CA Datacom® Fast Restore

User Guide

Version 14.02

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA Datacom® Datadictionary™
- CA Datacom® Fast Restore
- CA Datacom® SQL
- CA IPC
- CA Common Services for z/OS

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

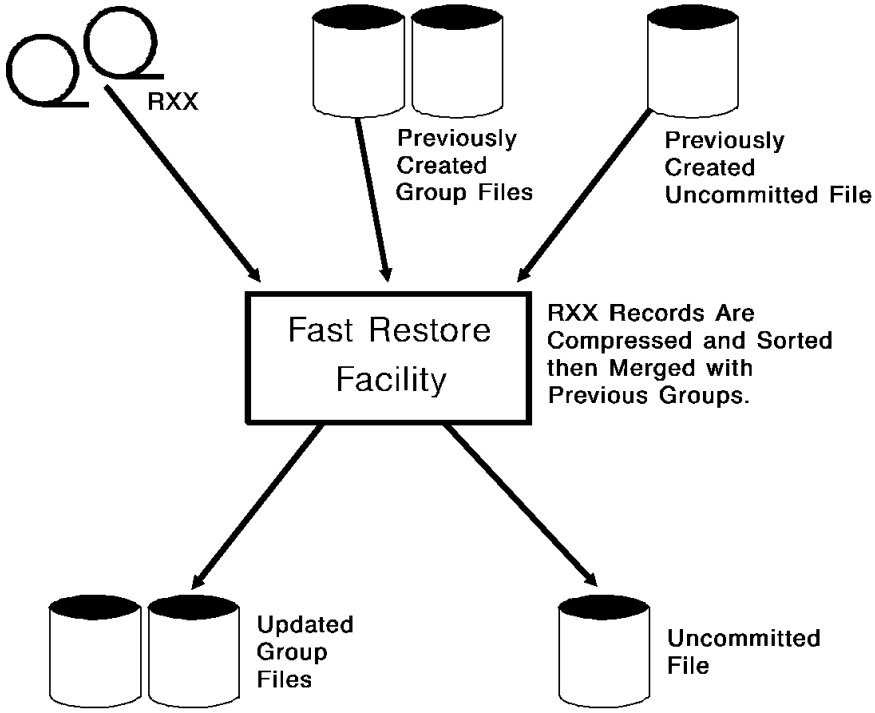
Contents

Chapter 1: Introduction

This guide provides specific information about using CA Datacom Fast Restore.

The primary purpose of CA Datacom Fast Restore is to reduce the time necessary to perform forward recovery. The conventional recovery utility must read a potentially large volume of recovery data and update affected data areas in an unpredictable sequence. CA Datacom Fast Restore creates change files sorted in performance-oriented sequence. After the DBUTLTY RECOVERY function processes these files, the time to recover is minimized.

CA Datacom Fast Restore also offers the convenience of less tape handling during the recovery process. The recovery information is compressed and stored in multiple user-defined files. Only the relevant files need be input to the DBUTLTY RECOVERY function. In many situations, these files can be disk files.



CA Datacom Fast Restore allows you to plan and prepare for possible recovery situations. New change files can be created and merged with previous change files on a regular schedule. CA Datacom Fast Restore eliminates time consuming processing during a crisis.

CA Datacom Fast Restore offers functionality that is not available with conventional recovery. Given a stable backup, recovery can be done to an arbitrary point of stability. The point is useful when recovery to a date and time is desired, or when the Recovery File (RXX) is incomplete (off-site recovery, for example). When this option is selected, all updates for uncommitted transactions are eliminated.

Reading Syntax Diagrams

Syntax diagrams are used to illustrate the format of statements and some basic language elements. Read syntax diagrams from left to right and top to bottom.

The following terminology, symbols, and concepts are used in syntax diagrams:

- Keywords appear in uppercase letters, for example, COMMAND or PARM. These words must be entered exactly as shown.
- Variables appear in italicized lowercase letters, for example, *variable*.
- Required keywords and variables appear on a main line.
- Optional keywords and variables appear below a main line.
- Default keywords and variables appear above a main line.
- Double arrowheads pointing to the right indicate the beginning of a statement.
- Double arrowheads pointing to each other indicate the end of a statement.
- Single arrowheads pointing to the right indicate a portion of a statement, or that the statement continues in another diagram.
- Punctuation marks or arithmetic symbols that are shown with a keyword or variable must be entered as part of the statement or command. Punctuation marks and arithmetic symbols can include the following:

,	comma	>	greater than symbol
.	period	<-	less than symbol
(	open parenthesis	=	equal sign
)	close parenthesis	¬	not sign
+	addition	-	subtraction
*	multiplication	/	division

Statement Without Parameters

The following is a diagram of a statement without parameters:

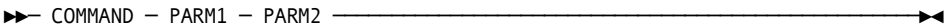
For this statement, write the following:

```
COMMAND
```

Statement with Required Parameters

Required parameters appear on the same horizontal line, the main path of the diagram, as the command or statement. The parameters are separated by one or more blanks.

The following is a diagram of a statement with required parameters:



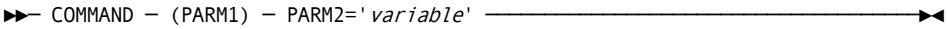
Write the following:

```
COMMAND PARM1 PARM2
```

Delimiters Around Parameters

Delimiters, such as parentheses, around parameters or clauses must be included.

The following is a diagram of a statement with delimiters around parameters:



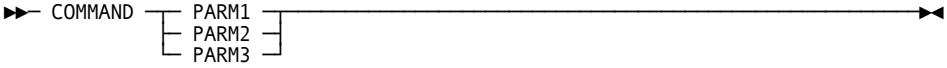
If the word *variable* is a valid entry, write the following:

```
COMMAND (PARM1) PARM2='variable'
```

Choice of Required Parameters

When you see a vertical list of parameters as shown in the following example, choose one of the parameters. This indicates that one entry is required, and only one of the displayed parameters is allowed in the statement.

The following is a diagram of a statement with a choice of required parameters:



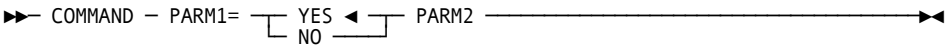
Choose one of the parameters from the vertical list, such as in the following examples:

```
COMMAND PARM1
COMMAND PARM2
COMMAND PARM3
```

Default Value for a Required Parameter

When a required parameter in a syntax diagram has a default value, the default value appears above the main line, and it indicates the value for the parameter if the command is not specified. If you specify the command, code the parameter and specify one of the displayed values.

The following is a diagram of a statement with a default value for a required parameter:



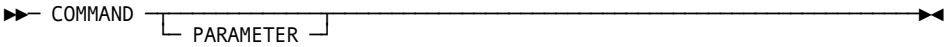
If you specify the command, write one of the following:

```
COMMAND PARM1=NO PARM2
COMMAND PARM1=YES PARM2
```

Optional Parameter

A single optional parameter appears below the horizontal line that marks the main path.

The following is a diagram of a statement with an optional parameter:



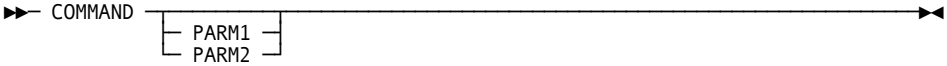
Choose (or not) to use the optional parameter, as shown in the following examples:

```
COMMAND
COMMAND PARAMETER
```

Choice of Optional Parameters

If you have a choice of more than one optional parameter, the parameters appear in a vertical list below the main path.

The following is a diagram of a statement with a choice of optional parameters:



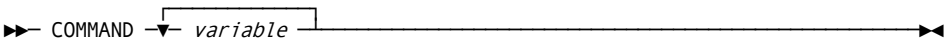
Choose any of the parameters from the vertical list, or you can write the statement without an optional parameter, such as in the following examples:

```
COMMAND
COMMAND PARM1
COMMAND PARM2
```

Repeatable Variable Parameter

In some statements, you can specify a single parameter more than once. A repeat symbol indicates that you can specify multiple parameters.

The following is a diagram of a statement with a repeatable variable parameter:



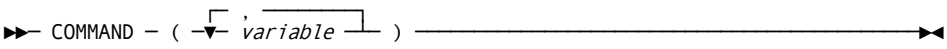
In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply, but it is also on the main path, which means that you are required to specify at least one entry. The repeat symbol indicates that you can specify a parameter more than once. Assume that you have three values named VALUEX, VALUEY, and VALUEZ for the variable. The following are some of the statements you can write:

```
COMMAND VALUEX
COMMAND VALUEX VALUEY
COMMAND VALUEX VALUEX VALUEZ
```

Separator with Repeatable Variable and Delimiter

If the repeat symbol contains punctuation such as a comma, separate multiple parameters with the punctuation. The following diagram includes the repeat symbol, a comma, and parentheses:

The following is a diagram of a statement with a separator with a repeatable variable and a delimiter:



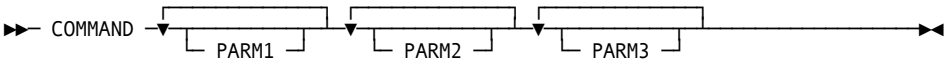
In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply. It is also on the main path, which means that you must specify at least one entry. The repeat symbol indicates that you can specify more than one variable and that you must separate the entries with commas. The parentheses indicate that the group of entries must be enclosed within parentheses. Assume that you have three values named VALUEA, VALUEB, and VALUEC for the variable.

The following are some of the statements you can write:

```
COMMAND (VALUEC)
COMMAND (VALUEB,VALUEC)
COMMAND (VALUEB,VALUEA)
COMMAND (VALUEA,VALUEB,VALUEC)
```

Optional Repeatable Parameters

The following diagram shows a list of parameters with the repeat symbol for optional repeatable parameters:



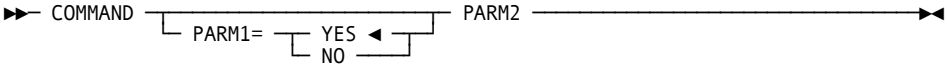
The following are some of the statements you can write:

```
COMMAND PARM1
COMMAND PARM1 PARM2 PARM3
COMMAND PARM1 PARM1 PARM3
```

Default Value for a Parameter

The placement of YES in the following diagram indicates that it is the default value for the parameter. If you do not include the parameter when you write the statement, the result is the same as if you had actually specified the parameter with the default value.

The following is a diagram of a statement with a default value for an optional parameter:

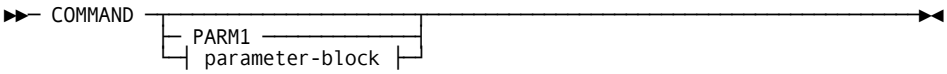


For this command, COMMAND PARM2 is the equivalent of COMMAND PARM1=YES PARM2.

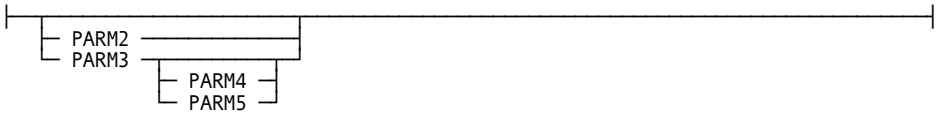
Variables Representing Several Parameters

In some syntax diagrams, a parameter-block represents a set of several parameters.

The following is a diagram of a statement with variables representing several parameters:



Expansion of parameter-block



The *parameter-block* can be displayed in a separate syntax diagram.

Choices you can make from this syntax diagram therefore include, but are not limited to, the following:

```
COMMAND PARM1
COMMAND PARM3
COMMAND PARM3 PARM4
```

Note: Before you can specify PARM4 or PARM5 in this command, specify PARM3.

Listing Libraries for Products in JCL

Guidelines to assist you in preparing your JCL are provided in this manual. The sample code provided in this document is intended for use as a reference aid only and no warranty of any kind is made as to completeness or correctness for your specific installation.

Samples for JCL and programs are provided in the install library.

- In z/OS, the default name for this library is CABDSAMP. CABDSAMP is an SMP/E target library so no changes to any member in this library can be saved in this PDS. You can copy the member to a different library to make any changes that you want to any member into a PDS that is not under SMP/E control. If you do make a change in the CABDSAMP library, be aware that any future SMP/E maintenance can write over your changes at in time.
- In z/VSE, sample PROCs allow you to use parameter substitution. You can copy and modify these samples for your specific requirements.

Any JOB statements must be coded to your site standards and specifications. All data set names and library names must be specified with the correct names for the installation at your site. In many examples, a REGION= or SIZE= parameter is displayed in an EXEC statement. The value displayed can be adequate in most instances, but you can adjust the value to your specific needs.

The libraries listed for searching must include the following in the order shown:

1. User libraries defined for specially assembled and linked tables, such as DBMSTLST, DBSIDPR, DDSRTLM, or User Requirements Tables
2. CA Datacom base libraries: CA Datacom/DB, CA Datacom Datadictionary, SQL
3. CA IPC libraries
4. CA Common Services for z/OS or CA CIS (Common Infrastructure Services) for z/VSE base libraries
5. Libraries for additional products

Chapter 2: Using CA Datacom Fast Restore

CA Datacom Fast Restore allows you to speed the forward recovery process thus reducing the time to recover from DASD failures. The process decreases input I/O and decreases I/O to the data area. The functions in CA Datacom Fast Restore allow you to consolidate, compress, and split Recovery File (RXX) data into highly efficient change units. You can also use CA Datacom Fast Restore to help ensure transaction level integrity on recovered data areas and make the recovery process more manageable.

When to Use

CA Datacom Fast Restore can be used in two ways:

- When a specific recovery is required
- As a regularly scheduled process

When a recovery is necessary, all relevant recovery files can be input to the SPLIT function. The SPLIT function creates CA Datacom Fast Restore files to be input to the DBUTLTY RECOVERY function. This process can be faster than the conventional recovery process. If there has been light update activity to the data areas to be recovered, then the overhead of the SPLIT function is not justified. However, if there has been heavy update activity, the SPLIT function overhead is more than offset by the increased I/O efficiency of CA Datacom Fast Restore recovery.

The SPLIT function can be used when recovery to an arbitrary point in time is needed. Unlike the conventional recovery function of DBUTLTY, the SPLIT function can be directed to remove uncommitted updates. The result is a recovery to transaction level stability. All transactions uncommitted at the date-time specified are rolled back.

Uncommitted updates could be a result of CA Datacom Fast Restore ignoring forced checkpoints or a restoration cycle that does not encompass the full length of time that a job was originally running.

The advantage of using CA Datacom Fast Restore as a part of normal operations procedure is that the site maintains the ability to respond quickly to a recovery situation. In a time of crisis, much of the work for recovery has been done, allowing systems to be restored in a way that minimizes down time.

For example, run the SPLIT function with TXUNDO=YES and provide a file for uncommitted updates. The group files produced, with the last backups, represent a stable state of the data areas specified. If the backup is restored and the group files are input to the DBUTLTY RECOVERY function, only committed updates are processed.

Once the SPLIT function has created group files and an uncommitted file, these files can be input to the next SPLIT execution. The updates from the previous group files are merged with the new committed updates from the Recovery File (RXX). The updates from the uncommitted file are merged if a commit is found on the new RXX. If an abort is found, the uncommitted updates are discarded. If no commit or abort is found, the updates are carried forward to the new uncommitted file.

This process can be repeated until new full backups are taken or new data is loaded or a reorganization occurs.

Use CA Datacom Fast Restore when you want to achieve the following:

- Have faster recovery by doing less I/O more efficiently.
- Use group files as incremental backups.
- Perform fewer backups on a regular basis.
- Have less tape handling for recovery (the group files can be placed on disk).
- Perform recovery to an arbitrary point of stability.
- Maintain a remote recovery site for off-site recovery.

Defining Groups

The smallest unit of recovery is the data area. Each CA Datacom Fast Restore group is composed of one or more data areas. A data area can be assigned to any group without restriction. Some examples of possible groupings are:

- One area per group
- All areas in a database
- All areas on a device
- All areas in an application

The more critical data areas can be grouped to minimize downtime, while the less critical can be assigned to a catchall default group.

How It Works

CA Datacom Fast Restore takes a set of Recovery Files (RXXs) or a previously created file or files as input. It condenses the log information to the bare essentials, sorts it, and splits it into separate files by data area or user-defined groups of areas. These files can then serve as input to the RECOVERY function of DBUTLTY. In other words, CA Datacom Fast Restore is a sort/merge facility for database updates. It sorts and compresses current updates and merges them with older updates from previous CA Datacom Fast Restore executions.

Condensing the Log Information and Records

Not all Recovery File (RXX) records are necessary for forward recovery, such as the control information used by RESTART, the Multi-User Facility, and user programs. CA Datacom Fast Restore discards these log records. Only records related to the update of CA Datacom/DB data areas are retained, such as ADDIT, UPDAT, and DELET records.

The next step involves condensing the records themselves. Each log record contains nonessential information. The entire "before" record is dropped because we are only concerned with forward recovery.

Consolidating Multiple Updates

CA Datacom Fast Restore further eliminates unnecessary data by consolidating multiple updates to the same data record into a single log record that generate a single update to the data area when processed by the RECOVERY function.

Finally, only the bytes of the data record that actually have changed are stored. Because of this high degree of compression, many of these files can be disk files. Indeed, they can function as incremental backups of the data areas. This single, consolidated update has no relation to CA Datacom/DB fields or data elements.

Saving I/O

Much input I/O is saved when the log information is condensed to this degree. The conventional RECOVERY function can be required to read thousands of records across multiple volumes of tape to process relatively few updates. With the new condensed and consolidated log file, the RECOVERY function reads only what is necessary and, in many cases, can use the more efficient disk I/O.

Sorting the Log Records

This process also minimizes I/O to the data area by sorting the log records into physical block number sequence. The RECOVERY function takes advantage of any locality of reference among the logged updates. Coupled with the pipeline and the multiple block write strategy can result in substantial I/O savings.

Reduced Tape Handling

CA Datacom Fast Restore provides more ease of use by reducing tape handling. This is especially valuable in a time of crisis. If CA Datacom Fast Restore routinely processes the Recovery File tapes, there is minimum input to recovery—only the CA Datacom Fast Restore file and any unprocessed Recovery File (RXX).

Assuring Data Integrity

Integrity can also be assured to a greater degree than available with the conventional RECOVERY function through an option to bypass uncommitted updates. Bypassing uncommitted updates allows a site to recover to a point of transaction level stability, even though all Recovery File tapes may not be available. However, the uncommitted updates need not be lost. They can be written to a special file which can be used during the next CA Datacom Fast Restore run. This capability can be useful for off-site recovery.

Operational Characteristics

CA Datacom Fast Restore takes in one or more Recovery File (RXX) tapes. CA Datacom Fast Restore also optionally accepts previously created CA Datacom Fast Restore files and an incomplete transaction file as input. The output of the process is a new CA Datacom Fast Restore file or files and, optionally, a new file of incomplete transactions.

All data sets created by CA Datacom Fast Restore are undefined, sequential data sets. Because the data sets created in one execution can be input in the next, using generation data groups (GDGs) can simplify JCL in z/OS environments.

Do not specify the same file for both group input and output.

No editing is done on the database record as in conventional recovery because there is no before record. Running recovery with CA Datacom Fast Restore input is similar to laying down an incremental backup.

Non-URI data areas are not supported.

The RECOVERY function accepts only one type of input per execution: an RXX file or CA Datacom Fast Restore files. Separate steps must be used when you have both a CA Datacom Fast Restore file and an RXX file to process. The CA Datacom Fast Restore file must be in the first step and followed by the RXX file in the second step. No changes are needed for the RECOVERY function control statement or the JCL. The type of processing is recognized from the type of input. When the RXX input is a CA Datacom Fast Restore file, any RECJOB statement is ignored.

Impact of "Safe" RECID Backups

When a backup with RECID=YES of a data area is done, updates saved on existing CA Datacom Fast Restore files are no longer needed. The updates are reflected in the backup and, therefore, the information is redundant. To avoid this redundancy, specify the date and time of the backup on the area SPLITDEF statement using the DATETIME= parameter. All completed updates (on both the CA Datacom Fast Restore file and the RXX) prior to the backup are dropped during the next SPLIT function execution. Updates contained in the backup are not written to the new CA Datacom Fast Restore file.

If a noncontinuous operation backup (SEQ=PHYSICAL,UPDATE=YES) was done, the DATETIME= value to be used should be taken from the CA Datacom/DB Utility Function History Report (REPORT TYPE=H). For more information, see to the *CA Datacom/DB DBUTLTY Reference Guide*.

During a continuous operation backup (SEQ=PHYSICAL,UPDATE=NO), the area can be open and updates can occur. Buffers can be in memory throughout the backup. To remove old updates, run the ACCESS or LOCK OPTION=MOVER functions to cause memory buffers to be written against the Multi-User Facility when the database is open. Use the date and time of the completion of either of these utility functions for the DATETIME= value.

Impact of Reorganizing a Data Area

When a data area is reorganized or when new data is loaded, existing CA Datacom Fast Restore files are no longer valid. The same is true for RXX records logged before the load. In this case, the area SPLITDEF DATETIME= parameter must be specified with the date and time of the load. This causes all invalid updates to be dropped from the new CA Datacom Fast Restore file during the SPLIT function execution.

Impact of Dropping a Table from a Group

Dropping a table from a group, such as moving the table from one area to another, is a barrier to recovery for that table. The updates on the CA Datacom Fast Restore file are dropped during the next cycle. The updates do not follow the table to the new group and the RECOVERY function ignores them.

Cautions on Using DBUTLTY RECOVERY

When using the DBUTLTY RECOVERY function to perform backward recovery or to recover forward to a point in time that does not include all updates to a data area, CA Datacom Fast Restore files and RXX records could be invalidated. To avoid erroneous updates, take a backup of the affected data areas after the recovery operation and before normal processing is resumed. Also, specify the date and time of the backup on the area SPLITDEF statement using the DATETIME= parameter. This assures that all logged updates correspond to the correct version of the data.

Similarly, if a data area is updated with logging temporarily turned off (as is sometimes done to speed batch processing), CA Datacom Fast Restore has been defeated. The effect is the same as a reorganizing LOAD. Take a new backup and discard the affected group files.

How to Use

The MUF must be enabled before you can use CA Datacom Fast Restore. To perform CA Datacom Fast Restore, supply the following commands to DBUTLTY:

- SPLITHDR
- SPLITDEF (optional)
- SPLITIN (optional)
- SPLITOUT (optional)
- SPLIT

The main function is SPLIT, which drives the utility that performs the action. The other functions must precede SPLIT when used.

The first-time CA Datacom Fast Restore is run, there is no group file input because the input file is produced as output from a previous SPLIT run. Therefore, specify IN=NO in the group header statement or supply properly formatted empty sequential data sets.

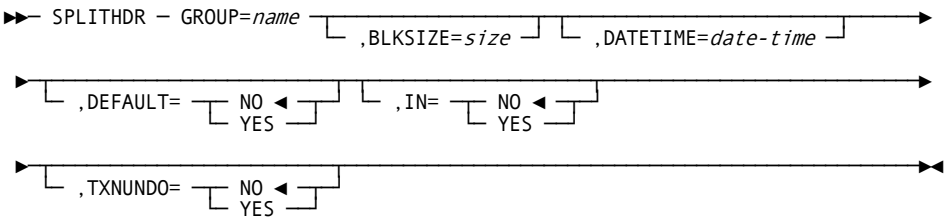
The SPLITHDR and SPLITDEF functions are for grouping. If you accept the default or specify DEFAULT=NO in the SPLITHDR statement, provide one or more related (same group name) SPLITDEF functions. If you specify DEFAULT=YES in the SPLITHDR function, you are not required to provide a SPLITDEF statement. You can supply up to 400 SPLITHDR or SPLITDEF statements in one execution of the function.

The SPLITIN function must be present if input of uncommitted transactions from a previous SPLIT run is processed. Omitting the SPLITIN function indicates no uncommitted input is necessary.

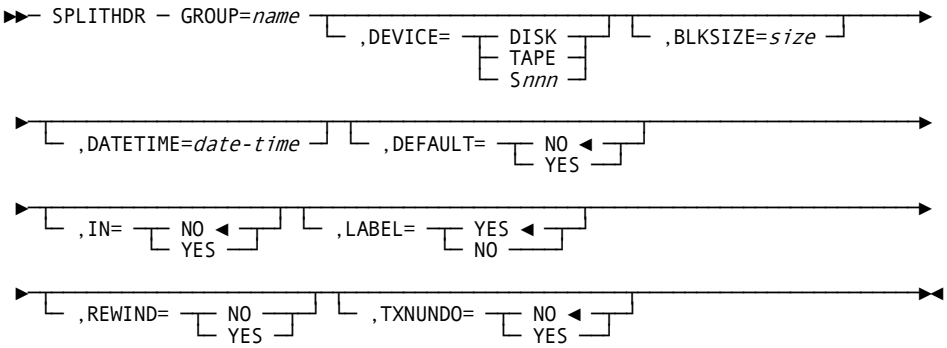
The SPLITOUT function must be present if output of incomplete transactions in this run is done. Omitting this function indicates that no incomplete transactions are split out to a separate file for later input to another SPLIT process. Uncommitted requests are discarded if TXNUNDO=YES is specified for the group in the SPLITHDR command.

SPLITHDR Command

SPLITHDR Command Syntax for z/OS Environment



SPLITHDR Command Syntax for z/VSE Environment



Note: Multiple groups can be specified in one JCL execution.

Command

SPLITHDR

Defines control information for this execution of the utility.

Required Keywords

,BLKSIZE=

Specifies the block size in bytes of the output group files and the SPLITOUT file. This block size overrides any specification in the JCL or the VTOC.

Valid Entries:

512 to 32767

Note: Using CA Sort in z/OS, the largest possible record you can use is 32700 (the largest record you can use with other sorts is unknown).

z/OS Default Value:

4096

VSE DISK Default Value:

4096

VSE TAPE Default Value:

32766

,DATETIME=

Specifies the date-time limit for RXX records. Only the records logged before the specified date/time are processed. All RXX records occurring after the date-time specified are ignored. This parameter is used to recover to a specific point in time.

Valid Entries:

Date and time in the format *ccyymmddhhmmss*

Default Value:

No date-time restriction

,DEFAULT=

Specifies the group to use as the default for all areas/bases not provided in any other group. You can specify YES in only one SPLITHDR statement in each execution of the SPLIT function, but you are not required to specify any default.

Valid Entries:

NO or YES

Default Value:

NO

,DEVICE=

(z/VSE) Specifies the device type for the input (if present) and output files for this group. If this parameter is provided in z/OS JCL, it is edited.

For those files residing on different tapes, you can assign a different device for each tape. Use DEVICE=S*nnn* to assign specific devices to files, where S is a constant and *nnn* is the SYS*nnn* assigned for the device. In the case of the SPLITHDR command with an IN=YES keyword, the S*nnn* value identifies the output device and CA Datacom Fast Restore increments your value by one to assign as the input device.

DEVICE=TAPE defaults to SYS005 for input and SYS010 for output.

Valid Entries:

DISK, TAPE, or S*nnn*

Default Value:

(No default)

GROUP=

Specifies the group name to relate this SPLITHDR to any SPLITDEF and to start the ddname or DLBL. The input file name, if IN=YES is specified, is this group name suffixed with an I. The output file name is this group name suffixed with an O. A group name on a SPLITHDR cannot be repeated.

Valid Entries:

1 to 6 characters

Default Value:

(No default)

Optional Keywords

,IN=

Specifies if an input file for this group is present. The input must be from a previous SPLIT run.

Valid Entries:

NO or YES

Default Value:

NO

,LABEL=

(z/VSE) Specifies whether the file is labeled. No action is taken if the device is disk. If this parameter is provided in z/OS JCL, it is edited.

Valid Entries:

NO or YES

Default Value:

YES

,REWIND=

(z/VSE) Specifies if the tape is to be rewound before and after open (YES) or not rewound before open nor after close (NO).

Valid Entries:

NO or YES

Default Value:

Rewind before open and with unload after close.

,TXNUNDO=

Specifies whether transaction boundaries are to be honored. This means that in splitting and merging, records for incomplete transactions are not to be written to the group output file, but instead are written to the jobs incomplete transaction file (for input to the next SPLIT process). NO indicates that all group records are to be written to the group output file regardless of transaction boundaries.

Note: If a SPLITOUT command is not provided, these uncommitted transaction records will be discarded.

Valid Entries:

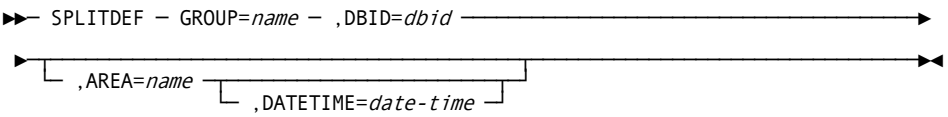
NO or YES

Default Value:

NO

SPLITDEF Command

SPLITDEF Command Syntax



Command

SPLITDEF

Identifies the group name, database, and areas that are to be included in this execution of the utility.

Required Keywords

,DBID=

Specifies the database ID of the database whose records are to be part of this group.

Valid Entries:

Valid database ID

Default Value:

(No default)

GROUP=

Specifies the group name to relate this SPLITDEF to the SPLITHDR.

Valid Entries:

1 to 6 characters

Default Value:

(No default)

Optional Keywords

,AREA=

Specifies the DATACOM-NAME of the area in the database to be part of this group. The area and database combination cannot be repeated in another SPLITDEF command in this execution of the SPLIT function. If this parameter is not specified, all areas in the database are processed.

Within one DBID, you cannot specify one SPLITDEF command with an AREA= parameter specified and another SPLITDEF command without an AREA= parameter in the same execution of the SPLIT function. You must specify an area before you can specify the DATETIME= parameter.

Valid Entries:

3-character AREA occurrence DATACOM-NAME

Default Value:

All areas in the database

,DATETIME=

Specifies the date and time of a backup. All records that were logged to the RXX file prior to this time are discarded during processing.

You must specify an area before you can specify this parameter.

Valid Entries:

Date and time in the format *ccyymmddhhmmss*

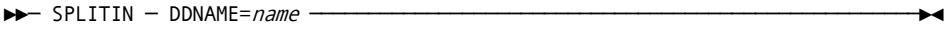
Default Value:

No date-time restriction

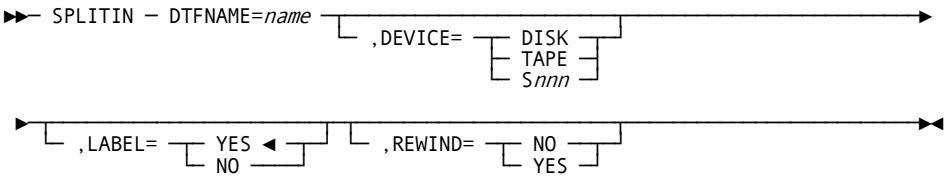
SPLITIN Command

If you omit the optional SPLITIN command, you are indicating that no prior incomplete transactions are processed.

SPLITIN Command Syntax for z/OS Environment



SPLITIN Command Syntax for z/VSE Environment



Command

SPLITIN

Specifies action for an input file.

Required Keywords

DDNAME=

Specifies the ddname of the input containing incomplete transactions.

A DDNAME is not acceptable for sequential input or output files if it is a name reserved for a CA Datacom area. Names with the following patterns are therefore not acceptable for DDNAME=:

- 3-byte names that end with XX, meaning they are reserved as either current or future CA Datacom control areas.
- 6-byte names that end with what could be a database ID from 001 through 999.
- 7-byte names that end with what could be a database ID from 1000 through 9999.

The DDNAME= value is verified for acceptability to protect you from unintentionally causing data corruption. The DDNAME check is the default but optional. You can prevent the DDNAME check by using a DBSIDPR parameter (DBUTLTY_EDIT_DATA_SET=) for individual MUF environments. However, we recommend that you allow the DDNAME check.

The data corruption risk involves not the DDNAME itself but the content of the data set. For example, suppose that you used the CXX DDNAME as the output of a backup. You then copied the CXX DD statement and changed the DDNAME of the copy to be acceptable, avoiding the DDNAME= error. The backup would, however, then overlay the CXX data set, which is not the intent of a backup.

If you specify an unacceptable name for DDNAME=, message DB10059E is generated. For more information about DB10059, see the *CA Datacom/DB Message Reference Guide.*

Note: We recommend that you allow DDNAME= check protection. You can, however, disable DDNAME= protection. To disable protection, assemble the DBSIDPR module used for this CA Datacom environment and specify NONE for the DBUTLTY_EDIT_DATA_SET= parameter. The default is DBUTLTY_EDIT_DATA_SET=FULL_1, which allows DDNAME= protection. For more information about DBSIDPR and DBUTLTY_EDIT_DATA_SET=, see the *CA Datacom/DB Database and System Administrator Guide.*

Valid Entries:

Valid ddname

Default Value:

(No default)

,DEVICE=

(z/VSE) Specifies the device type for the output file for this group. If this parameter is provided in z/OS JCL, it is edited.

For those files residing on different tapes, you can assign a different device for each tape. Use DEVICE=Snnn to assign specific devices to files, where S is a constant and nnn is the SYSnnn assigned for the device.

DEVICE=TAPE defaults to SYS005 for input and SYS010 for output.

Valid Entries:

DISK, TAPE, or Snnn

Default Value:

(No default)

,DTFNAME=

(z/VSE) Specifies the DTF name to be used for the output of incomplete transactions.

Valid Entries:

Valid DTF name

Default Value:

(No default)

Optional Keywords

,LABEL=

(z/VSE) Specifies whether the file is labeled. No action is taken if the device is disk. If this parameter is provided in z/OS JCL, it is edited.

Valid Entries:

NO or YES

Default Value:

YES

,REWIND=

(z/VSE) Specifies if the tape is to be rewound before and after open (YES) or not rewound before open nor after close (NO).

Valid Entries:

NO or YES

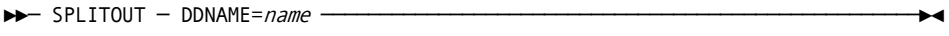
Default Value:

Rewind before open and with unload after close.

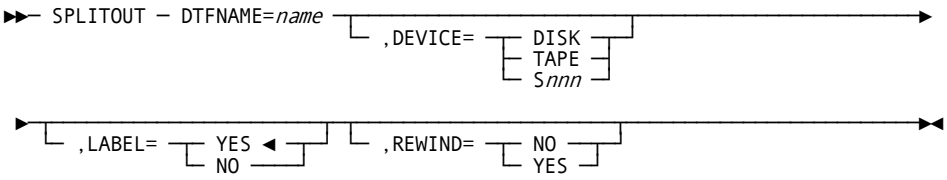
SPLITOUT Command

If you omit the optional SPLITOUT command, you are indicating that no incomplete transactions are to be saved and that any found will be discarded. Incomplete transactions can exist only if at least one SPLITHDR command has TXNUNDO=YES specified. You must specify at least one SPLITHDR command with TXNUNDO=YES to recognize incomplete transactions to write to an output file.

SPLITOUT Command Syntax for z/OS Environment



SPLITOUT Command Syntax for z/VSE Environment



Command

SPLITOUT

Specifies action for an output file.

Required Keywords

,DDNAME=

Specifies the ddname to be used for the output of incomplete transactions.

A DDNAME is not acceptable for sequential input or output files if it is a name reserved for a CA Datacom area. Names with the following patterns are therefore not acceptable for DDNAME=:

- 3-byte names that end with XX, meaning they are reserved as either current or future CA Datacom control areas.
- 6-byte names that end with what could be a database ID from 001 through 999.
- 7-byte names that end with what could be a database ID from 1000 through 9999.

The DDNAME= value is verified for acceptability to protect you from unintentionally causing data corruption. The DDNAME check is the default but optional. You can prevent the DDNAME check by using a DBSIDPR parameter (DBUTLTY_EDIT_DATA_SET=) for individual MUF environments. However, we recommend that you allow the DDNAME check.

The data corruption risk involves not the DDNAME itself but the content of the data set. For example, suppose that you used the CXX DDNAME as the output of a backup. You then copied the CXX DD statement and changed the DDNAME of the copy to be acceptable, avoiding the DDNAME= error. The backup would, however, then overlay the CXX data set, which is not the intent of a backup.

If you specify an unacceptable name for DDNAME=, message DB10059E is generated. For more information about DB10059, see the *CA Datacom/DB Message Reference Guide.*

Note: We recommend that you allow DDNAME= check protection. You can, however, disable DDNAME= protection. To disable protection, assemble the DBSIDPR module used for this CA Datacom environment and specify NONE for the DBUTLTY_EDIT_DATA_SET= parameter. The default is DBUTLTY_EDIT_DATA_SET=FULL_1, which allows DDNAME= protection. For more information about DBSIDPR and DBUTLTY_EDIT_DATA_SET=, see the *CA Datacom/DB Database and System Administrator Guide.*

Valid Entries:

Valid ddname

Default Value:

(No default)

,DEVICE=

(z/VSE) Specifies the device type for the output file for this group. If this parameter is provided in z/OS JCL, it is edited.

For those files residing on different tapes, you may assign a different device for each tape. Use DEVICE=Snnn to assign specific devices to files, where S is a constant and nnn is the SYSnnn assigned for the device.

DEVICE=TAPE defaults to SYS005 for input and SYS010 for output.

Valid Entries:

DISK, TAPE, or Snnn

Default Value:

(No default)

,DTFNAME=

(z/VSE) Specifies the DTF name to be used for the output of incomplete transactions.

Valid Entries:

Valid DTF name

Default Value:

(No default)

Optional Keywords

,LABEL=

(z/VSE) Specifies whether the file is labeled. No action is taken if the device is disk. If this parameter is provided in z/OS JCL, it is edited.

Valid Entries:

NO or YES

Default Value:

YES

,REWIND=

(z/VSE) Specifies if the tape is to be rewound before and after open (YES) or not rewound before open nor after close (NO).

Valid Entries:

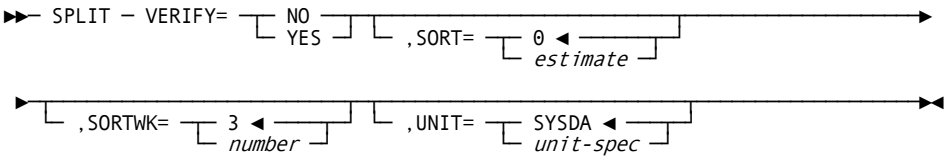
NO or YES

Default Value:

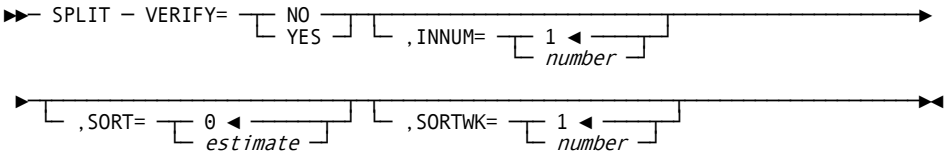
Rewind before open and with unload after close.

SPLIT Command

SPLIT Command Syntax for z/OS Environment



SPLIT Command Syntax for z/VSE Environment



Command

SPLIT

Initiates execution of the function.

Required Keywords

VERIFY=

Specifies whether all input records are to be accounted for in the output. Use YES to specify that any recovery record not accounted for is to cause a failure of this function. Use NO to specify that records not accounted for are not desired and will be ignored. This parameter is ignored if a default group is specified in the SPLITHDR command.

Valid Entries:

NO or YES

Default Value:

(No default)

Optional Keywords

,INNUM=

(z/VSE) When the Recovery File (RXX) input includes multiple files, the INNUM= keyword must be used to specify the number of input RXX files. When the end-of-file is detected, DBUTLTY checks the INNUM= keyword value to see if additional files are required. If they are, the current RXX file is closed and unloaded and an OPEN issued for the next file. This OPEN causes a message, 0P08t INTERV REQ, indicating the next file needs to be mounted. When the new file is opened, you can receive a message: 4132D ERROR IN FILE ID. If the correct tape is mounted, reply IGNORE.

Note: When CA Dynam/D for z/VSE is active, the processing option U (for unload) must be specified to allow CA Datacom Fast Restore to work properly. Errors result if U is not specified, because without U specified, the tape is rewound and the next open causes the same file to be read a second time.

Valid Entries:

1 to 3 digits

Default Value:

1

,SORT=

Specifies a record estimate for the sort. This is used to select sorting strategy and possibly in dynamic allocation of work areas.

Note: Nine digits are allowed but only provide more than 8 digits if the sort you are using allows 9 digits.

Valid Entries:

1 to 9 digits

Default Value:

0

,SORTWK=

Specifies the number of sort work areas for sorting.

Valid Entries:

1 digit

Default Value:

3

,UNIT=

Specifies the unit specification for the sort work areas.

Valid Entries:

1 to 8 characters

Default Value:

SYSDA

Example JCL and Reports

The following are examples of the format for the JCL to submit CA Datacom Fast Restore utility and the reports produced.

Note: Use the following as a guide to prepare your JCL. The JCL statements are for example, only. Lowercase letters in a statement indicate a value you must supply. Code all statements to your site and installation standards.

Example z/OS JCL

The following is an example of the format for the JCL to submit CA Datacom Fast Restore utility. Generation data groups (GDGs) can be used, but are not shown in this example.

```
 //jobname                 See the note above and Listing Libraries for Products
in JCL (see page 13).
 //DBUTLTY  EXEC PGM=DBUTLTY,REGION=2M
 //STEPLIB                 See the note above and Listing Libraries for Products
in JCL (see page 13).
 //RXX      DD DSN=rxx.data.set,DISP=SHR           Recovery data set
 //GROUP1I  DD DSN= ...                            Group input data set
 //GROUP1O  DD DSN= ...                            Group output data set
 //GROUP2I  DD DSN= ...                            Group input data set
 //GROUP2O  DD DSN= ...                            Group output data set
 //GROUP3I  DD DSN= ...                            Group input data set
 //GROUP3O  DD DSN= ...                            Group output data set
 //GROUP4I  DD DSN= ...                            Group input data set
 //GROUP4O  DD DSN= ...                            Group output data set
 //GROUP5I  DD DSN= ...                            Group input data set
 //GROUP5O  DD DSN= ...                            Group output data set
 //GROUP6I  DD DSN= ...                            Group input data set
 //GROUP6O  DD DSN= ...                            Group output data set
 //SYSOUT   DD SYSOUT=*
 //SYSPRINT DD SYSOUT=*                            Print Output
 //SYSIN    DD *                                   Command Input
          SPLITHDR GROUP=GROUP1,IN=YES
          SPLITDEF GROUP=GROUP1,AREA=AR1,DBID=991
          SPLITHDR GROUP=GROUP2,IN=YES
          SPLITDEF GROUP=GROUP2,AREA=AR2,DBID=991
          SPLITHDR GROUP=GROUP3,IN=YES
          SPLITDEF GROUP=GROUP3,AREA=AR3,DBID=991
          SPLITHDR GROUP=GROUP4,IN=YES
          SPLITDEF GROUP=GROUP4,AREA=AR1,DBID=990
          SPLITHDR GROUP=GROUP5,IN=YES
          SPLITDEF GROUP=GROUP5,AREA=AR2,DBID=990
          SPLITHDR GROUP=GROUP6,IN=YES
          SPLITDEF GROUP=GROUP6,AREA=AR3,DBID=990
          SPLIT VERIFY=NO
 /*
```

Example z/VSE JCL - Disk

The following is an example of the format for the JCL to submit CA Datacom Fast Restore utility when the device type is DISK.

```
 * $$ JOB ...          See the note above and Listing Libraries for Products in JCL
(see page 13).
 * $$ LST ...
 // JOB name
 // EXEC PROC=procname
 // ASSGN SYSnnn,DISK,VOL=volser,SHR
 // DLBL GROUP1I,'user.input.data.set'                  Input data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP1O,'user.input.data.set'                  Output data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP2I,'user.input.data.set'                  Input data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP2O,'user.input.data.set'                  Output data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP3I,'user.input.data.set'                  Input data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP3O,'user.input.data.set'                  Output data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP4I,'user.input.data.set'                  Input data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP4O,'user.input.data.set'                  Output data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP5I,'user.input.data.set'                  Input data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP5O,'user.input.data.set'                  Output data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP6I,'user.input.data.set'                  Input data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // DLBL GROUP6O,'user.input.data.set'                  Output data set
 // EXTENT SYSnnn,volser,,,rel trk/blk,number trks/blks
 // ASSGN SYS002,device type,VOL=volser,SHR
 // ASSGN SYS003,device type,VOL=volser,SHR
 // ASSGN SYS004,device type,VOL=volser,SHR
 // DLBL SORTWK1
 // EXTENT SYS002,volser,,,rel trk/blk,number trks/blks
 // DLBL SORTWK2
 // EXTENT SYS003,volser,,,rel trk/blk,number trks/blks
 // DLBL SORTWK3
 // EXTENT SYS004,volser,,,rel trk/blk,number trks/blks
 // TLBL RXX,'rxx.data.file'                            Recovery File
```

```
// EXEC DBUTLTY,SIZE=750K
         SPLITHDR GROUP=GROUP1,IN=YES,DEVICE=DISK
         SPLITDEF GROUP=GROUP1,AREA=AR1,DBID=991
         SPLITHDR GROUP=GROUP2,IN=YES,DEVICE=DISK
         SPLITDEF GROUP=GROUP2,AREA=AR2,DBID=991
         SPLITHDR GROUP=GROUP3,IN=YES,DEVICE=DISK
         SPLITDEF GROUP=GROUP3,AREA=AR3,DBID=991
         SPLITHDR GROUP=GROUP4,IN=YES,DEVICE=DISK
         SPLITDEF GROUP=GROUP4,AREA=AR1,DBID=990
         SPLITHDR GROUP=GROUP5,IN=YES,DEVICE=DISK
         SPLITDEF GROUP=GROUP5,AREA=AR2,DBID=990
         SPLITHDR GROUP=GROUP6,IN=YES,DEVICE=DISK
         SPLITDEF GROUP=GROUP6,AREA=AR3,DBID=990
         SPLIT VERIFY=NO
/*
/&
* $$ EOJ
```

Example z/VSE JCL - Tape

The following is an example of the format for the JCL to submit CA Datacom Fast Restore utility when the device type is TAPE.

```
 * $$ JOB ...          See the note above and Listing Libraries for Products in JCL
(see page 13).
 * $$ LST ...
 // JOB name
 // EXEC PROC=procname
 // ASSGN SYSnnn,TAPE,VOL=volser,SHR
 // ASSGN SYS001,TAPE
 // PAUSE MOUNT TAPE
 // TLBL GROUP1I,'user.input.data.set'                Input data set
 // TLBL GROUP1O,'user.input.data.set'                Output data set
 // TLBL GROUP2I,'user.input.data.set'                Input data set
 // TLBL GROUP2O,'user.input.data.set'                Output data set
 // TLBL GROUP3I,'user.input.data.set'                Input data set
 // TLBL GROUP3O,'user.input.data.set'                Output data set
 // TLBL GROUP4I,'user.input.data.set'                Input data set
 // TLBL GROUP4O,'user.input.data.set'                Output data set
 // TLBL GROUP5I,'user.input.data.set'                Input data set
 // TLBL GROUP5O,'user.input.data.set'                Output data set
 // TLBL GROUP6I,'user.input.data.set'                Input data set
 // TLBL GROUP6O,'user.input.data.set'                Output data set
 // ASSGN SYS002,device type,VOL=volser,SHR
 // ASSGN SYS003,device type,VOL=volser,SHR
 // ASSGN SYS004,device type,VOL=volser,SHR
```

```
// DLBL SORTWK1
// EXTENT SYS002,volser,,,rel trk/blk,number trks/blks
// DLBL SORTWK2
// EXTENT SYS003,volser,,,rel trk/blk,number trks/blks
// DLBL SORTWK3
// EXTENT SYS004,volser,,,rel trk/blk,number trks/blks
// TLBL RXX,'rxx.data.file'                          Recovery File
// EXEC DBUTLTY,SIZE=750K
         SPLITHDR GROUP=GROUP1,IN=YES,DEVICE=TAPE
         SPLITDEF GROUP=GROUP1,AREA=AR1,DBID=991
         SPLITHDR GROUP=GROUP2,IN=YES,DEVICE=TAPE
         SPLITDEF GROUP=GROUP2,AREA=AR2,DBID=991
         SPLITHDR GROUP=GROUP3,IN=YES,DEVICE=TAPE
         SPLITDEF GROUP=GROUP3,AREA=AR3,DBID=991
         SPLITHDR GROUP=GROUP4,IN=YES,DEVICE=TAPE
         SPLITDEF GROUP=GROUP4,AREA=AR1,DBID=990
         SPLITHDR GROUP=GROUP5,IN=YES,DEVICE=TAPE
         SPLITDEF GROUP=GROUP5,AREA=AR2,DBID=990
         SPLITHDR GROUP=GROUP6,IN=YES,DEVICE=TAPE
         SPLITDEF GROUP=GROUP6,AREA=AR3,DBID=990
         SPLIT VERIFY=NO
/*
/&
* $$ EOJ
```

Sample Report: SPLIT

This sample shows only selected pages from the full report. The following sample would be the result of the z/OS JCL execution. The report following a z/VSE JCL execution would also show the DEVICE=, LABEL=, and REWIND= parameters that may have been coded.

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
SPLITHDR GROUP=GROUP1,IN=YES

FUNCTION=SPLITHDR
   DEFAULT=NO
       GROUP=GROUP1
   IN=YES
```

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
SPLITDEF GROUP=GROUP1,AREA=AR1,DBID=991

FUNCTION=SPLITDEF
   AREA=AR1
       DBID=991
   GROUP=GROUP1
```

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
SPLITHDR GROUP=GROUP2,IN=YES

FUNCTION=SPLITHDR
   DEFAULT=NO
       GROUP=GROUP2
   IN=YES
```

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
SPLIT VERIFY=NO

FUNCTION=SPLIT
   VERIFY=NO
```

SPLITHDR GROUP=, SPLITDEF GROUP=, and SPLIT VERIFY=

The command exactly as entered.

FUNCTION=

An analysis of keywords encountered and expected. Any errors found are flagged with a note in the left margin.

(Blank area below FUNCTION=)

Any messages related to syntax processing.

Sample Report: SPLIT (Continued)

```
SUMMARY REPORT FOR GROUP GROUP2          FROM ccyy/mm/dd hh.mm.ss through ccyy/mm/dd hh.mm.ss:

    GROUP REQUEST TOTALS FOR AREA AR2 IN DATA BASE   991:

        TABLE K01 SORTED RXX RECORD TOTALS:     ADDIT:         0 DELET:         0 UPDAT:        10 RECORDS:         10
                  PREVIOUS GROUP INPUT TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INCLUDED TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  GROUP FILE OUTPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:        10 RECORDS:         10

        TABLE K02 SORTED RXX RECORD TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  PREVIOUS GROUP INPUT TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INCLUDED TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  GROUP FILE OUTPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0

    TOTAL NUMBER OF RECORDS OUTPUT FOR AREA AR2        10, MOVED RECORDS OUTPUT         0.
SUMMARY TOTALS FOR GROUP GROUP2:
    SORTED RXX RECORDS:                10      PREVIOUS GROUP INPUT RECORDS:          0
    UNCOMMITTED RECORDS INPUT:          0      UNCOMMITTED RECORDS INCLUDED:          0
    GROUP FILE RECORDS OUTPUT:         10      TOTAL RECORDS PROCESSED:              10
```

SUMMARY REPORT FOR GROUP

The group name. It can be followed by DEFAULT to indicate that the group is specified as the default in the SPLITHDR statement.

It is followed by the date and time of the first update request for the group after FROM and the date and time of the last update request for the group after THROUGH.

GROUP REQUEST TOTALS FOR AREA

The DATACOM-NAME of the area followed by the DBID of the database after IN DATABASE.

TABLE

The DATACOM-ID of the table.

SORTED RXX RECORD TOTALS:

The number of RXX records sorted, listed by type of request, and followed by a total.

PREVIOUS GROUP INPUT TOTALS:

The record counts from the previous CA Datacom Fast Restore file listed by type of request and a total. Since a single record can represent ADDIT, UPDAT, and DELET requests individually or in combination, the total of the values for each type of request may not always equate to the RECORDS: value.

UNCOMMITTED INPUT TOTALS:

The record count and request type totals for the uncommitted file input from the previous SPLIT execution.

UNCOMMITTED INCLUDED TOTALS:

The uncommitted record counts for this execution. The heading varies depending on where the requests are written.

UNCOMMITTED INCLUDED TOTALS:

The uncommitted requests were written to the CA Datacom Fast Restore file.

UNCOMMITTED SAVED TOTALS:

The uncommitted requests were written to an uncommitted file.

UNCOMMITTED DROPPED TOTALS:

The requests were not saved at all.

Sample Report: SPLIT (Continued)

```
SUMMARY REPORT FOR GROUP GROUP2          FROM ccyy/mm/dd hh.mm.ss through ccyy/mm/dd hh.mm.ss:

    GROUP REQUEST TOTALS FOR AREA AR2 IN DATA BASE   991:

        TABLE K01 SORTED RXX RECORD TOTALS:     ADDIT:         0 DELET:         0 UPDAT:        10 RECORDS:         10
                  PREVIOUS GROUP INPUT TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INCLUDED TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  GROUP FILE OUTPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:        10 RECORDS:         10

        TABLE K02 SORTED RXX RECORD TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  PREVIOUS GROUP INPUT TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  UNCOMMITTED INCLUDED TOTALS:  ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0
                  GROUP FILE OUTPUT TOTALS:     ADDIT:         0 DELET:         0 UPDAT:         0 RECORDS:          0

    TOTAL NUMBER OF RECORDS OUTPUT FOR AREA AR2        10, MOVED RECORDS OUTPUT         0.
SUMMARY TOTALS FOR GROUP GROUP2:
    SORTED RXX RECORDS:                10      PREVIOUS GROUP INPUT RECORDS:          0
    UNCOMMITTED RECORDS INPUT:          0      UNCOMMITTED RECORDS INCLUDED:          0
    GROUP FILE RECORDS OUTPUT:         10      TOTAL RECORDS PROCESSED:              10
```

GROUP FILE OUTPUT TOTALS:

The number of records and requests by type for the identified table that were written to the CA Datacom Fast Restore file. Because multiple requests are consolidated into a single output record, the totals for ADDIT, UPDAT, and DELET requests may not equate to the total number of records.

TOTAL NUMBER OF RECORDS OUTPUT FOR AREA

The sum of the totals for the tables in the identified area.

MOVED RECORDS OUTPUT

The number of records that had to be moved due to changes in size.

SUMMARY TOTALS FOR GROUP

The group name.

SORTED RXX RECORDS:

The total number of RXX records sorted for this group.

PREVIOUS GROUP INPUT RECORDS:

The total number of records input from a previously created CA Datacom Fast Restore file.

UNCOMMITTED RECORDS INPUT:

The total number of uncommitted updates input from a previously created uncommitted file.

UNCOMMITTED RECORDS INCLUDED:

The total number of uncommitted requests for this group for the current SPLIT execution.

GROUP FILE RECORDS OUTPUT:

The total number of records written to the CA Datacom Fast Restore file.

TOTAL RECORDS PROCESSED:

The total number of records processed for the group.

Sample Report: SPLIT (Continued)

```
SPLIT UTILITY PROCESSING TOTALS:

RXX RECORDS READ:          297 RXX RECORDS SORTED:          270 SPLIT RECORDS READ:             0
RXX BYTES READ:        108,162 RXX BYTES SORTED:         32,233 GROUP BYTES OUTPUT:         18,014 COMPRESSION:

RXX MAINTENANCE REQUESTS NOT BELONGING TO ANY GROUP:            0

RECORDS DROPPED BECAUSE OF GROUP DEFINITION CHANGES:
    GROUP RECORDS DROPPED:           0 UNCOMMITTED RECORDS DROPPED:           0

TIME PERIOD COVERED BY RXX:
FROM ccyy/mm/dd hh.mm.ss through ccyy/mm/dd hh.mm.ss
```

SPLIT Utility Processing Totals

RXX RECORDS READ:

The total RXX records read.

RXX RECORDS SORTED:

The total RXX records sorted.

SPLIT RECORDS READ:

The total previous CA Datacom Fast Restore records read.

RXX BYTES READ:

The total bytes of the RXX file read.

RXX BYTES SORTED:

The total bytes in the RXX file sorted.

GROUP BYTES OUTPUT:

The total bytes output to a new CA Datacom Fast Restore file.

COMPRESSION:

The percent compressed. If CA Datacom Fast Restore files are input, the percent compression can be over 100 percent. The percent is calculated using the size of the previous files. If the file shrinks or stays the same, compression may be indicated at 100 percent or more.

RXX MAINTENANCE REQUESTS NOT BELONGING TO ANY GROUP:

The number of maintenance requests that do not belong to the defined group(s) and are not placed in any CA Datacom Fast Restore files.

RECORDS DROPPED BECAUSE OF GROUP DEFINITION CHANGES:

The total number of records not written out because they no longer belong to the group assigned in the previous SPLIT execution.

TIME PERIOD COVERED BY RXX:

The date and time period beginning with the first RXX record and ending with the last RXX record.

Sample Report: SPLIT (Continued)

```
                                    REPORT OF UNCOMMITTED TRANSACTIONS
TSN      JOB      RUN UNIT START         END           REQUESTS

THERE WERE NO UNCOMMITTED TRANSACTIONS.
```

Report of Uncommitted Transactions

TSN

Transaction sequence number of the uncommitted transaction.

JOB

Job name of the uncommitted transaction.

RUN UNIT

Run unit number of the uncommitted transaction.

START

Date/time of the first maintenance request.

END

Date/time of the last maintenance request.

REQUESTS

Number of maintenance requests done by the uncommitted transaction.

Sample Report: RECOVERY

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
RECBASE DBID=990

FUNCTION=RECBASE
   DBID=990
```

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
RECBASE DBID=991

FUNCTION=RECBASE
   DBID=991
```

```
CONTROL CARD(S)
.........1.........2.........3.........4.........5.........6.........7.........8
RECOVERY OPTION=FORWARD

FUNCTION=RECOVERY
   OPTION=FORWARD
```

RECBASE DBID= and RECOVER OPTION=

The command exactly as entered.

FUNCTION=

An analysis of keywords encountered and expected. Any errors found are flagged with a note in the left margin.

(Blank area below FUNCTION=)

Any messages related to syntax processing.

```
DB13005I - NO RECOVERY JOB NAME - ALL JOBS WILL BE PROCESSED
```

DB130051

A message indicating that no RECJOB card was included in the job stream.

Sample Report: RECOVERY (Continued)

Selected lines from various pages of the report are presented here for illustration purposes.

```
                                        DATACOM/DB FAST RESTORE RECOVERY
TYPE UPDATE   DBID  AREA TABLE URI      BLOCK    L

UPD           991   AR2  K01   00000001 00000003 0
UPD           991   AR2  K01   00000002 00000003 0
UPD           991   AR2  K01   00000003 00000003 0
UPD           991   AR2  K01   00000004 00000003 0
UPD           991   AR2  K01   00000005 00000003 0
UPD           991   AR2  K01   00000006 00000003 0
UPD           991   AR2  K01   00000007 00000003 0
UPD           991   AR2  K01   00000008 00000003 0
UPD           991   AR2  K01   00000009 00000003 0
UPD           991   AR2  K01   0000000A 00000003 0

TOTAL RECORDS PROCESSED FOR GROUP GROUP2:          10 FROM ccyy/mm/dd hh.mm.ss through ccyy/mm/dd hh.mm.ss
```

```
                                        DATACOM/DB FAST RESTORE RECOVERY
TYPE UPDATE   DBID  AREA TABLE URI      BLOCK    L

+UPD DEL      990   AR1  A01   00000001 00000003 0
UPD           990   AR1  C01   00000003 00000003 0
+UPD DEL      990   AR1  A01   00000004 00000003 0
UPD           990   AR1  C01   00000006 00000003 0
+UPD DEL      990   AR1  A01   00000007 00000003 0
 .   .         .     .    .        .        .    .
 .   .         .     .    .        .        .    .
 .   .         .     .    .        .        .    .
```

TYPE UPDATE

The type of maintenance to the record. Each CA Datacom Fast Restore file record can represent the consolidation of multiple maintenance requests. The type is indicated as follows:

ADD

ADDIT

DEL

DELET

UPD

UPDAT

+UPD

The record represents multiple UPDAT requests.

DBID

The DBID of the database.

AREA

The DATACOM-NAME of the area.

TABLE

The DATACOM-NAME of the table.

URI

The Unique Row Identifier (URI) number in hex. For non-URI areas, this field is zero.

BLOCK

The block number in hex.

L

Zero if the area is URI. For non-URI areas, the logical number in hex.

```
                                    DATACOM/DB FAST RESTORE RECOVERY
TYPE UPDATE   DBID  AREA TABLE URI      BLOCK    L

ADD +UPD      990   AR1  X01   0000003B 00000008 0
ADD +UPD      990   AR1  X01   0000003C 00000008 0

TOTAL RECORDS PROCESSED FOR GROUP GROUP4:          60 FROM ccyy/mm/dd hh.mm.ss THROUGH ccyy/mm/dd hh.mm.ss
```

TOTAL...

Summary of processing for this group as follows:

TOTAL RECORDS PROCESSED FOR GROUP

The group name followed by the number of records processed for the group.

FROM

The date and time of the first update.

THROUGH

The date and time of the last update.

```
     ERROR SUMMARY
TOTAL ERRORS ENCOUNTERED-----0
```

ERROR SUMMARY

A summary of the errors encountered and a count of the total number of errors.

Chapter 3: Messages

DB00221E

SUPPORT FOR FAST RESTORE NOT REQUESTED IN THE MULTI-USER START UP OPTIONS

Reason:

The MUF used by the DBUTLTY function was not enabled with the option for CA Datacom Fast Restore. Without the option, this function does not execute.

Action:

The Multi-User Facility must be recycled with the option DATACOM FASTR including in the startup option.

DB08101I

**CA DATACOM FAST RESTORE R12 AT SERVICE PACK: * * * *

Reason:

This message provides the name and release level of the product and the service pack identification. The four asterisks shown are replaced by the service pack information. The release level of CA Datacom Fast Restore must be the same as CA Datacom/DB.

DB10022E

DEVICE REQUIRED WITH IN=YES

Reason:

In z/VSE, identify the device type (DISK or TAPE) when you specify an input file.

Action:

Add the DEVICE= parameter and resubmit.

DB10023E

AREA REQUIRED WHEN DATETIME PROVIDED

Reason:

In the SPLITDEF transaction, the DATETIME= parameter was specified, but the AREA= parameter was not.

Action:

Add the AREA= parameter or remove the DATETIME= parameter, and resubmit.

DB10026E

INVALID DATE, NOT ccyymmddhhmmss

Reason:

The value for the DATETIME= parameter was not specified within the following ranges:

cc

19—39

yy

00—99

mm

01—12

dd

01—31

hh

00—24

mm

00—59

ss

00—59

Action:

Correct the entry and resubmit.

DB13029E

TABLE ttt NOT DEFINED TO A SPLIT GROUP, REJECTED DUE TO VERIFY OPTION

Reason:

The VERIFY= parameter was specified in the SPLIT command and a Recovery File record was input for a table which is not defined to a group.

Action:

Either do not use the VERIFY option or change the group definitions, and resubmit.

DB13030W

LOG RECORDS HAVE BEEN IGNORED BECAUSE THEY WERE NOT IN A URI AREA

Reason:

A table is defined to a CA Datacom Fast Restore group. The CA Datacom Fast Restore utility does not support records in non-URI data areas.

Action:

Redefine the table or remove it from the CA Datacom Fast Restore group definition, and resubmit.

DB13031E

removed MOVED RECORD FOR TABLE ttt

Reason:

The SPLIT function detected a moved record. The CA Datacom Fast Restore utility does not support moved records.

Action:

Redefine the group to exclude the compressed table or remove compression from the table, and resubmit.

DB13032E

INVALID INPUT FILE FOR GROUP gggggg

Reason:

The data set presented as the previous CA Datacom Fast Restore file is not valid. This can be a JCL error.

Action:

Correct the error and rerun.

DB13033E

RXX OPEN FAILURE. ERROR CODE = x

Reason:

The Recovery File open has failed and an error code is displayed.

Action:

See DB18001C for an explanation of the error codes. Take appropriate action.

DB13034E

RXX READ FAILURE. ERROR CODE = x

Reason:

A Recovery File read failure has occurred.

Action:

Collect documentation and call CA Support.

DB13038E

AREA aaa NOT DEFINED FOR BASE dbid

Reason:

The specified area does not exist in the indicated database.

Action:

Correct the area name or the database DBID.

DB13041E

BASE dbid IS NOT DEFINED

Reason:

The specified database does not exist.

Action:

Correct the database DBID.

DB13047E

OUTPUT RECORD SIZE nnnnn IS GREATER THAN OUTPUT BLOCK SIZE nnnnn FOR GROUP gggggg

Reason:

An output record is too large for the specified block size.

Action:

Specify a larger block size.

DB13059E

AREA ccc IN BASE dbid IS DEFINED TO GROUP cccccc AND cccccc

Reason:

The specified area in the specified database has been defined to more than one group.

Action:

Change group definitions such that no area is assigned to more than one group.

DB13070E

SPLITHDR FOR GROUP cccccc IS DEFAULT=NO BUT NO SPLITDEF IS SPECIFIED

Reason:

A group is defined but no data areas are assigned to it.

Action:

Do one of the following:

- Define the group as DEFAULT=YES.
- Create SPLITDEF statements for the group.
- Delete the group.

DB13089E

SPLITDEF FOR GROUP cccccc NOT PRECEDED BY SPLITHDR

Reason:

A SPLITDEF statement must refer to a prior SPLITHDR statement.

Action:

Add a SPLITHDR for the group prior to any SPLITDEF statement for the group.

DB13100E

MULTIPLE DEFAULT GROUPS SPECIFIED

Reason:

One or more groups have DEFAULT=YES specified in the SPLITHDR command. Only one is allowed.

Action:

Specify DEFAULT=YES in only one group SPLITHDR command.

DB13117E

INVALID UNCOMMITTED FILE

Reason:

The input data set for uncommitted requests is not valid. This is usually a JCL error.

Action:

Correct the JCL or do not specify uncommitted input.

DB18010E

INTERNAL DBSPTPR PROGRAM ERROR

Reason:

Internal logic error.

Action:

Collect documentation and call CA Support.

Index

U

V