameter table CAPTURE_B075 and to display the initial values.

```
User:   ZFCRU LOCK-CAPTURE_B075

System: CSMP0097I 16.31.27 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0000I 16.31.27 TABLE LOCKED
        FCRU0000I 16.31.27 PARAMETER TABLE DISPLAY FOR CAPTURE_B075
          FUNCTION
          USABLE ECB START VALUE     20 PCT
          TAPE NAME                  BFA
          TARGET SYSTEM NAME         TPF _
          WID/ADR IMBED REFERENCES   YES
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   ALL
          NUMBER OF LOG MESSAGES     ALL
          RESTORE OPTION             REBUILD/NORELEASE
        FCRU0000I 16.31.27 END OF DISPLAY
```

2. The following example shows the ZFCRU SETUP command entered to modify parameter table CAPTURE_B075 to capture file ID B075.

```
User:   ZFCRU SETUP FUNCTION-CAPTURE REF-B075

System: CSMP0097I 16.32.18 CPU-B SS-BSS  SSU-HPN  IS-01 _
        FCRU0000I 16.32.18 MODIFIED
        FCRU0000I 16.32.18 PARAMETER TABLE DISPLAY FOR CAPTURE_B075
          FUNCTION                   CAP
          USABLE ECB START VALUE     20 PCT
          TAPE NAME                  BFA
          TARGET SYSTEM NAME         TPF
          WID/ADR IMBED REFERENCES   YES _
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   ALL
          NUMBER OF LOG MESSAGES     ALL
          REFERENCE ID               B075
        FCRU0000I 16.32.18 END OF DISPLAY
```

3. The following example shows the ZFCRU UNLOCK command entered to unlock parameter table CAPTURE_B075.

```
User:   ZFCRU UNLOCK-CAPTURE_B075

System: CSMP0097I 16.32.54 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0009I 16.32.54 OK TABLE - CAPTURE_B075 - UNLOCKED
```

4. The following example shows a BFA capture tape mounted.

```
User:   ZTMNT BFA 421 ao

System: CSMP0097I 16.36.34 CPU-B SS-BSS  SSU-HPN  IS-01
        COTM0310I 16.36.34 TMNT HPN    TAPE BFA MOUNTED ON DEVICE 421
                           VSN A00108 G  S0001 F38K  SL  NOBLK  NOCOMP
```

5. The following example shows the ZFCRU START command entered to verify parameter table CAPTURE_B075.

```
User:   ZFCRU START-CAPTURE_B075

System: CSMP0097I 16.37.20 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0021I 16.37.20 TABLE -CAPTURE_B075- ACTIVATED+
        CSMP0097I 16.37.20 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0000I 16.37.20 PARAMETER TABLE DISPLAY FOR CAPTURE_B075
          FUNCTION                   CAP
          USABLE ECB START VALUE     20 PCT
          TAPE NAME                  BFA
          TARGET SYSTEM NAME         TPF
          WID/ADR IMBED REFERENCES   YES _
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   ALL
          NUMBER OF LOG MESSAGES     ALL
          REFERENCE ID               B075
        FCRU0000I 16.37.20 REPEAT ENTRY TO START CRUISE
```

6. The following example shows the ZFCRU START command entered a second time to begin a capture. The state of the CRUISE function is running (in use).

```
User:   ZFCRU START-CAPTURE_B075

System: CSMP0097I 16.37.56 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0001I 16.37.56 TABLE -CAPTURE_B075- FUNCTION-CAP STARTED+
        CSMP0097I 16.37.56 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0001I 16.37.56 TABLE -CAPTURE_B075- FUNCTION-CAP STARTED+
        CSMP0097I 16.37.57 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0019I 16.37.57 00000059 RECORDS FOUND
                           00000059 RECORDS WRITTEN TO BFA+
        CSMP0097I 16.37.57 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0006I 16.37.57 TABLE -CAPTURE_B075- FUNCTION-CAP COMPLETED
                           00000000 ERRORS DETECTED+
        CSMP0097I 16.37.57 CPU-B SS-BSS  SSU-HPN  IS-01
        COTC0300A 16.37.57 TCLS HPN    REMOVE BFA FROM DEVICE 421
                           VSN A00111 G  S0001 F38K  SL  NOBLK  NOCOMP
```

To restore any data on a BFA restore tape:

1. The following example shows the ZFCRU LOCK command entered to create a parameter table and to display the initial values.

```
User:   ZFCRU LOCK-RESTORE

System: CSMP0097I 16.22.43 CPU-B SS-BSS  SSU-HPN  IS-01 _
        FCRU0000I 16.22.43 TABLE LOCKED
        FCRU0000I 16.22.43 PARAMETER TABLE DISPLAY FOR RESTORE
          FUNCTION
          USABLE ECB START VALUE     20 PCT
          TAPE NAME                  BFA
          TARGET SYSTEM NAME         TPF
          WID/ADR IMBED REFERENCES   YES _
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   ALL
          NUMBER OF LOG MESSAGES     ALL
          RESTORE OPTION             REBUILD/NORELEASE
        FCRU0000I 16.22.43 END OF DISPLAY
```

2. The following example shows the ZFCRU SETUP command entered to modify the parameter table.

```
User:   ZFCRU SETUP FUNCTION-RESTORE

System: CSMP0097I 16.23.23 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0000I 16.23.23 MODIFIED
        FCRU0000I 16.23.23 PARAMETER TABLE DISPLAY FOR RESTORE
          FUNCTION                   RST
          USABLE ECB START VALUE     20 PCT
          TAPE NAME                  BFA
          SET POOL ADDRESSES IN USE  NO
          WID/ADR IMBED REFERENCES   YES _
          BUILD STATISTICS           YES
          NUMBER OF PRINT MESSAGES   ALL
          NUMBER OF LOG MESSAGES     ALL
          RESTORE OPTION             REBUIL+
```

3. The following example shows the ZFCRU UNLOCK command entered to unlock the parameter table.

```
User:   ZFCRU UNLOCK

System: CSMP0097I 16.23.50 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0009I 16.23.50 OK TABLE - RESTORE - UNLOCKED
```

4. The following example shows a BFA restore tape mounted.

```
User:   ZTMNT BFA 421 ai

System: CSMP0097I 16.24.25 CPU-B SS-BSS  SSU-HPN  IS-01
        COTM0310I 16.24.25 TMNT HPN    TAPE BFA MOUNTED ON DEVICE 421
                           VSN A00108 G     S0001 F38K     SL  NOBLK
```

5. The following example shows the ZFCRU START command entered to verify the parameter table. The state of the CRUISE function is active.

```
User:   ZFCRU START-RESTORE

System: CSMP0097I 16.26.30 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0021I 16.26.30 TABLE -RESTORE- ACTIVATED+
        CSMP0097I 16.26.30 CPU-B SS-BSS  SSU-HPN  IS-01 _
        FCRU0000I 16.26.30 PARAMETER TABLE DISPLAY FOR RESTORE
          FUNCTION                 RST
          USABLE ECB START VALUE   20 PCT
          TAPE NAME                BFA
          SET POOL ADDRESSES IN USE  NO
          WID/ADR IMBED REFERENCES   YES
          BUILD STATISTICS           YES _
          NUMBER OF PRINT MESSAGES   ALL
          NUMBER OF LOG MESSAGES     ALL
          RESTORE OPTION             REBUILD/NORELEASE
        FCRU0000I 16.26.30 REPEAT ENTRY TO START CRUISE
```

6. The following example shows the ZFCRU START command entered a second time to begin restore. The state of the CRUISE function is running (in use).

```
User:   ZFCRU START-RESTORE

System: CSMP0097I 16.26.51 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0001I 16.26.51 TABLE -RESTORE- FUNCTION-RST STARTED+
        CSMP0097I 16.26.51 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0001I 16.26.51 TABLE -RESTORE- FUNCTION-RST STARTED+
        CSMP0097I 16.28.14 CPU-B SS-BSS  SSU-HPN  IS-01
        CYC00003I 16.28.14 POOL TYPE SDP DEVICE DEVA DIRECTORIES
              23 THRU       23 COUNTS     1590 IN USE+
        CSMP0097I 16.28.17 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0019I 16.28.17 00000059 RECORDS READ FROM TAPE BFA
                           00000059 RECORDS SELECTED
                           00000059 RECORDS RESTORED
                           00000000 ERRORS DETECTED+
        CSMP0097I 16.28.46 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0019I 16.28.46 FILE ADDRESS EXCHANGE
                           00000033 FROM REFERENCES PROCESSED
                           00000033 ADDRESSES EXCHANGED
                           00000026 FIX FILES RESTORED
                           00000000 ERRORS DETECTED+
        CSMP0097I 16.28.46 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0006I 16.28.46 TABLE -RESTORE- FUNCTION-RST COMPLETED
                           00000000 ERRORS DETECTED+
        CSMP0097I 16.28.47 CPU-B SS-BSS  SSU-HPN  IS-01
        COTC0087A 16.28.47 TCLS HPN    REMOVE BFA FROM DEVICE 421
                           VSN A00108  NOBLK
```

Index

File Number: S370/30XX-32
Program Number: 5706-196

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH31-0185-08