

UPDS - Partitioned Data Set Maintenance Utility (MVS only)

The Complete online utility UPDS allows you to monitor and maintain the members in a partitioned data set.

This chapter covers the following topics:

- Overview
 - Command Format
 - Using UPDS
 - Commands
-

Overview

Specifically, with UPDS you can:

- List the directory of a partitioned data set, optionally displaying user directory information;
- Display the contents of both a partitioned data set and a sequential data set;
- Print a PDS member or a PDS directory;
- Scratch (delete) one or more members in a partitioned data set;
- Rename a member of a partitioned data set;
- Assign an alias to a member of a partitioned data set;
- Display the two-character library identification codes and the associated DSNAME and VOLSER entries as defined by the UEDTB1 module;
- Submit a PDS member to RJE;
- Request a keyword display;
- While displaying a PDS member in UPDS, transfer control to the Complete Full Screen Editor;
- Modify a member of a partitioned data set.

Command Format

UPDS is a fully conversational online utility program. This means that you can select one of the various functions available either when you invoke UPDS, or any time thereafter.

The basic command format is:

*UPDS

This displays the UPDS menu:

```

11:58:16      TID   18      COM-5.1.      User MBE      05.10.97
      UPDS
      --- Partitioned Dataset Maintenance ---

Function ..... ID Operand(s)      Function ..... ID Operand(s)
-----
List PDS directory .... LS 1 (,4)      Edit Member ..... ED 1,2 (,4)
Display Member ..... DI 1,2 (,4)      Scratch Member ..... PG 1,2 (,4)
Submit Member ..... SU 1,2 (,4)      Rename Member ..... RN 1,2,3 (,4)
Submit Member, fetch UQ SQ 1,2 (,4)      Set Alias for Member ... AL 1,2,3 (,4)

Help ..... HE
-----

Select Function .....:      and Operands
  (1) Library .....:
  (1) or DSN .....:
  (2) Member .....:
  (3) Newname / Alias:
  (4) Volume .....:

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      End

```

This menu displays the UPDS functions available for selection together with important keywords and their defaults. You can invoke a function either by entering a single character, or pressing a PF Key. When you enter a command, the validity and syntax requirements are checked and a message is returned if any errors are found.

Note that the UPDS Menu does not include all commands and keyword combinations available within UPDS, but it does provide those combinations that are most frequently used. The COPY, FILES, HELP, KEYWORDS, LIST, PRINT, and ZAP commands are not supported as a menu option.

The valid UPDS commands are summarized in the following table.

Command	Description
<u>ALIAS</u>	Assigns an alias for a specified member name.
<u>COPY</u>	Prints the list of member names (the directory) in a partitioned data set.
<u>DISPLAY</u>	Displays a specific member of a partitioned data set.
<u>EDIT</u>	Transfers a PDS member in the Complete Full Screen Editor while displaying it in UPDS.
<u>FILES</u>	Lists the two-character library identification codes and their respective assignments as defined by the module UEDTB1.
<u>HELP</u>	Displays the UPDS Menu.
<u>KEYWORDS</u>	Displays the keywords and their current values.
<u>LIST</u>	Lists the member names in a partitioned data set.
<u>PRINT</u>	Prints the statements in a member of a partitioned data set.
<u>RENAME</u>	Renames a member of a partitioned data set.
<u>SCRATCH</u>	Scratches (deletes) a member from a partitioned data set.
<u>SUBMIT</u>	Submits a member from a partitioned data set to RJE.
<u>ZAP</u>	Modifies the currently displayed record.

You can enter all UPDS commands in their entirety or in the abbreviated form indicated in the preceding table by an underscore. Note that you must fully spell out the ALIAS, RENAME, SCRATCH, and ZAP commands in order to ensure against accidental execution of these commands. In addition, these four commands are not allowed as default functions.

You can also enter a command when invoking the UPDS utility to bypass the UPDS Menu.

The command format is:

```
*UPDS [command positional-argument,keyword]
```

where:

command	Specifies a UPDS command.
positional-argument	Specifies one or more UPDS positional arguments.
keyword	Specifies one or more UPDS keyword arguments.

Note:

If your site has a default command and keyword combination set up for this utility so that you cannot reach the menu with this method, you can type "*UPDS *" to override those defaults and reach the UPDS Menu.

Using UPDS

The functional considerations that must be taken into account when entering commands, positional arguments, and keywords are discussed in the following text.

Entering Commands

Since UPDS is fully conversational, you can enter any command function at any time while you're in conversation with UPDS. Once you have entered a command, that command is in effect until you enter another command request. You can enter additional argument(s) either positional or keyword, and the current command will be executed reflecting the changes imposed by your entry of the new argument(s).

For example, you could use the LIST command to list the members in a specific data set by entering the LIST command together with the data set name as a positional argument. After you have viewed the display, to cause the LIST command to be executed for a new data set, simply enter another data set name or a DSNNAME keyword argument.

There are, however, cases when the operation does not default to the last command function specified. These exceptions are:

- A ZAP operation resets the default to DISPLAY, so that you can inspect the zapped record simply by pressing ENTER once more.
- A SCRATCH, RENAME, or ALIAS operation resets the default operation to LIST and the MEMBER value to NEWNAME so that you can list the directory simply by pressing ENTER.

Positional Arguments

Positional arguments, if used, must always immediately follow the command function and precede any keyword arguments. Note that you must separate the command and the first positional argument by either a blank or a comma. In all illustrations in this chapter, a blank is used for this type of separator.

If you enter more than one positional argument, they must be separated by either a comma or a blank. In all illustrations on the UPDS Menu and in this chapter, a comma is used for this type of separator.

Except in the case of the positional argument NEWNAME, entering a positional argument always causes one or more keyword arguments to be initialized. If you enter a command that includes positional arguments, UPDS syntax checking processes positional arguments *after* it processes keyword arguments. Consequently, positional arguments are the final determining factor in the execution of a command function.

Specific considerations for entering positional arguments are given along with the description of each command in the latter sections of this chapter.

Keyword Arguments

You must specify keyword arguments in one of three ways:

- As the keyword argument only, with no command;

- As the only argument given with a command;
- Following all positional arguments;

Keyword arguments, when entered, must be separated from commands, positional arguments, and other keyword arguments by either a comma or a blank. In all illustrations in this chapter, a comma is used as the separator.

Each keyword argument consists of the keyword itself, followed by an equal sign, followed by the keyword data.

In the case of the DSNNAME keyword, the data can be a character string consisting of characters, embedded commas, embedded equal signs, and/or embedded blanks. Note that in these situations, the data entered should be enclosed by quotation marks. Furthermore, if the character string contains an embedded quote, the embedded quote should be entered twice. The following examples illustrate this concept:

```
DSNNAME='APPLE PIE'
```

The character string must be enclosed in quotes because it contains an embedded blank.

```
DSNNAME='JOE''S DATA SET'
```

The quotation marks following JOE must be entered twice because it is within a quoted character string.

```
DSNNAME=JOE'S
```

The quotation mark following JOE does not need to be entered twice because it is not within a quoted character string.

The available keywords for UPDS commands are summarized in the following table. The shortest possible abbreviations are indicated by underlining.

Keyword	Description
<u>C</u> VOL	Specifies a constant volume identification number to be used in multiple accesses. dVolume identification is required if the PDS has not been cataloged. See the chapter UDS - Data Set Maintenance Utility (OS Only) for cataloging procedures.
<u>D</u> ESTCODE	Specifies the screen-to-hardcopy device to be used for printout spooling requests.
<u>D</u> SNAME	Specifies the data set name of the partitioned data set to be accessed.
<u>F</u> ORMAT	Specifies a character, hexadecimal, or interpreted dump format for the DISPLAY and PRINT commands.
<u>L</u> IBRARY	Specifies a two-character library identification code.
<u>M</u> EMBER	Specifies the member name to be used when accessing the library.
<u>N</u> OTE	Specifies a one- to eight-character alphanumeric tag to be used for later reference.
<u>P</u> POINT	Specifies a tag previously defined using the NOTE keyword.
<u>R</u> ECORD	Specifies the desired position in the displayed data set.
<u>S</u> CAN	Specifies the value for a scan request.
<u>U</u> SERDATA	Specifies whether or not the optional user data or stow data is to be displayed after execution of the LIST command. Also specifies the format of the resulting display.
<u>V</u> OLSER	Specifies the volume identification number of the disk volume to be accessed.
<u>Z</u> ONE	Specifies the range of columns in the data set that are to be displayed.

Note that you can access a display of the current status of the keyword parameters by using the **KEYWORDS** command. For additional information, and for an example of the **KEYWORDS** command display, see the section **KEYWORDS Command** later in this chapter.

Special considerations must be made when using keywords. Among them are:

1. UPDS command processing always occurs in the following sequence:
 - Initializes the keyword arguments;
 - Reinitializes the keyword arguments based upon the positional arguments given;
 - Executes the command.
2. After a command has been successfully or unsuccessfully executed, you can use the **KEYWORDS** command to display all the initialized keyword arguments and their current values.
3. To initialize any keyword, simply enter the keyword with no associated command function or positional argument. The currently active command (that is, the command executed last) is then executed again using the new arguments that you entered as execution time arguments.

Once you have initialized a keyword, you can enter commands that use data from that keyword without needing to specify that keyword again; the initialized value will be used when you execute the command.

Initialized keyword values will remain in effect from command to command as long as:

- UPDS remains conversational and is not terminated, and:
- Entry of a command function does not alter one of the keyword values.

Note:

If you also give positional arguments, command entry may force an initialized keyword to be reinitialized.

4. If you enter more than one keyword in one command, Complete processes them from left to right. Therefore, if two or more keywords conflict in that they force initialization of a common keyword argument, the *last keyword entered* determines the initialized value to be in effect unless it is overridden again by a positional argument.
5. Initialization considerations for each keyword are given in detail below:

CVOL=cvol	Specifies the value with which the VOLSER argument is to be initialized when it is found to be blank.
DESTCODE=destcode	Specifies the destination routing code or TID to be used for UPDS PRINT commands.
	Note that destcode always defaults to the screen-to-hardcopy terminal defined for the terminal in use in the TIBTAB module, if any.
DSNAME=dsname	Specifies the data set name to be used when a two-character library code is not supplied. Note: Entry of this keyword argument or entry of a unique data set name causes the VOLSER and LIBRARY keywords to be initialized to blanks.

<p>FORMAT=format</p>	<p>Specifies the display format for the DISPLAY and PRINT commands. Valid formats are:</p> <table border="0" data-bbox="592 241 1388 472"> <tr> <td>CHAR</td> <td>character</td> </tr> <tr> <td>HEX</td> <td>hexadecimal</td> </tr> <tr> <td>INT</td> <td>interpreted hexadecimal dump</td> </tr> </table> <p>Note that for both the HEX and INT format options, one of three line formats will be selected, depending on line size. The formats have been designed around the standard line sizes of 40, 80, or 132 characters; however, the selection algorithm uses the minimum line sizes needed for each format:</p> <table border="0" data-bbox="609 693 1144 1113"> <thead> <tr> <th>Format</th> <th>Mm. Charts/Line</th> <th>Bytes/Line Shown</th> </tr> </thead> <tbody> <tr> <td rowspan="3">F=HEX</td> <td>125</td> <td>48</td> </tr> <tr> <td>77</td> <td>32</td> </tr> <tr> <td>40</td> <td>8</td> </tr> <tr> <td rowspan="3">F=INT</td> <td>112</td> <td>32</td> </tr> <tr> <td>59</td> <td>16</td> </tr> <tr> <td>33</td> <td>8</td> </tr> </tbody> </table> <p>Note: If less than 33 or 40 characters per line are available, the respective option may not be performed and the keyword value will be reset to CHAR.</p>	CHAR	character	HEX	hexadecimal	INT	interpreted hexadecimal dump	Format	Mm. Charts/Line	Bytes/Line Shown	F=HEX	125	48	77	32	40	8	F=INT	112	32	59	16	33	8
CHAR	character																							
HEX	hexadecimal																							
INT	interpreted hexadecimal dump																							
Format	Mm. Charts/Line	Bytes/Line Shown																						
F=HEX	125	48																						
	77	32																						
	40	8																						
F=INT	112	32																						
	59	16																						
	33	8																						
<p>LIBRARY=libcode</p>	<p>Specifies the two-character library code to be used when a command for which a library code has not been entered is processed.</p> <p>Note that entry of libcode initializes the keywords LIBRARY, DSNNAME, and VOLSER. The information with which these keywords are initialized is taken from the library code module UEDTB1 or from the volumes set by the UUTIL function UL.</p>																							

MEMBER=member	<p>Specifies the member name to be used when a command for which no member name has been entered is processed.</p> <p>In order to facilitate initialization of the appropriate keywords, member can also be specified in one of the following formats:</p> <ul style="list-style-type: none"> a. dsn(member)/volume b. dsn(member)volume c. libcode(member)
NOTE=name	<p>Specifies a one- to eight-character alphanumeric tag that allows rapid paging from one place in a member to another while displaying.</p> <p>When NOTE is issued, the current setting of the display (record number, format, and zone setting) is tagged with the name specified in the command. Later, when POINT is issued with the same name, you are returned to the setting where the NOTE command was entered for that name.</p> <p>Note that up to 64 names can be defined at any given time; a list of currently defined names is located at the bottom of the KEYWORDS display.</p> <p>The list of names is cleared when a new member/data set is displayed; no other provision is made to clear a name from the list. However, you can redefine a name simply by using it in another NOTE command.</p>
POINT=name	<p>Is a one- to eight-character alphanumeric tag previously specified using the NOTE keyword. It enables the rapid paging from one place in a member to another while displaying.</p>

RECORD=position	<p>Is used to position the display of a data set. Position can be specified as one of the following:</p> <ul style="list-style-type: none"> n Positions to the n th record of the data set/member. +n Positions forward by n records. -n Position backward by n records. ++ Positions to the last record of the data set/member. -- Positions to the first record of the data set/member. * Redisplays the current record from the beginning. <p>The number of the current record is displayed in the heading of the DISPLAY display and/or in the KEYWORDS display. If the request causes the record number to go below 1 or above 9999999, these values, respectively, will be substituted. In addition, if the requested record number is larger than the number of records in the data set/member, the last record will be substituted.</p> <p>You can perform the positioning function in one of two ways:</p> <ul style="list-style-type: none"> a. Include the RECORD keyword in the command string, e.g.: D UA(TIBTAB),RECORD=35 If you access a member other than the currently open one, the RECORD value will be reset to 1 unless you specify another RECORD value in the same command line as the display request (as shown above). b. If you do not intend to enter an operation or positional operand, you can enter RECORD keyword values without the preceding keyword, e.g.: 1 -300 ++ * You can also enter other keyword operands in the same line after a RECORD value without its keyword, e.g.: +5 F=I
-----------------	---

SCAN=value	Specifies the value for which a scan is to be performed.	
	Scan formats include:	
	SCAN=TEXTSTRING	This format is acceptable if the following characters are not part of the requested character string: blank, quote ('), plus (+), minus (-), or X. Note that +, -, and X can be part of the text string if they are not the first character of the string.
	SCAN='TEXTSTRING'	This format is required if blank, quote ('), plus (+), minus (-), or X is part of the string. Note that quotation marks within a string must be represented as double quotation marks.
	S=X.....	This format is required where the hex data must be an even number of nibbles.
	S=-TEXTSTRING S='-TEXTSTRING' or S=-X.....	Indicates a backwards scan. Note that a backwards scan requires more overhead than a forward scan.
	S=	Resumes the scan with the same argument.
	S=+	Changes the direction to forward and resumes.
	S=-	Changes the direction to backward and resumes.
	In character format, scanning starts at rec +/-1 (depending on the direction of the scan); in hex format scanning starts at rec, current position +1. The entire record is scanned left to right even in up scan mode (honoring the zone setting). The first occurrence located determines the new current record.	
Note that if EOF or TOF is reached before the scan succeeds, the current record setting is not changed. The current scan value will be displayed on the KEYWORDS menu.		
Attention interrupts may be used to terminate a runaway scan (every 500 records).		
The PF Key system supports PF9 as SCAN= for UPDS.		

USERDATA=indicator	Default: USERDATA=NO
	Specifies whether or not user stow information is to be displayed when directory information is requested.
	The value specified can be YES, NO, TTR, or HEX.
	USERDATA=YES indicates that directory information is to be included in the display. Note that if no directory information is present, only member names are displayed.
	USERDATA=NO indicates that directory information is not to be displayed.
	USERDATA=TTR indicates that TTR information is to be included in the display of member names. Note that TTR information is given in both hexadecimal and decimal forms in CCHHR format.
	USERDATA=HEX indicates that directory information is to be included in the display of member names in hexadecimal format.
VOLSER=volume	Specifies the volume on which the data set search is to be restricted.
	Volume searching conventions are determined by the value to which the keywords are initialized. For example, entering the DSNAME keyword argument clears the VOLSER keyword argument. Subsequent commands then cause VOLSER to be initialized from the CVOL keyword before command execution.
	Entering the LIBRARY keyword argument causes both the VOLSER and LIBRARY keyword arguments to be initialized. Volume searching sequence logic is summarized below:
	<ol style="list-style-type: none"> a. VOLSER=volume - Search the volume. b. LIBRARY=code - Search the catalog. c. CVOL=volume - Search the volume, if VOLSER=blank. d. Otherwise, search the catalog.

ZONE=range	Specifies the range of columns in the data set that are to be displayed.
	This keyword can be specified in one of four formats: ZONE=n1-n2 ZONE=Xn1-n2 ZONE=n1(n2 ZONE=Xn1(n2
	where:
	n1 Specifies the first column to be displayed.
	- Indicates "through".
n2 Specifies the last column to be displayed.	
(Indicates "specify the following number of columns".	
n2= Specifies the number of columns to be displayed.	
X Preceding n1 or n2, indicates that it is in hexadecimal.	
Note: Hex notation starts the record with a zero offset; decimal notation starts the record at column one.	

Paging Requests

If you request that a large amount of information be output from a hard copy terminal, Complete issues an attention interrupt, which causes the output to be interrupted. At this point, if you press ENTER (or its equivalent), one more line of output is generated and a prompting request for entry of a new command is then given.

If you are using a 3270-type terminal or compatible device, press ENTER to continue the display of output that you cannot view in a single display.

PF Key Assignments

The default PF key assignments for UPDS display function are described in the following table:

PF-KEY	Description
PF1	Page backward.
PF2	Go to the top of the member.
PF3	Go to the bottom of the member.
PF4	Page forward.
PF5	Scroll backward one line.
PF6	Scroll forward one line.
PF7	Scroll backward 10 lines.
PF8	Scroll forward 10 lines..
PF9	SCAN = (repeat the SCAN)
PF10	Submit the member.
PF11	Edit the member.
PF12	Display keywords.

Note that the system administrator can modify the PF key assignments for all users. In addition, you can also redefine your own default PF key assignments using the FK function of the UUTIL Utility.

To display an overview of your current PF Key assignments, enter the immediate command:

***P.ALL**

For further information concerning immediate commands, see the section on Immediate Commands in **COM-PASS - Parallel Transaction Utility**.

Commands

The UPDS commands are discussed in the remainder of this chapter.

ALIAS Command

The ALIAS command enables you to assign an alias to a member of a partitioned data set.

The command format is:

ALIAS member, name2

where *member* can be any one of the following formats:

member	Specifies the one- to eight-character member name to be renamed. Note: If member is specified, the DSNNAME keyword must be initialized.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be renamed. If member is omitted, the currently initialized member is assumed to be the member name.
dsn(member)/volume	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be renamed. If member is omitted, the enclosing parentheses must either remain or be replaced by the optional slash (/).
	volume specifies the volume identification of the disk volume to be searched when locating dsn.

The positional argument *name2* specifies the one- to eight-character name that is to become the alias for member.

COPY Command

The COPY command enables you to obtain a hard copy listing of all member names in a partitioned data set.

The command format is:

COPY [**destcode**]

where *destcode* is either a Terminal Identification number (TID) or a message switching destination code that identifies the terminal to which the hard copy listing will be queued.

If *destcode* is omitted, the value determined by the keyword function DESTCODE determines the printout spool destination.

The printout spool listing is printed with 54 lines of data to a page. The top of each page contains summary information generated by UPDS that identifies the fact that the listing was generated by UPDS. Included in this heading is such information as the terminal user ID, the originating TID, the library name, and other applicable information.

DISPLAY Command

The DISPLAY command enables you to obtain a listing of either a member in a partitioned data set or a sequential data set.

The command format is:

DISPLAY [*member*]

where *member* can be any one of the following formats:

member	Specifies the one- to eight-character member name to be displayed. Note: If member is specified, the DSNNAME keyword must be initialized.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be displayed. If member is omitted, the currently initialized member is assumed to be the member name.
dsn(member)/volume	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be displayed. If member is omitted, the enclosing parentheses must either remain or be replaced by the optional slash (/).
	volume specifies the volume identification of the disk volume to be searched when locating dsn.

In each of the above formats, if *member* is omitted, the member to be displayed is identified by the keyword argument MEMBER.

EDIT Command

The EDIT command enables you to transfer control to the Complete Full Screen Editor (UEEDIT) while you are displaying a PDS member under UPDS.

The command format is:

EDIT *member*

where *member* may be any one of the following formats:

member	Specifies the one- to eight-character member name to be edited.
	Note: If member is specified, the DSNAME keyword must be initialized.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be edited. If member is omitted, the currently initialized member is assumed to be the member name.
dsn(member)/volume	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be edited. If member is omitted, the enclosing parentheses must either remain or be replaced by the optional slash (/).
	volume specifies the volume identification of the disk volume to be searched when locating dsn.

FILES Command

The FILES command enables you to obtain a listing of the entries in the two-character library code table UEDTB1.

The command format is:

FILES

Any additional arguments that you enter *must* be keyword arguments. Note, however, that entering keyword arguments only causes initialization of the appropriate keyword functions; it has no effect on the output of the FILES display.

The FILES command display consists of a listing of the various libraries defined by UEDTB1. For each library listed, the two-character library identification code is given together with the volume identification of the disk volume on which the library resides and the data set name in the following format:

<pre> F ID...DSNAME...(ON...VOLSER) </pre>
--

HELP Command

The HELP command enables you to obtain a display of the UPDS menu.

The command format is:

HELP

You may optionally enter the HELP command as:

?

Any additional arguments that you enter *must* be keyword arguments.

If you include keyword arguments with the HELP command, it will cause the initialization of those keyword entries; to view the new initialized values, use the KEYWORD display (see the following section). You can subsequently use the initialized values simply by entering a command function with no arguments.

KEYWORDS Command

The KEYWORDS command enables you to obtain a display of the keywords and their current values.

The command format is:

KEYWORDS

The following figure illustrates the format of the keywords display.

```

KEYWORD PARAMETER CURRENT VALUES:
L IBRARY =
D SNAME =
V OLSER =
M EMBER =
R ECORD =
C VOL =
F ORMAT = (CHCAR, HEX, INT)
U SERDATA = (YES, NO, TTR, HEX)
S CAN =
Z ONE =
DE STCODE =
N OTE = NAME (REMEMBER CURRENT SCREEN)
P OINT = NAME (SHOW REMEMBERED SCREEN)
CURRENTLY DEFINED NAMES:

```

For additional information on the KEYWORDS command, see the section **Using UPDS** above.

LIST Command

The LIST command enables you to obtain a listing of the member names of a partitioned data set.

The command format is:

LIST [member]

where *member* can be in one of the following formats:

(member)	Specifies the one- to eight-character name to be used as the collating reference point with which member names will be listed.
	Note that the parentheses are required. Note: The DSNNAME keyword must be initialized in order for this command to execute.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be used as the collating reference point with which member names are listed. If member is omitted, the enclosing parentheses may also be omitted.
dsn(member)/volume	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character name to be used as the collating reference point with which member names are listed. If member is omitted, the enclosing parentheses must either remain or be replaced by the optional slash (/).
	volume specifies the volume identification of the disk volume to be searched when locating dsn.

In each of the formats described above, if member is omitted, the first member to be displayed is determined by the keyword argument MEMBER.

Note:

The LIST command with nothing following ("L") picks up the LIST function from the previous reference; a LIST command with one blank following ("L ") returns the list to the beginning of the directory.

PRINT Command

The PRINT command enables you to obtain a hard copy listing of either a sequential data set or a partitioned data set member.

The command format is:

```
PRINT destcode NNNNNN,ATTACH,CC,NOPAGE,NOHEADER
```

The optional arguments are defined as follows:

destcode	Specifies either a Terminal Identification number (TID) or a message switching destination code that identifies the terminal(s) to which a hard copy listing will be queued.
	<p>Notes:</p> <ol style="list-style-type: none"> 1. destcode must always be the first positional operand. 2. If you don't specify DESTCODE, but do specify other operands, you must indicate the absence of DESTCODE by a comma. <p>If destcode is omitted, the value determined by the keyword function DESTCODE determines the printout spool destination. The default for the keyword argument DESTCODE is the screen-to-hardcopy device of the terminal in use, if any. If SCHC=0 (no default hard copy is assigned), then the calling terminal is the default destcode.</p>
NNNNNN	Specifies the number of records to be printed, starting with the current record in the display.
	If NNNNNN is not specified, the printout will contain the entire data set or member.
	Note that NNNNNN is recognized as the only numeric operand beyond DESTCODE.
ATTACH	Specifies that asynchronous spooling of long printouts is to occur. The Attach argument calls an identical copy of UPDS, which runs as an asynchronous task.
	<p>Note:</p> <p>This function will not work properly if UPDS has been invoked under a user program.</p>
CC	Specifies that ASA carriage control characters in position one of data records are to be used.
	<p>Note:</p> <p>With the CC option, no other formatting is performed and no headings are provided by UPDS.</p>
NOPAGE	Specifies that data is to be printed without page formatting. Only the header and EOF message will be produced.
NOHEADER	Specifies that data is to be printed without using any of the formatting options. Only the data will be printed.

Any of the optional operands, except DESTCODE can be specified in any order.

If you specify more than one paging option, the following rules apply:

- CC is considered to include NOHEADER, so CC takes effect (and no headers will be printed).
- NOHEADER is considered to supersede NOPAGE, so it takes effect and (and no page formatting will occur).

- CC and NOPAGE are considered to be in conflict and cause the generation of an error message.

If you do not specify any paging option, data will be printed in a single-spaced module with page headers, 54 lines of data to a page. The page header contains summary information generated by UPDS identifying the fact that UPDS generated the listing, as well as information such as the terminal TID, the originating TID, the library name, and other applicable information.

Wherever it is permitted by the terminal access methods, you can use an ATTN interrupt to abort print requests. On spooled printouts, a message is added to indicate this condition. If output is interrupted by use of the BREAK key (or equivalent on hard copy terminals), the message:

UPD0000 - ENTER UPDS COMMAND

is displayed, and new input will be accepted.

RENAME Command

The RENAME command enables you to rename a member of a partitioned data set.

The command format is:

RENAME *member* , *newname*

where *member* can be any one of the following formats:

member	Specifies the one- to eight-character member name to be renamed.
	Note: If member is specified, the DSNNAME keyword must be initialized.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be renamed. If member is omitted, the currently initialized member is assumed to be the member name.
dsn(member)/volume	dsn specifies either a fully qualified data set name or its two-character library code.
	member specifies the one- to eight-character member name to be renamed. If member is omitted, the enclosing parentheses must either remain or be replaced by the optional slash (/).
	volume specifies the volume identification of the disk volume to be searched when locating dsn.

The positional argument *newname* specifies the one- to eight-character name to which *member* is to be changed.

SCRATCH Command

The SCRATCH command enables you to delete one or more member names from the directories of one or more partitioned data sets.

The command format is:

```
SCRATCH member [ ,member , . . . ]
```

where *member* can be any one of the following formats:

member	Specifies the one- to eight-character member name to be scratched. Note: If member is specified, the DSNNAME keyword must be initialized.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code. member specifies the one- to eight-character member name to be scratched. If member is omitted, the currently initialized member is assumed to be the member name.
dsn(member)/volume	dsn specifies either a fully qualified data set name or its two-character library code. member specifies the one- to eight-character member name to be scratched. If member is omitted, the enclosing parentheses must either remain or be replaced by the optional slash (/). volume specifies the volume identification of the disk volume to be searched when locating dsn.

In each of the formats described above, if *member* is omitted, the member to be scratched is identified by the keyword argument MEMBER.

SUBMIT Command

The SUBMIT command enables you to submit a sequential data set or partitioned data set member containing JCL statements to be scheduled for batch execution via Com-plete's RJE facility.

The command format is:

```
SUBMIT member
```

where *member* can be any one of the following formats:

member	Specifies the one- to eight-character member name to be submitted. Note: If member is specified, the DSNNAME keyword must be initialized.
dsn(member)	dsn specifies either a fully qualified data set name or its two-character library code. Note that member specifies the one- to eight-character member name to be submitted. If member is omitted, the currently initialized member name is assumed.
dsn(member)/volume	dsn specifies either the fully qualified data set name or its two-character library code. member specifies the one- to eight-character member name to be submitted. If member is omitted, the enclosing parantheses must either remain or be replaced by the optional slash (/). volume specifies the volume identification of the disk volume to be searched when locating dsn.

An alternative format of the SUBMIT command is:

```
SUQ dsn(member) | volume
```

Use of this format performs an automatic fetch to "*UQ A JB=xxx", where xxx represents the name of the first JCL statement (assuming that it is a job statement).

ZAP Command

The ZAP command enables you to modify the currently displayed record.

The command format is:

```
ZAP disp, verdata, repdata
```

where:

disp	Specifies the displacement into the record at which the modification is to occur. Note that the value specified can be either in decimal format (the first byte of the record is specified as column one), or in hexadecimal format (hex digits preceded by the character "X"; the first byte of the record is specified as offset X0).
verdata	Specifies the value to be verified at the location "disp" in the record.
repdata	Specifies the value to be moved into the record at location "disp".

You can specify both *verdata* and *repdata* as either character data or hex digits following these rules:

- If you specified *disp* as a decimal value, *verdata* and *repdata* must be character strings (which must be enclosed in quotation marks only if they contain commas, blanks, equal signs, or leading quotation marks).

- If you specified *disp* as a hexadecimal value, you must also specify *verdata* and *repdata* as hexadecimal numbers (consisting of an even number of digits not preceded by the character "X") or as character strings enclosed in quotation marks.

Note:

The ZAP command is not applicable to load modules in PDSEs, as this dataset type does not allow update in place.