

TPF Database Facility



Utilities

Release 1

Contents

Part 1. Concepts 1

CRUISE 3

Why use CRUISE 3

When to use CRUISE 3

CRUISE States 4

Where to use CRUISE 4

Recoup 5

Why use TPFDF recoup 5

Multigroup chain chase processing 6

When to use TPFDF recoup 6

Where to use TPFDF recoup 6

Data collection 7

Why use data collection 7

When to use data collection 7

Where to use data collection 7

Maintenance and support 9

Why use maintenance and support 9

When to use maintenance and support 10

Where to use maintenance and support. 10

Part 2. Tasks 11

Run CRUISE 13

Prepare to create a parameter table 13

Create and modify a parameter table. 14

Run a capture function 16

Run a restore function 17

Run a verify function. 18

Run a pack function 18

Index 21

Part 1. Concepts

CRUISE

The TPFDF capture/restore utility, information and statistics environment (CRUISE) is a process of recovering and validating data, controlling pool space, and creating statistics.

CRUISE is based on file information that is stored in the TPFDF database definition (DBDEF). A parameter table contains the values that define how a CRUISE function is processed. A CRUISE function processes a parameter table when you use a ZFCRU command. A database administrator prepares, creates and maintains individual parameter tables. A parameter table index must be initialized and the global default values defined before a parameter table is created, named and set up. Use the ZFCRU HELP command for additional help.

See “Prepare to create a parameter table” on page 13 for more information about creating a parameter table index and setting up the default values. See “Create and modify a parameter table” on page 14 for more information about setting up a parameter table. See the following for more information about processing parameter tables:

- “Run a capture function” on page 16
- “Run a restore function” on page 17
- “Run a verify function” on page 18
- “Run a pack function” on page 18.

Why use CRUISE

Use CRUISE to minimize the exposure of software and hardware malfunctions, diagnose database corruption, and reduce DASD usage. CRUISE provides database administrators and system programmers with a versatile and easy-to-use method of database management. Some benefits include:

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

When to use CRUISE

Use CRUISE to selectively backup your database and recover data in the event of database corruption. CRUISE can also verify the integrity of your data structures and compact your databases to reduce DASD usage. Use a ZFCRU command to specify a CRUISE function. The following lists and describes the CRUISE functions:

| | |
|----------------|---|
| Capture | The data is captured to tape. |
| Pack | The data structures are packed according to the DBDEF macro. |
| Restore | The data is restored from tape. |
| Verify | The data structures are verified; broken chains and references that are not valid are reported. |

CRUISE States

Each CRUISE parameter table can be in one of several states at any time. Enter the ZFCRU DISPLAY command with the ALL parameter specified to display a list of parameter tables and their current state. The following lists and describes the CRUISE parameter table states:

| | |
|----------------|---|
| Active | A parameter table that is verified after you enter the initial ZFCRU START command. |
| Running | A parameter table that is processing a specified CRUISE function after you enter a second ZFCRU START command. |
| | Note: A CRUISE function in running state is in use. |
| Locked | A parameter table that is locked to the existing terminal address when you use the ZFCRU LOCK command. If a parameter table does not exist, it is created by being locked. |
| Paused | A parameter table in running state that is suspended while processing a CRUISE function successfully. Use the ZFCRU PAUSE command to put a parameter table in a paused state. |
| Stopped | A parameter table processing a CRUISE function that is immediately stopped when you use the ZFCRU STOP or ZFCRU ABORT command. |

Where to use CRUISE

Use CRUISE in a TPF system or Airline Control System (ALCS) environment.

Recoup

TPFDF recoup is a real-time database validation utility that runs as an extension to TPF or Airline Control System (ALCS) recoup phase 1. Recoup processing verifies fixed records and the chains of pool records attached to them. See *TPF Database Reference* for more information about TPF recoup.

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.

Why use TPFDF recoup

Use TPFDF recoup to help maintain database integrity and validate long-term records. Unused blocks can be returned to the TPF system or ALCS environment to be used again.

Recoup functionality is expanded on a TPF system or in an ALCS environment with TPFDF installed. The following TPFDF ZRECP commands are available in addition to the TPF ZRECP commands:

- ZRECP LOG
- ZRECP MPLOG
- ZRECP PRT
- ZRECP Q RESET
- ZRECP REPORT
- ZRECP STA.

TPFDF ZRECP commands provide the following support:

- A log file of the primary processor of phase 1 recoup processing, which contains information about files processed and the number of errors detected from the last recoup run.
- A chain chase status summary of TPFDF records on a processor.
- The ability to write information about broken chains and recoup processing for specific file IDs to a specified printer (for example, recoup retrieval error messages or TPFDF activity reports).
- A queue control indicator, which specifies whether a file address can pass references between subfiles during recoup.
- A specialized activity report displayed during recoup processing.

Note: The following shows an example of the message that can be displayed after each file ID is chain chased during TPFDF recoup.

```
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
```

- Additional reports that display the status and statistics of recoup processing.

Multigroup chain chase processing

Multigroup chain chase processing significantly reduces chain chase processing time. Multigroup chain chase processing allows you to chain chase as many as eight databases simultaneously on a processor. When chain chase processing has started for the last fixed record ordinal of a database, chain chase processing can start on another database. This is how multigroup chain chase processing is designed for both TPF and TPFDF recoup. Each database has its own timeout values, ID counters, and broken chain counters.

When to use TPFDF recoup

Use TPFDF recoup regularly to validate and recover usable long-term records, especially when running low on available pool blocks.

Where to use TPFDF recoup

Use TPFDF recoup on a TPF system or 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 installing the recoup user exit code.

Data collection

TPFDF data collection is an effective method to access information that relates to TPFDF system usage.

Why use data collection

Use data collection to help you manage your TPFDF resources by displaying the following statistics:

- File ID activity for as many as 30 data collections
- File ID activity for one or all file IDs in a data collection
- TPFDF macro call rates and counts
- TPF and ALCS macro rates and counts issued by TPFDF macros.

When to use data collection

Use TPFDF data collection when you need to analyze system performance, including I/O rates and macro usage. Data collection statistics help a system administrator make adjustments to the system to optimize performance.

Where to use data collection

Use data collection on a TPF system or in an ALCS environment.

Data collection is available on the current processor, or on all processors. Data collection on all processors is only available on a TPF system. You can also use data collection for any subsystem user (SSU). Data collection for all SSUs is only available on a TPF system.

Maintenance and support

The ZUDFM commands are a utility for TPFDF maintenance and support to help you manage and optimize the performance of the TPFDF product.

To simplify using the ZUDFM commands, you can use either the traditional or standard form. The following table lists these commands.

Table 1. ZUDFM commands

| 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 |

Why use maintenance and support

ZUDFM commands are used during setup to:

- Initialize files
- Initialize and set up a help utility
- 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.

When to use maintenance and support

Use the ZUDFM commands to perform database and system maintenance for the TPFDF product. Many of the ZUDFM commands must be linked to a subfile to ensure database and system integrity. ZUDFM commands must be linked to a subfile to add, delete, or replace an LREC, or to copy or pack a subfile. Linking is unique to a terminal address. A modified subfile must be re-linked before another operation is performed. A restriction table specifies the terminal addresses from which certain ZUDFM commands can be entered.

The ZUDFM commands also provide setup support for the TPFDF product. Use the ZUDFM commands to:

- Rebuild the DBDEF table to maintain TPFDF attributes
- Build the MLS database online
- Edit the restriction table.

Where to use maintenance and support

Use the ZUDFM commands for TPFDF maintenance and support on a TPF system or in an Airline Control System (ALCS) environment.

Part 2. Tasks

Run CRUISE

TPFDF capture/restore utility, information and statistics environment (CRUISE) uses the information stored in a parameter table to determine what processing occurs when running a CRUISE function. See the following list of procedures for information (with examples) about running a CRUISE function:

- “Prepare to create a parameter table”
- “Create and modify a parameter table” on page 14
- “Run a capture function” on page 16
- “Run a restore function” on page 17
- “Run a verify function” on page 18
- “Run a pack function” on page 18.

Use the ZFCRU HELP command for additional help.

Prepare to create a parameter table

Before you can create a CRUISE parameter table, you must create a parameter table index and set up the parameter table default values. The following steps show you how to create a parameter table index and set the default parameter table values:

1. Enter the ZFCRU EXPAND command to create the parameter table index; for example:

```
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. Enter the ZFCRU DEFAULT command to set the default parameter table values. This is required even if you do not need to make any changes to the parameter table values. Use the ZFCRU DISPLAY command with the DEFAULT parameter specified to view the existing default values.

These default values are used as the initial settings when creating parameter tables. These default values also override the defaults created when CRUISE was compiled with the c\$cruusr header file. See *TPFDF Installation and Customization* for more information about the c\$cruusr header file.

In the following example, default value USABLE ECB START VALUE is set to 20%. This means that up to 20% of the entry control blocks (ECBs) allocated to the system will 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

```

You are now ready to use CRUISE to create and set up a parameter table. See the following for information about parameter table processing:

- “Create and modify a parameter table”
- “Run a capture function” on page 16
- “Run a restore function” on page 17
- “Run a verify function” on page 18
- “Run a pack function” on page 18.

Create and modify a parameter table

The following steps show you how to create and modify a parameter table that runs a CRUISE function for the database indexed by record ID B075:

1. A parameter table must be locked before you can change the values. If a parameter table does not exist, it is created by being locked. Use the ZFCRU LOCK command to create and lock TABLE1; for example:

```

User:   ZFCRU LOCK-TABLE1

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 TABLE1
        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

```

The CRUISE function is in locked state.

Note: As with other parameter table commands, the ZFCRU LOCK command displays all the current parameter table settings.

2. Use the ZFCRU SETUP command to set the parameter table values. In the following examples, the FUNCTION and REFERENCE ID are set to capture TABLE1 to a BFA tape.

```

User:   ZFCRU SETUP FUNCTION-CAPTURE  TABLE1

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 TABLE1
          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

```

You can now process TABLE1 by unlocking the parameter table (see step 3) and running a CRUISE function. See the following for information about running a CRUISE function:

- “Run a capture function” on page 16
- “Run a restore function” on page 17
- “Run a verify function” on page 18
- “Run a pack function” on page 18.

To modify an existing parameter table after running a CRUISE function, use the ZFCRU LOCK (see step 1 on page 14) and ZFCRU SETUP (see step 2 on page 14) commands. The following example shows TABLE1 modified to restore data from a BFA tape.

```

User:   ZFCRU SETUP FUNCTION-RESTORE TABLE1

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 TABLE1
          FUNCTION                RST
          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

```

Note: In the previous examples, you did not have to set the TAPE NAME value because BFA was the default.

3. Before parameter table processing is started, use the ZFCRU UNLOCK command to unlock the parameter table; for example:

```

User:   ZFCRU UNLOCK-TABLE1

System: CSMP0097I 16.32.54 CPU-B SS-BSS  SSU-HPN  IS-01
        FCRU0009I 16.32.54 OK TABLE - TABLE1 - UNLOCKED

```

During unlock processing, the parameter table values are verified. If the values do not pass verification, an error message is issued and the table is remains unlocked. The CRUISE function is in stopped state.

Run a capture function

The following steps show you how to start the TABLE1 parameter table. Before you run a CRUISE capture function, set up a parameter table for the capture function and mount the BFA tape. See “Create and modify a parameter table” on page 14 for more information about setting up a parameter table.

1. 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
```

2. Starting a parameter table is a two-step process.
 - a. The first step verifies the parameter table values by using the ZFCRU START command; for example:

```
User:   ZFCRU START-TABLE1

System: CSMP0097I 16.37.20 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0021I 16.37.20 TABLE - TABLE1 - ACTIVATED
        CSMP0097I 16.37.20 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0000I 16.37.20 PARAMETER TABLE DISPLAY FOR TABLE1
              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
```

The CRUISE capture function is in active state.

- b. If the values pass verification, you can start the capture function by using the ZFCRU START command again; for example:

```
User:   ZFCRU START-TABLE1

System: CSMP0097I 16.37.56 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0001I 16.37.56 TABLE -TABLE1- FUNCTION-CAP STARTED
        CSMP0097I 16.37.56 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0001I 16.37.56 TABLE -TABLE1- 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 -TABLE1- 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
```

The CRUISE capture function is running (in use) state during parameter table processing. Now that you have successfully captured TABLE1 to the BFA tape, the CRUISE capture function is in stopped state.

Run a restore function

The following steps show you how to start the TABLE1 parameter table. Before you run a CRUISE restore function, set up a parameter table for the restore function and mount the BFA tape. See “Create and modify a parameter table” on page 14 for more information about setting up a parameter table.

1. 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
```

2. Starting a parameter table is a two-step process.
 - a. The first step verifies the parameter table values by using the ZFCRU START command; for example:

```
User:    ZFCRU START-TABLE1

System: CSMP0097I 16.26.30 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0021I 16.26.30 TABLE -TABLE1- ACTIVATED
        CSMP0097I 16.26.30 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0000I 16.26.30 PARAMETER TABLE DISPLAY FOR TABLE1
                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
```

The CRUISE restore function is in active state.

- b. If the values pass verification, you can start the restore function by using the ZFCRU START command again; for example:

```
User:    ZFCRU START-TABLE1

System: CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0001I 16.26.51 TABLE -TABLE1- FUNCTION-RST STARTED
        CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0001I 16.26.51 TABLE -TABLE1- 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 -TABLE1- 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
```

The CRUISE restore function is running (in use) state during parameter table processing. Now that you have successfully restored data from the BFA tape, the CRUISE restore function is in stopped state.

Run a verify function

The following steps show you how to start the TABLE1 parameter table. Before you can run a CRUISE verify function, set up a parameter table for the CRUISE verify function. See “Create and modify a parameter table” on page 14 for more information about setting up a parameter table.

1. Starting a parameter table is a two-step process. The first step verifies the parameter table values by using the ZFCRU START command; for example:

```
User: ZFCRU START-TABLE1

System: CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-VER STARTED
CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-VER STARTED
CSMP0097I 16.28.17 CPU-B SS-BSS SSU-HPN IS-01
FCRU0126I 02/06/12 16.28.17 CPUB 00003358 RECORDS FOUND
CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-VER COMPLETED
00000000 ERRORS DETECTED
CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-VER COMPLETED
00000000 ERRORS DETECTED
```

The CRUISE verify function is in active state.

2. If the values pass verification, you can start the verify function by using the ZFCRU START command again; for example:

```
User: ZFCRU START-TABLE1

System: CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-VER STARTED
CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-VER STARTED
CSMP0097I 16.28.17 CPU-B SS-BSS SSU-HPN IS-01
FCRU0126I 02/06/12 16.28.17 CPUB 00003358 RECORDS FOUND
CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-VER COMPLETED
00000000 ERRORS DETECTED
CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-VER COMPLETED
00000000 ERRORS DETECTED
```

The CRUISE verify function is in running (in use) state during parameter table processing. Now that you have successfully verified data, the CRUISE verify function is in stopped state.

Run a pack function

The following steps show you how to start the TABLE1 parameter table. Before you can run a CRUISE pack function, set up a parameter table for the CRUISE pack function. See “Create and modify a parameter table” on page 14 for more information about setting up a parameter table.

1. Starting a parameter table is a two-step process. The first step verifies the parameter table values by using the ZFCRU START command; for example:

```

User:   ZFCRU START-TABLE1

System: CSMP0097I 16.26.30 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-PAC STARTED
        CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-PAC STARTED
        CSMP0097I 16.28.17 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0128I 02/06/12 16.28.17 CPUB 00003358 RECORDS FOUND
                                00000026 FILES PACKED
        CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-PAC COMPLETED
                                00000000 ERRORS DETECTED
        CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-PAC COMPLETED
                                00000000 ERRORS DETECTED

```

The CRUISE pack function is in active state.

2. If the values pass verification, you can start the pack function by using the ZFCRU START command again; for example:

```

User:   ZFCRU START-TABLE1

System: CSMP0097I 16.26.30 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-PAC STARTED
        CSMP0097I 16.26.51 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0100I 02/06/12 16.26.51 TABLE -TABLE1- FUNCTION-PAC STARTED
        CSMP0097I 16.28.17 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0128I 02/06/12 16.28.17 CPUB 00003358 RECORDS FOUND
                                00000026 FILES PACKED
        CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-PAC COMPLETED
                                00000000 ERRORS DETECTED
        CSMP0097I 16.28.46 CPU-B SS-BSS SSU-HPN IS-01
        FCRU0132I 02/06/12 16.28.46 CPUB TABLE -TABLE1- FUNCTION-PAC COMPLETED
                                00000000 ERRORS DETECTED

```

The CRUISE pack function is in running (in use) state during parameter table processing. Now that you have successfully packed data, the CRUISE pack function is in stopped state.

Index

A

ALCS

- running TPDFDF recoup 5

C

Capture Restore Utility, Information and Statistics Environment (CRUISE)

- information about 3
- parameter table preparation
 - steps for 13
- run a CRUISE capture function
 - steps for 16
- run a CRUISE pack function
 - steps for 18
- run a CRUISE restore function
 - steps for 17
- run a CRUISE verify function
 - steps for 18

concepts

- what is a data collection 7
- what is CRUISE 3
- what is maintenance and support 9
- what is recoup 5

D

data collection

- information about 7

details

- parameter table preparation 13
- run a CRUISE capture function 16
- run a CRUISE pack function 18
- run a CRUISE restore function 17
- run a CRUISE verify function 18

E

examples

CRUISE set up

- create a parameter table 14
- modify a parameter table 14
- parameter table preparation
 - create a parameter table index 13
- run a CRUISE capture function
 - process a parameter table 16

examples (continued)

- run a CRUISE pack function
 - process a parameter table 18
- run a CRUISE restore function
 - process a parameter table 17
- run a CRUISE verify function
 - process a parameter table 18

M

maintenance and support

- information about 9

R

recoup

- information about 5
- running on ALCS 6
- running on TPF 5

T

tasks

- create a CRUISE parameter table
 - steps for 14
- modify a CRUISE parameter table
 - steps for 14
- parameter table preparation
 - steps for 13
- run a CRUISE capture function
 - steps for 16
- run a CRUISE pack function
 - steps for 18
- run a CRUISE restore function
 - steps for 17
- run a CRUISE verify function
 - steps for 18

TPF

- running TPDFDF recoup 5

Z

ZUDFM commands

- standard 9
- traditional 9