Com-plete

	Version 6.3.1	Utilities

software AG

This document applies to Com-plete Version 6.3.1 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© Copyright Software AG 2003
All rights reserved.

The name Software AG and/or all Software AG product names are either trademarks or registered trademarks of Software AG. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

Overview of all Utilities . . . 1
Overview of all Utilities . . . 1
Conventions . . . 4
Conventions . . . 4
UATTACH - Start Background Program . . . 6
UATTACH - Start Background Program . . . 6
COM-PASS - Parallel Transaction Utility . . . 7
COM-PASS - Parallel Transaction Utility . . . 7
COM-PASS Features - Overview . . . 7
COM-PASS Standard Mode: The COM-PASS Menu . . . 9
COM-PASS Expert Mode . . . 11
COM-PASS TPF Mode . . . 13
Parallel Transaction Handling . . . 14
COM-PASS Commands . . . 17
SUSPEND/RECALL/JUMP Commands . . . 17
Direct Commands . . . 19
Immediate Commands . . . 21
UCOPY - PRINT/SAVE Screen Utility . . . 25
UCOPY - PRINT/SAVE Screen Utility . . . 25
Command Format . . . 25
Functional Considerations . . . 25
UDEBUG - Application Debugging . . . 26
UDEBUG - Application Debugging . . . 26
Overview . . . 26
Restrictions on the use of UDEBUG . . . 34
Execution of Breakpoints . . . 34
Instruction Stepping . . . 35
UDEBUG User Interface . . . 35
UDEBUG Session Startup . . . 35
Error and Information Messages . . . 36
Program Function Keys . . . 37
Customizing Characters and Options . . . 37
UDEBUG Screens . . . 38
The UDEBUG Session Information Screen (UDB0) . . . 40
The UDEBUG Dump Storage Screen (UDB1) . . . 43
The UDEBUG Error Messages Screen (UDB2) . . . 44
The UDEBUG Breakpoint Information Screen (UDB4) . . . 45
The UDEBUG Symbol Display Screen (UDB5) . . . 47
The UDEBUG Breakpoint List Screen (UDB6) . . . 52
UDEBUG Commands . . . 54
Entering Commands . . . 54
Recalling Commands . . . 54
Command Description . . . 54
UDD - DASD Display Utility (Control User) . . . 67
UDD - DASD Display Utility (Control User) . . . 67
Command Format . . . 67
Header Record Display . . . 68
Data Record Display . . . 69

Modifying Disk Address 69
UDS - Data Set Maintenance Utility (z/OS only) 71
UDS - Data Set Maintenance Utility (z/OS only) 71
Command Format 71
Menus 71
Example 1 73
Example 2 73
UDS Functions 74
ALLOC Function 74
CATLG Function 75
DISP Function 76
FIND Function 76
HELP Function 77
LIST Function 77
PURGE Function 82
RECATLG Function 82
RENAME Function 83
SCRATCH Function 84
UNCATLG Function 84
UDS VSAM Services (UDVS for VSE) 85
UDUMP - Conversational Dump Display Utility 86
UDUMP - Conversational Dump Display Utility 86
Overview 86
Command Format 87
UDUMP Commands 89
Addressing Command Options 90
Display Format Options 93
Address Relocation Options 96
PGMS command 97
Print Dumps 98
UDYN - Dynamic Dataset Allocation and Deallocation (z/OS only) 99
UDYN - Dynamic Dataset Allocation and Deallocation (z/OS only) 99
UDZAP - DASD Modification Utility (Control User) 101
UDZAP - DASD Modification Utility (Control User) 101
Command Format 101
Selecting a Volume 102
UCB/PUB Lookup Display 102
Zap Volume Display 103
UED - Line-Oriented Source Editor Utility 105
UED - Line-Oriented Source Editor Utility 105
Command Format 105
UED Commands 106
CHANGE Command 109
COPY Command 109
DEFINE Command 110
DELETE Command 114
END Command 115
FIND Command 115
HALT Command 116
HELP Command 117
INSERT Command 117

LIST Command 118
MACRO Command 118
MOVE Command 120
NUMBER Command 121
PRINT Command 122
QUERY Command 124
READ Command 124
RECOVER Command 126
REPEAT Command 126
RESUME Command 126
SAVE Command 126
SUBMIT Command 128
TAB Command 128
Forward Paging Command 129
Backward Paging Command 129
Last Statement Paging Command 129
First Statement Paging Command 130
Tag Paging Command 130
Statement Modification Command 130
UEDIT - Text Editor Utility 131
UEDIT - Text Editor Utility 131
Getting Started 131
Opening a Member for Editing 132
Entering Commands on the Command Line 132
Entering Prefix Commands 134
Starting an Edit Session 138
Full Screen Editor Menu 138
Edit Direct Call 145
Full Screen Editing 150
Screen Areas 150
Tags 150
Strings 151
SET ZONE Command 151
SET SCAN Command 151
Processing Priorities 152
Prefix Commands 152
Immediate Prefix Commands 152
Block Prefix Commands 156
A/B Positioning Commands 158
Examples 158
Scrolling Commands 159
Examples 160
User Profiles 160
PF Key Usage 161
Edit Session Suspend Levels 161
CommandFormat 161
Repeat a Command 162
Redisplay a Command 162
Command Overview 162
Command Description 164
BOTTOM Command 164

CCHANGE Command . . . 164
CHANGE Command . . . 165
COPY Command . . . 167
DELETE Command . . . 168
END Command . . . 169
FIND Command . . . 169
GLOBALS Command . . . 171
HELP Command . . . 171
HEX Command . . . 172
INCLUDE Command . . . 172
LEAVE Command . . . 174
LEFT Command . . . 175
MOVE Command . . . 175
NOTE Command . . . 176
POINT Command . . . 176
PRINT Command . . . 177
PROFILE Command . . . 178
QUERY Command . . . 179
READ Command . . . 181
RESET Command . . . 183
RIGHT Command . . . 183
RSAVE Command . . . 184
SAVE Command . . . 185
SCAN Command . . . 187
SET Command . . . 189
STATUS Command . . . 198
SUBMIT Command . . . 198
TOP Command . . . 199
Moving PDS Members to and from a Personal Computer (PC) . . . 199
Downloading a PDS Member to a PC File . . . 199
Example 1: Downloading to a Diskette . . . 200
Example 2: Downloading to a DOS Directory . . . 200
Uploading a PC File to a PDS Member . . . 201
ULIB - Program Catalog Maintenance Utility . . . 203
ULIB - Program Catalog Maintenance Utility . . . 203
Program Parameters . . . 203
Defining a Default Parameter Set . . . 205
Command Format . . . 205
Function OV - Program Overview . . . 206
Function CA - Catalog a Program . . . 207
Function MO - Modify a Program Catalog Entry . . . 207
Function DE - Delete a Program Catalog Entry . . . 208
Function RF - Refresh a Program . . . 208
ULOG - LOGON/LOGOFF UTILITY . . . 209
ULOG - LOGON/LOGOFF UTILITY . . . 209
Logon Procedure . . . 209
External Security Systems . . . 211
ULOG Commands . . . 212
UM - Message Switching / Printout Spooling Utility . . . 214
UM - Message Switching / Printout Spooling Utility . . . 214
Destination Codes . . . 215

Class Codes 215
Message Queue 217
Message Delivery and Receipt 217
Message Interruption and Recovery 218
Message Text 219
Message Routing 219
Alternate Terminals 219
Disabled Terminals 220
Inoperative Terminals 220
Menu Operation 220
Function SM - Send Message to User(s) or TID(s) 221
Function SA - Send Message to all Logged on Users 222
Function SS - Send Message to User(s) (from Menu): 222
Function SG - Send Message to Accounting Groups: 223
Function ST - Send Message to TIB Groups: 223
Direct Command Operation 224
Send to Terminal(s): 225
Display Message Status 226
Display a Message 227
Delete a Message 227
Hold a Message 227
Release a Message 228
Purge all Messages 228
Enable Terminal 228
Disable Terminal 228
Reset Terminal I/O 229
Set Alternate Terminal 229
Remove Alternate Terminal 229
Change Receiving Class Codes (Control User) 229
Change Sending Class Codes (Control User) 230
UMAP - Terminal Mapping Utility 232
UMAP - Terminal Mapping Utility 232
Command Format 232
Direct Command Support 232
Using UMAP 233
Map Storage 233
Map Retrieval 233
Termination 234
UMAP Main Menu 234
Mapname 235
Map Retrieval from Com-plete’s Load Library Chain 236
Field Indicators 236
UMAP Main Menu Functions 237
HELP 237
Modify Map 238
New Map Creation 238
Copy Map 240
Delete Map from SD File 240
Show Map 241
Terminal Control Codes 242
Edit Copy Code 244

Load GDDM Symbol Sets 246
UMAP Modify Map Menu 247
Field Indicators 248
New Field Default Field Description Codes 248
UMAP Modify Map Menu Functions 248
Full Screen Modify 249
Modify with Prompt 252
Save Map into Library 258
Dynamic Test 258
UMATH - Mathematics Utility 262
UMATH - Mathematics Utility 262
Command Format 262
Math Mode 262
Example 262
Function Mode 263
Example 263
UP - Terminal Paging Utility 264
UP - Terminal Paging Utility 264
UPDS - Partitioned Data Set Maintenance Utility (z/OS only) 266
UPDS - Partitioned Data Set Maintenance Utility (z/OS only) 266
Overview 266
Command Format 266
Using UPDS 269
Entering Commands 269
Positional Arguments 269
Keyword Arguments 269
Paging Requests 278
PF Key Assignments 278
Commands 279
ALIAS Command 279
COPY Command 280
DISPLAY Command 280
EDIT Command 281
FILES Command 282
HELP Command 282
KEYWORDS Command 283
LIST Command 283
PRINT Command 284
RENAME Command 286
SCRATCH Command 287
SUBMIT Command 287
ZAP Command 288
UQ - System Job Queue Display Utility 290
UQ - System Job Queue Display Utility 290
Overview 290
Display Operations 290
Control Operations 291
Security 291
Control 291
Com-plete Initialization 291
Comment Control Statements 292

User-Written Exit 293
Command Format 293
UQ Menu 293
UQ Commands 294
A Command 295
D Command 298
L Command 301
M Command 302
O Command 302
Q Command 305
S Command 308
T Command 310
V Command (z/OS Only) 311
UQ Keywords 313
Keyword Values 313
AD Keyword 315
CL Keyword 316
DC Keyword 317
DE Keyword 318
DS Keyword 318
IN Keyword 319
JB Keyword 320
JC Keyword z/OS/XA (JES3) Only 321
JO Keyword z/OS/XA (JES3) Only 321
LL Keyword 322
LR Keyword 322
OC Keyword 323
OP Keyword 323
PP Keyword 323
PT Keyword 324
RL Keyword 324
RR Keyword 325
SC Keyword 325
SD Keyword 327
SI Keyword 327
SQ Keyword 328
WR Keyword 329
Entering Operator Commands (Control User) 329
Example 330
USERV - Library Service Utility (VSE Only) 331
USERV - Library Service Utility (VSE Only) 331
Overview 331
Command Format 331
Using USERV 333
Entering Commands 333
Positional Arguments 333
Keyword Arguments 334
Paging Requests 340
PF Key Assignments 340
Commands 341
DISPLAY Command 341

FILES Command 342
HELP Command 343
KEYWORDS Command 344
LABEL Command 345
LIST Command 346
PRINT Command 349
SPACE Command 350
VTOC Command 350
USPOOL - Printout Spooling Utility 353
USPOOL - Printout Spooling Utility 353
General PF Key Assignments 353
USPOOL Functions 353
List Queue 354
Printer Overview 357
Operate Printer (OPERATE Function) 359
Display Printout on Screen (SHOW or QUEUE Function) 360
USTOR - Storage Display Utility (Control User) 362
USTOR - Storage Display Utility (Control User) 362
Command Format 362
Main Storage Display 363
Main Storage Addressability Features 363
USTOR Commands 366
UTIMER - Com-plete Timer Services Utility 374
UTIMER - Com-plete Timer Services Utility 374
Command Format 374
Using UTIMER 375
Entering Operands 375
Serving Timer Requests 375
UTIMER Functions 375
Send Message 375
Handle Job (OS systems only) 375
Log Me Off 376
List Personal Services 376
List All Services 377
Restore Last Menu 377
Timer Exits 377
UZAP - Load Module Modification Utility 378
UZAP - Load Module Modification Utility 378
Command Format 378
The UZAP Menu 379
Reestablish Zaps from Previous Session 380
UUTIL - Menu-Driven Utility Functions 382
UUTIL - Menu-Driven Utility Functions 382
Command Format 382
Examples: 382
Overview of Functions 382
Function AI - Accounting Display 384
Function AI - Accounting Display 384
Function FK - PF Key Definition 386
Function FK - PF Key Definition 386

Function FM - File (DDN) Catalog Maintenance (Administrator) 388
Function FM - File (DDN) Catalog Maintenance (Administrator) 388
Access Method Options 388
ISAM Considerations (z/OS only) 388
BDAM Considerations (z/OS only) 389
VSAM Considerations 389
Using the File Catalog Maintenance Function 390
File Overview 391
Display Catalog Entry 393
Catalog a File Entry 393
Modify a Catalog Entry 400
Remove a Catalog Entry 401
Set File To BATCH Status 401
Set File To ONLINE Status 401
Open a VSAM File 401
Close an Open File 401
SHOWCB ACB Information 402
Function HC - Hardcopy Device Definition 403
Function HC - Hardcopy Device Definition 403
Setting the Hardcopy Definition 404
Function HE - Help System 405
Function HE - Help System 405
Function HM - Help Maintenance (Administrator) 407
Function HM - Help Maintenance (Administrator) 407
Listing Help Items 408
Help Menus 409
Help Text / Glossary 410
Function GL - Display Global Lib IDs 412
Function GL - Display Global Lib IDs 412
Function MM - Message Maintenance (Administrator) 413
Function MM - Message Maintenance (Administrator) 413
Update Message 414
Select Messages from a List 415
Function MO - Monitor Control (Control User) 417
Function MO - Monitor Control (Control User) 417
AI Subfunction 419
AS Subfunction 419
BA Subfunction 421
CM Subfunction 422
CS Subfunction 423
D2 Subfunction 424
FB Subfunction 425
Buffer % Used 429
FP Subfunction 430
IP Subfunction 430
OP Subfunction 431
PG (Task ("Processor") Groups) Subfunction 432
PL (Task ("Processor") List) Subfunction 434
PR Subfunction 436
QO (Queue Overview) Subfunction 437
RA Subfunction 439

RP Subfunction 440
SG (Thread Subgroups) Subfunction 441
SO Subfunction 443
TL (Thread List) Subfunction 444
TO Subfunction 446
TG (Thread Groups) Subfunction 447
TS Subfunction 449
TU Subfunction 450
UA Subfunction 452
US Subfunction 454
VS Subfunction 456
NQ Function - Monitor ENQs (z/OS and VSE only) (Administrator) 459
NQ Function - Monitor ENQs (z/OS and VSE only) (Administrator) 459
Thread ENQs / LOCKs 459
System ENQs 461
Selection Modes 464
Display Modes 467
Displaying Enqueue/Dequeue Conflicts 468
Function PW- User Password Maintenance 470
Function PW- User Password Maintenance 470
Function SD - SD File Display Utility (Administrator) 471
Function SD - SD File Display Utility (Administrator) 471
Function SI- User Site Information 474
Function SI- User Site Information 474
Function TO - Trace Options Maintenance (Administrator) 477
Function TO - Trace Options Maintenance (Administrator) 477
Function TT - TIBTAB Maintenance (Administrator) 478
Function TT - TIBTAB Maintenance (Administrator) 478
List 479
The TIB Parameter Menu 481
Show 482
Add 482
Update a TIB Definition 483
Update a TIB in the Active TIBTAB 483
Delete a TIB Definition 483
Free a "Deleted" TIB from the Active TIBTAB 483
Copy Contents of a TIB from the Active TIBTAB to a TIB Definition 483
Function TU - Terminal / User ID Group Maintenance (Administrator) 484
Function TU - Terminal / User ID Group Maintenance (Administrator) 484
Function UD - Maintain User Defaults 486
Function UD - Maintain User Defaults 486
Function UL - Maintain User Lib IDs 487
Function UL - Maintain User Lib IDs 487
Function UM - User Maintenance (Administrator) 489
Function UM - User Maintenance (Administrator) 489
User Maintenance Functions 489
Creating a New User Profile 490
Maintain COM-PASS Menu 495
Function UP - PF Key Defaults (Administrator) 497
Function UP - PF Key Defaults (Administrator) 497

Function ZA - Zap and Applymod Display (Administrator) 499
Function ZA - Zap and Applymod Display (Administrator) 499
ZO - Zap Overview 499
ZS - Display Status of a Single Zap 500
AL - Display Applymod List 501
AU - Display Used Applymods 502

Overview of all Utilities

This is a reference guide for all users of Com-plete, whether they be application programmers, terminal operators, system programmers, computer operators or system administrators. It describes all Com-plete online utility programs used to maintain your individual environment, access system objects, and maintain the Com-plete system as a whole, including all privileged functions.

Many of the utilities and functions described in this documentation require that you are specially authorized. A distinction is made beween the following:

- **General User:**
  All users of Com-plete have access to utilities and functions available to the general user. If a utility or function is not marked otherwise in this documentation, it is available to the general user;

- **Control User:**
  In order to have access to certain utilities and functions, you need control status. Those utilities and functions requiring control status are marked "Control User".

- **Administrator:**
  Users with administrator status have access to all utilities described in this documentation. Those utilities and functions restricted to the administrator are marked "Administrator" and require knowledge of the maintenance password.

Note that if the accounting/logon option is used, control status is assigned to you through the user ID. If the accounting/logon option is not used, control status is assigned through TIBTAB. The accounting facility and the use of the TIBTAB are described in the Com-plete System Programming documentation.

	Utility	Authorization	Description
	UATTACH - Start Background Program	General	starting an asynchronous (background) program to run in a thread in the Com-plete address space.
	COM-PASS - Parallel Transaction Utility	General	Maintain several parallel sessions with Com-plete at the same time.
	UCOPY - Print/Save Screen Utility	General	Print a copy of current screen at a printer or terminal.
	UDEBUG - Application Debugging	General	assists you in locating problem and enables you to bypass a specific problem temporarily
	UDD - DASD Display Utility (Control User)	General	Display of the contents of any record on a DASD device.
	UDS - Data Set Maintenance Utility (z/OS Only)	General	Examine and modify characteristics of disk-resident data sets and system catalog entries (z/OS only).
	UDUMP - Conversational Dump Display Utility	General	Display the content of online dumps.

	Utility	Authorization	Description
	UDYN - Dynamic Dataset Allocation and Deallocation (z/OS only)	General	Allocate and deallocate datasets to/from the Com-plete address space without having to restart Com-plete.
	UDZAP - DASD Modification Utility (Control User)	Control	Perform modifications to records on DASD devices
	UED - Line-Oriented Source Editor Utility	General	Edit source text using any available terminal device type with both input and output capability. This utility is designed specifically for use with hardcopy type devices.
	UEDIT - Text Editor Utility	General	Edit source text and submit jobs from local and remote terminals.
	ULIB - Program Catalog Maintenance Utility	General/Control	Maintain the Com-plete program catalog.
	ULOG - Logon/Logoff Utility	General	Establish and terminate sessions with Com-plete; define your password.
	UM - Message Switching/Printoutspooling Utility	General	Send messages to any terminal in the Com-plete network. Maintain and control the message switching and printout operations.
	UMAP - Terminal Mapping Utility	General	Dynamically create, modify, and test maps on a 3270-type or compatible terminal.
	UMATH - Mathematics Utility	General	Perform mathematical computations.
	UP - Terminal Paging Utility	General	View data in a terminal paging data set created by an online program.
	UPDS - Partitioned Data Set Maintenance Utility (z/OS Only)	General	Monitor and maintain members in a partitioned data set.
	UQ - System Job Queue Display Utility	General/Control	Obtain system information. Enter operator commands and display operator WTO and WTOR messages.
	USERV - Library Service Utility (VSE Only)	General	Display library members and partition labels (VSE).
	USPOOL - Printout Spooling Utility	General	Manage and distribute output to any online printer in the network.
	USTOR - Storage Display Utility (Control User)	Control	Display and modify storage in the Com-plete system.
	UTIMER - Com-Plete Timer Services Utility	General	Schedule certain functions.

	Utility	Authorization	Description
	UUTIL - Menu-Driven Utility Functions		Access functions via selection menus. Perform certain system monitoring and maintenance functions. The UUTIL facility replaces the following utilities: UDTIB, UHELP, UMSGF, UCTRL, UTHG, ULOGM, UZAP, UTERM, UPROF, UPF, USCHC, UGLIB, UPWD, UINFO, UDEF, USDLIB and ULIBID (see *Overview of Functions*)
	UZAP - Load Module Modification Utility	Administrator	Display information about currently implemented fixes. Display / modify applymod settings for the current Com-plete session.

Conventions

Requests from a terminal to execute an online program must normally be preceded by an asterisk (*). You can use different characters when invoking programs under COM-PASS. (See the section entitled *COM-PASS Terminal User Information*.) The asterisk as the first character signals to Com-plete that the information following is a request for execution of a user or Com-plete program. Note that utilities are considered to be Com-plete programs.

The general format for executing a program from a Com-plete terminal is:

```
*program
```

where *program* is the one- to eight-character name of the program to be executed. Note that this program must reside in the Com-plete program library.

In some cases, arguments can be passed to an online program when the program is initially called for execution, in the format:

```
*program argument
```

where *argument* represents one or more arguments to be passed.

In this documentation, each command that invokes a Com-plete utility program is shown in a statement illustrating the proper format for that command. Note that program names and arguments in upper-case must be entered exactly as illustrated. Lower-case arguments indicate that a substitution must be made if that argument is used. Valid substitutions for each argument are described in the appropriate section for each command.

Brackets, [ ], curly braces, { } , and vertical bar(s), |, are used to illustrate command syntax in this documentation.

Note:
Do not type brackets, curly braces, or vertical bars as part of the command.

Brackets, [ ], indicate that the keyword, argument, or value is optional. When more than one option is given within one set of brackets, you can select one or more.

Braces, { }, indicate that a choice of enclosed items must be made. Note that one, and only one, of the values between the braces must be supplied.

The vertical bar, |, separates the selectable options.

Example 1

The format for a typical ULIB request is:

```
*ULIB DIS,PGM=program
```

When you enter this command at a terminal, you must enter the upper-case words and symbols as shown. In addition, you must supply the name of the program for which information is requested.

An example of the use of this command is:

```
*ULIB DIS,PGM=USTOR
```

Example 2

The format for a ULIB catalog request for programs is:

```
*ULIB CAT,program[,RG=nK][,TL=n]
```

The square brackets indicate that the RG and TL keywords can be omitted.

A typical command to catalog a program is:

```
*ULIB CAT,DPBEAN,RG=12K
```

UATTACH - Start Background Program

This utility program can be used for starting an asynchronous (background) program to run in a thread in the Com-plete address space. UATTACH purely provides an interface for entering the data necessary for running the program:

- the name of the program,
- a parameter string to be passed to the program,
- the number of instances (how many times to start the program in parallel,
- a user ID for the asynchronous program to run with. Security is controlled by RACF / ACF2 / TopSecret based on surrogate authorization.

```
17:07:01        TID    12          Com-plete         User MBE          13.11.00
                    --- Start an Asynchronuous Program ---                UATT



Program name.....:
Parameter string.:


Start program.....   1 time(s)


If the program should run with a userID different from your own:
(You must be an authorized surrogate!)

User ID..........:





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

Note:
Programs that use device-dependent terminal-write functions must not be run asynchronously. See the description of function ATTACH in the *Com-plete Application Programming* documentation for more information about running asynchronous programs.

COM-PASS - Parallel Transaction Utility

COM-PASS parallel transaction handling means that you are able to start, suspend, and restart a transaction without the need to shut down or end other transactions. Note that when you suspend a transaction, it is temporarily saved and restored without loss of information. You can suspend and restart transactions by either using a function key or issuing a command.

You can have up to nine suspended transactions and one active transaction. The COM-PASS Main Menu allows you to monitor which transactions are suspended and enables you to restart suspended transactions.

This chapter covers the following topics:

- COM-PASS Features - Overview
- COM-PASS Standard Mode: The COM-PASS Menu
- COM-PASS Expert Mode
- COM-PASS TPF Mode
- Parallel Transaction Handling
- COM-PASS Commands

COM-PASS Features - Overview

The special features of COM-PASS are outlined in the following paragraphs.

COM-PASS Menus

The front-end utility menus, which provide you with an enhanced interface to the Com-plete utilities, are a major feature of the COM-PASS utility. A typical COM-PASS utility menu displays all of the commands applicable to the utility, as well as all operands and syntax requirements. The commands that you select and issue from the menus are syntax checked and an error message is returned if invalid syntax is detected.

The standard COM-PASS Main Menu, the Expert Mode Menu, as well as the TPF Mode Menu are discussed later in this section.

Online Help Facility

The Help facility consists of an overview of all commands and operands for a function, the syntax rules for its use, and a glossary of all terms that are associated with the function. All help information is related to the current version of Com-plete.

Note that you can extend the Help facility with your own user-written help system. For additional information, see the section *UUTIL - Menu-driven Utility Functions*.

User Profile Definition

COM-PASS allows each user to be defined to Com-plete in the form of a user profile. Your profile consists of:

- The Com-plete characteristics that would be defined if COM-PASS was not installed;
- The extent to which you can access and use Com-plete, COM-PASS, and user functions. You can also have a startup transaction defined that will be invoked immediately after you log on (defaults to USTACK);
- PF key settings;
- Default parameter settings;

Note that you can change only the PF key and default parameter settings in your user profile, unless you are a control user.

User Site Information

To select and display online information concerning an installation where COM-PASS is installed, you can use the SI function of the UUTIL utility. This information can be defined and modified by control users. For additional information on site-specific information, see the section *UUTIL - Menu-driven Utility Functions*.

The Security System

All COM-PASS users are defined to COM-PASS by a user profile, as described above. Part of your profile contains your transaction authority, which defines to what extent you are allowed to access and use COM-PASS, Com-plete, and user functions.

Each user can have up to 27 individual functions defined. These consist mainly of menus containing Com-plete and utilities that consist of several transactions. Your COM-PASS menu lists the functions defined for you, together with an information line concerning each function's use.

You will either be restricted to the up to 27 functions that can be defined for display on your COM-PASS main menu, or be allowed to use other functions as well.

The startup transaction is part of the security system and simplifies control over COM-PASS, Com-plete, and user functions. To take you to the main menu, the startup program is defined to be a special COM-PASS program called USTACK. For additional information on USTACK, see the section *Function UM - User Maintenance* in *UUTIL - Menu-driven Utility Functions*.

Mailboxes and Message Switching

COM-PASS also provides facilities for sending messages to a single user, a single terminal, a group of users, or a group of terminals. See the section *UM - Message Switching Utility* for more information.

The mailbox message system ensures that messages you cannot receive immediately are retained and will be delivered when you log on.

Entire Connection

If the appropriate version of Entire Connection is installed on a workstation, then COM-PASS supports the use of windows to display the current active and suspended programs. Please see the Entire Connection documention for further details.

Mode Type

Within COM-PASS, you can define different modes. This means you can select one of three menus, each with a different layout and a different subset of available functions. Select a mode using the M=<type> command. Possible mode types are:

M=S	Standard mode. The COM-PASS Main Menu offering Com-plete services available to you.
M=E	Expert mode. Alternative to standard mode, making work easier for you if you use a certain subset of utilities frequently.
M=T	TPF mode. Reserved for users of ADABAS TPF.

The following sections describe the three available modes listed above in detail.

COM-PASS Standard Mode: The COM-PASS Menu

In standard mode, once you have logged on successfully, the COM-PASS Main Menu is displayed (if your startup program is set to USTACK). An example of a COM-PASS Main Menu is shown in the following figure.

```
16:58:06        TID     6          COM-5.1          User MBE          21.04.97
                                -- COM-PASS --                                USTS
        Suspended Programs                          Program Services
 ---------------------------------       ---------------------------------------
 Programs     Name     C Level  PF       Service Description    Programs  ID  PF
 --------   --------   - -----  --       --------------------   --------  --  --
                          1              Storage display        USTOR      A
                          2              PDS functions          UPDS       B
                          3              Online control         UCTRL      C
                          4              JES functions et al    UQ         D
                          5              THE Editor             UEDIT      E
                          6              Online dumps           UDUMP      F
                          7
                          8
                          9              Map definition         UMAP       I

 Enter Input:                                                   HELP       ?

 ------------------------------------------------------------------------------
 LU Name: SHRDAEN       HC TID:               Recall: =         Language: 001
 Devtype: 3270 VS       Device:              Suspend: <  PA1    Case    : Upper
 Lines  : 24               Key: PA2             Jump: NO NO
 Columns: 80
```

The Main Menu is your personal view of Com-plete.

The fields pertaining to the suspended programs/transactions on the left side of this menu are explained in the following table:

Field	Meaning
PROGRAMS	Your installation’s name for each suspended program.
NAME	The (optional) name you can supply for a suspended program. **Note:** The character string you use for the name must not start with a "-"; this would cause the program to be removed from the suspended program list. See the subsection *Deleting a Suspended Transaction*.
C	Case setting of the level (L=lower, U=upper). You can modify this.
LEVEL	The level number of the suspended programs. When you invoke the "LEVEL=" function, COM-PASS marks the level you select with "<", and the last suspended level with "*".
PF KEY	The Program Function (PF) key setting for the program. You can specify the PF Key setting either on this screen or on the User Profile definition screen (using function UM of UUTIL).

The right side of the COM-PASS screen lists a standard set of program services available to you. Note that the system administrator at your site defines the programs that you can use.

Field	Meaning
SERVICE DESCRIPTION	A description of the program/function.
PROGRAMS	The name of the program/function.
ID	A single character identifier for the program/function.
PF KEY	The optional PF key assigned to the program/function.

The terminal-specific information at the bottom of the COM-PASS Main Menu screen is summarized in the following table.

Field	Meaning
LU Name	The user terminal name; if a VTAM network, the name of the VTAM device.
Device Type	The terminal classification as defined by Com-plete.
Upper/Lower Case	Indicates the default case; (you can modify this by using the UP and LOW commands). For additional information, see the section *Direct Commands*.
Lines	The number of addressable lines on your terminal.
Columns	The number of addressable columns on your terminal.
Suspend	The character you use to suspend one transaction and initiate another, and the key you can use to suspend the current transaction.
Hard Copy Device	The name of the hard copy device assigned in UUTIL.
Jump	The jump character and key used to suspend the current transaction and automatically move to the next.
HC TID	The TID of the hard copy device assigned in UUTIL.
Recall	The character you use to recall a suspended transaction and suspend your current transaction

COM-PASS Expert Mode

COM-PASS Expert Mode is an alternative way to work with COM-PASS. If you work within a subset of Com-plete utilities and use those utilities frequently, expert mode allows more complete tracking of utility usage.

Entering "MODE E", "M E", or "M=E" in the command line places you in this mode; entering "MODE S", "M=S", or "M S" takes you back to the standard COM-PASS Main Menu. The COM-PASS Expert Mode menu is shown in the following figure.

```
17:01:54        TID     6          COM-5.1          User MBE          21.04.97
                                -- COM-PASS --                              USTE
                                                                          001
 Level       Suspended                                                  Menu
 Id          Programs   Last UEDIT     Last UQ        Last UPDS        Programs
 ----------  --------   -------------  -------------  -------------    - --------
 1                                                                      A USTOR
 2                                                                      B UPDS
 3                                                                      C UCTRL
 4                                                                      D UQ
 5                                                                      E UEDIT
 6                                                                      F UDUMP
 7
 8
 9                                                                      I UMAP

 Enter Input:                                                  HELP       ?

 ------------------------------------------------------------------------------
 LU Name: SHRDAEN       HC TID:                Recall: =         Language: 001
 Devtype: 3270 VS       Device:               Suspend: <  PA1    Case    : Upper
 Lines  : 24               Key: PA2              Jump: NO NO
 Columns: 80
```

In Expert Mode, the last used parameters of the UEDIT, UQ, and UPDS utilities are displayed for each level at which they were called.

Note that Expert Mode does not have all the features of the standard mode. If you need to use the standard mode features, simply change the mode back to standard (with the MODE S, M=S, or MS command). Note also, however, that when you switch between the two modes, no information is lost.

Last Default System

If you log off while in Expert Mode (using LOGOFF or *ULOG OFF), your last defaults (that is, the fields you marked in the Expert Mode menu) are automatically saved to a "last default" SD file. When you log on next time, you will still be in Expert Mode, and the defaults marked on the menu are still in place.

Note that you can force a reading of your last set of defaults while in Expert Mode by using the READ command, or force a saving of the defaults by using the SAVE command. For additional information on the READ and SAVE commands, see the section *Direct Commands*.

Special Features

Expert Mode also gives you the ability to change the columns "LAST UEDIT", "LAST UQ", and "LAST UPDS".

If you place the cursor in one of these fields, and no program was suspended on that level, the corresponding utility will be started as follows:

UEDIT	as "UEDIT XX(YYYYYYYY)", when XX(YYYYYYYY) is specified in the column, where XX is the LIB ID and YYYYYYYY is the member.
UQ	as "UQ Q,JB=XX", when XX is specified in the field.
UPDS	as "UPDS L,XX", when XX is specified in the field, or: "UPDS D,XX(YYYYYYYY)", when XX(YYYYYYYY) is specified in the column, where XX is the LIB ID and YYYYYYYY is the member.

COM-PASS TPF Mode

COM-PASS TPF Mode is the mode reserved for ADABAS TPF users. The menu shows the PROGRAM SERVICES programs available to the user. Only the programs shown on the menu as it appears to you are available, for example:

```
17:05:32        TID     6          COM-5.1          User MBE          21.04.97
                                -- COM-PASS --                              USTT
                                                                           001

                 Service Description    Programs   ID   PF
                 --------------------   --------   --   --
                 Storage display        USTOR       A
                 PDS functions          UPDS        B
                 Online control         UCTRL       C
                 JES functions et al    UQ          D
                 THE Editor             UEDIT       E
                 Online dumps           UDUMP       F


                 Map definition         UMAP        I

 Enter Input:                                                 HELP       ?

 ------------------------------------------------------------------------------
 LU Name: SHRDAEN       HC TID:               Recall: =         Language: 001
 Devtype: 3270 VS       Device:              Suspend: <  PA1    Case    : Upper
 Lines  : 24               Key: PA2             Jump: NO NO
 Columns: 80
```

Working in TPF mode, certain restrictions apply:

- When you log on, you are presented with the TPF mode menu. You cannot change the mode or select any user-defined menu using the M= command. If you are in Standard or Expert mode, you can select TPF mode (M=T), but once in TPF mode, you cannot switch back, you must log off first.

  Note:
  To log off from TPF mode, use the LOGOFF command; ULOG OFF is not available.

- No suspended programs appear on the TPF mode menu, as you cannot suspend programs in TPF mode and return to the TPF mode menu, but you can use the Jump feature to move among programs listed in the menu. The Suspend and Recall characters should therefore not be set and used, as they may lead to unpredictable results.

Parallel Transaction Handling

As a COM-PASS user, you can carry out the following operations from any COM-PASS, Com-plete, or user function (but note the restrictions for TPF mode users mentioned above):

- Start a transaction from the COM-PASS Main Menu;
- Start a transaction from another transaction, automatically suspending the current transaction;
- Suspend a transaction and return to the COM-PASS Main Menu;*
- End a transaction and return to the COM-PASS Main Menu;
- Restart a suspended transaction from the COM-PASS Main Menu;*
- Restart a suspended transaction from another transaction;
- Delete a suspended transaction;*
- Temporarily deactivate COM-PASS.*

  * Not available in TPF mode.

Note:
When you are using UEDIT or UED on several levels, COM-PASS ensures that you can recover the edit files on each level (including the level on which you are working) after a system failure.

Note that in this section, the term "transaction" refers to any utility, program, or function.

Starting a Transaction from the COM-PASS Main Menu

COM-PASS now makes up to 27 menu programs available for each user under the heading "Program Services" on the Main Menu. If you have more than nine programs, COM-PASS displays the current page and the total number of pages just to the right of "Program Services". For example, "1/3" means that you are on page one of three.

Note:
In TPF mode, only those program services displayed on the menu are available to you.

Provided that your User Profile permits the use of the transaction, there are five ways to start a transaction from the COM-PASS Main Menu:

1. Enter "*XXXXXX YYYYYY" on the command line under the *Enter Input* prompt. For the result, see the section *Direct Commands*;
2. Enter "n=XXXXXX YYYYYY" on the command line (not available in TPF mode). For the result, see the section *Direct Commands*;
3. Enter one of the characters from the "ID" column of "Program Services" on the command line. This starts the program whose name is to the left of that letter;

4. Press one of the PF Keys listed in the "PF" column under "Suspended Programs". This starts the program whose name is defined in that line;

5. Place the cursor in a field in the "Service Description" or "Programs" column of "Program Services", which contains a transaction you want to initiate, and press ENTER. This starts the program that is defined in that line.

Starting a Transaction from a Transaction

You can invoke a transaction and automatically suspend your current transaction using one of the following three methods:

Suspend *	Suspend Operation - To suspend (and save) the current transaction and invoke a new transaction, enter "<name", where "name" represents the transaction you wish to invoke. Note that if you do not supply a valid name, the current program will be restarted.
Recall *	Recall Operation - To suspend the current transaction and restart a suspended transaction, enter "=name", where "name" is either a level number (1 through 9), a name assigned to the level number, or a transaction name.
Jump	Jump Operation - to jump from the current transaction to the next transaction, enter ">" (jump char) or press the jump key. In mode S and E, jumping is only possible from one suspended transaction to the next suspended transaction. In mode T, jumping is possible from one program service to the next. The transaction from where you jumped to the next transaction is suspended internally, so every information will be available, if you jump in next time.

* Not available in TPF mode.

Note that using the suspend and recall operations differs from invoking another transaction by utility name (for example, invoking *ULIB while in UEDIT). Invoking a utility in this way *will not* suspend/save the current transaction; instead, the new transaction will overwrite the current transaction.

Suspending a Transaction and Returning to the COM-PASS Main Menu

Not available in TPF mode, but see the description of the Jump feature above.

To suspend a transaction and return to the Main Menu, press the SUSPEND key (PA1 or TRQ), or enter the SUSPEND ONLY command. Note that you can restart the suspended program at any time (see the subsection *Restarting a Suspended Transaction from the COM-PASS Main Menu* below).

You can also suspend the transaction at a specific level by assigning the transaction to a level before you start the transaction using the "LEVEL=x" special command (see the section *Direct Commands*).

Ending a Transaction and Returning to the COM-PASS Main Menu

Provided that USTACK is the startup transaction defined for you, you may end most transactions and return to the Main Menu by pressing the CLEAR key or entering "*EOJ" in the command line. If you are in NATURAL, enter "FIN" at any "MORE" or "NEXT" prompt .

Restarting a Suspended Transaction from the COM-PASS Main Menu

Not available in TPF mode, but see the description of the Jump feature above.

Suspended transactions are shown on the left half of the COM-PASS Main Menu. You can restart any one of them using one of the following three methods:

1. Enter one of the level numbers from the "Level" column of "Program Services" on the command line under the *Enter Input* prompt. This restarts the transaction whose name is in that line;

2. Press one of the PF Keys listed in the "PF" Column under "Suspended Programs". This restarts the transaction whose name is defined in that line;

3. Place the cursor in a non-empty field in the "Program" or "Name" column under "Suspended Programs", and press ENTER. This restarts the transaction whose name is defined in that line.

In TPF mode, use the Jump feature to return to an internally suspended program service from another.

In Standard and Expert mode, you can also restart (recall) a suspended transaction while in another transaction by using the convention "=name", where "=" is the default recall character. Note that "name" can be either:

- A level number;

- A name assigned to a number;

- A transaction name.

If no program is located that corresponds to "=name", COM-PASS interprets the command as "<name" (that is, SUSPEND command), and the last suspended program is restarted.

For additional information on the RECALL and SUSPEND commands, see the section *SUSPEND/RECALL/JUMP Commands*.

Deleting a Suspended Transaction

Not available in TPF mode.

To delete a suspended transaction displayed on the COM-PASS screen, simply type a "-" over the first character of the suspended transaction name.

Deactivating/Activating COM-PASS

Not available in TPF mode.

To temporarily deactivate COM-PASS, enter the "DISABLE" command on the COM-PASS Main Menu.

Note:
You can only use the DISABLE command if there are no programs suspended.

- To reactivate COM-PASS, enter "*USTACK".

COM-PASS Commands

There are three types of COM-PASS commands:

1. SUSPEND/RECALL/JUMP commands;
2. Direct commands;
3. Immediate commands.

The COM-PASS commands are discussed in the following sections.

SUSPEND/RECALL/JUMP Commands

The SUSPEND and RECALL commands are used during parallel transaction handling to suspend the current transaction and either invoke a new transaction (SUSPEND) or restart another suspended transaction (RECALL). These commands are discussed in the following table.

Command	Meaning
RECALL	Suspends the current transaction and invokes a suspended transaction.
	Consists of the recall character ("=" is the default) optionally followed by either:
	a. A level number, indicating that the transaction on that level is to be restarted. b. A user-defined name, indicating that the transaction associated with this name is to be restarted. c. A utility name, indicating that the transaction identified by the utility name is to be restarted.
	Note: If you only enter "RECALL" on the COM-PASS Main Menu, the last suspended transaction will be restarted.
	Note that if the transaction cannot be identified, the RECALL command will have the same effect as the SUSPEND command. (See below.)
RECALL LAST	Suspends the current transaction and recalls the last suspended transaction.
	Consists of the recall character entered twice, e.g., "==".
	Note that repeated use of the RECALL LAST command toggles between the two last-suspended transactions.
SUSPEND	Results in the saving/suspension of the present transaction and the execution of the reminder of the data supplied (for example, "<UQ Q" results in suspension of the present transaction and execution of the utility UQ with the argument Q).
	Consists of the suspend character ("<" is the default) followed by any named Com-plete transaction call (that is "*utility argument(s)").
SUSPEND ONLY	Suspends and temporarily saves the present transaction, and starts the COM-PASS startup transaction (the default is USTACK).
	Consists of the suspend character entered twice: "<<".
SUSPEND KEY	Pressing the SUSPEND key (either PA1 or TRQ) is equivalent to entering the "SUSPEND ONLY" command. (See above.)
JUMP CHARACTER	Results in saving/suspending of the present transaction and the execution of the next suspended transaction. (for mode TPF in execution of the next transaction from the "program services".)
JUMP KEY	Pressing the JUMP KEY is equivalent to entering the "SUSPEND CHARACTER". (See above.)

Note:
The RECALL and SUSPEND commands are not available in TPF mode.

For additional information on these functions, see the section *Parallel Transaction Handling*.

Direct Commands

On the COM-PASS Main Menu, you can enter direct commands on the COM-PASS command line under "Enter Input:". Note that you need not be concerned with case here, because COM-PASS transforms all input on the command line to upper case.

Note:
You can always omit the "=" between the keyword and parameter or replace it by a blank, if the keyword can be uniquely identified.

The following table contains all the available COM-PASS direct commands. The column headed Mode indicates for which mode type the command is available: S (Standard), E (Expert) and/or T (TPF).

Command	Mode	Description
ENTER-key	S,E,T	Moves one page of Suspended Programs forward (if you have more than nine programs suspended). **Note:** If the STAY mode is on, you will remain on the same page. (The STAY mode is indicated by a "*" before the page number, e.g., *1/3.) See the STAY command later in this table.
*XXXXXX YYYYYY	S,E,T	Starts the program XXXXXX on the first free level and (optionally) passes the parameter(s) YYYYYY to it.
A YYYYYY -	S,E,T	Starts the program marked with a character I YYYYYY A through I in the "ID" column on the right side of the COM-PASS Main Menu and (optionally) passes the parameter(s) YYYYYY to it.
1 - 9	S,E	Recalls the program marked with a number 1 through 9 in the "Level" column on the left side of the COM-PASS Main Menu. **Note:** In Expert Mode, if no program is suspended on that level, but the "Last Parameters" field on that line contains parameter information, the appropriate utility is started using that parameter.
1=XXXXXX YYYYYY	S,E	Starts program XXXXXX on a free level (1 9) that you assign it, and (optionally) passes parameter(s) YYYYYY to it.
9=XXXXXX YYYYYY		

Command	Mode	Description
+ or ENTER-key	S,E	Moves one page of Suspended Programs forward (if you have more than nine programs suspended). Note that this command also works in STAY mode.
+nnn	S,E	Moves nnn page(s) of Suspended Programs forward (if you have more than nine programs suspended).
-	S,E	Moves one page of Suspended Programs backward (if you have more than nine programs suspended).
-nnn	S,E	Moves nnn page(s) of Suspended Programs backward (if you have more than nine programs suspended).
CONTROL or CTRL	S,E,T	Starts the control facility *UCTRL.
DISABLE	S	Disables the stacking of programs. **Note:** This is only possible if no program is suspended. when you use this command, the COM-PASS screen disappears. To enable stacking again, enter "*USTACK".
HC=XXXXXX	S,E,T	Sets the hard copy device for this session. Note that XXXXXX can be either the hard copy device name or its TID number.
HC ?	S,E,T	Calls *USCHC, which enables you to choose a hard copy device from a list.
HC OFF	S,E,T	Removes the hard copy setting.
LEVEL=n	S,E	Sets the current suspend level to n (a number 1 through 9). COM-PASS marks this level with "<=" in the "Level" column. The next program is started on that level.
LOGON	S,E	Performs a Com-plete logon to the XXXXXX,YYYYYY userid XXXXXX. YYYYYY is the password and can be omitted. If so, you will be prompted for a password by *ULOG. **Warning:** **This command ends the session you are currently in.**
LOGOFF	S,E,T	Performs a Com-plete logoff (*ULOG OFF).
LANG=nnn	S,E,T	Sets the national language to be used for COM-PASS menu and messages. Valid values are 1-225.
LOW	S,E,T	Sets the terminal to lower case.
UP	S,E,T	Sets the terminal to upper case.
MODE E, M=E, M E	S	Invokes COM-PASS Expert Mode.

Command	Mode	Description
MODE S, M=S, M S	E	Invokes the COM-PASS standard mode. **Note:** When you switch modes betwen E and S, all stacking information remains as it was. The only thing that changes is the map layout.
MODE T, M=T, M T	S,E	Invokes COM-PASS TPF Mode. **Note:** When you switch to mode T you cannot switch back to any other mode until you log off.
PAGE=n or P=n	S,E	Sets the page number of Suspended Programs to the number "n" (if you have more than nine programs suspended).
READ	E	Reads the last defaults for the Expert Mode menu from the last defaults SD file.
RECALL	S	Recalls the last suspended program.
SAVE	E	Saves the last defaults from the Expert Mode menu into the last defaults SD file.
STAY or STAY ON	S,E	Disables the automatic scrolling of pages of Suspended Programs. When STAY is on, a "*" will appear before the page number, e.g., *1/3.
STAY OFF	S,E	Enables the automatic scrolling of pages of Suspended Programs.
ALT	S,E,T	Sets the terminal to alternate screen size (if defined).
NALT	S,E,T	Reset the terminal to non-alternate screen size (if defined).
NTCMW ON/NTCMW=ON	S,E,T	Sets NATURAL Workstation Communication mode.
NTCMW OFF/NTCMW=OFF	S,E,T	Resets NATURAL Workstation Communication mode.
SETnn -	S,E,T	Terminates the indicated (nn) suspend level. This is equivalent to typing a "-" over the first character of the suspended program.

Immediate Commands

Com-plete also provides a series of utility functions that you can execute at any time. You can type in these immediate commands either while on the COM-PASS Main Menu, or while a utility program is in operation. Note that each function executes only once, so when you press the ENTER key a second time, you are returned to the processing environment that was previously active, that is, to the COM-PASS Main Menu or the utility program.

Each immediate command has the following format:

`*command.parameter`

where *command* is a single letter that represents the command to be executed. Valid commands are described below.

The command identifier is followed by a period (.), which is followed by a parameter. The form of the parameter depends on the command used. Each parameter is defined in the following sections with its corresponding command.

C Command

The C command should be followed by a two- or four-hexadecimal digit parameter. The digits are interpreted as an IBM System/370 Operation Code and the following information is displayed:

- The name of the command;
- The command mnemonic;
- The hexadecimal operation code;
- The command format (RR, RX, RS, SE, S, or SS);
- The operand format [R1, D1 (L,B1), D2 (X2,B2), etc.];
- Attributes.

Note that this information is displayed using the conventions from the *IBM System/370 Principles of Operation* documentation.

Command letter:	C
Parameter:	A two- or four-digit hexadecimal number
Example:	*C.1A

Output:	`NAME: ADD REGISTER`
	`MNEMONIC: AR`
	`OPCODE: 1A`
	`FORMAT: RR`
	`OPERANDS: R1, R2`
	`ATTRIB: CONDITION CODE IS SET`

Note:
The RRE and SSE instruction formats are not supported.

In the case of S-format instructions, you should input a four-digit op-code. In all other cases, a two-digit op-code is required.

E Command

The E command displays the hexadecimal digits that denote the EBCDIC values of the characters in the parameter string.

Command letter:	E
Parameter:	From 1 to 40 printable characters
Example:	*E.ABCXYZ
Output:	`CIC2C3E7E8E9`

Note:
The system may change letters from lower to upper case.

N Command

The N command displays the hexadecimal equivalent of the decimal number given as parameter.

Command letter:	N
Parameter:	An unsigned decimal number from 0 to 2147483647
Example:	*N.667646
Output:	`A2FFE`

Note that the parameter must not be longer than 31 bits.

P Command

The P command displays the PF key assignments (if any) for the level on which it is issued.

Command letter:	P
Parameter:	ALL (optional)
Examples:	*P.ALL *P.

Output:	`PF01/D = LR=LAST-10`
	`PF02/D = LR=FIRST`
	`PF03 = DS=1`
	`PF04 = SI=CC`
	`...`
	`PF24/D = Q,JB=_____`

Notes:

1. A "/D" following a PF key name indicates that this key’s data is displayed on the terminal before execution. The data can be modified, and the command is executed when ENTER is pressed.
2. When a PF key with the display option contains underscores (see the PF24 key in the above example output), the cursor is automatically positioned to the first underscore when the

data is written to your terminal.

3. If the system defaults set by the system administrator have not been overridden using the FK function of UUTIL, the system defaults are displayed.

UCOPY - PRINT/SAVE Screen Utility

The online utility UCOPY is designed for use with 3270-type terminals. It enables you to print a copy of a screen on a printer terminal (or any terminal) or save the screen in an editor work file.

The PRINT and SAVE functions of UCOPY are controlled by the USCHC utility.

This chapter covers the following topics:

Command Format

To invoke UCOPY, press the PA2 key, or use the command format for UCOPY, shown below:

```
*UCOPY
```

Functional Considerations

For a printout, the default screen-to-hardcopy terminal is used for output. To save the screen in a work file, the HC function of the UUTIL utility should have the field "Save screen for edit" set to Y (yes). You can then recover the screen using the Full Screen Editor menu. See *UUTIL - Menu-driven Utility Functions* for more information.

You can use the utility UCOPY at any time. Note that since it uses the message switching/printout spooling task of Com-plete, execution of UCOPY does not cause the online program in use to terminate.

Note:
When you are using a program such as UEDIT, press ENTER first in order to avoid losing screen input before invoking UCOPY.

Example

While displaying data with UEDIT, you wish to print the currently displayed screen. The UUTIL default TID for your terminal is TID 11. The following command request will, without terminating UEDIT, cause the displayed screen to be printed on TID 11:

```
*UCOPY
```

UDEBUG - Application Debugging

The Com-plete Application Debugger UDEBUG assists you in locating problem areas within an application and, where applicable, enables you to bypass a specific problem temporarily by altering various parts of the program's environment, such as storage contents, register contents or the actual program code itself.

Highlights of UDEBUG functionality include:

- Stop programs at specific points in their execution
- Alter storage
- Give instruction steps
- Trace MCALL
- Disassemble assembler code (not yet implemented)
- Modify assembler code (not yet implemented)
- Full screen conversational interface
- Powerful DSECT display facilities

This chapter covers the following topics:

- Overview
- Restrictions on the use of UDEBUG
- UDEBUG User Interface
- UDEBUG Commands

Overview

What follows is a description of the terms associated with UDEBUG, followed by a description of how to use UDEBUG.

The UDEBUG Session

The UDEBUG session is simply the running of the program UDEBUG on a terminal attached to Com-plete. UDEBUG itself is a Com-plete application program with the ability to set dynamic hooks into another Com-plete application. In the case of a COM-PASS user, it must run as one of the user levels, in which case the application to be debugged may run on a different level or even a different terminal. In the case of a non-COM-PASS user, only one program may be run at a time. In this case, UDEBUG is this program and can only test an application to be debugged on a different terminal.

The Debugger

The Debugger is the session which is debugging an application program.

The Testing Terminal

The Testing Terminal is the terminal running the UDEBUG session. If the Testing Terminal is running COM-PASS, then there can be more than one UDEBUG session active on different levels. These different sessions can have their own individual Test Terminals and/or Test Levels.

The Test Terminal

The Test Terminal is the terminal running the program being debugged. A terminal can only be a Test Terminal for one UDEBUG user, even if the Test Terminal is running COM-PASS. However, if the Test Terminal is running COM-PASS, the Testing Terminal can test on more than one level. This simply requires that the Testing Terminal have a UDEBUG session active for each of the levels on the Test Terminal which are to be debugged.

The Test Level

The Test Level is the level running the program to be debugged. Only one debug session can be active on a particular Test Level at a time. When the Debugger is running COM-PASS and the Testing and Test Terminals are the same, the Test Level can be any level on the terminal except the level upon which the Debugger is running and level 0. If the Test and Testing Terminals are different, the Test Level can be any valid level for the Test Terminal. If the session with the program being debugged is not running COM-PASS, then the test level must be zero.

The Test Session

The Test Session is the Test Terminal / Level combination where testing is or will take place.

Breakpoints

A breakpoint is a point in a program’s execution at which the program should be stopped to enable the Debugger to determine what the status of the program’s operating environment is. At this point, the Debugger can change the program’s environment, change the program itself or change the logical flow of the program.

UDEBUG Breakpoints

As stated previously, UDEBUG enables the user to stop the application program at certain points in the coding. This is done by setting breakpoints in the code. When a breakpoint is set at a certain point in the code, the Debugger receives control immediately before the instruction on which the breakpoint is set is executed. At this point, you can check the environment at the point at which the code was stopped and alter storage or the actual code before the code is executed. You can also restart the program being debugged at another address.

Breakpoints can be set to execute a certain number of times before giving control to the Debugger, and can also be set to stop giving control to the Debugger after it has been executed a certain number of times. They can be set in RESIDENTPAGE programs and in programs which are in the thread or will be loaded into the thread at some point in the execution of the program being debugged.

A breakpoint always has an owning TID/LEVEL combination and a Test TID/LEVEL combination. The owning combination identifies who set the breakpoint while the Test Combination identifies for which terminal/level combination the breakpoint has been set that is, the Test Terminal and the Test Level. If the program being debugged terminates for any reason, the breakpoints set by the Debugger will remain intact. If the Debugger terminates for any reason, the breakpoints are either deleted, or, if this is nmot possible, flagged for later deletion. Any user running on anything other than the Test Terminal Level combination who reaches the breakpoint is dispatched normally, though with extra CPU overhead to bypass the breakpoint.

When one or more breakpoints have been set, the Test Terminal and Level for that UDEBUG session will be unmodifiable until all the breakpoints are again deleted. The Debugger can then change the Test Terminal and/or the Test Level.

Warning About Breakpoints

Basically, the only way it can be known that a piece of storage will be executed is when the area of storage is fetched by instruction fetch processing. This applies even if the area is not even a valid instruction. Rather than restrict where a breakpoint may be set, UDEBUG simply ensures that the address where a breakpoint will be set is halfword aligned. This can conceivably cause problems, for example, if a breakpoint was set on the "to" address of an MVC instruction, it would change the location to where the MVC would take place. This will at best cause a storage exception or at worst, it will cause writing to an unknown storage area and enable the program to continue processing as if a certain piece of storage has been set. For this reason, care must be taken when a breakpoint is set to ensure that the breakpoint does not corrupt the program being tested.

Implicit Breakpoints

As stated previously, the user can set points in the program at which the program should stop. These are explicit breakpoints. We speak of implicit breakpoints when you have requested a function of UDEBUG which requires the program to be stopped at a specific point.

Currently, an implicit breakpoint is set for the following UDEBUG functions:

- Instruction Tracing
- MCALL tracing

Generally speaking, the implicit breakpoint only exists when the breakpoint is required, and it is deleted when the debugged program comes off the breakpoint. This means that, for example, during MCALL tracing, when the program issues a MCALL, an implicit breakpoint is set, causing the Debugger to get control. When the program is next dispatched and comes off the breakpoint, the breakpoint is deleted.

Breakpoints in Storage

It is also possible to set a breakpoint in storage obtained by an application program in the thread. This facility is available for high level language compilers that provide a trace function which simply branches to a storage area, in which there is simply a branch back. Another reason could be to enable the debugger to restart a program after a wild branch. This type of breakpoint must be set with care and the following should be noted carefully:

- The breakpoint can only be set in an area which has been getmained in the thread. An attempt to set a breakpoint in a free area of the thread results in the breakpoint being flagged as invalid.
- Please be aware that setting a breakpoint in a storage area requires moving an SVC instruction to this location, overwriting the contents of the area for two bytes. By the same token, when the breakpoint is removed, the "instruction" moved from this area is moved back to leave the area in the state in which UDEBUG found it.
- It is your responsibility to ensure that the location the breakpoint is set is not altered while the breakpoint is active. If this occurs, the UDEBUG breakpoint SVC is deleted and therefore the breakpoint has no effect.
- When the Test Program terminates, any breakpoints that are set in storage are flagged as "Dormant". This means that the control block identifying the breakpoint is still there, but,the breakpoint is not set until the Debugger reactivates the breakpoint by issuing the AT command. This is to ensure that the test program can get the storage and build the environment before UDEBUG attempts to set the breakpoint again.

Redispatching a Breakpointed Program

When a program being debugged is on a breakpoint, if the Test and Testing Terminals are not the same, the only way to restart a breakpointed program is by issuing the UDEBUG command GO. However, if the Test and Testing Terminals are the same, you can issue the GO command or, if you have stacked the UDEBUG session, you can simply restart the program by using any of the commands from the COM-PASS screen to select the suspended Test Level.

MCALL Tracing

When MCALL tracing is active for a Test Session, when the program being debugged is about to execute an MCALL, the Debugger is given control. That is to say that the Debugger is given control BEFORE the MCALL is executed, giving the Debugger the chance to change the parameter options or even to bypass the MCALL altogether. Please note that when MCALL tracing is active, ANY program running on the Test Session will cause control to be passed to the Debugger when a MCALL is issued.

Instruction Tracing

Instruction tracing enables the Debugger to step through a program instruction by instruction. That is to say that once instruction tracing is activated, the Debugger gets control prior to the execution of every assembler instruction. As this is simply an implicit breakpoint, you can make any changes during an instruction trace which can be made at an explicitely defined breakpoint.

Instruction tracing can be active with no effect. That means that the Debugger may have set the option for the Test Session, but nothing happens. This is because instruction tracing can only be effected when the program to be debugged has terminated at a breakpoint, be it implicit or explicit. The UDEBUG code checks when coming off the breakpoint if the option is set or not. When it is set, it is then activated for the next instruction after the breakpoint.

The instruction tracing can only go as far as tracing within application programs. This means, that if the application issues a SVC call or an internal Com-plete nucleus call, this cannot be traced, as breakpoints cannot be set in these places. In these cases, UDEBUG sets the next "instruction trace" breakpoint at the instruction following the instruction that cannot be followed.

UDEBUG Symbols

To make programs more meaningful, various sysmbols or labels can be defined. In the same way, to make certain addresses and displays more meaningful in a UDEBUG test environment, symbols can be set in various ways. During the following description, reference is made to level 1, 2 and 3 symbols. How these categories related to UDEBUG is described later.

Standard Equates

A user can set up equates within a session using the EQUATE command (described later). This command enables an eight character identifier to be set to a particular address in storage. When this equate is later referenced alone or within an expression, UDEBUG uses the address to which the identifier was equated to resolve the request. An equate can also be given a length so that during disassembly or when listing the equate, UDEBUG "knows" the length the data that the equate describes. In UDEBUG terms, an equate is a level 1 type symbol.

Load Module Equates

Many load modules consist of more than one CSECT which can cause them to be very large. In any case, working with a linked module can be tiresome given the fact that initially, only the actual module name will be known. UDEBUG enables you to "know" each of the CSECT within a linked module, so that when the load module name is resolved, you can reference the various CSECTS within the module directly. This is achieved by using the LMODULE command (described later). This causes the load module to be read from a specified load library and an internal equate set up for each of the CSECTS within the load module.

The CSECT equates are set up relative to the start of the actual load module, so once the load module is not relinked, it can be loaded anywhere and UDEBUG succeeds in addressing the various CSECTs correctly. This is particularly useful in the case of a relocatable load module. In UDEBUG terms, the load module name which is loaded is a level 1 symbol, while the actual CSECT equates themselves are level 2 symbols.

Testran Defined Symbols

Current assemblers running under z/OS enable you to produce what is called "testran" records. These records describe each of the various DSECTs and CSECTs that exist within a module and the various labels within the DSECT or CSECT. UDEBUG can read the testran symbols using the READ command (described later). The member name being read becomes a UDEBUG level 1 symbol, while each DSECT or CSECT becomes a UDEBUG level 2 symbol. The fields defined within a DSECT or CSECT then become UDEBUG level 3 symbols.

The symbols defined as a result of the READ command can then be used to see the structure of DSECTS and/or CSECTS including field names, lengths and offsets. When a particular CSECT or DSECT is resolved, the contents of the various fields can also be seen. Resolution of CSECTS and DSECTS is discussed later.

Local and Global Symbols

The number of symbols you wish to define may run into thousands. It would not be practical to keep all these in the UDEBUG thread, therefore the concept of global and local symbols is available. A globally defined symbol is available to all UDEBUG users running within that Com-plete until it is deleted. In this way, commonly used control blocks and/or load modules can be defined globally.

You can define global symbols. However, in the case where one DSECT has changed while all of the "common" DSECTS available globally have not, you can simply define the changed DSECT locally. UDEBUG will always use local definitions before attempting to use global definitions. Therefore, with local symbols, you can effectively "front end" the globally defined symbols.

UDEBUG Symbol Levels

UEBUG "sees" all symbols as being of a certain level. This level determines whether a symbol is already defined or not. For example, a load module name is a level 1 type symbol and therefore no other load module of that name can appear on level one. Of course the same load module name can exist locally and globally as the global level 1 and the local level 1 are seen as being different.

As an example of how the second level functions, assume two NATURAL V21 linked nuclei named TSTNUC1 and TSTNUC2. Both of these have a CSECT named ACMDRIV and both can be defined, as the level 1 names are different. However, TSTNUC1 could not have the CSECT ACMDRIV linked twice (even though this makes no sense it is technically possible). In this case, UDEBUG only honours the first occurrence, and issues an error message about the second. In the same way, level 3 labels can only exist once within the level 1/2 combination, meaning, for example, that many DSECTs can have the same field name.

Resolution of Symbols

Where a symbol is not equated directly to an address, UDEBUG attempts to find where the address is. Firstly, UDEBUG searches for an equate for the highest level name for the symbol. For example, if a CSECT is referenced, UDEBUG attempts to find an equate for the load module or testran member to which the CSECT relates. If no equate is found, UDEBUG attempts to find a using statement for the entity. If it is a level 3 field, a level 2 CSECT or DSECT is searched for. In the case of a level 2 DSECT or CSECT, the section itself is searched for. When symbols cannot be resolved either via an equate or a using, depending on the circumstances, you are either informed, or the area displayed relating to the symbol contains the "not resolved" UDEBUG character.

As various different modules and/or members used to create symbols may have the same symbols defined as a result of the LMODULE or READ command, the user has the ability to specify which particular symbol is being referred to. This is done as follows:

```
Lev1name.Lev2name
```

where

Lev1name	is the name of the module which was read by the LMODULE command or the member name read by the READ command.
Lev2name	is the symbol you want to reference.

If the symbol is entered on its own and exists more than once, the first occurrence is taken.

If you wish to always work with the same level 1 name as opposed to having to type in "Lev1name.Lev2name", if a Test Program name is entered, UDEBUG uses a symbol related to that program. When it fails to find this, it simply uses the first occurrence as before.

Addressing Mode

In systems capable of 31 bit addressing, there is always a question as to how to interpret an address. When a program is on a breakpoint, UDEBUG uses the AMODE in which the program is running to determine how it should interpret addresses. However, when the program is not at a breakpoint, all addresses are interpreted as 31 bit addresses.

Implicitly Defined Symbols

To avoid having to set certain standard symbols, UDEBUG sets some defaults at various points in the execution. The following symbols are set at startup and re-evaluated periodically to ensure that they are still correct.

DCOMREG	COMREG for the Com-plete where UDEBUG is running
DTIB	The TIB on which the UDEBUG session is running
Resident programs	For each resident program defined in the Com-plete system, UDEBUG sets up an equate for the name of the program, its address and length. As these are continuously updated, additions to the residentpage list via the PGM operator command, or any refreshes of programs should be reflected almost immediately.

The following symbols are defined as soon as a program is active on the test TID/Level. They are deleted as soon as the level is freed.

DUPCB	UPCB of the program being debugged.
THDS	Start of thread for the program being debugged.
THDE	End of thread for the program being debugged.
THXS	Start of thread extension for the program being debugged.
THXE	End of thread extension for the program being debugged.
programs	The root program name and any modules loaded into the thread will be set up as equates.

When a breakpoint is reached, UDEBUG deletes and sets the following symbols. If these names are already used, they are deleted and set as indicated below.

BPPSW	Breakpoint PSW address
BPRx	Breakpoint register, where ’x’ is ’0’ to ’9’ or ’a’ to ’f’.
BPTIBA	The TIB buffer related to the program under test.

Address Expressions

When attempting to address a storage area, UDEBUG can read various types of expression to enable the Debugger to find the address space. The given expression is evaluated and the result of that evaluation is used as the absolute address.

Of course, absolute addresses can also be used. All numeric data entered must be preceded by the hex character to indicate hexadecimal data, or the decimal character, which indicates a decimal number. Hex is the default. If the value is preceded by the relocate character, the resultant value is calculated relative to the relocated address as set by the RELOC command. The decimal, hex and relocate characters and how to set them are described later in this section. The following indications can be used in an expression.

Hex/decimal values	Absolute values
Hex/decimal addresses	Absolute addresses
Hex/decimal offsets	Relative offsets from a point set by the user
Symbols	Globally or privately defined symbols
Arithmethic operators	Plus (+) and Minus (-) enable the Debugger to add or subtract from addresses.
Brackets "(’ and ’)"	When brackets are specified, the result of the expression within the brackets is used as the address of the fullword. This value is then used in the expression in place of the brackets and its contents.

Note:
The arithmetic operators and brackets mentioned above can be customized to suit your requirements. This procedure is described later.

Storage Display and Modification

When a breakpoint is set, the program code will contain the UDEBUG SVC. In this case, the Debugger is not concerned with what is really at this point but what is logically at this point, that is, the replaced instruction. Therefore, when UDEBUG displays such a piece of storage, the logical contents of the storage are displayed, that is, the storage is displayed with the replaced instruction, and not the UDEBUG breakpoint SVC.

By the same token, if you wish to update a piece of storage and a breakpoint exists at the point where the modification is to take place, UDEBUG handles this by updating the breakpoint in such a way that the instruction to be exectuted is modified in core. This becomes apparent when the debugged program leaves the breakpoint.

Modifying thread storage is a little more complex. If UDEBUG is running with no program active on the test level, absolute addresses are treated as such, that is, if you enter an address which is contained in a Com-plete thread, you will see the actual data which is there at that point in time. However, when a program is active on the level, and you enter an address which is in the program’s thread somewhere, this is resolved logically in the rolled out image of the thread.

There are cases where you receive the message that the storage is not available. This can occur if the image of the test program could not be written to the roll buffer; in this case contact your system programmer. It can also occur if the UDEBUG session was started on a test level AFTER that level was last active. In this case, Com-plete would not have known that the test level should have been retained in the roll buffer. This can be corrected by simply activating the

test level once.

Abend of a Test Program

When a program which is being tested abends, the Debugger is notified and can look into the thread. However, the program cannot currently be restarted after an abend. When a Com-plete dump is written to the Com-plete SD dataset, the thread will have been rebuilt to how it logically looked. This means that instructions upon which a breakpoint is set will have been rebuilt. It also means that bad breakpoints which were set in the middle of an instruction, for example, will also have been reset. If one of these bad breakpoints has caused the abend, it will not be obvious from the dump.

If any other dump of storage is taken other than a Com-plete dump, for example, if applymod 73 is on, the thread will be dumped as it was at the time of the abend, thus all UDEBUG SVC instructions will still be in place. If confusing results are being obtained, the problem can be further traced with such a dump as it will reflect the true status of storage at the time of the abend.

Confirmation Processing

UDEBUG provides full screen interfaces to display any information that the Debugger may require. In most of these screens, you can update some or all of the areas. When this occurs, if confirm processing is on, the changed field will be highlighted and protected and you are asked to confirm the change. If the CONFIRM UDEBUG command is then issued, the updates are made as requested. If anything else is entered, the updates are forgotten. This facility can be turned on or off using the SET command.

If you want to use Confirm processing, but do not wish to have to go through two input operations, you can enter the CONFIRM command on the command line when the updates are made. This is then taken as confirmation that the updates should be done. The easiest way to do this is to set the CONFIRM command on a PF Key. In this way, the updates can be made and the PF Key pressed to cause the updates to be accepted. Any changes made inadvertently followed by an enter causes normal confirm processing to take place.

Restrictions on the use of UDEBUG

There are a number of restrictions and warnings that users must take into account while using UDEBUG. Some of these situations are actually disallowed by UDEBUG, however, in certain cases, UDEBUG cannot determine that this is the case and may accept and perform a certain task. However, the results will then not be as expected.

Execution of Breakpoints

A breakpoint may only be activated if the area of code where the breakpoint has been set is executed. When a breakpoint is requested by a user for a program, UDEBUG has no way of knowing whether an instruction will be executed or not as the only way to determine this is to execute the program. Where a breakpoint has been set and the code not executed, you should first ensure that that section of code is actually being executed in your application.

It would be possible to at least ensure that the address where the breakpoint is set an instruction. This was not implemented, as many people use an invalid hex instruction code to cause a program to abend. In some cases, users may wish to set their breakpoint on this 'instruction' to take another course of action. It is also possible that a breakpoint could be set on the address operator of an instruction which could represent a valid instruction code. This means that it is impossible to accurately determine if the storage at a location is actually an instruction so to avoid ambiguities whereby sometimes it would fail a request and other times not, it was decided not to include any checks whatsoever.

Note also that the setting of a breakpoint based on an offset into a module to be loaded into the thread can only occur when the module is actually loaded into the thread. In the event that the offset does not exist in the module, this will only be known when the module is loaded and as such, can only be marked as an invalid offset at that time. If such a breakpoint fails to be triggered, check if the breakpoint has been flagged as invalid using the BPLIST function.

Instruction Stepping

The instruction step functionality provides the facility to step through a program on an instruction by instruction basis. For each application program instruction, a breakpoint is triggered which will force the thread image to be rolled out of the thread. This will cause problems with programs which are event based whereby they issue a request for which they expect to be posted back when the request is completed.

If the ECB in use is inside the thread, it is likely that the serving posting back the indication that a request has completed will post the storage location in the actual thread storage area and not the actual ECB in the user's rolled out thread copy. This will result in the POST being lost to the expectant application due to the fact that it's copy of the ECB is never updated.

This may also cause other problems for the current occupier of the thread whose storage at the equivalent location will be overwritten by the post intended for the application program being debugged.

There is no way to know or protect against this. Therefore, application programs of this nature must be tested with extreme care and no breakpoints either explicit or implicit set between the requesting of a service and the wait for that service to complete.

UDEBUG User Interface

This section explains the user interface to UDEBUG. It describes the general layout of the screens along with the available options. The various UDEBUG maps are then shown and described in detail.

UDEBUG Session Startup

UDEBUG must be started with the command

```
*UDEBUG
```

UDEBUG sets up the standard user environment as follows:

1. If you are a COM-PASS user, then the current terminal is set up as the Test Terminal if another user is not testing on it. In the case where another user is testing on the terminal or if you are not a COM-PASS user, the Test Terminal is set to 0 and you must set it to a valid terminal before testing can begin.

2. For a COM-PASS user, the next available level is set as the Test Level. This means that the UDEBUG level plus one is taken to the maximum levels available. When this is exceeded, it wraps around to take level 1. If a Test Session already exists on this level, or you are a non COM-PASS user, the Test Level is set to zero.

3. Symbols are set up for the following:

   - All residentpage programs (symbol names will be the program names)
   - COMREG (symbol name DCOMREG)
   - The UDEBUG Session TIB address (symbol name DTIB)

4. The hardcopy TIB for the Testing Terminal is set as the UDEBUG hardcopy device when available.

5. The UDEBUG nucleus is loaded.

6. The profile named after the your user ID is executed from the default profile DD/DLBL (COMDBPRF) if the DD/DLBL exists and a profile exists for you. For more details, see the description of the PROFILE command.

7. Any data entered after the *UDEBUG is assumed to be command data and is passed to the UDEBUG command handler.

When the above has completed, you will generally be presented with the UDEBUG session information screen UDB0. This will NOT be the case if data entered along with the *UDEBUG command or a command in the profile has a different map to be selected. For example, if a DUMP command was contained in your profile, the first screen you will see is the storage display map UDB1.

The UDEBUG program can optionally be specified as a STARTUPPGM (see the sysparm by that name) which will cause UDEBUG to be attached with the user ID SYSUSR. You must then add a member name SYSUSR to the profile dataset which can contain UDEBUG commands to build global symbols that are available to all UDEBUG users. The profile must always end with the EOJ command to ensure that the program terminates cleanly. The following example will cause the Com-plete nucleus name COMPLETE to be read and each CSECT name in the nucleus to be set up as a global symbol.

```
LMOD COMPLETE * GLOBAL
EOJ
```

See also the description of the LMODULE command below.

Error and Information Messages

At UDEBUG startup and for various command combinations while in the UDEBUG session, more than one message may be required. Depending on the number of lines on the terminal, UDEBUG saves each message to the maximum available lines on the UDEBUG message screen UDB2. When more than one message is required, the first message is displayed in the UDEBUG map message line with ’+++’ as the message ID and not ZDB. To see the additional messages, simpy enter MESSAGES on the command line. If more messages than can fit on the messages screen, this is indicated by the last message on the screen. All error messages following the last displayed message are discarded.

Program Function Keys

Generally, when ENTER-Key is pressed from a screen, anything typed on the command line is first interpreted and the relevant command executed. The screen handler then gets control to handle input from the screen (if any) and the screen is redisplayed with any appropriate messages and/or updates. This of course will not be the case if any of the commands entered cause the display screen to be changed.

When you press a program function key, either the user-specific keys are used, or the global PF keys for the system. Global PF keys are maintained by the system adminstrator, while you can add personal PF keys using the UPROF utility. When a PF key is defined, the data as defined for the PF key is taken as a command and passed to the UDEBUG command processor. If you press an undefined PF key, an appropriate message is displayed. The bottom line of the screen displays the first 5 characters of the command allocated to the PF key. When the PF key is not defined, no text appears under the PF key header. If the PF key is defined with the "display" option, the contents set for the PF key are displayed on the UDEBUG command line.

Customizing Characters and Options

For all functions, UDEBUG has default characters and options. However, you can set these as required. This can be done via your user profile. The following lists the operand name for the SET command which enables the character to be set together with the standard default for the character. When the operand ON is given for one of the following options, the default is set when possible. When OFF is specified, the character is disabled or x’00’ is displayed when the option is a display option.

For the following characters, all of the characters must be unique within the list.

Operand	Default	Description
Decchar	Z	Identifies a numeric string as decimal
Hexchar	X	Identifies a string as hexadecimal
Relchar	#	Identifies a string as relative to the address set by the RELOC command
Parmdel	.	Delimits parameters within an operand, for example, NATCOM21.ACMDRIV identifies NATCOM21 as the level 1 and ACMDRIV as the level 2 symbol within an expression
Opdel	,	Operand delimiter. This character or blank must be used to delimit operands within a UDEBUG command
Cmddel	;	Command delimiter. This character delimits UDEBUG commands on the same command line
Pntldel	(	Pointer left delimiter. This indicates the start of an expression which is used to get an address. The expression between this pointer and the Pntrdel delimiter is evaluated and the address that the expression points to is used as the result, as opposed to the result of the expression itself
Pntrdel	)	Pointer right delimiter. This closes the procedure started by the Pntrdel character
Taddrid	*	When the current top address is to be used in an expression, this character can be used to avoid the need of retyping the address in full
Addchar	+	Indicates that two values are to be added together
Subchar	-	Indicates that two values are to be subtracted from each other

The following characters do not need to be unique as they are used in a different context to the above characters.

Operand	Default	Description
DEFchar	*	When a UDEBUG command has more than one positional operand and you wish to use the default for one or more of the operands, this character can be used in the operand position
NOTRes	-	If UDEBUG attempts to display storage based on a symbol and the symbol cannot be resolved, this character is used to fill the area
NOTAlloc	.	If UDEBUG attempts to display storage and the storage is not allocated, this character is used to fill the area
NOTACocc	=	If UDEBUG attempts to display storage and you cannot access the storage, this character is used to fill the area

UDEBUG Screens

General Format

The following is the format of all UDEBUG screens on a 3270 model 2 device (24 lines by 80 columns). Each of the areas which appear on all screens are described once here.

```
Message line ..............................................................
Time         TID -tid"              Inst-ID         User -userid"         Date
                             --- -Screen Name" ---
Screen Headings
Command line ..............................................................
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Function keys assignments .................................................
```

Message Line	This line is used by error and informational messages. UDEBUG messages have the format ZDBnnnnn, where nnnnn is a message number. If UDEBUG has more than one message to display, the first message is displayed in this line with an identifier of '+++nnnnn'. This indicates that there are more messages which can be seen by displaying the Messages Screen.
Time	This field contains the time of day in the format HH:MM:SS that the screen was sent to the terminal.
Tid	This fields contains the Terminal ID Number in Com-plete on which the UDEBUG session is running.
Instid	This field contains the Installation ID for the Com-plete running the UDEBUG session. This ID is set at Com-plete startup by the INSTALLATION sysparm.
Userid	This field contains the user ID of the user logged onto the terminal runing the UDEBUG session.
Date	This field contains the date that the screen was sent to the terminal. If applymod 61 is not set, the date has the format: MM/DD/YY. If applymod 61 is set, the date has the format DD.MM.YY.
Screen Name	This field contains the title of the UDEBUG screen currently displayed. Each UDEBUG screen has a title (see the example screens later in this section).
ID	This is the UDEBUG map identifier for the screen currently displayed. This ID has the format UDBx, where x is an internal identifier for the screen.
Screen Headings	When the individual UDEBUG screens require headings for the various fields. they are displayed here.
Command Line	UDEBUG commands can be entered in this line. When there are no errors in a screen display, the custor is positioned here.
Function Key Settings	You can use the globally defined PF keys or a set of PF keys customized set for yourself. When defined, UDEBUG takes the first five characters of the command defined for each PF Key and displays it here. If nothing is defined, nothing is displayed. The Enter key is always used to enter data.

The UDEBUG Session Information Screen (UDB0)

This screen is displayed as the result of the *UDEBUG call from COM-PASS and contains displays relevant data related to your UDEBUG session currently being run. You can modify various fields to change the criterea for the current session.

```
08:22:26        TID    17           COM-5.1.          User MBE          04/08/97
                          ---  Session Information  ---                     UDB0
 Test Information-   User Settings----------------   Miscellaneous-------------

 TID            17   MCALL Tracing                N   Defined Symbols         42
 Luname   SHRDAEN    Instruction Tracing          N   Defined Breakpoints      0
 Level           2   Confirm                      N
 Program             Bump Storage                 N
                     Default Character            *
                     Hardcopy Luname
                     Hardcopy TID                15 ‡‡‡
```

MCALL racing	This indicates whether MCALL tracing is active for the TestSession or not. "Y" indicates that it is active and "N" indicates that it is not active. This can be altered at any time during the UDEBUG session. However, it will only be effective the next time a MCALL is issued from the Test Session.
Instruction Tracing	This indicates whether UDEBUG instruction tracing is active for the Test Session or not. "Y" indicates that it is active and "N" indicates that it is not active. This can be altered at any time during the UDEBUG session. However, it will only be effective if the Test Session is on a breakpoint or after the next time the Test Session reaches a breakpoint.
Confirm	This indicates whether UDEBUG Confirm processing is active for updates. "Y" indicates that it is active and "N" indicates that it is not active. This can be altered at any time during the UDEBUG session and is effective immediately.
Bump Storage	This indicates whether UDEBUG Bump processing is active for UDEBUG Screens. "Y" indicates that it is active and "N" indicates that it is not active. With UDEBUG Bump Processing active, on screens where there is more than one screen page of information, if you press ENTER-Key without typing in any data, the following screen of data is displayed. This can be altered at any time during the UDEBUG session and is effective immediately.
Hardcopy Luname	This is the name of the hardcopy device to which UDEBUG will route any hardcopy output it may generate as a result of UDEBUG commands. This can be altered at any time during the UDEBUG session and is effective immediately.
Hardcopy TID	This is the Terminal ID of the hardcopy device to which UDEBUG will route any hardcopy output it may generate as a result of UDEBUG commands. This can be altered at any time during the UDEBUG session and is effective immediately.

Miscellaneous

This column contains any other data relevant for the UDEBUG session. Fields in this column cannot be updated.

Defined Symbols	This is the number of symbols that are locally defined for this UDEBUG session. This will never exceed the maximum defined at Com-plete startup by the SYMTAB sysparm.
Defined Breakpoints	This is the number of breakpoints that are currently defined for this UDEBUG session. It does not include implicit breakpoints or breakpoints that have been flagged for deletion. It contains breakpoints that are dormat. Please refer to the section on Breakpoints for more details.

The UDEBUG Dump Storage Screen (UDB1)

This screen is displayed when you press the appropriate PF key (in our example above, PF6). The display shows your storage in hex and character format with the current breakpoint registers for reference, if the Test Session is currently on a breakpoint. It also enables you to change this storage.

```
08:23:44        TID    17          COM-5.1.           User MBE          04/08/97
                              ---  Dump Storage  ---                        UDB1
 Address  Relative Hex contents----------------------- Char contents--- BP Regs-
 00000000          040C0000 810B58F0 00000000 80000000 ..  a..0    .
 00000010          00FDD030 00000000 070E0000 00000000  .ü.    ..
 00000020          078C2000 0001DCF4 078C2000 0001D8B4 ...  ..4...  .Q.
 00000030          00000000 00000000 070E0000 00000000         ..
 00000040          00000000 00000000 00000000 00FDD030              .ü.
 00000050          00000000 00000000 040C0000 810AD4A8         ..  a.My
 00000060          040C0000 00CA5EF0 00080000 87DC51D0 ..   .;0 .  g..ü
 00000070          00080000 87DC61B0 040C0000 810AF700  .  g./...  a.7
 00000080          00000000 00011202 000400C8 00040011      ... . H . .
 00000090          0001D000 00052000 00D3D512 00000105  .ü  ..  LN.  ..
 000000A0          00000000 010B2F88 00000000 00000000     ...h
 000000B0          00000000 00000000 0001022F 00FD67E8          ... ..Y
 000000C0          28000000 00000000 00000000 00000000 .
 000000D0          00000000 00000000 00000000 00000000
 000000E0          00000000 00000000 00000000 00000000
 000000F0          00000000 00000000 00000000 00000000
 00000100          00000000 00000000 00000000 00000000                  Key 0FP

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

To cause the first address to be displayed to change, simply enter expressions on the command line or use cursor positioning. When the cursor is placed in either the Address or Relative line, this line becomes the top line and the top address is altered accordingly. When the cursor in entered in either of the contents fields, the byte on which the cursor has been placed becomes the top address for the screen.

Meaning of the information by column heading:

Address

This is the address of the storage which is being displayed. The fields in this column cannot be altered.

Relative

This is the offset of the storage relative to the relocation address provided via the RELOC command. If the relocation address is 0, that is, if the relative address is equal to the actual address, nothing is displayed in this column. The fields in this column cannot be altered.

Hex Contents

This column contains the hex representation of the storage found at the address indicated in the Address column. For easier, it is represented as four eight byte areas, each representing four bytes of storage. When the storage does not exist, the field or portion of the field is filled with the UDEBUG "not allocated" character (default is ".") and is set using the SET command.

The hex data fields can be modified to cause the storage area itself to be updated by UDEBUG. In a case where storage does not exist, or only a portion of the four bytes to of a hex field exist, the field is protected. That means, that the field is only available for update when all four bytes exist.

Character Contents

This column contains the character representation of the storage found at the address indicated in the "Address" column. This output is translated according to the utility output translation table for the terminal. When the storage does not exist, the field or portion of the field is filled with the UDEBUG "not allocated" character (default is ".") and can be set using the SET command.

The data can be modified to cause the storage area itself to be updated by UDEBUG. In a case where storage does not exist, or only a portion of the sixteen bytes exists, the field is protected. That means that the field is only available for update when all sixteen bytes exist. Care must be taken when updating storage via this field, because for any update, all sixteen bytes are written back to the appropriate storage area. When the sixteen bytes contain non-character output and have been translated, it is possible that invalid hex values are written back to storage. It is therefore recommended that storage only be updated via the character representation when all sixteen bytes are valid characters.

BP Regs

This column contains sixteen 8 byte fields which display the hex representation of the contents of each of the breakpoint registers when the user is on a breakpoint. The top line represents R0, the next R1, and so on, up to RF. When you are not on a breakpoint, these fields are blank. These fields are not modifiable. In order to modify the breakpoint registers, you must go to the Breakpoint Information screen.

Key

This field gives information about the storage key of the page corresponding to the first address displayed on the screen. There are three positions to this field. The first is always filled and contains the storage protect key of the storage represented by "0" to "F". The second position indicates if the storage is fetch protected or not. When the storage is fetch protected, it contains the character "F", and when it is not fetch protected, it contains a blank. The third position contains a "P" if the storage area is page protected, and a blank if it is not. If the storage being displayed on the screen relates to two pages, the information only relates to the first page displayed.

The UDEBUG Error Messages Screen (UDB2)

When UDEBUG has more than one message to be displayed, only the first is displayed on the message line. You can then go to this screen by pressing the appropriate PF key (in our example, PF9) to see the other messages. Each information line on the display contains one message. No input options are available on this screen.

```
08:25:03        TID    17          COM-5.1.          User MBE          04/08/97
                            ---  Error Messages  ---                        UDB2

 COMZDB0007 (A) Invalid operand ABC for SET command
 COMZDB0052 (A) RELOC command issued successfully
 COMZDB0006 (A) Missing operands for AT command









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

The UDEBUG Breakpoint Information Screen (UDB4)

This screen is used to display information about a breakpoint when the user program is on a breakpoint. It can be used for information purposes, but also to change the running environment of the Test Program.

```
COMZDB0030 (A) Currently on breakpoint id 'BP1' at UCTRL+x00000060
16:46:04        TID    13          COM-5.1.           User JPO          02/03/97
                     ---  Breakpoint Information   ---                     UDB4
Breakpoint data-----  BP  Contents   Data-------------------------------------
Addressing Mode  24   PSW 07CC0000 00309060 FFFFFFFFFFFFFFFF * 90ECD00C18CF41E0
State            S    R0  00000000   040C0000810B58F000000008000000000FDC63000
Protect Key      C    R1  00008038   0000803E000000016140404040404040404040404000
Exception Masks       R2  00008000   0034F000006400A000016E48008DC4B8008D429800
 Fixed Point     N    R3  0021B000   070C000084710910847108020004837000008000000
 Decimal         N    R4  0000B8B8   078C0000847042DE000000000005510000008000000
 Exponent        N    R5  0034F820   0000000030C000010000000000000308301002010000
 Significance    N    R6  00000000   040C0000810B58F000000008000000000FDC63000
Condition code   0    R7  00000000   040C0000810B58F000000008000000000FDC63000
                      R8  00000000   040C0000810B58F000000000000000000FDC63000
Id          BP1       R9  00000000   040C0000810B58F000000000000000000FDC63000
Information           RA  00000000   040C0000810B58F000000000000000000FDC63000
                      RB  00000000   040C0000810B58F000000000000000000FDC63000
                      RC  00000000   040C0000810B58F000000008000000000FDC63000
                      RD  0030F6C0   0000000000000000000000000000000000000000
                      RE  0000802E   1F110AC800004C000A0E0000803E00000001614000
                      RF  00309000   47F0F060E4C3E3D9D3404040F4F5F040F0F561F100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

Meaning of the information according to column header:

Breakpoint data

This column contains information relating to the breakpoint on which the test program is currently sitting, together with information about the state of the user.

Addressing Mode	This indicates the address mode that the user program is running in. Under non-XA capable systems, this is always 24. You can optionally change the addressing mode when the test program is on a breakpoint. When this is done, the next time the test program is dispatched, it will have the new address mode as set by you.	
State	This indicates the state that the user program is running in.	
	S	Supervisor state
	P	Problem Program state
	You can change this, causing the test program to be in the specified state the next time it is dispatched.	
Protect Key	This indicates the protect key that the user program is running in. This contains a value from "0" to "F", depending on the protect key found in the PSW. You can change this, causing the test program to be in the specified set protect key the next time it is dispatched.	
Exception Masks	These fields indicate whether various exceptions will cause a program interrupt or not. Possible values:	
	N	if the exception occurs, it is ignored
	Y	an exception will cause a program interrupt. These exceptions relate to the four exception masks found in the PSW. You can change one or more of these settings to cause the program to run with the exception condition active or inactive the next time it is dispatched.
Condition Code	These field contains the last condition code set by the test program. It can be a number from 0 to 3. You can set this condition code to one of the desired values to cause the program to see a different condition code the next time it is dispatched.	

Id	This field contains the identifier of the breakpoint on which the test program is sitting. For an explicit or user-defined breakpoint, this is the ID specified on, or generated by the AT command. For implicit breakpoints such as those for instruction tracing or MCALL tracing, the following values may be found here:	
	INSTTRCE	Indicates an instruction tracing breakpoint.
	BP-MCALL	Indicates an MCALL breakpoint.
	BP-OSSVC	During user program processing, some OS SVCs area is trapped and satisfied by Com-plete. When MCALL tracing is active, these will also cause a breakpoint to occur before the request is satisfied.
	*ABEND**	Indicates a user program abend breakpoint.
Information	For various breakpoints, additional information is provided in this field as follows:	
	BP-MCALL	The MCALL request being issued is displayed.
	BP-OSSVC	The OS Macro name associated with the SVC or the SVC number is displayed.
	*ABEND**	The abend code is displayed.

BP Contents

This column contains the contents of the PSW and registers for the breakpoint as indicated on each line. The PSW cannot be explicitly changed on this screen. However, alterations in the Breakpoint Data column or a parameter on the GO command will obviously cause the PSW to be altered. The register contents can be altered, the changes take effect when the test program is next dispatched.

Data

This column shows the storage at the location pointed to by the PSW or register on the appropriate line. The register contents are interpreted according to the address mode of the user program. If the address does not exist, the UDEBUG "not allocated" character is displayed.

The UDEBUG Symbol Display Screen (UDB5)

This screen displays locally and globally defined symbols as a result of the SYMBOLS command. As the screen must display many different symbols, all the headings do not make sense for all symbols. A number of examples are therefore given, one for a display of level 1 symbols, one for a display of level 2 symbols and one for a display of level 3 symbols. A description of the fields as they are displayed for the first example with any differences noted for the subsequent examples.

Apart from the FORWPAGE and BACKPAGE commands, the top line to be displayed can be selected by placing the cursor anywhere on the desired line and pressing ENTER.

Level 1 Example Display:

```
16:51:03        TID    13          COM-5.1.          User JPO          02/03/97
                           ---  Symbol Display  ---                        UDB5
Type        D/Csect            Label            Equ/Lmod/Memb           Scope G
Symbol-- Mult--- Type Lngth- Disp---- Addr---- Contents------------------------
U2DBMAP4       1 EQU  2600   00000000 04CE45D8 47F0F060E4F2C4C2D4C1D7F4F4F5F4 >
U2DBMAP5       1 EQU  6384   00000000 04CE2710 47F0F060E4F2C4C2D4C1D7F5F4F5F4 >
U2DBMAP6       1 EQU  3496   00000000 04D29220 47F0F060E4F2C4C2D4C1D7F6F4F5F4 >
U2DBNXTO       1 EQU   352   00000000 0470A040 47F0F060E4F2C4C2D5E7E3D6F4F5F3 >
U2DBOFF        1 EQU  1024   00000000 04CE14E8 47F0F060E4F2C4C2D6C6C640F4F5F3 >
U2DBOSIO       1 EQU  1136   00000000 00337240 47F0F060E4F2C4C2D6E2C9D6F4F5F3 >
U2DBOSLD       1 EQU  1816   00000000 04CE18E8 47F0F060E4F2C4C2D6E2D3C4F4F5F3 >
U2DBPRLG       1 EQU   264   00000000 047002A8 47F0F060E4F2C4C2D7D9D3C7F4F5F3 >
U2DBPROF       1 EQU  1328   00000000 04715340 47F0F060E4F2C4C2D7D9D6C6F4F5F3 >
U2DBREAD       1 EQU  2752   00000000 04BE1330 47F0F060E4F2C4C2D9C5C1C4F4F5F3 >
U2DBSET        1 EQU  1392   00000000 047003B0 47F0F060E4F2C4C2E2C5E340F4F5F3 >
U2DBSTOR       1 EQU  1384   00000000 04714658 47F0F060E4F2C4C2E2E3D6D9F4F5F3 >
U2PRINT        1 EQU  1104   00000000 00706608 47F0F060E4F2D7D9C9D5E340F4F5F0 >
U2STHD         1 EQU   328   00000000 0003C018 47F0F060E4F2E2E3C8C44040F4F5F0 >
CCOMBLKS       1 MEM     0   00000000          -
COMPLETE       1 LMOD  130 K 00000000          --------------------------------

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

Meaning of the information by column heading:

Type

This field can be used to search for a specific type of level 1 object or restrict the display to certain types of objects. The following values are valid for this field:

Equ	Equates
Lmod	Load modules
Memb	Members

D/Csect

When displaying symbols that have been built as a result of the reading of Testran output from the assembler, there will be one or more sections defined. You can select which section is to be displayed by entering the name in this field. The third example in this section shows an example of selecting the DCOMREG DSECT built from a member called CCOMBLKS.

Label

This field enables the user to specify criteria which determine the first symbol in a list to be displayed. You can specify an absolute name, which must exist to be displayed, or a generic string, by entering a prefix following by an asterisk. For example, if you enter "ABCDE", the label ABCDE must exist, but if you enter "ABCDE*" in this field, the first symbol found starting with the letters ABCDE are the first to be displayed.

Equ/Lmod/Memb

Here you can enter the name of the level 1 symbol with which you want to work. The display depends on what sort of symbol is first found. For a more restrictive search, you can specify a type in the TYPE field. For example, if an Equate exists for ABCDE and ABCDE is also a load module which has been processed via the LMOD command, two level one symbols will exist. The first may be the Equate and the second the Load module. If this is entered with no type, a list of equates is displayed starting with ABCDE. However, if it is entered with the type LMOD, the level two items for the load module ABCDE are displayed as in the example for LMOD COMPLETE in this section.

Scope

This indicates the scope of the first symbol displayed on the screen. Possible values:

G	Global symbol
L	Local symbol.

This is purely for information and cannot be changed.

Symbol

This column contains the name of the symbol for which information is being displayed on the same line.

Mult

This is the multiplication factor. For non-level 3 symbols, it is meaningless and is set to 1. For level 3 symbols, this contains the multiplication factor for the storage type being displayed. An example of where this is meaningful can be seen on the display for the DSECT DCOMREG.

Type

This describes the type of symbol being displayed. The following are the possible mneumonics and what they indicate.

MEM	Member name read using the READ command.
LMOD	Load module read using the LMODULE command.
EQU	Equate defined explicitely or implicitely.
USNG	Using statement defined by the USING command.
CSCT	Code Section (CSECT).
DSCT	Data Section (DSECT).
COMN	Common Section.
INST	Program instruction.
CCW	Channel Command Word.
CHAR	Character data.
DBCS	Double Byte Character Set data.
HEX	Hexadecimal data.
BIN	Binary data.
FW	Fullword data.
HW	Halfword data.
FLPS	Floating point (short) data.
FLPL	Floating point (long) data.
FLPE	Floating point (extended) data.
ADDR	Address type data.
Y	’Y’ type data.
S	’S’ type data.
VCON	V-Constant data.
PD	Packed decimal data.
ZD	Zoned decimal data.

Length

This shows the length that the data was defined with. For example, a load module’s length is the total length of all CSECTs in the load module. When then length is greater than 9999 bytes this is represented in Kbytes, indicated by a K. If it is greater than 9999 KB, it will be represented in Mega Bytes and indicated by an M.

Disp

This shows the displacement from the base area. For a field in a DSECT or CSECT, it is the displacement from the section. For a module within a load module, it is the displacement from the first module.

Addr

When the symbol can be resolved, this contains the address where the symbol starts. If it cannot be resolved, this field is blank.

Contents

When the symbol can be resolved and the address exists, the contents of the area described by the symbol are displayed here in as meaningful a fashion as possible. For symbols that are longer than the area available, a ’>’ sign indicates that there is more data to be displayed. In the case of data fields, they are displayed according to their type. For example, a CHAR-type field is displayed in character, a HEX type field in hexadecimal and so on. When there is no longical display format for a type, a hexadecimal dispaly is given. In the case of data fields with a multiplier greater that one, only the first iteration is displayed. When the symbol cannot be resolved, the UDEBUG "not resolved" character (Default is "-") is displayed. If the symbol is resolved but the storage not found, the "not allocated" symbol is displayed.

Level 2 Example Display:

The following is an example of a display of level 2 symbols. In this case, the symbols were defined by issuing the LMODULE command for the module COMPLETE and requesting that the symbols be available globally.

```
16:52:40        TID    13          COM-5.1.           User JPO         02/03/97
                           -  Symbol Display  -                        UDB5
Type LMOD   D/Csect            Label            Equ/Lmod/Memb COMPLETE  Scope G
Symbol- Mult  - Type Lngth- Disp  - Addr-  - Contents-                        -
TLCOMREG       1 CSCT 4096   00000000 00008000  .0  . . .>. .D. ..q..=&.yFü  .>
TLVSAM         1 CSCT 7100   0000F888 00017888 .00%TLVSAM  450 05/17/91 / 21.4>
TLAMVTAM       1 CSCT 8192   00009710 00011710 TLAMVTAM.0ü..0D..0I..0.D. ¿..0.>
TLAMVTEX       1 CSCT 4360   000165F0 0001E5F0 TLAMVTEX450 05/17/91 / 21.06.0.>
TLAMVDEV       1 CSCT 4096   00012320 0001A320 .00-TLAMVDEV450 05/17/91 / 21.0>
TLAMVTRC       1 CSCT 5010   0000B710 00013710 .00-TLAMVTRC450 05/17/91 / 21.0>
TLAMMODS       1 CSCT  240   00007B18 0000FB18 DTAB .üy .*h ..h         ... ..>
TLSRDKAC       1 CSCT  564   0001EAE8 00026AE8 .00-TLSRDKAC450 05/17/91 / 21.3>
TLSRCTRL       1 CSCT 3730   0001DAD8 00025AD8 .0¿.TLSRCTRL450 05/17/91 / 21.3>
TLOCOLOG       1 CSCT 3669   0001AAC0 00022AC0 . . . .  .b.&..M        ..+q  .>
TLDSKCHR       1 CSCT 1078   0000E808 00016808 .<.<.<  3330     .._ g . . ....>
TLSRROLE       1 CSCT  628   0001EFD8 00026FD8 .00-TLSRROLE450 05/18/91 / 00.5>
TLMSCHED       1 CSCT 1784   0000E010 00016010 .00-TLMSCHED450 05/17/91 / 21.1>
TLMSGCQ        1 CSCT 1454   000137A0 0001B7A0 .. ... ... ... ... ... .&.ü.&üü>
TLMSDQCQ       1 CSCT 1012   000133A8 0001B3A8 .00-TLMSDQCQ450 05/17/91 / 21.1>
TLOPADAB       1 CSCT 2770   0001C738 00024738 .00-TLOPADAB450 05/17/91 / 21.3>

Enter-PF1-  PF2 - PF3 - PF4 - PF5 - PF6 - PF7 - PF8 - PF9 - PF10 -PF11- -PF12 -
      Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

For an explanation of the fields, see the example of a level 1 symbol display above.

Level 3 Example Display

The following is an example of a display of level 3 symbols. In this case, the symbols were defined by issuing the READ command for the member CCOMBLKS and requesting that the symbols be available globally. The CCOMBLKS member was created by assembling a module containing COPY statements for various Com-plete control blocks and specifying the TEST assembler parameter to cause it to generate TESTRAN symbols.

```
16:54:49        TID    13          COM-5.1.          User JPO         02/03/97
                          ---  Symbol Display  ---                        UDB5
Type MEM    D/Csect DCOMREG    Label            Equ/Lmod/Memb CCOMBLKS  Scope G
Symbol-- Mult--- Type Lngth- Disp---- Addr---- Contents-----------------------
ATIBTAB        1 ADDR    4   00000000 00008000 0034F000
CNOTIBS        1 ADDR    2   00000004 00008004 0064
CTIBLEN        1 ADDR    2   00000006 00008006 00A0
CQSTART        1 VCON    4   00000008 00008008 00016E48
APVT           1 ADDR    4   0000000C 0000800C 008DC4B8
ATCSTCB        1 ADDR    4   00000010 00008010 008D4298
ATCSSBCB       1 ADDR    4   00000014 00008014 04717E50
CRESPGML       1 ADDR    4   00000018 00008018 04A8C6E0
ATCWSCAN       0 FW      4   0000001C 0000801C
ATCWS          1 VCON    4   0000001C 0000801C 00009004
CF4            1 ADDR    4   00000020 00008020 00000004
ATCWE          1 VCON    4   00000024 00008024 0000905C
CNOTHRDS       1 ADDR    2   00000028 00008028 0004
CNOTASKS       1 ADDR    2   0000002A 0000802A 0007
COSEXIT        1 INST    6   0000002C 0000802C 0A03
CEOJ           0 HW      2   0000002E 0000802E

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

For an explanation of the fields, see the example of a level 1 symbol display above.

The UDEBUG Breakpoint List Screen (UDB6)

This screen displays a list of breakpoints which have been defined for the UDEBUG session. To display breakpoints, use the command BPLIST. If no breakpoints are defined, this screen is empty.

Apart from the FORWPAGE and BACKPAGE commands, the top line to be displayed can be selected by placing the cursor anywhere on the desired line and pressing ENTER.

```
COMZDB0030 (A) Currently on breakpoint id 'BP2' at UCTRL+x00000060
16:48:54        TID    13          COM-5.1.          User JPO         02/03/97
                       ---  Breakpoint List  ---                         UDB6
   Number- ID------  Status-  Module-- Program- Offset  Exec'd- Maxexec Preexec
         5 BP5       Reset    UCTRL    UCTRL    000080        0 9999999       0
         4 BP4       Reset    $THREAD$          002B70        0 9999999       0
         3 BP3       Set      UCTRL    U2CTAI   000000        0 9999999       0
         2 BP2       Active   UCTRL    UCTRL    000060        1 9999999       0
         1 BP1       Set      UCTRL    UCTRL    000000        1 9999999       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
         0                                                    0       0       0
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help              Retur Confi Dump  Backp Forwp Messa BP    Go    Recal
```

Meaning of the information according to column header:

Number

This is the number assigned to the breakpoint when it is defined with the AT command. Each defined breakpoint is allocated a sequential number which is unique within that particular UDEBUG session. If the breakpoint is an implicit breakpoint, this field shows "0".

ID

This is the ID of the breakpoint as specified, or defaulted on the AT command. For implicit breakpoints, this ID indicates the purpose of the breakpoint.

Status

This field indicates the status of the breakpoint as follows:

Reset	The breakpoint has been defined but is not currently set. This occurs when breakpoints are to be set in thread storage as Com-plete must first rollin the test session to physically set the breakpoint.
Set	The breakpoint has been defined and the UDEBUG SVC has been set in the appropriate place.
Active	The test program is currently sitting on the breakpoint.
Dormant	The breakpoint has been defined. However, due to the termination of the test program, it could not be automatically set again. To reactivate the breakpoint, simply issue an AT command for the same breakpoint.
Deleted	The breakpoint has been deleted logically. However, it could not be physically deleted. This can occur in a number of cases, for example, when the test program is sitting on the breakpoint. In this case, the breakpoint is physically deleted when the test program is next activated. In other cases, physical deletion will take place as soon as is possible.
Invalid	The breakpoint has been defined. However, when it was defined it was not possible to determine that it was valid. This occurs for breakpoints set within the test program’s thread. A breakpoint is marked as invalid if it is defined as an offset from a program and the program is not that large, or if it is defined in a storage area in thread and the storage area has not been acquired by the test program.

Module

This is the name of the module within which the breakpoint is set or is to be set when it can be determined. When the string "$THREAD" appears in this column, it indicates that the breakpoint has been requested in thread at an offset from the start of the thread. When the breakpoint is set, if it exists within a module, the breakpoint will be updated to reflect the module name which will then be seen here.

Program

Where module is made up of one or more progams, the LMODULE command can be issed to build symbols for the various CSECTs contained in the module. When this is available, the name of the CSECT in the module where the breakpoint is set is displayed here.

Offset

This is the offset from the module or program where the breakpoint is set. If a program name exists, it is the offset from the program. If only a Module name exists, it is the offset from the module. If the module name is ’$THREAD$’, it is the offset from the start of the user program area in thread.

Exec’d

This is the number of times that the breakpoint has been executed.

Maxexec

This is the maximum number of times that the breakpoint should be executed. After this limit is reached, the breakpoint no longer causes the test program to give control to the debugger.

Preexec

This is the number of times a breakpoint should be executed without giving control to the debugger. After it has executed this number of times, the debugger receives control.

UDEBUG Commands

Entering Commands

More than one command can be entered at a time by using the command delimeter (default is semi-colon ";"). Each operand for a command must be separated by a blank or the operand delimiter (default is comma ","). The commands are processed in turn from left to right. As PF keys and profiles are passed through the same command processor, this applies equally when defining them. All commands and operands can be abbreviated to a short form that makes them unique. In the following descriptions, the upper case characters indicate the minimum which must be entered to uniquely identify a command or option.

Recalling Commands

Each command is entered is stored in a UDEBUG buffer. This means that the commands can be recalled using the UDEBUG command RECALL. When first entered, the RECALL command redisplays the last command entered. If entered again immediately, it displays the command entered previous to that and so on. The location pointer for RECALL processing is reset as soon as a command other than RECALL is entered. A maximum of 30 commands are buffered, the oldest is lost when the buffer fills. Please note also that this processing also applies to PF keys and Profiles.

Command Description

In the following description of the commands, the shaded command format indicates the shortest possible abbreviation of the command keyword. Keywords typed in capitals must be entered as is. Parameter keywords in italics must be substituted with valid values.

AT - Set a Breakpoint

This command enables you to explicitly set a breakpoint. Before this command can be issued, a valid Test Terminal and Level must have been set.

```
AT offset/addr program bpid execute-no  ignore-n0
```

Where:

offset/addr	Determines where the breakpoint will be set. This expression is first checked to determine if it is within a residentpage program. If so, any program name entered is ignored. If it is not within a residentpage program, it is taken to be an offset into breakpoint program name.
program	If an offset is provided for the breakpoint name, a program name is required from which this offset is taken. If program is not supplied, or the default character is specified, a previously set Test Program is used. If this is not set, then the breakpoint cannot be set and an error message is issued. If a program name is established, the program must be cataloged to Com-plete.
bpid	Optional. This is an eight character ID which you can supply to identify the breakpoint. This is displayed whenever the breakpoint is referenced in any way. If no ID is specified, the breakpoint ID defaults to "BPnnnnnn", where nnnnnn is the zoned representation of the internal breakpoint number.
execute-no	Optional. The number of times the breakpoint is to be executed, after which time it will become dormant (defined but ignored whenever hit). This must be a number less than x’7FFFFFFF’. If it is not specified, it will default to x’7FFFFFFF’, which in practice means that it is executed every time it is hit.
ignore-no	Optional. The number of times the breakpoint is to be ignored before being taken. This must be a number less than x’7FFFFFFF’. If this is not specified, the breakpoint is taken the first time that it is hit.

Examples:

1. AT 0 UCTRL

   This causes a breakpoint to occur when the program UCTRL is started on the test level.

2. AT RESPGM+50 * RESBP

   This adds a breakpoint with an ID of "RESBP" which is hit when a program on the test level hits the instruction at offset x50 into the resident program "RESPGM".

Notes:

1. When an error in the parameter value occurs, the breakpoint is NOT set, even if the parameter is an optional one.
2. If an offset/program combination is provided, it cannot be ensured at the time it is defined that the offset is valid. This can only be established when the program is loaded for the Test

Session ,when it can be verified if firstly the module length can contain the offset and secondly that a valid instruction exists at this location. If it is found to be invalid, the breakpoint entry is flagged as having an invalid offset, which will be seen if the breakpoint is displayed.

3. If a new address and/or amode is specified, the amode in which the user program is to be dispatched must be consistent with the restart address. For example, a 31 bit mode address cannot be specified if the restart amode is specified or defaults to 24 bit mode.
4. The amode parameter only applies to operating systems that are capable of running in 31 bit mode.
5. Breakpoints can only be deleted by the Terminal/Level combination which added them.
6. When a UDEBUG session terminates either normally or abnormally, all breakpoints set by that session are removed.
7. If the EOJ command is contained in a profile, it will cause immediate termination of the execution of the profile AND of your session.
8. Global symbols that need to be defined each time Com-plete comes up can be defined using this functionality. If UDEBUG is started as a Com-plete STARTUPPGM, it will have a user ID of SYSUSR. If a profile name of SYSUSR exists on the COMDBPRF defined dataset, this will be executed when the program is attached. To avoid an abend from this task when it finishes, the EOJ command must be the last command in the profile.
9. When the TESTRAN outout is created in a PDS, the assembler generally punches out the module text records following them. For this reason, the READ command simply reads the member until it finds the first non-TESTRAN record when it finishes. If the command finishes correctly the number of TESTRAN records read will be indicated in the message.
10. Global symbols can be defined at startup using the SYSUSR profile as explained for the PROFILE command.

BACKPAGE - Page Backward on the Current Screen

On screens where information can be scrolled, this command causes the display to scroll backward one logical page. On screens where scrolling is not necessary, it has no effect.

BA

Note:
This command is usually assigned to PF7.

BP - Show Breakpoint Information

This command causes the breakpoint information screen to become the current screen. If the user program is not on a breakpoint, an error message is issued and the command rejected.

BP

Note:
This command is usually assigned to PF10.

BPLIST - Give a Breakpoint List

This command causes the Breakpoint List screen to become the current screen.

BPL

C - Relocate the Top of Screen Address (24 bit mode)

This command causes UDEBUG to take the fullword pointed to by the top address on the current screen, clear the high order byte and make this address the new top address.

```
C
```

CONFIRM - Confirm a Previously Entered Update

When an update is made in full screen mode, and confirm processing is active, the screen processor highlights and protects the changed fields, and requests that you confirm your changes. This command indicates that you wish the changes to take place. If this command is NOT the next command issued after such an update request, UDEBUG clears any record of the update request.

```
CO
```

DELETE - Delete Symbols

With this command, you can delete a symbol defined using the EQUATE, LMODULE or READ UDEBUG commands.

```
DE type name g|l
```

Where:

type	Is Equate, Lmodule or Member depending on what is to be deleted. Note that when Lmodule or Member is specified, all symbols created for that symbol will also be deleted.
name	Is the name of the Equate, Lmodule or Member which you wish to delete.
g|l	Optional. Specify either "g" for global delete of the symbol, or "l" for local delete. Local is the default.

Example

1. DE Equate MYPOINT

   This command causes the equate MYPOINT to be deleted.

2. DE Lmodule MYLOAD

   This command causes the load module MYLOAD and all CSECT symbols relating to this load module to be deleted.

DUMP - Dump Storage

This causes the dump storage display to become the current screen. If no parameter is used, the previous top address from the last DUMP command or DUMP processing is used.

```
DU expression
```

Where:

expression	is an expression determining the new address to be used as the top of screen.

Example

DU DCOMREG

This causes the data around the the address where COMREG is found to be displayed.

EOJ - Terminate the UDEBUG Session

This command causes the UDEBUG session to be terminated. All breakpoints owned by this TID/Level combination are either deleted or flagged for deletion.

```
EOJ
```

EQUATE - Define Private Symbols

With this command, you can define symbols within your private symbol table area.

```
EQ name expression length
```

Where:

name	Required. The name of the symbol. Maximum length is 8 characters.
expression	Optional. The expression indicating where the symbol should be equated to. If this is not specified, the address at the top of the screen address is taken by default.
length	Optional. The length of the symbol being defined. If this is not specified, the symbol length is zero.

Example

1. EQ MYCOM DCOMREG

   This causes the symbol MYCOM to be defined with a pointer to the address described by DCOMREG which is the address of Com-plete’s COMREG area.

2. EQ PSA 0

   This causes the symbol PSA to be defined with a pointer to the address 0.

FORWPAGE - Page Forward on the Current Screen

On screens where information takes up more than one screen page, this command scrolls the display forward one logical page. On screens where scrolling is not necessary, it has no effect.

```
FOR
```

Note:
This command is usually assigned to PF8.

GO - Restart a Breakpointed User Program

When the program being debugged is sitting on a breakpoint, this command causes it to be restarted at the next instruction. A new restart address can also be supplied as a parameter to the GO command, as well as a new address mode for the user.

```
GO expression amode
```

Where:

expression	Optional. The expression indicating the address where the program to be debugged is to restart. If this is not specified, the program is restarted at the breakpointed instruction in the AMODE it was in when the breakpoint was hit.
amode	Optional. Possible options are 31 or 24 to force the program to be restarted in the appropriate addressing mode. If this is not specified, the AMODE the program was in when the breakpoint was taken is used.

Example

GO *+8 24

This causes execution to continue at the current address at the top left of the screen plus 8 bytes in 24 bit mode.

Notes:

1. If a new address and/or amode is specified, the amode in which the user program is to be dispatched must be consistent with the restart address. For example, a 31 bit mode address cannot be specified if the restart amode is specified or defaults to 24 bit mode.
2. The amode parameter only applies to operating systems that are capable of running in 31 bit mode.
3. Breakpoints can only be deleted by the Terminal/Level combination which added them.
4. When a UDEBUG session terminates either normally or abnormally, all breakpoints set by that session are removed.
5. If the EOJ command is contained in a profile, it will cause immediate termination of the execution of the profile AND of your session.
6. Global symbols that need to be defined each time Com-plete comes up can be defined using this functionality. If UDEBUG is started as a Com-plete STARTUPPGM, it will have a user ID of SYSUSR. If a profile name of SYSUSR exists on the COMDBPRF defined dataset, this will be executed when the program is attached. To avoid an abend from this task when it finishes, the EOJ command must be the last command in the profile.
7. When the TESTRAN outout is created in a PDS, the assembler generally punches out the module text records following them. For this reason, the READ command simply reads the member until it finds the first non-TESTRAN record when it finishes. If the command finishes correctly the number of TESTRAN records read will be indicated in the message.
8. Global symbols can be defined at startup using the SYSUSR profile as explained for the PROFILE command.

HELP or ? - Provide UDEBUG Help

This will cause the Com-plete help utility to be called to display the available UDEBUG help information.

```
HE?
```

Note:
This command is usually assigned to PF1.

LMODULE - Read a Load Module and Equate its CSECTs

Under z/OS, when a load module is created, a list of CSECTs contained in the load module are placed in formatted records at the start of the module with details of offsets and lengths. The LMODULE command causes UDEBUG to read these records and set up symbols, indicating the offset and length of the CSECT names. When the original module name is resolved either via an EQUATE or using the CSECTs within, the module will be also be addressible.

```
LM name dd scope
```

Where:

name	Required. The name of the load module to be read.	
dd	Optional. The name of DD which should be used to locate the appropriate module. If this is not specified, the default COMPLIB will be used.	
scope	Possible options:	
	LOCAL	the symbols will only be set up for the UDEBUG session issuing the READ command. When this session is terminated, the symbols are lost.
	GLOBAL	the symbols defined as a result are available to all users of UDEBUG under that Com-plete, and remain for the lifetime of the Com-plete region.

Example

LM COMPLETE * GLOBAL

This will cause the Com-plete nucleus module COMPLETE to be read from the current default LOADDD (COMPLIB, unless you change it), and a CSECT equate set up for each CSECT found in the load module. The resultant symbols are available globally.

MESSAGES - Show Current List of Messages

This causes the UDEBUG messages screen to become the current screen. You can issue this command if there is more than one message to be displayed.

```
ME
```

Note:
This command is usually assigned to PF9.

OFF - Remove Breakpoints

This command deletes a breakpoint, or flags a breakpoint for deletion, indicating that deletion has been deferred for some reason. In the latter case, the system ensures that the breakpoint is physically deleted at the appropriate time.

```
OFF bpid|ALL
```

Where:

bpid	This is either the eight byte breakpoint ID for the breakpoint, or the internal numberic ID by which the breakpoint to be removed is known.
ALL	All breakpoints are to be removed.

Example

OFF MYBPID

This causes the breakpoint ID "MYBPID" to be deleted.

Notes:

1. Breakpoints can only be deleted by the Terminal/Level combination which added them.
2. When a UDEBUG session terminates either normally or abnormally, all breakpoints set by that session are removed.
3. If the EOJ command is contained in a profile, it will cause immediate termination of the execution of the profile AND of your session.
4. Global symbols that need to be defined each time Com-plete comes up can be defined using this functionality. If UDEBUG is started as a Com-plete STARTUPPGM, it will have a user ID of SYSUSR. If a profile name of SYSUSR exists on the COMDBPRF defined dataset, this will be executed when the program is attached. To avoid an abend from this task when it finishes, the EOJ command must be the last command in the profile.
5. When the TESTRAN outout is created in a PDS, the assembler generally punches out the module text records following them. For this reason, the READ command simply reads the member until it finds the first non-TESTRAN record when it finishes. If the command finishes correctly the number of TESTRAN records read will be indicated in the message.
6. Global symbols can be defined at startup using the SYSUSR profile as explained for the PROFILE command.

PROFILE - Execute a Profile

Profiles must be added as members of a PDS, and this PDS must be allocated to Com-plete with a specific DD name. Profile members can contain any valid UDEBUG command that can be entered on the UDEBUG command line. At UDEBUG startup, the PROFILE command is issued implicitly for you to enable you to customize the environment automatically.

```
PRO name dd
```

Where:

name	Optional. The name of the profile to be executed. This must exist as a PDS member in the dataset pointed to by the applicable DD name. If this is not specified, your user ID is used as the profile name.
dd	Optional. The name of DD which should be used to locate the appropriate profile member. If this is not specified, the default COMDBPRF is used.

Example

PROfile TESTPROF TESTDD

This causes each record in the member TESTPROF from the DD/DLBL name TESTDD to be read and passed in turn to the UDEBUG command processor.

Notes:

1. If the EOJ command is contained in a profile, it will cause immediate termination of the execution of the profile AND of your session.
2. Global symbols that need to be defined each time Com-plete comes up can be defined using this functionality. If UDEBUG is started as a Com-plete STARTUPPGM, it will have a user ID of SYSUSR. If a profile name of SYSUSR exists on the COMDBPRF defined dataset, this will be executed when the program is attached. To avoid an abend from this task when it finishes, the EOJ command must be the last command in the profile.
3. When the TESTRAN outout is created in a PDS, the assembler generally punches out the module text records following them. For this reason, the READ command simply reads the member until it finds the first non-TESTRAN record when it finishes. If the command finishes correctly the number of TESTRAN records read will be indicated in the message.
4. Global symbols can be defined at startup using the SYSUSR profile as explained for the PROFILE command.

READ - Read TESTRAN Symbols

The various assemblers can produce what is called TESTRAN output for the module they are assembling when the TEST parameter is specified. TESTRAN records contain the details of all CSECTs and DSECTs in the assembled module. UDEBUG can read these from the applicable DD name and build tables containing the applicable information. In this way, DSECTs can be displayed online and it is hoped to be able to disassemble a module to provide the most readable output possible.

```
REA name dd scope
```

Where:

name	Required. The name of the member to be read containing the TESTRAN records.	
dd	Optional. The name of DD which should be used to locate the appropriate member. If this is not specified, the default COMDBTXT is used.	
scope	Possible options:	
	LOCAL	the symbols will only be set up for the UDEBUG session issuing the READ command. When this session is terminated, the symbols are lost.
	GLOBAL	the symbols defined as a result are available to all users of UDEBUG under that Com-plete, and remain for the lifetime of the Com-plete region.

Example

MYSYMS * GLOBAL

This causes the member MYSYMS to be read from the current default TEXTDD (COMDBTXT unless you change it) and the Testran records interpreted from that member. The resultant symbols are available globally.

Notes:

1. When the TESTRAN outout is created in a PDS, the assembler generally punches out the module text records following them. For this reason, the READ command simply reads the member until it finds the first non-TESTRAN record when it finishes. If the command finishes correctly the number of TESTRAN records read will be indicated in the message.
2. Global symbols can be defined at startup using the SYSUSR profile as explained for the PROFILE command.

RECALL - Display the Command Entered Last on the Command Line

This command causes the command entered last to be redisplayed on the command line. When entered repeatedly without intervening commands, it will cause UDEBUG to progressively display the previous commands to the last command displayed.

REC

Note:
This command is usually assigned to PF12.

RELOC - Set the Relocation Base Address

During dump and disassembly processing, you can see the actual address of displayed data, together with a relative address. By default these are equal. However, this command enables you to set the base address against which the relative address will be calculated.

Note:
This is only effective when on the DUMP storage screen or the DISASSEMBLY screen. Also, a seperate relocation factor can be set for the DISASSEMBLY screen without effecting the

DUMP screen and vice versa.

```
REL expresion
```

Where:

expression	Optional. The expression indicating the address to which the current address should be made relative. If this is not specified, the top address for the screen is taken by default.

Example

RELOC DCOMREG

This causes all relative address to be displayed/used relative to Com-plete's COMREG address.

SESSINFO - Show the Current Session Information

This causes the UDEBUG session information screen to be displayed.

```
SES
```

SET - Set Various UDEBUG Options

The SET command enables you to set options and various character and default values for the UDEBUG session. This facility is provided primarily to enable you to customize your UDEBUG session via your Profile.

```
SET option name ON|OFF value
```

Where:

option	The option to be turned on or off. Valid options are:	
	MTRACE	MCALL tracing
	ITRACE	Instruction tracing
	BUMP	bumping of screens when ENTER is pressed without data being entered
	CONFIRM	Confirm processing

name	The name of a default character or value which is to be changed. The characters and values that can be changed are:
	PROFDD — Default profile DD/DLBL name for PROFILE command. TEXTDD — Default text DD/DLBL name for READcommand. LOADDD — Default load DD/DLBL name for LMODULE command. DEFCHAR — Character to indicate default for positional operands for a command. NOTRES — Character to indicate unresolved storage. NOTALLOC — Character to indicate unallocated storage. NOACCESS — Character to indicate unavailable storage. DECCHAR — Character to indicate decimal values. HEXCHAR — Character to indicate hexadecimal values. RELCHAR — Character to indicate relative values. TADDRID — Character to represent top address on screen. CMDDEL — Character to delimit commands in a line. OPDEL — Character to delimit operands in a command. PARMDEL — Character to delimit parameters in an operand. PNTLDEL — Pointer indicator left delimiter. PNTRDEL — Pointer indicator right delimiter. ADDCHAR — Character to indicate addition. SUBCHAR — Character to indicate subtraction.
ON/OFF	Indicates if the specified option is to be turned on or off. ON is the default. If ON is specified for a character or value, the default is set. OFF is invalid for default values. However, for character defaults, OFF causes the character not to be used.
value	The character or string to which the character or value should be set.

Examples

SET ITRACE	Sets instruction tracing on
SET ITRACE OFF	Sets instruction tracing on
SET DECCHAR Y	Sets the character to identify decimal to "Y"
SET TEXTDD	Sets the default Text DD/DLBL name to the default (COMDBTXT).

SYMBOLS - Show the Symbol Display Screen

This causes the UDEBUG symbol display/list screen to become the current screen.

SY

X - Relocate the Top of Screen Address (31 bit mode)

This command causes UDEBUG to take the fullword pointed to by the top address on the current screen, clear the high order bit and make this address the new top address.

X

UDD - DASD Display Utility (Control User)

The UDD utility program enables the terminal user to obtain a display of the contents of any record on any DASD device.

A display of record information is requested by specifying the absolute address where the information is to be found. Note that the display is given in both hexadecimal and character format.

This chapter covers the following topics:

- Command Format
- Header Record Display
- Data Record Display
- Modifying Disk Address

Command Format

Record information can be displayed with UDD only if the absolute address (cylinder, track, record) for the desired information is known. This information can be obtained using the online utility programs UDS and UPDS in z/OS or USERV in VSE. Once this information is known, the record contents can be displayed with the command format:

```
*UDD cuu|vvvvvv cccchhhhrr|pbn
```

where:

cuu	Specifies the CUU address of the DASD device containing the record to be displayed where: c — channel number uu — unit address In z/OS/XA systems, specifies the three-digit hexadecimal device number.
vvvvvv	Specifies the six-character volume serial number of the desired volume (which can replace the value CUU).

plus, for CKD devices:

cccc	Specifies the hexadecimal cylinder number of the cylinder containing the desired record.
hhhh	Specifies the hexadecimal head number (track number) of the track within the cylinder containing the desired record.
rr	Specifies the hexadecimal record number of the desired record on the track previously identified.

plus, FBA devices:

pbn	Specifies the hexadecimal six-digit Physical Block Number of the desired record.

The UDD utility program is fully conversational. Once a record has been displayed, additional records can be displayed by modifying the character string displayed as the first line at the terminal:

```
cuu vvvvvv cccchhhhrr pbn
```

The information displayed at the terminal consists of a maximum of 23 lines plus a header record. Each data line displays 32 characters of data in hexadecimal format and the translated 16 characters in decimal format. Each header line identifies the CUU address of the disk on which the data set resides, the volume serial number of the disk, and the record count/PBN information.

Header Record Display

The format of the header record is:

```
cuu COUNT=cccchhhhrrkkllll vvvvvv
```

or

```
cuu PBN=pbn vvvvvv
```

where:

cuu	Specifies the CUU address of the disk volume.
COUNT=	Is a constant indicating that the following information is count information.
cccc	Specifies the hexadecimal cylinder number.
hhhh	Specifies the hexadecimal track number.
rr	Specifies the hexadecimal record number.
kk	Specifies the hexadecimal key length.
llll	Specifies the hexadecimal record length.
vvvvvv	Specifies the volume identification number.
pbn	Specifies the six-digit hexadecimal Physical Block Number.

Data Record Display

Execution of the UDD program creates a display showing both the header record and the data portion of the desired information. The following figure illustrates a typical display as a result of executing UDD. Note the header record on line one.

```
468 COUNT=00D0000001080100   DISK01
0000 4040404040404040 4040404040404040  *                *
0010 4040404040C7D9D6 E4D740C6C9C5D3C4  *     GROUP FIELD*
0020 E240C1D9C540D9C5 E3D9C9C5E5C5C440  *S ARE RETRIEVED *
0030 C2E2404040404040 4040404040404040  *                *
0040 4040404040404040 F0F0F1F3F9F2F0F0  *        00139200*
0050 4040404040404040 4040404040404040  *                *
0060 4040404040404040 40D6D5C5407CD6C3  *         ONE Ú§OC*
0070 C3E4D9D9C9D5C761 E2C5D3C5D4C5D5E3  *CURRING/SELEMENT*
0080 61E2C17C7C404040 4040404040404040  */SAÚ§Ú§           *
0090 4040404040404040 F0F0F1F3F9F3F0F0  *        00139300*
00A0 4040404040404040 4040404040404040  *                *
00B0 4040404040C6C9C5 D3C440C6D6D3D3D6  *     FIELD FOLLO*
00C0 E6C5C440C2E840D6 D5C5404040404040  *WED BY ONE      *
00D0 4040404040404040 4040404040404040  *                *
00E0 4040404040404040 F0F0F1F3F9F4F0F0  *00139400        *
00F0 4040404040404040 4040404040404040  *                *
0100 4040404040404040 407CD6C3C3E4D9D9  *         Ú§OCCURR*
0110 C9D5C761E2C5D3C5 D4C5D5E361E2C27C  *ING/SELEMENT/SBÚ§*
0120 7C40C6C9C5D3C440 4040404040404040  *Ú§ FIELD         *
0130 4040404040404040 F0F0F1F3F9F5F0F0  *        00139500*
0140 4040404040404040 4040404040404040  *                *
0150 4040404040C6D6D9 40C140E3D6E3C1D3  *     FOR A TOTAL*
0160 40D6C640C6C9C6E3 E840D6C3C3E4D9D9  * OF FIFTY OCCURR*
```

In this display, cylinders and tracks are numerically addressed relative to zero. The data lines are numbered according to their hexadecimal addresses relative to zero. The hexadecimal addresses for each data line appear in the left margin of the display.

On any given record display, if the contents of the record do not fit on one screen page, press ENTER to scroll down. When the end of the record is reached, the final display is:

END OF REC cccchhhhrrkllllpbn

where the definition of the character string is the same as that for the header record described above. To find the next record, enter another UDD display request.

Note that error messages can appear while using UDD. These normally occur because either the call to UDD was entered with an invalid format or a ’no record found’ condition was encountered. In this case, reenter the call or adjust the cylinder, head, and/or record data.

Modifying Disk Address

Once a record has been displayed, the disk address can be manipulated using the terminal’s Program Function (PF) keys. These keys alter the current disk address as follows:

CKD	PFK	FBA
Add 1 to the current cylinder.	1	Add 352 to the pbn.
Add 1 to the current head.	2	Add 32 to the pbn.
Add 1 to the current record.	3	Add 1 to the pbn.
Subtract 1 from the current cylinder.	4	Subtract 352 from the pbn.
Subtract 1 from the current head.	5	Subtract 32 from the pbn.
Subtract 1 from the current record.	6	Subtract 1 from the pbn.

Note:
Altering the current cylinder resets the current head to zero and the current record to one. Altering the current head resets the current record to one.

Example

A data set on disk volume DISK01 is believed to contain erroneous data. The volume resides on the device whose CUU is 350. The record believed to be in error is located at record one on track five of cylinder three. To display the desired record, enter:

```
*UDD 3500003000501
```

The following figure illustrates the resulting display.

```
350 COUNT=0003000501080100   DISK01
0000 4040404040404040 4040404040404040 *                *
0010 4040404040C7D9D6 E4D740C6C9C5D3C4 *     GROUP FIELD*
0020 E240C1D9C540D9C5 E3D9C9C5E5C5C440 *S ARE RETRIEVED *
0030 C2E2404040404040 4040404040404040 *AS              *
0040 4040404040404040 F0F0F1F3F9F2F0F0 *        00139200*
0050 4040404040404040 4040404040404040 *                *
0060 4040404040404040 40D6D5C5407CD6C3 *         ONE Ú§OC*
0070 C3E4D9D9C9D5C761 E2C5D3C5D4C5D5E3 *CURRING/SELEMENT*
0080 61E2C17C7C404040 4040404040404040 */SAÚ§Ú§           *
0090 4040404040404040 F0F0F1F3F9F3F0F0 *        00139300*
00A0 4040404040404040 4040404040404040 *                *
00B0 4040404040C6C9C5 D3C440C6D6D3D3D6 *     FIELD FOLLO*
00C0 E6C5C440C2E840D6 D5C5404040404040 *WED BY ONE      *
00D0 4040404040404040 4040404040404040 *                *
00E0 4040404040404040 F0F0F1F3F9F4F0F0 *00139400        *
00F0 4040404040404040 4040404040404040 *                *
0100 4040404040404040 407CD6C3C3E4D9D9 *         Ú§OCCURR*
0110 C9D5C761E2C5D3C5 D4C5D5E361E2C27C *ING/SELEMENT/SBÚ§*
0120 7C40C6C9C5D3C440 4040404040404040 *Ú§ FIELD         *
0130 4040404040404040 F0F0F1F3F9F5F0F0 *        00139500*
0140 4040404040404040 4040404040404040 *                *
0150 4040404040C6D6D9 40C140E3D6E3C1D3 *     FOR A TOTAL*
0160 40D6C640C6C9C6E3 E840D6C3C3E4D9D9 * OF FIFTY OCCURR*
```

Press ENTER to display the remaining portion of the records.

UDS - Data Set Maintenance Utility (z/OS only)

The Com-plete online utility UDS enables you to examine and modify characteristics of disk-resident data sets and system catalog entries.

Note:
Some UDS options may not be available to you if you have included a user-written routine which disables certain options.

This chapter covers the following topics:

- Command Format
- UDS Functions
- UDS VSAM Services (UDVS for VSE)

Command Format

The primary method of invoking UDS is entry of the command:

```
*UDS function
```

where *function* represents any of the available UDS commands together with its arguments. If you omit the *function* parameter, either the UDS menu for COM-PASS users or the UDS options menu for non-COM-PASS users is displayed.

Note:
If you have defined a default operation using UUTIL, that operation takes precedence and the menu mentioned above is not displayed.

Menus

COM-PASS Users

The following two figures illustrate the UDS menu screens used by COM-PASS users.

```
17:04:27     TID    12          COM-5.1.             User ID ADMIN    08/31/97
                     -- Disk Space Maintenance --                        Page 1
       Function           ID    PFK          Operands                     UDS1
---------------------     --     ---  ----------------------------------------
    Allocate Data Set     AD       1  DSN,Volume
     Display Data Set     DD       2  DSN
        Find Data Set     FD       3  DSN
       Purge Data Set     PD       4  DSN,Volume(s)
      Rename Data Set     RD       5  DSN,Volume(s),NewDSN
     Scratch Data Set     SD       6  DSN,Volume(s)
        Vsam Services     VS       7
          List DSName     LD       8  DSN(,Volume)(,Hex)
         List Catalog     LC       9  (All)(,Index)(,Volume)  for CVOLS
           List Space     LS      10  Volume
            List VTOC     LV      11  Volume

      Select Function:        or PFK
         and Operands
         DSN or Index:
               NewDSN:
            Volume(s):
              DEVtype:
       'Hex' OR 'All':                                 (Enter for next page)
```

```
17:04:39     TID    12          COM-6.2.            User ID ADMIN     08/31/02
                      -- Disk Space Maintenance --                        Page 2
Function                  ID     PFK          Operands                     UDS2
-----------------------   --     ---  ----------------------------------------
       Catalog Data Set   CD       1   DSN,Volume(S),DEVtype
     Recatalog Data Set   RD       2   DSN,Volume(S),DEVtype
     Uncatalog Data Set   UD       3   DSN









     Select Function:        or PFK
        and Operands
        DSN or Index:




 Volume(s) OR NewVOL:
             DEVtype:                                  (Enter for next page)
```

Note that only LIST VTOC, LIST SPACE, and LIST DSNAME are allowed as default functions defined with the UD function of UUTIL.

Non-COM-PASS Users

The following figure illustrates the UDS menu screen used by non-COM-PASS users.

```
ENTER   <OP> AND OPERAND.       < > MEANS OPTIONAL.
OP......MEANS................. OPERAND
FIND    SEARCH VTOCS.           DSN
LIST    LIST DSN INFORM.        'DSNAME',DSN,<VOL>,<'HEX'>
LIST    LIST VTOC.              'VTOC',VOL
LIST    LIST FREE SPACE.        'SPACE',VOL
LIST    LIST CATALOG.           'CTLG', <'ALL' >, <INDEX > ,<VOL >
ALLOC   ALLOCATE A DSN.         DSN,VOL
RENAME  RENAME.                 DSN,NEWDSN,VOL,VOL
SCRATCH SCRATCH.                DSN,VOL,VOL
PURGE   SCRATCH,PURGE.          DSN,VOL,VOL
DISP    SEARCH CATALOG.         DSN
CATLG   CATALOG DSN.            DSN,DEVTYP,VOL,VOL
UNCATLG UNCATALOG DSN.          DSN
RECATLG RECATALOG DSN.          DSN,DEVTYP,VOL,VOL
HELP    REDISPLAY MENU.
```

This UDS menu lists all the argument options available with UDS. Note that arguments within < > are optional; arguments enclosed within single quotation marks must use the exact characters as represented.

You can abbreviate all function codes and arguments with the smallest number of letters that make them unique. In the command function descriptions below, the minimum abbreviation required for each command function is indicated by underscoring of the required characters.

Note:
You must replace any optional arguments that are omitted with commas, unless the argument being omitted is the last argument in the list.

Example 1

After displaying the menu, you must enter the arguments in one statement line as illustrated in the following sample UDS request:

```
LIST DSNAME,WORK.DATASET,PACK01
```

This sample UDS command requests a display of allocation information for the data set WORK.DATASET. The command format assumes that the data set is not cataloged, and directs the search to be restricted to disk volume PACK01.

Example 2

UDS is a fully conversational program, i.e., you may enter UDS commands when you invoke UDS or at any time thereafter.

The next example illustrates the specification of the UDS LIST command at UDS invocation.

```
*UDS LIST DSNAME,WORK.DATASET,PACK01
```

UDS Functions

After you have invoked UDS and used one of the options, you can display or redisplay the menu by pressing ENTER. Note that you can make a new UDS request at any time, either while the menu is displayed or while the output from a previous UDS command is displayed.

Normally, all UDS requests must include an operation code followed by the list of applicable arguments; however, if you enter a data set name without an operation code, the operation code DISP is assumed.

To terminate UDS, press the CLEAR key, or enter the Com-plete program termination command *EOJ.

Note:
Since UDS is fully conversational, entry of a call request for execution of another program or utility automatically terminates UDS.

The functions available from the UDS menus are summarized in the following table and discussed in detail in the remainder of this section. The shortest possible abbreviations are indicated by underlining.

Command	Description
ALLOC	Dynamically allocates a data set.
CATLG	Dynamically catalogs data set names.
DISP	Dynamically locates a data set.
FIND	Searches all online disk volume devices for the specified data set name.
HELP	Returns to the UDS function menu.
LIST DSNAME	Lists DSCB information for a specified data set.
LIST VTOC	Lists VTOC information for a specified disk volume.
LIST SPACE	Lists free space information for a specified disk volume.
LIST CTLG	Lists system catalog information for a specified high-level index.
PURGE	Dynamically purges a specified data set.
RECATLG	Dynamically changes system catalog entries without uncataloging and cataloging the entire entry.
RENAME	Dynamically changes the name of a specified data set.
SCRATCH	Dynamically scratches a specified data set.
UNCATLG	Dynamically removes catalog entries from the system catalog.

ALLOC Function

The ALLOC function enables you to dynamically allocate a data set.

The command format is:

ALLOC dsname,vol

The arguments are:

dsname	Required.
	Indicates the name of the data set to be allocated.
	Note that this name is entered in the volume table of contents for the specified volume only; it is not entered into the catalog.
vol	Required.
	Specifies the volume identification number of the disk volume on which the data set is to be allocated.

On the next screen you can enter the necessary DCB and SPACE parameters:

```
09:57:22        TID    14          COMTEST2          User ADMIN        14.02.02
                              UDS Dataset Allocation                        UDS3

 Data Set Name  . . . :  MY.DATASET
 Volume . . . . . . . :
 Catalog DSN  . . . . :  Y               (Y or N)

 Management class . . :                  (Blank for system default)
 Storage class  . . . :                  (Blank for system default)
 Data class . . . . . :                  (Blank for system default)

 Dsorg  . . . . . . . :                  (DA, PS, PO, POE, DAU, PSU, POU)
 Recfm  . . . . . . . :
 Blksize  . . . . . . :      0
 Lrecl  . . . . . . . :      0

 Allocation unit  . . :                  (TRK, CYL or average block length)
 Primary  . . . . . . :         0
 Secondary  . . . . . :         0
 Directory  . . . . . :         0
 Form . . . . . . . . :                  (CONTIG, MXIG, ALX or blank)
 Round  . . . . . . . :                  (RO or blank)
 Expiration date  . . :                  (YYYYDDD)
```

Once you have entered all the requested DCB and SPACE information, the data set is allocated and a display equivalent to that obtained from entering the LIST function is returned. (See the *LIST Function*).

CATLG Function

The CATLG function enables you to dynamically catalog data set names in the system catalog and automatically create the appropriate indices.

The command format is:

CATLG dsn,<device>,vol1[,vol2,...]

The arguments are:

dsn	Required.
	Specifies the fully qualified name of the data set to be cataloged.
	Note that all necessary indices are automatically created if they do not already exist.
device	Required when volume(s) not online. Ignored when volume(s) online.
	Specifies the device type of the volume to contain the data set. **Note:** When the volume is not available online, the generic device type must be specified (for example, 3380). Do not specify group names like SYSDA which may consist of a mixture of different device types!
vol1	Required.
	Specifies the volume identification number of the disk volume containing the data set (dsn) to be cataloged.
vol2	Optional.
	If the data set to be cataloged resides on multiple volumes, specifies the volume identification numbers on which the various extents reside.
	Note that multiple volumes must be specified if the data set resides on more than one volume.

UDS creates a catalog entry in the appropriate catalog for each of the volumes indicated.

DISP Function

The DISP function enables you to dynamically locate a data set by means of its catalog entry. This is the default function assumed when you enter a data set name but omit the operation or function.

The command format is:

```
DISP dsn
```

The argument is:

dsn	Required.
	Specifies the fully qualified name of the data set to be located.

UDS locates the data set requested and gives the same display as the FIND function; it lists the data set by name and identifies the volume and device type on which it resides. (See the *FIND Function*).

FIND Function

The FIND function enables you to search all online disk volume devices for the data set name specified.

The command format is:

FIND dsname

The display returned gives the volume identification number of the disk containing the data set, the device type of the volume, and the CUU. A typical display received in response to the FIND operation is:

```
VOLUME(S):
DISK01 (3330-1) ON 460
FIND OPERATION SUCCESSFUL
```

Note that the FIND function returns information about the requested data set after searching the online disk volumes, *not* the system catalog. In addition, if more than one data set with the same name exists on more than one volume, the FIND function finds only the first data set encountered in the search. This search is performed on an ascending CUU basis.

HELP Function

The HELP function enables you to return to the UDS function menu any time after you have entered a UDS request.

The command format is:

HELP

No arguments are provided or accepted with the HELP function. Only the UDS function menu is displayed.

LIST Function

The LIST function enables you to:

- Display allocation information from the DSCB for a specified data set.
- Obtain a listing of the data sets residing on a specific disk volume.
- Obtain a summary listing of the available free space on a specific disk volume.
- Display catalog entries for a specified high-level catalog index.

  The command format for the LIST function varies depending upon the list request entered. The following subsections discuss the four types of the LIST function.

List Data Set Information

The LIST DSNAME function enables you to list DSCB information for a given data set.

The command format is:

LIST DSNAME,dsn,[vol],[HEX]

The arguments are:

DSNAME	Required.
	Indicates that data set DSCB information is to be displayed.
dsn	Required.
	Specifies the fully qualified data set name of the data set for which DSCB information is to be listed.
vol	Optional.
	Specifies the volume identification number of the disk volume on which the data set resides.
	If vol is omitted, the system catalog is used to locate the data set. **Note:** The absence of this argument must be indicated by a comma.
HEX	Optional.
	Specifies that the requested display information is to be given in hexadecimal format only.

The initial information displayed is the summary information as taken from the format 1 DSCB. If the data set is an ISAM data set, pressing ENTER displays the information from the format 2 DSCB. If the data set has more than three extents, pressing ENTER displays information about the remaining extents as taken from the format 3 DSCB.

The following figure illustrates a typical response that might be received after entering a LIST request for a data set named COM.DEMO.

```
LIST DSNAME,COM.DEMO
VOLUME(S):
WORK01 (3330-1) ON 460
F1DSCB = 0404 00 35       0194 00 23(HEX)
CR DATE     EX DATE     NOBDB  DSIND
78 214      00 000       000    80
EXTS DSORG  RECFM  BLKSIZE    LRECL OPTCD
01   PO      FB      2480       80    00
ALLOCATION = CYLS  CONTIG
EXTNT TYPE    LOW       HIGH   SPACE   LOW(HEX)    HIGH(HEX)
00   PRIME  0253 00   0254 14  92 00   00FD 00     00FE 0E
TOTAL SPACE =      95 CYLS 00 TRKS
PERCENT USED      33
```

A description of the above information may be found in the IBM publication describing the utility IEHLIST.

List VTOC Information

The LIST VTOC command enables you to list VTOC information for a given disk volume.

The command format is:

LIST VTOC,vol

The arguments are:

VTOC	Required.
	Indicates that disk volume VTOC information is to be displayed.
vol	Required.
	Specifies the volume identification number of the disk volume for which VTOC information is to be listed.

The information displayed is a listing of the data sets residing on the volume. The actual information in the display is taken from the disk volume table of contents and includes summary information such as:

- Data set names
- Type of data set
- Type of allocation
- Number of extents
- Size of primary extent
- Size of secondary extent(s)

The following figure illustrates a typical response that might be received after entry of a LIST request for VTOC information for the disk volume WORK01.

```
 LIST VTOC,WORK01
 DATASET NAME           TY UNIT EX PRIM SECN CREAT EXPDT LAST  PW

ADABAS.DB137.ASSOR1     DA  CYL 01   15    0 87006       87201
NATURAL.NATPROC.LOAD    PS  CYL 01   15    0 87062 98365 87255
NETPASS.V2.SYSTEM       PS  CYL 01   10    0 87300       87305 RW
```

where:

TY	Specifies the TYpe of data set organization.
UNIT	Specifies the allocation UNIT in CYLinders, TRacks, etc.
EX	Specifies the number of allocated EXtents.
PRIM	Specifies the PRIMary allocation size.
CREAT	Specifies the CREATion date in the format YYDDD
EXPDT	Specifies the EXPiration DaTe
LAST	Specifies the LAST date accessed (if the F1 DSCB contains this value).
PW	Specifies the PassWord status (R = read-protect, W = write-protect).

List Free Space Information

The LIST SPACE command enables you to list free space information for a given volume.

The command format is:

`LIST SPACE,vol`

The arguments are:

SPACE	Required.
	Indicates that a listing of volume free space is to be given.
vol	Required.
	Specifies the volume identification number of the disk volume for which the free space display is to be given.

The volume free space display includes such items as:

- The relative track address for each free extent
- The size of each free extent
- A summary of the total free space residing on the volume

The following figure illustrates a typical display that might be given after entering a LIST request for the free space on a given volume.

```
LIST SPACE,WORK01
 REL ADD    CYLS  TRACKS
       1       0       1
     456       6       0
    1425     320       0
    7581       1       0
  TOTALS     327      18
LIST OPERATION SUCCESSFUL
```

List System Catalog Information

The LIST CTLG command enables you to list system catalog information for a given high-level index.

Note:
The LIST CTLG command is valid for non-ICF catalogs only. For additional information, see the section *UDS VSAM SERVICES*.

The command format is:

`LIST CTLG,[ALL],[index],[vol]`

The arguments are:

CTLG	Required. Indicates that system catalog information is to be displayed.
ALL	
	Optional. Specifies that all index information is to be listed in addition to the list of data set name catalog entries.
	If ALL is omitted, only cataloged data set names are listed.
index	Optional. **Note:** If this argument is omitted, a listing of the entire catalog is given.
	Specifies the high-level index(es) for which the catalog display is to be restricted. For example, if a data set is cataloged as SAMPLE.DATA.SET, the value SAMPLE. or SAMPLE.DATA. or SAMPLE.DATA.SET. could be specified for index and the data set would be listed.
	Note that the ending period must be included.
vol	Optional.
	Specifies the volume identification number of the disk volume for which catalog information is to be listed.
	Note that the volume specified must be a user volume on which a secondary system catalog exists.
	If vol is omitted, the resident system volume is assumed.

The information displayed is a listing of the data set catalog entries. If you specify the optional argument ALL, all index levels are included in the display.

Note:
The UDS LIST CTLG function only lists information from an z/OS system catalog and the VSAM master catalog, *not* from a VSAM user catalog.

The following figure illustrates a typical response received after entry of a LIST request for catalog information.

```
LIST CTLG,,COM.
INDEX IS A CVOL POINTER TO SAG01
COM.COB.LOAD               3330-1 SAG01
COM.DASDS                  3350   PACK01
COM.MESSAGES               3350   PACK02
COM.PGMLIB                 3350   PACK01
COM.SD                     3330-1 SAG01
LIST OPERATION SUCCESSFUL
```

PURGE Function

The PURGE function enables you to dynamically purge a specific data set. If the data set is cataloged, you can use the UDS UNCATLG function to remove the catalog entry.

The command format is:

```
PURGE dsn,vol1[,vol2,...]
```

The arguments are:

dsn	Required.
	Specifies the fully qualified name of the data set to be purged.
vol1	Required.
	Specifies the volume identification number of the disk volume containing the data set to be purged.
	Note that the system catalog is not referenced during the purge operation.
vol2	Optional.
	If the data set to be purged resides on multiple volumes, specifies the volume identification numbers on which the various extents reside.

Note:
UDS scratches the data set from each of the volumes indicated, regardless of expiration date status.

RECATLG Function

The RECATLG function enables you to dynamically change catalog entries in the system catalog without needing to uncatalog and catalog the entire entry.

The command format is:

```
RECATLG dsn,<device>^,vol1[,vol2,...]
```

The arguments are:

dsn	Required.
	Specifies the fully qualified name of the data set whose catalog entry is to be changed.
device	Required when volume(s) not online. Ignored when volume(s) online.
	Specifies the device type of the device on which the new catalog entry is to be created. **Note:** When the volume is not available online, the generic device type must be specified (for example, 3380). Do not specify group names like SYSDA which may consist of a mixture of different device types!
vol1	Required.
	Specifies the volume identification number of the disk volume containing the data set to be cataloged.
vol2	Optional.
	If the data set being cataloged resides on multiple volumes, specifies the volume identification numbers on which the various extents reside.

UDS creates a new catalog entry for the specified data set. The new device types and volumes entered are reflected in the new catalog entry.

RENAME Function

The RENAME function enable you to dynamically change the name of a specific data set.

The command format is:

```
RENAME dsn1,dsn2,vol1[,vol2,...]
```

The arguments are:

dsn1	Required.
	Specifies the fully qualified name of the data set to be renamed.
dsn2	Required.
	Specifies the fully qualified name to which the data set is to be renamed.
vol1	Required.
	Specifies the volume identification number of the disk volume containing the data set to be renamed.
	Note that the system catalog is not referenced during the rename operation.
vol2	Optional.
	If the data set to be renamed resides on multiple volumes, specifies the volume identification numbers on which the various extents reside.

UDS renames the data set on each of the volumes indicated. Note that if the data set resides on more than one volume, you must specify multiple volumes.

SCRATCH Function

The SCRATCH function enables you to dynamically scratch a specific data set. If the data set is cataloged, you can use the UDS UNCATLG function to remove the catalog entry.

The command format is:

```
SCRATCH dsn,vol1[,vol2,...]
```

The arguments are:

dsn	Required.
	Specifies the fully qualified name of the data set to be scratched.
vol1	Required.
	Specifies the volume identification number of the disk volume containing the data set to be scratched.
	Note that the system catalog is not referenced during the scratch operation.
vol2	Optional.
	If the data set to be scratched resides on multiple volumes, specifies the volume identification numbers on which the various extents reside.

UDS scratches the data set from each of the volumes indicated*only*if the data set expiration date has expired.

UNCATLG Function

The UNCATLG function enables you to dynamically remove catalog entries from the system catalog and automatically remove the correct indices.

The command format is:

```
UNCATLG dsn
```

The argument is:

dsn	Required.
	Specifies the fully qualified name of the data set to be uncataloged. All necessary indices are automatically removed.

UDS uncatalogs the data set from the system catalog and uncatalogs any unnecessary indices. Note that the highest-level index is not removed.

UDS VSAM Services (UDVS for VSE)

Com-plete's VSAM SERVICES allow COM-PASS users to use the interactive execution of IBM's IDCAMS utility program.

The command format is:

VS

The VSAM SERVICES menu, shown in the following figure, then appears.

```
11:59:16     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                          -- Vsam Services --                              UDV1

  Select Function   :
  Name or Level     :
  Catalog Name      :
  List Options      : S   (S for short, D for detail)
                    :     (P for Printout)                   Printer:  PSG

  Functions
                      LL  --  Listcat Level(level)
                      LE  --  Listcat Entry(name )


  Direct Access Method Services Command:
  >                                                                      <
  >                                                                      <
  >                                                                      <
  >                                                                      <
  PF10 = Scroll Left  PF11 = Scroll right
```

Note:
Usage of the "Direct Access Method Services Commands" can be restricted by your systems administrator.

UDUMP - Conversational Dump Display Utility

The Com-plete utility UDUMP allows you to access dumps in the online dump library and display them at the terminal. You can generate a hard copy printout of a specific UDUMP dump using either the UDUMP PRINT command or the Com-plete batch utility program TUDUMP.

This chapter covers the following topics:

- Overview
- Command Format
- UDUMP Commands

Overview

Online programs running under Com-plete occupy main storage areas called threads. When a program executing in a thread terminates abnormally, the contents of both the thread and the registers are written to the online dump data set residing in the SD library. Other information added to each dump as it is written to the dump or data set includes:

Contents of the following control blocks: UPCB, COMREG, THCB, TIB, CUOW;

- The program name;
- The date and time of the dump;
- The Terminal Identification number (TID) of the terminal from which the program was executing;
- The number assigned to the dump in the dump library;
- The error message associated with the abnormal termination;

  The maximum number of dumps kept on the dump library is 32. The dump library is a wraparound library; that is, generated dumps are numbered sequentially beginning with one. The 33rd dump generated replaces the first dump generated.

  The display features available with UDUMP allow you to:

- Display different portions of a dump;
- Alter the display format;
- Relocate addresses used within the dump;
- Display all dumps in the dump library.

Command Format

The primary command format for invoking UDUMP is:

```
*UDUMP argument
```

where *argument* can be one of the following:

ALL	A constant specifying a summary display of all dumps residing in the dump library (see the following figure).
n	An integer from 1 to 32 designating the specific dump to be displayed.
-n	A negative integer. This number is added to the number of the most recent dump in the dump library and the dump with the resulting number assignment is displayed.
name	The name of the abnormally terminated online program that created a dump. If multiple dumps exist in the dump library for the same program, the most recent dump is displayed.
null	Specifies a display of the dump most recently written to the dump data set.

If you request the ALL argument, a list of all dumps in the dump library is displayed, for example:

```
NO  NAME     DATE  TIME    TID ERR(COM)  NO  NAME     DATE  TIME    TID ERR(COM)
01 USTACK    9/06 141717    23 .ROL0012     NO DUMP
02 IPCS      9/06 150750    19 .ABS0006     NO DUMP
03 IPCS      9/06 151425    19 .ABS0006     NO DUMP
04 NAT22     9/06 164301    11 .ABS0006     NO DUMP
05 NAT22     9/06 164652    23 .ABS0006     NO DUMP
06 SKGETM    9/06 173827    19 .ABS0006     NO DUMP
07 SKGETM    9/06 174134    19 .ABS0006  27 UEDIT     9/05 112903    21 .MPO0022
08*NDB23     9/06 211552    11 .ABS0006  28 SDTEMP    9/05 131051    21 .SDF0008
```

This menu display is used to identify the specific dump to be selected. Note that the most recent dump is marked with an asterisk.

Once you have identified the dump you wish to display, select the specific dump by entering the appropriate dump number in the input field top left. When a specific dump is displayed, two primary formats result. The first format is the register summary and dump contents display as illustrated in the following figure:

```
  NAME=UDS       DATE= 11/04  TIME= 163648  DUMP=   5 TID=     8
COMABS0006-* PROGRAM UDS ABEND S0C4 PSW=078C1000800284CC

    PSW  078C1000 800284CC
R 0- 7   00000000 00039A48 0000F000 0A8A6000 0001F000 008B9C50 00000002 00F3B2E0
R 8-15   00039A00 00FD4B18 800284BE 0001238C 00028318 00053840 00000322 00000000

FR0- 6   00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

UPCB     0A8A6000-0A8A7000
SQA2     001F5000-001F6000
REGION1  0022B000-00233B30
REGION2  0A8A7000-0A8A7B00
COMREG   0000F000-00010000
THCB     0001F000-0001F400
TIB      09FCE600-09FCE6C0
CUOW     09C29000-09C29290

Program  EntryPnt LoadPnt  End      Length   Location PSW

UDS      8022B000 0022B000 002333A8 000083A8 thread
```

Further in this section, this display will be referred to as "the REGS page". Along with the register contents, the address ranges of the storage areas (sections) contained in the dump are displayed. These represent the Com-plete control blocks and user region(s) related to the program that has terminated abnormally. In this display, you can either position the cursor on the line indicating the area you want to be displayed, or use the commands described below in this section to specify the address to be displayed, as well as additional options.

The following figure illustrates a display of storage contents at a given address:

```
  PSW  078C1000 800284CC RELOC=COMREG   NAME=UDS      DUMP=   5 TID=    8
R 0- 7   00000000 00039A48 0000F000 0A8A6000 0001F000 008B9C50 00000002 00F3B2E0
R 8-15   00039A00 00FD4B18 800284BE 0001238C 00028318 00053840 00000322 00000000
Absolute Relative   COMREG  +00000000
0000F000 00000000   09FCE000 006400C0 09C1E300 008D5808    *..Ö–  . äAT  ...*
0000F010 00000010   008F95C0 89BFE5D0 09C19FC0 00010004    * .näi.Vü.A.ä . .*
0000F020 00000020   00000004 00010020 00010007 0A031F11    *   . . . . .....*
0000F030 00000030   0AC80000 4C000A0E 0000F03E 00000001    *.H  < ..  0.   .*
0000F040 00000040   61404040 40404040 40404040 40404040    */               *
0000F050 00000050   09C19040 89BFA990 000000E5 F5F1F024    *.A. i.z.   V510.*
0000F060 00000060   808C3020 00780000 00000000 89C0F138    *.... .      ä1.*
0000F070 00000070   B20A0000 89B2A008 89BFE040 00000000    *..  i...i.Ö     *
0000F080 00000080   89B08320 005C0000 B20A0080 00015F50    *i.c. *  .. . .^&*
0000F090 00000090   09FD2BC0 00017000 1997307F 00343A45    *...ä .. .p.” ...*
0000F0A0 000000A0   D6E24040 D4F6F0F3 96720004 89B0A2E8    *OS  M603o. .i.sY*
0000F0B0 000000B0   09FACCD8 00000000 89B0B020 89B06038    *...Q    i...i.-.*
0000F0C0 000000C0   89BEE1E8 00000000 89BE61E8 89C004F8    *i..Y    i./Yiä.8*
0000F0D0 000000D0   89BF9508 89C102F8 89B2A7E0 808C3678    *i.n.iA.8i.xÖ....*
0000F0E0 000000E0   00016190 89C0E0C0 89BFAEB0 89BAD000    * ./.iäÖäi...i.ü *
0000F0F0 000000F0   00008938 89BF3578 89B29D50 89BF8040    *  i.i...i..&i.. *
0000F100 00000100   89BEE9E0 89EC04A8 89B2B000 89B2C0B0    *i.ZÖi..yi.. i.ä.*
0000F110 00000110   00000000 89B106B0 89C00030 00012240    *    i...iä . .. *
0000F120 00000120   89B79018 00012478 89BE2D50 89B43020    *i... ...i..&i...*
0000F130 00000130   89B930A8 89B100B8 89C062B8 89C030C8    *i..yi. .iä..iä.H*
```

To view subsequent pages within a specific dump display, pressc ENTER.

UDUMP Commands

Dumps displayed by UDUMP are similar in content and format to SYSUDUMP dumps produced by batch programs. Three categories of display commands are available to examine a dump:

- Commands to address (examine) portions of the dump;
- Commands to change the display format;
- Commands to change the relative addresses in the left margin of the display.

These categories and the command functions within each category are summarized in the following tables.

Command/Function	Description
address	Displays the location beginning at the specified address.
C* / X*	Displays the location specified by the 24/31 bit address stored in the current location.
CRn / XRn	Displays memory at the 24/31 bit address in general register n.
cursor placement	Displays the location beginning at the location indicated by the cursor.
displacement	Displays the location at the specified number of bytes from the current location.
null entry	Displays the next page.
P	Displays the beginning of the program area.
PSW	Displays the location addressed in word 2 of the PSW.
S nnnnnnnn	Scans for the specified hexadecimal digits (maximum of eight).
T	Displays the beginning of the thread.
section name	Displays the beginning of the section as indicated on the REGS page.

Command	Description
HEX	Displays the dump in hexadecimal format only.
INT	Displays the dump in both character and hexadecimal format.
REGS	Displays the REGS page.
PGMS	Displays the programs that were loaded at the time of the error.

Command	Description
R=M	Sets the addresses relative to the beginning of the machine.
R=P	Sets the addresses relative to the beginning of the program, if this is in the dump.
R=*	Sets the addresses relative to the current location being displayed.
R=section name	Sets the addresses relative to the beginning of the specified section.
R=T	Sets the addresses relative to the beginning of the thread, that is, of the UPCB.

These commands are described in the following subsections.

In addition, the PRINT command is described, which allows you to spool the currently displayed dump to a terminal printer.

Addressing Command Options

The addressing command options enable you to request dump displays either beginning at specific addresses within the dump or relative to specific locations.

ADDRESS Command

Specific addressing enables you to display the online dump beginning at a specified address.

The command format is:

```
address
```

where address is any string of up to eight characters representing the hexadecimal address to be used as the beginning address for the next display.

Note:
The address must refer to a location within the area available for display under the relocation option in effect; otherwise, an addressing error will occur.

C* and X* Commands

The C* and X* commands enable you to display the online dump beginning at the address specified by the first fullword in the current display. The contents of this fullword are treated as a 24 or 31 bit address respectively.

The command format is:

```
C*  or
X*
```

The contents of the first fullword in the current display (current at the time you enter the command) are used as the beginning address for the next display.

Note:
The address must refer to a location within the area available for display under the relocation option in effect; otherwise, an addressing error will occur.

CRn and XRn Commands

The CRn and XRn commands enable you to display the online dump beginning at the address contained in a general register. The register contents are treated as a 24 or 31 bit address respectively.

The command format is:

```
CRn  or
XRn
```

where the variable *n* must be a hexadecimal number from 0 to F. The value you select designates the general register whose content is to be used as the starting point in the dump display.

Cursor Placement Function

By positioning the cursor to any byte displayed on the screen and pressing ENTER, you can scroll the display forward so that the indicated byte moves to the top of the screen.

DISPLACEMENT Command

Displacement addressing enables you to display the online dump beginning at an address relative to the address of the currently displayed area.

The command format is:

```
displacement
```

where *displacement* represents the number of bytes before the current address (-*displacement*) or after the current address (+*displacement*) at which the next display is to begin.

Note:
The displacement must refer to a location within the area available for display under the relocation option in effect; otherwise, an addressing error will occur.

The displacement value can be a decimal or hexadecimal number. Note that if you use a hexadecimal number, the number must be preceded by the character X.

Typical addressing commands relative to the currently displayed address are:

Decimal:	+50, -28
Hexadecimal:	+X25C, -XD3F

You can also request displacement addressing relative to an address other than the current address. For example:

Relative to PSW:	PSW-50
Relative to program:	P+X5C3
Relative to user region 2	REGION2 + 64

Null Entry Function

Null entry refers to the pressing of ENTER without providing a command function. This is the primary paging function of UDUMP: it displays successive pages of dump memory (a page is 23 lines of output). Addresses displayed are in increments of X'20' if the HEX display format is in effect, or increments of X'10' if the INT display format is in effect.

P Command

The P command enables you to display the online dump beginning at the load address of the application program, if this is in the dump.

The command format is:

```
P
```

Note that the display begins at the load address of the application program regardless of the relocation option in effect.

PSW Command

The PSW command enables you to display the online dump beginning at the address specified by the second fullword in the current PSW.

The command format is:

```
PSW
```

The second fullword in the current PSW is used as the beginning address for the next display.

Note:
he address must refer to a location within the area available for display under the relocation option in effect; otherwise, an addressing error will occur.

S Command

The S command enables you to display the online dump beginning at the address of a specified character string.

The command format is:

```
S nnnnnnnn
```

where *nnnnnnnn* is a string of hexadecimal digits representing the hexadecimal internal machine format of the characters whose address will be the beginning address for the next display. Note that *nnnnnnnn* can be a maximum of eight hex digits.

Note:
The characters to be scanned must be within the area available for display under the relocation option in effect; otherwise, a scan failure will occur.

The scan proceeds beginning with the address currently displayed. A comparison is made with each successive set of internal hexadecimal digits, left to right, until either an equal condition is met or the end of the dump is found. If an equal condition is met, the address of the character string is used as the current address in the next display.

Note that because internal hexadecimal digits are scanned,*nnnnnnnn*must be an even number of hexadecimal digits (0 through F).

For example, the command:

```
S F1F0
```

is correct, while the command:

```
S F1F
```

is incorrect and generates an error message.

T Command

The T command enables you to display the online dump beginning at the address >of the thread, that is, the UPCB.

The command format is:

```
T
```

The address of the beginning of the thread is used as the beginning address for the next display.

Section Name Command

The section name command allows you to display the dump from the start of any of the sections indictated on the REGS page.

The command format is:

```
Section Name
```

The starting address of the section specified is used as the beginning address for the next display.

Display Format Options

The display format options enable you to change the display format to the following formats:

- Hexadecimal format only;
- Decimal and hexadecimal format.

HEX Command

The HEX command enables you to change the display format to hexadecimal character format only.

The command format is:

HEX

The format of the current display is changed to reflect a one-column format. The column is displayed in internal hexadecimal format. Note that all addresses, regardless of the relocation option in effect, are in increments of 32 (X’20’).

The following figure illustrates a typical hex display format.

```
  PSW  07EC2000 0020DA64 RELOC=COMREG   NAME=UDS      DUMP=  20 TID=   12
R 0- 7   00000001 00000000 0000F000 00203000 000130C8 00096900 A020D92A 00211F14
R 8-15   00211660 0020F2D8 5006EDBC 0006FDBB 0020CFB0 0020D670 00211868 00000000
COMREG  +00000000
00000000 00096000 006400C0 000162E8 008EE468 008FF2E0 00000000 05D40B40 00010004
00000020 00000004 00010058 00060007 0A031F11 0AC80000 4C000A0E 0000F03E 00000001
00000040 61404040 40404040 40404040 40404040 000050C8 00030001 000000E5 F4F6F024
00000060 808DB808 00780000 00000000 85DF8000 B20A0000 85D26008 85DEB930 0001C610
00000080 85D07068 005C0080 B20A0080 0001FF50 0009ABC0 00017EE8 0094251F 0034292A
000000A0 C5E2C140 F0F3F840 30900004 85D07BD8 0603EAD0 00000000 85D09F20 85D021D0
000000C0 85DE7050 00000000 85DE9058 0001B0A0 0001B0A0 0001B9B4 0001BD8A 85D25CA0
000000E0 0001A010 85DF60B0 85DE6020 85D5F730 85D061F8 85DEFA40 85D0D7F0 0001596C
00000100 85DE79E0 86072898 85DE3038 85D3D080 00000000 85DDB0E0 85DEA078 00015D18
00000120 00000000 00015F50 85D09838 85DEB018 05DFF000 85DDC028 85DE5220 85DF32B8
00000140 85DE6C30 85DE36F8 85D06CD8 05D0B000 00005978 00000000 85D0DCF8 85DF7830
00000160 85D09B60 85D0CC18 85D276E0 85D76098 85D3E0F0 85D02068 85DE8110 85D92000
00000180 00000000 85D0D038 0000F9C0 00000000 00000000 0000D500 85DF7020 0000C600
000001A0 0603EE80 00000000 00000000 8009BA60 8009C610 0009000A 000B000C 000A0000
000001C0 860AE240 00000000 860AD730 00000000 00004E20 00000000 860AF980 00000000
000001E0 85DC3000 00000000 00000000 86072280 85E2E168 05D42180 05D42380 00000000
00000200 C4000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000220 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000240 000007F8 00000000 0000057C 00000000 00000000 00000000 00043300 00001000
00000260 002BC000 00190000 060536D0 00014628 00000000 00110012 00014C7E 6C6E0009
```

INT Command

The INT command enables you to change the display format to both internal hexadecimal and external decimal format.

The command format is:

INT

The format of the current display is changed to reflect a two-column format. The leftmost column is internal hexadecimal format, whereas the rightmost column is external decimal format. Note that all addresses, regardless of the relocation option in effect, are in increments of 16 (X’10’).

The following figure illustrates a typical INT display format.

```
  PSW  07EC2000 0020DA64 RELOC=COMREG   NAME=UDS      DUMP=  20 TID=   12
R 0- 7   00000001 00000000 0000F000 00203000 000130C8 00096900 A020D92A 00211F14
R 8-15   00211660 0020F2D8 5006EDBC 0006FDBB 0020CFB0 0020D670 00211868 00000000
COMREG  +00000000
00000000 00096000 006400C0 000162E8 008EE468      * .-  . ä ..Y .U.*
00000010 008FF2E0 00000000 05D40B40 00010004      * .2Ö    .M.  . .*
00000020 00000004 00010058 00060007 0A031F11      *   . . . . .....*
00000030 0AC80000 4C000A0E 0000F03E 00000001      *.H  < ..  0.   .*
00000040 61404040 40404040 40404040 40404040      */               *
00000050 000050C8 00030001 000000E5 F4F6F024      *  &H . .   V460.*
00000060 808DB808 00780000 00000000 85DF8000      *.... .      e.. *
00000070 B20A0000 85D26008 85DEB930 0001C610      *..  eK-.e... .F.*
00000080 85D07068 005C0080 B20A0080 0001FF50      *eü.. * ... . ..&*
00000090 0009ABC0 00017EE8 0094251F 0034292A      * ..ä .=Y m.. ...*
000000A0 C5E2C140 F0F3F840 30900004 85D07BD8      *ESA 038 .. .eü#Q*
000000B0 0603EAD0 00000000 85D09F20 85D021D0      *...ü    eü..eü.ü*
000000C0 85DE7050 00000000 85DE9058 0001B0A0      *e..&    e... ...*
000000D0 0001B0A0 0001B9B4 0001BD8A 85D25CA0      * ... ... .¨.eK*.*
000000E0 0001A010 85DF60B0 85DE6020 85D5F730      * ...e.-.e.-.eN7.*
000000F0 85D061F8 85DEFA40 85D0D7F0 0001596C      *eü/8e.. eüP0 ..%*
00000100 85DE79E0 86072898 85DE3038 85D3D080      *e.'Öf..qe...eLü.*
00000110 00000000 85DDB0E0 85DEA078 00015D18      *    e..Öe... .).*
00000120 00000000 00015F50 85D09838 85DEB018      *     .^&eüq.e...*
00000130 05DFF000 85DDC028 85DE5220 85DF32B8      *..0 e.ä.e...e...*
```

REGS Command

The REGS command enables you to display the general registers and PSW at the time of program failure.

The command format is:

REGS

The information displayed includes:

- The general registers;
- The current PSW;
- UDUMP header information;
- The floating point registers;
- An overview of the sections of storage contained in the dump.

The following figure illustrates a typical REGS display format.

```
  NAME=UDS       DATE= 11/04  TIME= 163648  DUMP=   5 TID=     8
COMABS0006-* PROGRAM UDS ABEND S0C4 PSW=078C1000800284CC

    PSW  078C1000 800284CC
R 0- 7   00000000 00039A48 0000F000 0A8A6000 0001F000 008B9C50 00000002 00F3B2E0
R 8-15   00039A00 00FD4B18 800284BE 0001238C 00028318 00053840 00000322 00000000

FR0- 6   00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

UPCB     0A8A6000-0A8A7000
SQA2     001F5000-001F6000
REGION1  0022B000-00233B30
REGION2  0A8A7000-0A8A7B00
COMREG   0000F000-00010000
THCB     0001F000-0001F400
TIB      09FCE600-09FCE6C0
CUOW     09C29000-09C29290

Program  EntryPnt LoadPnt  End      Length   Location PSW

UDS      8022B000 0022B000 002333A8 000083A8 thread
```

Address Relocation Options

The address relocation options enable you to set the addresses used in the display relative to either:

- The machine
- The thread
- The program
- The current address
- A specified dump section

When you specify a relocation command, the current display remains the same, but Com-plete adjusts the addresses displayed in the left margin according to the command specified.

R=M Command

The machine address relocation command enables you to display the addresses relative to absolute zero (machine addresses).

The command format is:

R=M

R=P Command

The program address relocation command enables you to display the addresses relative to the program load address, if this is the dump.

The command format is:

```
R=P
```

R=* Command

The specific address relocation command enables you to display the addresses relative to the address that is current when you enter this command.

The command format is:

```
R=*
```

R=T Command

The thread address relocation command enables you to display the addresses relative to the beginning of the thread, that is, of the UPCB.

The command format is:

```
R=T
```

R=section name

The dump section relocation command enables you to display the addresses relative to the beginning of the specified dump sections.

Functional Considerations

Com-plete interprets the various addressing commands available with UDUMP relative to the relocation option in effect. For example, when program relocation is in effect, you can only display addresses located within the program area. The address you specify may be either an actual machine address or an address relative to the program load point. The location you address, however, must be within the bounds of the program.

For example, if the thread address is X’E0000’ and the program load address is X’E4000’, the display returned is the same whether you specify X’4A0’ or X’E44A0’. A request for address X’E3480’ is not honored because it precedes the program load point, but a request for address X’3480’ displays that relative location in the program (actual machine address X’E7480’).

When a specific address relocation is in effect, you can only display address locations greater than the specified address. Thus, if the specific starting address was X’A48’ when you requested specific address relocation, only locations from X’A48’ to the end of the thread will be available for display.

Note:
Display requests using the CRn, XRn or PSW commands require that the address is in the register specified, or that the PSW is within the area under the relocation option in effect; otherwise, an addressing error will result.

PGMS command

This command displays a list of the programs that were loaded on behalf of the application when the error occurred. For each program, its name, location and size are indicated. If the error PSW points into one of these modules, the displacement is shown.

```
  PSW  078C1000 800284CC                NAME=UDS      DUMP=   5 TID=    8
R 0- 7   00000000 00039A48 0000F000 0A8A6000 0001F000 008B9C50 00000002 00F3B2E0
R 8-15   00039A00 00FD4B18 800284BE 0001238C 00028318 00053840 00000322 00000000

Program  EntryPnt LoadPnt  End      Length   Location PSW

UDS      8022B000 0022B000 002333A8 000083A8 thread
```

Print Dumps

The PRINT command enables you to spool the currently displayed dump to a terminal printer. The device is typically a printer, but can be any device in the Com-plete network. Note that the receiving device must be able to print a 132-character line in order to format the dump properly.

The command format is:

`PRINT [destination]`

where destination is the one- to eight-character destination code of the device(s) on which the dump is to be printed.

The destination code can be a Terminal ID (TID) or groupname to which one or more devices are assigned. If you specify an invalid destination code, an error message is issued and you are prompted to reenter the PRINT command. If you do not specify a destination, the default screen-to-hardcopy device defined for the terminal is used. This default terminal is defined for each terminal in the Terminal Information Block Table (TIBTAB) but can be altered by using the HC function of the UUTIL utility.

UDYN - Dynamic Dataset Allocation and Deallocation (z/OS only)

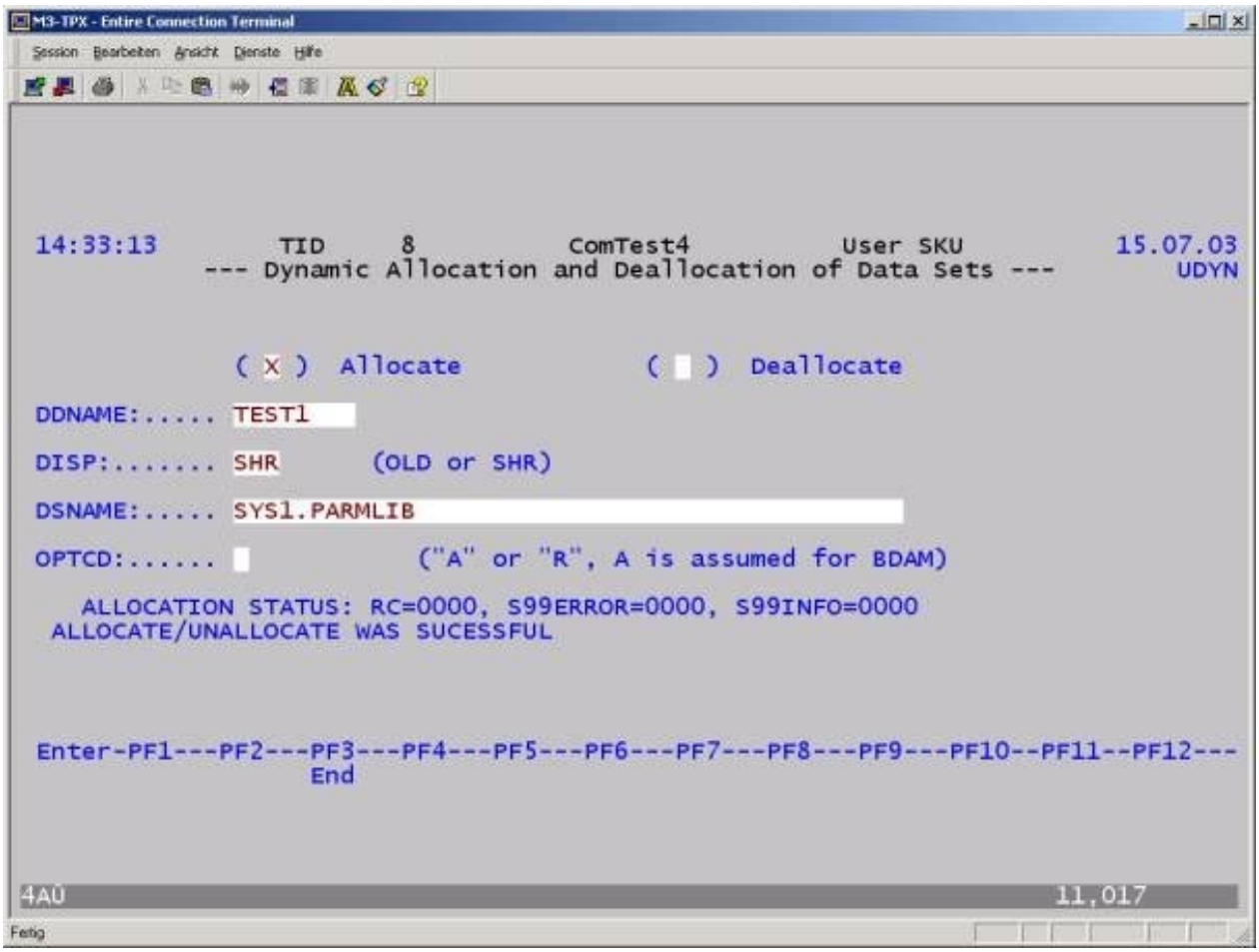
This utility program can be used for allocating and deallocating datasets to/from the Com-plete address space without having to restart Com-plete. While UDYN uses the operating system service DYNALLOC, it supports only the following basic parameters of DYNALLOC:

Allocation

- DD name (required),
- Disposition (required, only the values “OLD” and “SHR” are supported),
- Dataset name (required),
- OPTCD (optional).

Deallocation

- DD name (required). Any other parameters entered are ignored with the deallocation function.



UDYN – Dynamic Dataset Allocation and Deallocation

Upon successful or unsuccessful completion of the operation, the DYNALLOC return code, error code, and information code are displayed, along with a short description text.

Note:
The deallocation function will deallocate any valid DD name entered without further warning. No check is performed whether or not that file might be vital for Com-plete to continue running.

UDZAP - DASD Modification Utility (Control User)

The UDZAP utility, designed for use with 3270-type terminals, enables you to perform specific CCHHR zaps to a specified DASD volume. A UCB routine locates the specified volume by either:

- CUU address;
- Volume identification name.

Note:
VSE installations must have, for each disk device accessed by UDZAP, a logical unit assignment (VSE LUB) in the VSE partition in which Com-plete resides.

This chapter covers the following topics:

- Command Format

Command Format

UDZAP is entered by issuing the request:

```
*UDZAP
```

This call displays the following menu:

```
CHOOSE FUNCTION UCB LOOKUP -- PF1
                ZAP DISK   -- PF2
```

Once this menu is displayed, you can perform additional functions. Optionally, the additional functions can be invoked as part of the UDZAP call.

The initial menu display explicitly specifies that one of two major functions can be performed:

UCB LOOKUP	Performs a UCB/PUB lookup to obtain information about a specific volume and displays its UCB/PUB.
ZAP DISK	Displays a menu used to enter specific CCHHR/PBN information and data to be verified and replaced at the CCHHR/PBN.

The function to be performed can be selected in one of three ways:

- Press the appropriate PF key as illustrated in the menu.
- Enter the characters that describe the function, as follows:

  ```
  UCB/PUB
  ZAP
  ```

- Specify the function when calling UDZAP as follows:

```
*UDZAP UCB/PUB
*UDZAP ZAP
```

Selecting a Volume

The volume on which the chosen function is to be performed can be identified by CUU or volume identification name. The choice is initially indicated in the display obtained after a specific UCB/PUB or ZAP function is selected, as shown in the following figure.

```
ENTER CUU ___ OR VOL ID______
```

To choose which CUU or volume to reference, enter the desired CUU value or volume identification in the appropriate input field.

The following pages illustrate and describe typical displays.

UCB/PUB Lookup Display

Note that this display does not permit the changing or entering of data. Its only function is to allow you to view UCB information.

z/OS Display

```
UCB ADDR 1A3C UCB 0048FF8C 046A0020 013A0300 00F4F6C1 3050200D ......
    F1401049 1DFC0100 00000000 00000000 00000001 ...... VOL   ID SAG01
DEV ADDR--46A
UCB TYPE--MODEL CODE 0 OPT FEAT 50  DEV CLASS 20 .....
PF1= +1 CUU  PF2= -1 CUU  PF3= NEW PACKID ENTERED .....
CUU ENTERED
```

VSE Display

```
PUB ADDR 015E
PUB 0048FF8C 046A0020 013A0300 00F4F6C1 3050200D ......     F1401049 1DFC0100
00000000 00000000 00000001 ......

VOL   ID DOSVSE
DEV ADDR--15F
PUB TYPE--PUB  OPT  00 CHN FLAG FF  JCL FLAGSFC  UNIT TYPE 65
PF1= +1 CUU  PF2= -1 CUU  PF3= NEW PACKID ENTERED .....
CUU ENTERED
```

The PF key definitions listed in the display enable you to either continue the UCB lookup function or to switch to the ZAP function. Specific PF key meanings and usage are:

PF1	Causes the display of the next sequential CUU. Either: a. Press the PF1 key. b. Enter the characters "PF1".
PF2	Causes the display of the preceding CUU. Either: a. Press the PF2 key. b. Enter characters "PF2".
PF3	Causes the UCB/PUB of the designated volume to be displayed. Either: a. Enter the volume identification of desired volume and press the PF3 key. b. Enter the characters "PF3=vvvvvv", where vvvvvv is the desired volume identification.
PF4	Causes the function selection menu to be displayed. Either: a. Press the PF4 key. b. Enter the characters "PF4".
PF5	Causes the UCB of the designated CUU to be displayed. Either: a. Enter the CUU of the desired UCB and press the PF5 key. b. Enter the characters "PF5=cuu", where cuu is the desired CUU value.

Zap Volume Display

The following figure illustrates a typical zap volume display.

```
FOR CUU 150 VOL ID DISK01
ENTER CYL/PBN___HEAD___REC___LGH-KRR     VER DATA________ ________
    REP DATA________ ________
ASSUME DECIMAL EXCEPT VER AND REP DATA (PRECEDE HEX ... )
```

You can enter data in the appropriate input fields. As indicated in the last line of the zap display, VER and REP data is assumed to be hexadecimal. Other data entered is assumed to be decimal. If hexadecimal is desired, the entry must be preceded by an "X". Note that it is not necessary to zero-fill any fields.

When zap information is entered and ENTER is pressed, the above screen is redisplayed either showing the accepted zap, or, in the case of an error, giving notification of the verify/rejection and permitting another attempt at the zap entry.

The following display illustrates a typical response received for a successful verify/replacement. The CCHHR value illustrated is for cylinder 1, head 1, record 1, offset 0.

```
FOR CUU 356 VOL ID DISK01 ENTER CYL/PBN__HEAD 1__REC __OFFSET 0__  LGH-KRR
002A28     VER DATA 90ECD00C 185D05F0
    REP DATA 90EFD00C 185D05F0
 ASSUME DECIMAL EXCEPT VER AND REP DATA (PRECEDE HEX ...
DONE. NEW DATA+ENTER=ZAP. PF1=NEW CUU. PF2=NEW FUNCTION.
```

The functions described at the bottom of the preceding display are:

NEW DATA	Tab the cursor to the input fields in the display and reenter additional CCHHR and verify/replace data.
PF1	Causes the CUU/VOLUME selection menu to be displayed. Either: a. Press the PF1 key. b. Enter the characters "PF1".
PF2	Causes the function selection menu to be displayed. Either: a. Press the PF2 key. b. Enter the characters "PF2".

When a zap is verified, the replacement zap is performed and a message summarizing the activity is sent to the default Com-plete logging terminal. The following figure shows a typical response received at a hard copy terminal.

```
MSG ID:   30, SENT 2/11/87 AT 1227, FROM TID   1
ZAP VOL DISK01 AT 0001000101 +0 ___

                   VER 90ECD00C 185D05F0
                   REP 90EFD00C 185D05F0
```

If the verify/replacement is unsuccessful, the entry screen is redisplayed indicating an error condition. To correct the error, tab to the input fields and reenter the correct information.

The following figure illustrates the display obtained from an unsuccessful attempt to modify data.

```
FOR CUU 356 VOL ID DISK01
ENTER CYL 1__HEAD 1__SEC 1__OFFSET 0__
LGH-KRR 002A28     VER DATA 90ECD00C 185D05F0
    REP DATA 90EFD00C 185D05F0
 ASSUME DECIMAL EXCEPT VER AND REP DATA (PRECEDE HEX ...
VER DATA DOESN’T MATCH RECORD: REENTER
```

UED - Line-Oriented Source Editor Utility

UED is an online utility with which you can edit source text using any available terminal device type with both input and output capability. Specifically, UED can be used with hard copy type terminals such as TTY devices and 2741 terminals.

With UED, all editing operations are performed in editor work files allocated within Com-plete SD files. First, an SD file is allocated; then the library or member is read into the SD file. Next, you edit the data in the SD file, and finally use the SAVE command to write the data back to the indicated library member.

In addition to dynamic allocation of sequential or partitioned data sets, the UED utility includes these features:

- You can use any library containing source text as input or output to UED.
- For z/OS, input and output data sets are restricted to the following DCB conventions:

  ```
  RECFM=F or RECFM=FB
  DSORG=PS or DSORG=PO
  ```

- In z/OS, BLKSIZE value of any input or output library is restricted only by the size of the thread in which UED is to execute. If files with large BLKSIZE are encountered while using UED, editing may not be possible. In this situation, you will need the assistance of the system programmer responsible for Com-plete maintenance.
- In z/OS, you can use any sequential data set containing source text as input or output to UED.
- You can select statements from other members and other libraries for insertion into the text you're editing.
- Editing is either statement number-oriented or tag-oriented, depending upon your selection.
- You may use a full range of editing commands using a well-defined syntax structure.
- A user-written subroutine can be used for establishing security restrictions.
- A table of library identification codes enables you to use a two-character reference instead of entering the full file name.

This chapter covers the following topics:

- Command Format
- UED Commands

Command Format

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

The basic command format is:

```
*UED
```

Once you have entered this command, conversation is established with UED and you can then enter the various command functions.

Optionally, you can use the command format:

```
*UED function
```

where *function* is any of the command functions defined for UED.

Note:
Unlike the UEDIT utility, termination of UED requires that you understand the termination command END as described later in this chapter. Pressing the CLEAR key on a 3270-type terminal or "control/D" on a TTY-type terminal terminates UED *only* if you have not modified the SD work file in use.

UED Commands

You can abbreviate all UED commands. In the command descriptions that follow, the minimum abbreviations allowed are indicated by an underscore.

Function	Description
CHANGE	Changes one or more occurrences of a given character string to another character string.
COPY	Copies a designated set of statements.
DEFINE	Defines or resets the default characteristics assumed by UED during the editing session.
DELETE	Deletes one or more statements.
END	Terminates the editing session.
FIND	Finds a designated character string.
HALT	Halts the execution of a command or suppresses output from UED to the terminal.
HELP	Displays the syntax for a given command (same as ? command).
INSERT	Inserts one or more statements.
LIST	Lists one or more statements.
MACRO	Defines the instream UED functions to be executed.
MOVE	Relocates one or more statements.
NUMBER	Resequences one or more statements.
PRINT	Obtains a hard copy listing of one or more statements.
QUERY	Displays the current default values (set by using the DEFINE command).
READ	Reads or obtains from a library one or more statements or members.

Function	Description
RECOVER	Recovers the UED work file after a system failure.
REPEAT	Repeats a line or command as previously entered.
RESUME	Resumes typing (after a HALT command).
SAVE	Saves all changes made during an editing session.
SUBMIT	Submits a job stream to batch for execution.
TAB	Lists the current tab settings.
+n	Pages forward n statements.
-n	Pages backward n statements.
++	Pages to the last statement.
--	Pages to the first statement.
/tag	Pages to the statement containing a specified value (tag), where tag is any string of characters that begins in the designated tag column.
n	Pages to or modifies the statement with sequence number n.
?	Synonym for HELP.

Note that you can request more than one command function at a time, provided that each function is separated by the command delimiter.

Note:
If an error results while a series of commands is executing, the command function in error is terminated and the subsequent commands are ignored.

You can dynamically change most default values used by UED, including the command delimiter (the default is a semicolon), at any time during execution of UED or at invocation of UED by using the DEFINE command.

The descriptions of the various UED commands in the following sections refer to subscripted items such as *num*, *addr*, and *st*. Note that the following definitions apply to all such references unless otherwise specifically stated:

num	Any positive integer, zero, or the word ALL, which specifies all applicable occurrences.
addr	Any of the following values:
	number — Specifies any statement sequence number. You need not supply zeros unless the sequence number is more than ten digits. string — Specifies any string preceded by the designated tag character (the default is a slash). -- — Specifies the first statement of the file to be edited. ++ — Specifies the last statement of the file to be edited. * — Specifies the current statement. You can also use one of the following values with +n or -n (where n is an integer) as a replacement for addr in order to establish relative addressing: +n — indicates forward relative addressing by n statement(s). -n — indicates backward relative addressing by n statement(s).
st	Any string of characters.
	Delimiter UED command delimiter for strings If the character string contains one or more delimiters (for example, blanks, semicolons, commas) or is a null character string, the string must be enclosed by string delimiters (for example, single quotation marks). If the string contains a quotation mark, the default string delimiter and the quotation mark must appear twice.
	Note that the default string delimiter (single quotation marks) is overridden using the DEFINE command.

Note that several arguments in the following command descriptions are defined as *addr* arguments but their descriptions indicate that they may be *num* (numbers) as well. Such arguments must be preceded by the default number identifier # in order to ensure that the number attribute is attached.

Note also that all commands except DEFINE have positional arguments. You must replace any intermediate argument that is to be omitted for a given command by an operand delimiter (for example, comma). If you omit trailing arguments, however, a replacement operand delimiter is not required.

You can use the SAVE command to replace an existing member of a source library. If the member does not exist, a new member is created.

In z/OS, you can overwrite an existing sequential file with the SAVE command.

CHANGE Command

The CHANGE command enables you to change one or more occurrences of a given character string within a statement to another character string. You can define the number of occurrences per statement (one line) as well as the number of statements.

The command format is:

```
CHANGE st1,st2[,num][,addr1][,addr2]
```

The arguments are:

st1	Required.
	Specifies the character string to be changed.
st2	Required.
	Specifies the replacement character string.
num	Optional.
	Default: ALL
	Specifies the maximum number of occurrences per statement of the character string for which the change will occur.
	Note that the change occurs left to right, per statement, for this number of occurrences.
addr1	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the first statement for which the change is to occur.
addr2	Optional.
	Default: #1
	Specifies the address of the last statement for which the change occurs or, if stated as a number (e.g., #25), the number of statements to be affected.

Note that you can only change a character string when the entire string is contained within the following ranges:

- Before the sequence columns;
- In the sequence columns;
- After the sequence columns.

COPY Command

The COPY command enables you to copy one or more statements.

The command format is:

```
COPY [addr1][,addr2][,addr3][,num]
```

The arguments are:

addr1	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the first statement to be copied.
addr2	Optional.
	Default: #1
	Specifies the address of the last statement to be copied or, if stated as a number, (e.g., #25), the number of statements to be copied.
addr3	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the statement after which the copied statement(s) is to be placed. If "--" is specified, the copy is placed before (not after) the first line.
num	Optional.
	Default: 1
	Specifies the number of copies to be generated. Use this argument to repeat one or more statements.

Note that the COPY command uses the currently allocated SD work file to capacity before generating an error message. For example, if you request 10 copies of 8 lines and there is only space for 60 lines, the 60 lines will be created, after which an error message is generated: "SD FILE TOO SMALL".

DEFINE Command

The DEFINE command enables you to change the default edit characteristics assumed by UED (as opposed to the QUERY command, which is used to display existing default values).

The command format is:

```
DEFINE key1=[value1][,key2=[value2]]...[,keyn=[valuen]]
```

where each *keyn* is one of the keywords described in the following text. Note that if you give a keyword argument with no value specified, the keyword is initialized to the standard default value.

Available keywords and their default values are:

CMDDEL	Optional.
	Default: CMDDEL=;
	Specifies the command delimiter character that is to be used to separate commands when entering multiple commands.

CON	Optional.
	Default: CON=YES
	Specifies whether or not the terminal device buffer size is the limitation factor that is to be placed upon the amount of output to be generated by any one command function.
	Possible values are YES and NO.
	CON=YES specifies that the buffer size for the terminal is to be considered as one output display. For a TTY device, the size is 2000 characters; for a 3277 model 2 device, 23 statements. To continue output, press ENTER.
	CON=NO specifies that all output will be written continuously.
DSN	Optional.
	Default: DSN=null value
	Specifies the name of the file to be used when the READ command function is executed.
	If this keyword is initialized to a file name, all READ commands entered without arguments obtain their input from this file.
	If this keyword is initialized to a null value, note that all READ commands entered must specify the source file that is to be used for input.
LISTCOL	Optional.
	Default: LISTCOL=73-80,1-72
	Specifies the order of the columns to be displayed when using the LIST command.
LRECL	Optional.
	Default: LRECL=80
	Specifies the logical record length that is to be assumed for the SD work file while editing. If a SAVE command is entered, the LRECL is not changed in the output file.
	Note that the value specified must be between 2 and 247. If an odd value is specified, the actual LRECL used is rounded to the next largest even value. **Notes:** 1. Padding with blanks occurs on both input and output files if the input and output files have an LRECL smaller than the LRECL specified for UED. 2. Truncation occurs on both input and output files if LRECL is less than the LRECL of the input and output file.

MEMB	Optional.
	Default: MEMB=null value
	Specifies the name of the member in a source library to be used for input to UED.
	If this keyword is initialized to a specific value, all READ command functions will default to this member name if the member name is not given in the READ command.
	If this keyword is initialized to null values and a source library is to be edited, note that all READ commands must specify a member name.
NUMCHAR	Optional.
	Default: NUMCHAR=#
	Specifies the character to be used as the prefix when command arguments as numbers are indicated.
OPDEL	Optional.
	Default: OPDEL=,
	Specifies the character to be used as the argument delimiter for commands.
SEQCOL	Optional.
	Default: SEQCOL=73-80
	Specifies the column numbers to be used for sequence numbers.
	Note that the value specified must be in the format nn-nn.
	SEQCOL=N indicates that sequence columns are not to be used.
SIZE	Optional.
	Default: SIZE=1500
	Specifies the number of records to be used when allocating the SD work file. The actual number of records used depends upon both the device type (3330, 3350, etc.) on which the SD data set is to be stored (full track utilization always occurs) and the LRECL value specified for the SD work file.
STRDEL	Optional.
	Default: STRDEL=’
	Specifies the character to be used when character strings containing other delimiters (e.g., blanks, semicolons, commas, or a null string) are delimited.
TABCHAR	Optional.
	Default: TABCHAR=%
	Specifies the character to be used when setting tab stops for the terminal.

TABCOL	Optional.
	Default: TABCOL=10,16,41,72,73
	Specifies the default columns at which tabs are to be placed when the TAB command function is used.
	Note that the maximum number of tab settings that can be used is 10. The tabs must be specified one at a time.
	TABCOL=N indicates that tabs are not to be used.
TAGCHAR	Optional.
	Default: TAGCHAR=/
	Specifies the character to be used as the prefix when the data being edited for a given tag is searched.
TAGCOL	Optional.
	Default: TAGCOL=1-8
	Specifies the columns to be assumed by UED as the beginning and ending columns when a tag command function request is processed.
	Note that the value specified should be in the format n-n.
TRI	Optional.
	Default: No translation occurs
	Specifies the translation to occur on characters input from the terminal during editing.
	The value entered should be in the format:
	where the quotation marks must be entered. Each xx represents the hexadecimal format of a character input from the terminal. Each yy represents the hexadecimal format of the resulting translation.
TRO	Optional.
	Default: No translation occurs
	Specifies the translation that is to occur on characters output to the terminal during editing.
	The value entered should be in the format:
	where the quotes must be entered. Each xx represents the hexadecimal format of a character prior to output to the terminal. Each yy represents the hexadecimal format of the resulting translation on output.

VER	Optional.
	Default: VER=ON
	Specifies whether or not statement verification is to be on.
	Two values are permitted: ON and OFF.
	VER=ON indicates that every statement modified during editing is displayed at the terminal for modification verification. The current line is also displayed after every command that alters the current line pointer.
	VER=OFF indicates that statements modified during editing will not be displayed at the terminal for modification verification. The current line is not displayed when the current line pointer is changed.
VOL	Optional, except as noted below for VSE.
	Default: VOL=null value
	Specifies the volume identification number of the direct access volume on which a file being read with the READ command function will be located.
	Note:/tab/If this keyword is initialized to a specific volume, that volume and only that volume will be searched for the input READ command file.
	In z/OS, if this keyword is initialized to a null value, the system catalog will be used to resolve which volume to locate the READ command file on (unless the volume is indicated in the READ command).
	In VSE, this keyword is required for the READ command.

DELETE Command

The DELETE command enables you to delete one or more of the data statements that you are editing.

The command format is:

```
DELETE [addr1][,addr2]
```

The arguments are:

addr1	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the first statement to be deleted.
addr2	Optional.
	Default: #1
	Specifies the address of the last statement in a range of statements to be deleted or, if stated as a number (e.g., #25), specifies the number of statements to be deleted.

If the current line is not deleted, the current line pointer is not changed; otherwise, the current line pointer is set to the address of the line immediately following the second argument address. One exception is when the second argument specifies the address of the last line; in this case, the current line pointer is set to the new last line.

To clear the SD work file without terminating UED, specify:

```
DELETE --,++
```

END Command

The END command enables you to terminate the UED utility program.

Note:
Termination of UED depends upon the status of the SD work file.

The command format is:

```
END [n]
```

where n is an optional argument with one of the following values:

I	Terminates UED immediately and deletes the SD work file.
H	Terminates UED immediately but holds or does not delete the SD work file. The SD work file may subsequently be recovered using the RECOVER command function of UED.

If you use the END command function with no arguments, UED is terminated only if the SD work file has not been modified since the last SAVE or SUBMIT function.

Note that use of either the CLEAR key on a 3270-type terminal, "control/D" on a TTY-type terminal, or the Com-plete program termination command *EOJ is identical in function to use of the END command with no arguments.

FIND Command

The FIND command enables you to find or locate one or more occurrences of a designated character string.

The command format is:

```
FIND st[,num][,addr1][,addr2]
```

The arguments are:

st	Required.
	Specifies the character string to be found.
num	Optional.
	Default: 1
	Specifies the number of occurrences of the designated character string to be found.
	ALL specifies that all occurrences are to be found.
addr1	Optional.
	Default: *+1
	Specifies the address of the statement in which the search is to begin.
addr2	Optional.
	Default: ++
	Specifies the address of the statement in which the search is to terminate or, if stated as a number (e.g., #25), specifies the number of statements that are to be searched.

The FIND command sets the current line pointer to the line where the last occurrence of the designated string was found. If the string is not found, no change in the line pointer will occur.

HALT Command

The HALT command enables you to terminate command execution within UED or terminate output of UED (for example, at a hard copy terminal).

The command format is:

```
HALT [xxx]
```

where the argument *xxx* can be:

TYPING	Specifies that all UED output is suppressed until the end of the current command line is reached or until a RESUME command is entered.
	This argument can be abbreviated with T.
EXECUTION	Specifies that execution of the current UED command function (e.g., LIST) will be terminated and all other command functions entered as part of the same command line will be ignored.
	This argument can be abbreviated with X.

If you do not enter any arguments with the HALT command, the TYPING argument is taken as default.

If you specify CON=YES with the DEFINE command function, the HALT command is the only command function that terminates command execution at the end of one screen of terminal output when more output is to be displayed.

HELP Command

The HELP command enables you to display syntax information about a given command or display a list of all commands available for use with UED.

The command format is:

```
HELP [command]
```

where the optional argument *command* represents any command name recognized by UED. If you enter a command name, a syntax description of that command is displayed. If you do not specify argument, a list of available command function names is generated.

You can optionally execute the HELP command by using the format:

```
? [command]
```

where the functions performed and the description of the command are the same as described above. The question mark (?) is a synonym of the character string HELP.

INSERT Command

The INSERT command enables you to insert one or more lines after a designated line.

When you enter the INSERT command, you are placed in INSERT mode and remain in this mode until you enter a null statement (that is, press ENTER without entering additional data).

The command format is:

```
INSERT [addr][,S]
```

The arguments are:

addr	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the statement after which the insertion will be made.
	Note that if "--" is specified, the insertion is made before (not after) the first line.
S	Optional.
	Default: None
	Specifies that, as each additional statement is added, the characters entered preceding the first blank statement are to become the sequence number. The characters after the first blank become the data starting in column one or, if the sequence columns start in column one, the first column after the sequence columns.

LIST Command

The LIST command enables you to display one or more statements without changing the current line pointer.

The command format is:

```
LIST [addr1][,addr2]
```

The arguments are:

addr1	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the first statement to be listed.
addr2	Optional.
	Default: #1
	Specifies the address of the last statement in the set of statements to be listed or, if stated as a number (e.g., #25), specifies the number of statements to be listed.

MACRO Command

The MACRO command enables you to define a string of commands that are to be executed each time you enter the name of the macro definition.

Note:
Macro statements are valid only for the duration of one execution of UED. As soon as UED terminates, all macro definitions for that execution will be lost.

Each macro definition must fully reside in one statement. If you wish to use more than one statement, you must create multiple macro definitions.

Macro statements, by definition, specify UED commands to be executed when the macro statement is issued by name. Since a macro statement is also a UED command, it can issue another macro statement, or even issue itself. Thus, you can chain macro statements together by having each macro execute one or more other macro statements.

Three command formats are available for a macro definition.

DEFINE

The first MACRO command format enables you to define a specific macro statement.

The command format is:

```
MACRO DEFINE name cmd1[;cmd2]...[;cmdn];MEND
```

The arguments are:

DEFINE	Required.
	Specifies that the DEFINE function of the MACRO command is to be executed.
name	Required.
	Specifies the name by which this macro statement is to be known and referenced.
	Note that name must follow standard naming conventions (that is, eight characters or less, the first character being alphabetic).
cmd1	Required.
	Specifies the UED command to be executed when the macro name is used as a command function.
cmdn	Optional.
	Default: None
	Specifies one or more additional UED command functions to be executed when the macro name is entered as a command function.
	Note that each command function must be separated by the command delimiter defined by the CMDDEL keyword of the DEFINE command.
MEND	Required.
	Is a constant designating the end of the macro definition.

Example:

The following example defines a macro command that, when executed, will list statements 200 through 400. In addition, 800 statements will be scanned beginning with statement 500, and a listing of those statements will be given for those containing the character string "programmer":

```
MACRO DEFINE A L 200,400;F PROGRAMMER,1,500,#800;MEND
```

You could then execute this macro command simply by entering:

```
A
```

LIST

The second format for the MACRO command enables you to obtain a listing of all currently defined macro names or a listing of one or more macro definitions.

The command format is:

```
MACRO LIST [name1][,name2]...[,namen]
```

The arguments are:

LIST	Required.
	Specifies that the list option of the MACRO command is to be executed.
name	Optional.
	Default: All current macro names as defined by MACRO commands are listed.
	Specifies the name of a previously defined macro definition. If one or more macro names are given, each macro definition is listed.

Example:

To obtain a list of all currently defined macro statements, you would enter:

```
MACRO LIST
```

The output of this request lists the macros currently defined by name.

To obtain a display of a specific macro definition (macro A in this example), issue the command:

```
MACRO LIST A
```

DELETE

The third format for the MACRO command function enables you to delete a previously entered macro definition.

The command format is:

```
MACRO DELETE name[,namen]...
```

The arguments are:

DELETE	Required.
	Specifies that the delete option of the MACRO command function is to be executed.
name	Required.
	Specifies the name of the macro definition to be deleted.
namen	Optional.
	Specifies one or more additional macro definitions to be deleted.

MOVE Command

The MOVE command enables you to move one or more statements within the data being edited.

The command format is:

```
MOVE [addr1][,addr2][,addr3]
```

The arguments are:

addr1	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the first statement to be included in the move operation.
addr2	Optional.
	Default: #1
	Specifies the address of the last statement to be included in the move operation or, if stated as a number (e.g., #25), specifies the number of statements to be moved.
addr3	Optional.
	Default: * (indicates the current statement)
	Specifies the address of the statement after which the move or relocation will occur.
	Note that if "--" is specified, the move is made before (not after) the first statement.

If the current line address is not included in the range of statements you are moving, the current line address is not changed; otherwise, the current line address becomes that of the statement following the last statement indicated by the second argument (addr2).

NUMBER Command

The NUMBER command enables you to change the statement sequence numbers for the statements being edited.

The command format is:

```
NUMBER [num1][,num2][,label][,N][,addr1][,addr2]
```

The arguments are:

num1	Optional.
	Default: 10
	Specifies the sequence number to be used for the first statement.
num2	Optional.
	Default: Same as that specified for num1.
	Specifies the increment to be used when the statements are numbered.
label	Optional.
	Default: None
	Specifies the label or identifier that is to be placed in the leftmost position of the sequence number.
	Note that label can be any identifier. If the constant DATE is chosen as a replacement for label, the current date is placed in the leftmost five positions of each sequence number in the format YYDDD.
N	Optional.
	Default: All statements will be renumbered.
	Specifies that only those statements that have been modified or inserted since the beginning of the session will be renumbered.
addr1	Optional.
	Default: --
	Specifies the address of the first statement that is to be renumbered.
addr2	Optional.
	Default: ++
	Specifies the address of the last statement in a range of statements to be renumbered or, if stated as a number (e.g., #25), indicates how many statements are to be renumbered.

PRINT Command

The PRINT command enables you to obtain a hard copy printout of the data being edited.

The command format is:

```
PRINT [code][,format][,addr1][,addr2]
```

The arguments are:

code	Optional.
	Default: The screen-to-hardcopy terminal assigned to the terminal in use (if any); otherwise, the default is the same terminal. The screen-to-hardcopy terminal can be defined in the Com-plete Terminal Information Block Table (TIBTAB) or altered by using the HC function of UUTIL.
	Specifies the Terminal Identification number (TID) of the terminal to receive the printout or terminal name.
format	Optional.
	Default: FORMAT
	Specifies the format to be used when printing.
	This argument can have one of three values: FORMAT, CC, and NOCC.
	FORMAT specifies that all output is arranged in pages, 60 statements per page, with a standard page header identifying the user ID, originating TID, library name, member name, date, and time.
	Note that all pages are sequence numbered.
	CC specifies that column one of each statement being edited contains a standard ASA carriage control character to control formatting of the printout. The standard ASA carriage control characters are:
	blank advance 1 line and print 0 advance 2 lines and print - advance 3 lines and print 1 advance to a new page and print + suppress spacing
	Note that no header is provided and pages are not sequence numbered.
	NOCC specifies there are no carriage control characters in column one of the statements being edited. Page overflow printing is in effect and all statements print with no carriage control.
addr1	Optional.
	Default: --
	Specifies the address of the first statement with which printing begins.
addr2	Optional.
	Default: ++
	Specifies the address of the last statement in a range of statements to be printed or, if stated as a number (e.g., #25), specifies the number of statements to be printed.

QUERY Command

The QUERY command enables you to display the default UED editing characteristics as defined or defaulted by the DEFINE command.

The command format is:

```
QUERY [key][,key]...
```

The arguments are:

key	Optional.
	Default: If no key is given, a list of all default keyword values will be given.
	Represents a keyword argument, less the equal sign, as used in the DEFINE command. For each keyword argument listed, a display of the current value assigned to that argument will be given.
	As with the DEFINE command, the keywords can be abbreviated to the shortest number of characters that uniquely identifies them.

READ Command

The READ command enables you to read or obtain data statements from a library for editing purposes.

The data being read can be a member of a source library or (in z/OS) a sequential file. Note that if you specify a source library, you must either specify a member or default to the member initialized with the MEMB keyword in the DEFINE command.

You can read an entire member or portions thereof. All data read is placed in the SD work file, if it already exists, in the location that you specify. If the SD work file does not currently exist, a new one is created.

Note:
If you issue a READ command, subsequent READ commands you issue without terminating UED will concatenate additional data statements to those originally read. To avoid this, as well as the need to terminate UED, make sure to delete all statements from the SD work file before you perform a subsequent READ by using the DELETE command.

Before using the READ command, you must be familiar with the DSN, MEMB, and VOL keyword options of the DEFINE command function.

The format of the READ function is:

```
READ [library][(member)][/volume][,num1][,num2][,addr]
```

Note that the first three argument values (library, member, and volume), if entered, must not be separated by spaces or commas.

The arguments are:

library	Optional.
	Default: None, unless the DEFINE command function is used to initialize the keyword DSN.
	Specifies the name of the input source library.
	If two-character library identification codes are used (as is done with the UEDIT utility program), the data set name need not be entered; simply enter the two-character code. Note, however, that these codes must be either defined to UED in the table of library identification codes (UEDTB1) or defined by the UL function of the UUTIL utility.
member	Optional.
	Default: None, unless the DEFINE command function is used to initialize the keyword MEMB.
	Specifies the member in a source library to be used as input.
	Note that only the first parenthesis is required.
volume	Optional.
	Default: None, unless the DEFINE command function is used to initialize the keyword VOL.
	Specifies the volume identification number of the direct access volume to be searched for the library.
	Note that the slash (/) must be included.
	In z/OS, if this argument is not used and no default value is defined, the system catalog is used to locate the file.
num1	Optional.
	Default: 1
	Specifies the relative statement number (relative to one) of the first statement to be included in the read operation.
num2	Optional.
	Default: ALL
	Specifies the number of statements to be read.
addr	Optional.
	Default: ++
	Specifies the address of the statement within the current SD work file after which the statement(s) being read will be placed. Note that if "--" is entered, the statements read are placed in front of the first statement line.

Example

The following example illustrates the use of the READ command. In this example, the Com-plete initialization procedure "COMPLETE" is read from the library SYS1.PROCLIB and only the first three statements of the procedure are to be read:

```
READ SYS1.PROCLIB(COMPLETE),1,3
```

RECOVER Command

The RECOVER command enables you to resume editing of the data in a specific SD work file after either a system failure occurs or after UED terminates with the H option of the END command function.

If Com-plete is initialized with the accounting option in effect, or if you log onto Com-plete before the initial editing session, the SD work file is recovered based upon your user ID.

If Com-plete is initialized without the accounting option in effect, and if you do not log on to Com-plete before the initial editing session, the SD work file is recovered based upon the Terminal Identification (TID) number. In this situation, you can only recover from the original terminal.

Note:
The RECOVER command only works if you use it before you edit any other file or member.

The command format is:

```
RECOVER
```

REPEAT Command

The REPEAT command enables you to repeat the execution of the previously entered command line.

Note:
You cannot issue a REPEAT command if the previous input command line contains either a REPEAT command or a macro statement containing a REPEAT command.

The command format is:

```
REPEAT
```

RESUME Command

The RESUME command enables you to resume output or typing after it has been halted by a HALT command.

The command format is:

```
RESUME
```

SAVE Command

The SAVE command enables you to save or write the data being edited to an output source library member or library. All or some of the statements being edited may be saved.

The command format is:

```
SAVE [library][(member)][/volume][,addr1][,addr2]
```

The arguments are:

library	Optional, except as indicated below.
	Default: The library originally used in the initial READ command function, if any. If this is new data being entered (that is, a READ command function was not entered), an output library must be specified.
	Specifies the name of the output library to be used when the SAVE function is performed.
	If two-character library identification codes are used (as is done with the UEDIT utility program), the data set name need not be entered; simply enter the two-character code. Note, however, that these codes must be defined to UED in the table of library identification codes (UEDTB1) or defined by the UL function of the UUTIL utility.
member	Optional, except as indicated below.
	Default: The member name specified in the READ command function initially executed, if any. If this is a new member being entered (that is, a READ command function was not entered), and an output source library is used, an output member must be specified.
	Specifies the name of the output member to be used when the save operation is performed.
	Note that only the first parenthesis is required.
volume	Optional, except as indicated below.
	Default: None. If the output library is not cataloged, this argument must be specified.
	Specifies the volume identification number of the direct access volume to be searched for the library.
	Note that the slash (/) must be included if this argument is used.
	In z/OS, if this argument is not used, the system catalog is used to locate the file. If no arguments are given, however, the file included in the original read is not cataloged, and the save occurs in the same library.
	Note that if a library code is used when the library is referenced in a read and the library code table contains the volume serial number, the volume argument is not required.
addr1	Optional.
	Default: --
	Specifies the address of the first statement to be included in the save operation.

addr2	Optional.
	Default: ++
	Specifies the address of the last statement to be included in the save operation or, if stated as a number (e.g., #25), specifies how many statements are to be saved.

Note:
If you use the SAVE command function to save only a range of statements rather than the entire SD work file, the range of statements entirely replaces an existing member (that is, the existing member is not updated).

SUBMIT Command

The SUBMIT command enables you to submit a range of statements via the RJE facility of Com-plete to be scheduled for batch execution.

You can submit either the entire SD work file or portions thereof for scheduling.

The command format is:

```
SUBMIT [addr1][,addr2]
```

The arguments are:

addr1	Optional.
	Default: --
	Specifies the address of the first statement in the SD work file to be submitted.
addr2	Optional.
	Default:++
	Specifies the address of the last statement within the SD work file to be included in the submit operation or, if stated as a number (e.g., #25), specifies how many statements to submit.

TAB Command

The TAB command enables you to display an edit scale that indicates the available columns in which you can perform editing. This scale also contains the character T in every tab position.

The command format is:

```
TAB
```

No arguments are required or provided.

The output displayed by this command is illustrated in the following figure:

```
T.|...8 |...|...1T0...T..2|0...|...3|0... (...)
```

The following notes apply to the items in the preceding display:

- Columns 72-80 are listed first. These are the sequence number columns that normally appear in the output of a LIST command.
- A space separates the sequence columns from the data columns.
- The character T is entered in every column for which a tab is set; in this example, columns 72, 10, and 16.
- Column numbers divisible by five are indicated with a vertical bar (|). Column numbers divisible by ten are indicated by having the numerical digits representing those columns separated by a vertical bar (e.g., column 30 is represented as "3|0", where the 3 is actually in column 29 and the 0 is actually in column 31).

Forward Paging Command

The Forward Paging command enables you to advance the current line pointer address forward a designated number of lines.

The command format is:

```
+num
```

where *num* is any integer larger than zero.

This command causes the current line pointer address to move forward the indicated number of lines.

Backward Paging Command

The Backward Paging command enables you to decrement the current line address a designated number of lines.

The command format is:

```
-num
```

where *num* is any integer larger than zero.

This command causes the current line pointer address to move backward the indicated number of lines.

Last Statement Paging Command

The Last Statement Paging command enables you to advance the current line pointer address to the last statement.

The command format is:

```
++[-num]
```

Use of the *-num* argument positions to "num" lines before the last statement (e.g., ++-5 would position to five lines before the end of file).

First Statement Paging Command

The First Statement Paging command enables you to change the current line pointer address to the first statement.

The command format is:

```
--[+num]
```

Use of the +*num* argument positions you to "num" lines after the first statement (e.g., --+5 would position to five lines after the top of file).

Tag Paging Command

The Tag Paging command enables you to change the current line pointer address to that of the statement containing a designated tag. A tag is a character string ranging from 1 to *n* characters.

The search for a tag is normally in columns one through eight of each statement; however, you may override this with the TAGCOL keyword option of the DEFINE command function.

The command format is:

```
/tag[+num]
```

where you must enter the slash (/) before the tag itself.

Use of the +*num* or -*num* argument positions you to "num" lines after or before, respectively, the tag statement (e.g., /tag+5, /tag-5).

Statement Modification Command

The Statement Modification command enables you to advance the current line pointer address forward to a designated statement and optionally modify the contents of that statement.

The command format is:

```
num[data]
```

where *num* is any string representing the desired statement sequence number.

Note that data (that is, any characters specified after the sequence number) will replace the data currently contained in the statement identified by the specified statement number.

UEDIT - Text Editor Utility

This section describes Com-plete's text editor utility, UEDIT.

- The first section briefly describes how to use UEDIT. After reading this section, you will have a basic understanding of editing with UEDIT;
- The second and third parts of the section describe in much greater detail how to use the editor;
- The fourth section contains detailed explanations of UEDIT commands;
- The final part of the section describes UEDIT upload and download functions to the personal computer.

This chapter covers the following topics:

- Getting Started
- Starting an Edit Session
- Full Screen Editing
- Processing Priorities
- Prefix Commands
- A/B Positioning Commands
- Scrolling Commands
- User Profiles
- PF Key Usage
- Edit Session Suspend Levels
- CommandFormat
- Command Overview
- Command Description
- Moving PDS Members to and from a Personal Computer (PC)

Getting Started

This section explains how to start a UEDIT session in a very short time. For more details of the UEDIT functions, see the subsequent parts in this section.

Remember that the explanations in this section are designed to only get you started. Please read this entire section for a complete understanding of UEDIT.

Opening a Member for Editing

To edit a member that already exists, enter the following command:

```
*UEDIT dsn(membername
```

where:

dsn	Specifies the name of the data set where the member is located.
membername	Specifies the name assigned to the member when it was saved earlier.

If you are creating a member, then enter:

```
*UEDIT dsn(membername CR
```

Note:
Whether you are editing an existing member or creating a member, the member name does not have to be followed by a right parenthesis.

Upon entering the editor, you will see a screen that contains a rule line. If you are editing an existing member, the first 23 lines of text will be displayed beneath the rule line.

Entering Commands on the Command Line

The command line is actually part of the rule line. When you type in commands, that part of the rule line till be overtyped. Commands entered on the command line allow you to page forward and backward through the text and perform functions on the member as a whole. Following are the descriptions for some of these commands.

Paging Commands

The following commands allow you to page forward and backward within a text member.

--	Two hyphens entered on the command line move you to the top of the member.
++	Two plus signs entered on the command line move you to the bottom of the member. This is a quick way to reach the end of the member if you want to add additional text there.
+n	A plus sign followed by a number entered on the command line will move you forward in the text n lines. A plus sign entered alone moves you forward in the text one line.
-n	A minus sign followed by a number entered on the command line moves you backward in the text n lines. A minus sign entered alone moves you backward in the text one line.

Functional Commands

The following commands allow you to perform functions on the entire member.

SAVE	The SAVE command stores the member including all the changes you made during a session. You are returned to the Full Screen Editor menu. To exit the UEDIT utility from there, press the CLEAR-key.
END	The END command also takes you out of the member to the Full Screen Editor menu, but any changes you made to the member during the current session are not stored. Pressing the CLEAR-key also takes you out of a member without saving any changes.
SCAN	The SCAN command allows you to search the member for a specific string of characters.
	For example, to scan for the word "UEDIT", enter the SCAN command with its operand on the command line as shown in the following figure.

```
Scan UEDIT....+....2....+....3....+....4....+....5....+....6....+....7..
This section will enable you to start using the editor in a very short time. For
greater detail and explanation of the functions for using the UEDIT text editor,
refer to the sections that follow in this chapter.

Remember that the explanations contained in this section are designed to only
get you started. Please read this entire chapter for a complete understanding of
the text editor.

Opening a Member for Editing
```

	The SCAN function will search the text (beginning at the top line currently displayed) for the string UEDIT and then redisplay the text with the line that contains the string at the top of the screen. In the preceding example, the string searched for occurs in the third line of the screen where the command was entered. After you press ENTER, the screen will appear as shown in the following figure.

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
using the UEDIT text editor, refer to the sections that follow in this chapter.

Remember that the explanations contained in this section are designed to only
get you started. Please read this entire chapter for a complete understanding of
the text editor.

Opening a Member for Editing

To edit a member that already exists, enter this command.
```

	You can use the SCAN command to scan forward, scan backward, or return the number of occurrences of a string. To familiarize yourself with all the options available, read the description of the SCAN command later in this section.
	To find out how you can change a character string for an entire member, refer to the CCHANGE command section later in this section.

Entering Prefix Commands

Prefix commands are entered on the screen on any line *below* the command line. There are two types of prefix commands. Immediate prefix commands are used to affect single lines of text. Block prefix commands are used to affect multiple lines of text.

Prefix commands are preceded by an escape character (the default is "#"). Note that in this chapter, "#" is used as the escape character unless otherwise noted. For information on how to change the escape character, see the section *User Profiles* and the description of the SET command later in this chapter.

Enter prefix commands on the editor screen, beginning with the "#", in the first column of the text line you wish to affect. Space once after typing in the last character of the command to ensure that none of the text in the existing line is interpreted by UEDIT to be part of the command.

Note that entering a prefix command (with the correct escape character in the first column) "on top of" the first characters of a text line will not alter those characters. When you press ENTER, the command is executed, and the characters at the beginning of the text line are returned to their original state.

A/B Positioning Commands

Some prefix commands are used in conjunction with the "A" and "B" positioning commands. For example, with the COPY command, you would key in a "#A" or "#B" on the particular line that you want the marked text to be copied after or before, respectively.

Immediate Prefix Commands

Immediate prefix commands are single commands that apply only to single lines of text. Following are the descriptions of three immediate prefix commands that you will probably use often.

C	Enter a "#C" on the beginning of the line to mark the line for the COPY function. Identify where you want the line to be copied by entering a "#A" or "#B" positioning command on the line that you want the copied line to appear after or before, respectively.
	For example, let’s say you were entering the text at the beginning of this chapter. To save keystrokes, you decided to copy a line instead of keying in practically the same information twice.
	The following figure illustrates how you would mark the line for the COPY function and how you would mark the position of the line to be copied.

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
To edit a member that already exists, enter this command.

#C         *UEDIT dsn(membername

where:

     DSN is the name of the data set where the member is located.

     Membername is the name assigned to the member when it was saved previously.

If you are creating a member, then enter
#A
```

On the preceding screen, the line "*UEDIT DSN(membername" has been marked to be copied on the line below the line marked with a "#A". When you press ENTER, the screen will appear as shown below.

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
To edit a member that already exists, enter this command.

          *UEDIT dsn(membername

where:

     DSN is the name of the data set where the member is located.

     Membername is the name assigned to the member when it was saved previously.


If you are creating a member, then enter

       *UEDIT dsn(membername
```

D	Enter a "#D" at the beginning of a line and press ENTER to delete the line.
I	Enter a "#I" at the beginning of a line and press ENTER to insert a blank line immediately below that line. You can also specify the number of lines to insert by placing the number after the command as follows: "#In".

Block Prefix Commands

Block prefix commands consist of a series of commands that apply to several lines (a block) of text. Mark the block of data that you wish to perform a function on by keying the block prefix commands in the first character positions of the beginning and ending lines of the block.

Following are the descriptions of three block prefix commands that you may use often.

<table>
<tr><td rowspan="2">CC</td><td>Enter a "#CC" in the first and last line of the block to mark it for the COPY function. Identify where you want the block copied by entering a "#A" or "#B" positioning command on the line where you want the block to appear after or before, respectively.</td></tr>
<tr><td>The following figure illustrates how you would mark a block of text to be copied and how you could mark the position of where you want the block to appear after or before.

<pre>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7
..   *UEDIT DSN(membername

#CC e:
     DSN is the name of the data set where the member is located.
     Membername is the name assigned to the member when it was saved previously.
#CC
If you are creating a member, then enter
          *UEDIT DSN(membername
#A
</pre></td></tr>
<tr><td></td><td>The following figure shows the result of the block prefix commands entered on the previous screen.

<pre>
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..

         *UEDIT DSN(membername

where:

     DSN is the name of the data set where the member is located.

     Membername is the name assigned to the member when it was saved previously.

If you are creating a member, then enter

       *UEDIT DSN(membername

where:

     DSN is the name of the data set where the member is located.

     Membername is the name assigned to the member when it was saved previously.
</pre></td></tr>
<tr><td>DD</td><td>Enter a "#DD" on the first and last line of the block to mark it for the DELETE function. Once you’ve marked the block, press ENTER to delete it.</td></tr>
</table>

	The following figure illustrates how you would mark a block of text for deletion.
	
	`....+....1....+....2....+....3....+....4....+....5....+....6....+....7..`
	`*UEDIT DSN(membername`
	
	`where:`
	
	`DSN is the name of the data set where the member is located.`
	
	`Membername is the name assigned to the member when it was saved previously.`
	
	`If you are creating a member, then enter`
	
	`*UEDIT DSN(membername`
	
	`#DD e:`
	
	`DSN is the name of the data set where the member is located.`
	
	`Membername is the name assigned to the member when it was saved`
	`#DD previously`

MM	Enter a "#MM" on the first and last line of the block to mark it for the MOVE function. Identify where you want the block moved by entering a "#A" or "#B" positioning command on the line where you want the block to appear after or before, respectively.

	The following figure illustrates how you would mark a block to be moved and how you would mark where you would like the block moved.
	
	`....+....1....+....2....+....3....+....4....+....5....+....6....+....7.`
	`This section will enable you to start using the editor in a very short time. For greater detail and explanation of the functions for using the UEDIT text editor, refer to the sections that follow in this chapter.`
	
	`#MM mber that the explanations contained in this section are designed to only get you started. Please read this entire chapter for a complete understanding of the text editor.`
	`#MM`
	`Opening a Member for Editing`
	
	`To edit a member that already exists, enter this command.`
	
	`*UEDIT DSN(membername`
	
	`where:`
	
	`DSN is the name of the data set where the member is located.`
	
	`Membername is the name assigned to the member when it was saved previously.`
	
	`#A`

	The following figure shows the result of the block prefix commands entered on the previous screen.
	
	`....+....1....+....2....+....3....+....4....+....5....+....6....+....7..`
	`This section will enable you to start using the editor in a very short time. For greater detail and explanation of the functions for using the UEDIT text editor, refer to the sections that follow in this chapter.`
	
	`Opening a Member for Editing`
	
	`To edit a member that already exists, enter this command.`
	
	`*UEDIT DSN(membername`
	
	`where:`
	
	`DSN is the name of the data set where the member is located.`
	
	`Membername is the name assigned to the member when it was saved previously.`
	
	`Remember that the explanations contained in this section are designed to only get you started. Please read this entire chapter for a complete understanding of the text editor.`

RR	Enter a "#RR" on the first and last line of the block to mark it for the REPEAT function. Identify where you want the block repeated by entering a "#A" or "#B" positioning command on the line where you want the block to appear after or before, respectively.

Starting an Edit Session

You can begin a UEDIT session by either issuing a direct call or invoking the Full Screen Editor menu. Once you are familiar with the UEDIT parameters, you can find it easier to start a session by issuing a direct call. Until then, you may use the Full Screen Editor menu to enter the necessary information. The Full Screen Editor menu also allows you to specify sequence numbers and the PANVALET/ LIBRARIAN options to be used. Each of these start up methods is described below.

Full Screen Editor Menu

To invoke the Full Screen Editor menu, issue the following command:

`*UEDIT`

z/OS UUDIT screen:

```
17:12:01     TID    12           COM-4.6.          User ID ADMIN      08/31/94
                        -- FULL SCREEN EDITOR --                           UEDT
Edit Object
 Library:        Member:            Newname:            Profile: ADMIN
     DSN:                                                Volume:

Function        PFK or Mark                 Edit Option Settings
 Edit             1  =>                      Work File Size:  1500
 Create           2  =>                       Record Length:  80
 Recover          3  =>
 File Maintenance 4  =>                            ASM Tabs:
 Delete           -  =>                           PL/I Tabs:
 Rename           5  =>                          Cobol Tabs:
 Submit           6  =>                        Fortran Tabs:
 SD-File Summary  7  =>                        Special Tabs:
 Help            10  =>                             Setting:  0  00  0  0  0
 Edit Profile    11  =>
 Create Profile  12  =>                    Sequence Numbers:
                                                Start Value:   100
Panvalet/Librarian                                Increment:   100
 Includes expanded:                             In Columns: 73 -80
             Level: 000000
     Date and Time: 999999 000000            Blank Col 73-80:
```

VSE UEDIT screen:

```
13:53:38     TID     7          COM46-F2          User ID ADMIN46      09/09/94
                        -- FULL SCREEN EDITOR --                           UEDD
 Edit Object
   ULIBID:       Member:               Type:              VOLUME:
  LIB/DSN:                                                 Sublib:
                 Newname:              Newtyp:                Profile:  ADMIN46
 Function        PFK or Mark                 Edit Option Settings
  Edit             1  =>                      Work File Size:  1200
  Create           2  =>                       Record Length:   0
  Recover          3  =>
  File Maintenance 4  =>                            ASM Tabs:
  Delete           -  =>                           PL/I Tabs:
  Rename           5  =>                          Cobol Tabs:
  Submit           6  =>                        Fortran Tabs:
  SD-File Summary  7  =>                        Special Tabs:
  Help            10  =>                             Setting:  0  0  0  0  0  0
  Edit Profile    11  =>
  Create Profile  12  =>                    Sequence Numbers:
                                                Start Value:   100
 Panvalet/Librarian                                Increment:   100
  Includes expanded:                             In Columns: 73 - 80
              Level:      0
      Date and Time: 999999      0            Blank Col 73-80:
```

The Full Screen Editor menu is divided into four areas:

- An Edit Object section;
- A Function/PFK or Mark section;

- An Edit Option Settings section;
- A Panvalet/Librarian section.

The following text describes each of these sections.

Note:
The fields contained on the Full Screen Editor menu may vary from one environment to another. For example, some VSE environments have more identification fields in the Edit Option section. All of the fields that you may see, depending on the environment, are described here in their appropriate sections. Fields that are unique to certain environments are noted.

Edit Object

Use the fields listed under the heading "Edit Object" on the Full Screen Editor menu to define what you wish to edit.

The various items listed are described in the following table.

Field	Description
LIBRARY=id	Optional, unless DSN is not specified.
	Default: The data set specified for DSN (see below).
	Specifies the library in which the member to be edited is located, or, if a new member is being created, the library into which the new member is to be placed.
	Note that id is identified by a two-character code which must be currently defined in either the user’s UL table or the Com-plete User File Table (UEDTB1).
MEMBER=name	Required.
	Default: None
	Specifies the name of the member to be edited or created.
	Note that name can be from one to eight characters, and must comply with the rules for member name construction as required by the operating system in use.
NEWNAME=newname	Optional, except when used with the RENAME function. (See the following section.)
	Default: None
	Specifies the new name to be assigned to the member. **Notes:** 1. NEWNAME is used only in conjunction with the RENAME function. 2. Under VSE, both NEWNAME and NEWTYP must be given for the RENAME function.

Field	Description
DSN=dsn	Optional, unless LIBRARY is not specified.
	Default: The library specified for LIBRARY (see above).
	When used in conjunction with the EDIT function, specifies the name of the file in which the member to be edited is located.
	When used in conjunction with the CREATE function, specifies the name of the data set into which the newly created member is to be placed.
VOLUME=volser	Optional, except required when the file name is not cataloged.
	Default: Specifies the volume serial number of the volume on which the file name to be used is contained.
PROFILE=profile	Optional.
	Default: The Com-plete LOGON user ID of the user is used as the Profile identifier.
	Specifies the User Profile to be used (executed) before the edit session is started.
	Note that setting this field to blanks causes the Profile feature to be deactivated for the session.
TYPE=member-type	Required (VSE only).
	Default: None
	Specifies the type of the member to be edited.
NEWTYP=member-type Optional (VSE only).	
	Specifies the new type of the member to be renamed.
SUBLIB=sublibrary	Sublibrary name, up to eight characters long (VSE only).
	In VSE, this field can be up to eight characters long.
NEWTYP=	Optional, except when used with the RENAME function (VSE only). new-member-type
	Default: None
	Default: Specifies the new type to be to be assigned to the member.

Function/PFK or Mark

The items listed under the heading "Function PFK or Mark" on the Full Screen Editor menu are for specifying the function you wish to perform.

You can select the function you want by either pressing the appropriate PF Key (see the section *PF Key Usage* later in this chapter) or entering any character in the input field to the right of the function on the screen and then pressing ENTER.

Note that you may only select one function at a time.

The functions available are described in the following table.

Function	Description
Edit (PF1)	Begins an edit session in which an existing member is to be edited.
Create (PF2)	Begins an edit session in which a new member is to be created.
Recover (PF3)	Recovers a member that was being edited when a system failure occurred.
File Maintenance (PF4)	Enters the file maintenance utility (UPDS in z/OS; USERV in VSE).
Delete (no PF Key assigned)	Deletes a member from a library.
Rename (PF5)	Renames an existing member.
Submit (PF6)	Submits a member for the purpose of executing a job in batch mode.
SD-File Summary (PF7)	Displays a summary of current edit work files for this user.
Help (PF10)	Invokes the UEDIT HELP function, which provides online documentation assistance.
Edit Profile (PF11)	Edits an existing User Profile.
Create Profile (PF12)	Creates a new User Profile.

Note:
Pressing ENTER with the Full Screen Editor menu displayed is interpreted as the EDIT (PF1) function.

Edit Option Settings

The fields under the heading "Edit Option Settings" on the Full Screen Editor menu are for specifying the edit settings that are to be used.

This section of the Full Screen Editor menu is further divided into four sections. These sections allow you to:

- Specify edit file sizes.
- Select screen tab settings.
- Specify starting sequence numbers and their increments.
- Specify whether or not columns 73 through 80 will contain blanks.

To change the Work File Size and the Edit Record Length to be used during the current session, overwrite the defaults specified and press ENTER.

Note:
The "Library", "Member", and "DSN" fields must be blank while you overwrite the "Work File Size" and/or "Record Length" fields.

To select one of the listed screen tab settings to be used during the current session, enter any character in the input field next to the desired tab setting.

To specify whether or not you want sequence numbers, enter any character in the input field next to "Sequence Numbers". In the remaining three fields, you can also specify the starting point and the increment of the sequence numbers, and in what columns you want the sequence numbers to appear.

You can specify that you want blanks to appear in columns 73 through 80 by entering any character in the input field next to "Blank Col 73-80".

Each of the Edit Option Settings fields is summarized in the following table.

Field	Description
Work File Size	Specifies the maximum number of text lines to be permitted during the edit session.
	Default: The existing member size if known; otherwise, the default work file size defined within Com-plete.
Edit Record Length	Specifies the record length of the edit work file to be used. **Note:** If the length specified is not equal to that of the file to be edited, the editor will truncate or pad the records on the right with blanks, as appropriate.
ASM Tabs	Indicates that tab settings that correspond to Assembler language coding requirements are to be used.
PL/I Tabs	Indicates that tab settings that correspond to the PL/I language coding requirements are to be used.
Cobol Tabs	Indicates that tab settings that correspond to the COBOL language coding requirements are to be used.
Fortran Tabs	Indicates that tab settings that correspond to the FORTRAN language coding requirements are to be used.
Special Tabs	Indicates that user-specified tab settings are to be used.
Setting	Displays the column numbers of the current tab settings.
Sequence Numbers	Indicates that the editor is to provide sequence numbers for each line.
Start Value	Specifies the starting sequence number.
Increment	Specifies the increment in the sequence numbers.
In Columns	Specifies the columns in which the sequence numbers are to be placed.
Blank CC 73-80	Indicates that columns 73 through 80 of each statement line are to be set to blanks immediately before the member is saved.

PANVALET/LIBRARIAN Options

Fields under the heading "Panvalet/Librarian" on the Full Screen Editor menu are for specifying the PANVALET/LIBRARIAN options to be used.

Note:
These options are valid only if the PANVALET or LIBRARIAN system is being used.

The items listed are described in the following table.

Field	Description
Expanded Includes	Indicates that all LIBRARIAN "-INC" or PANAVELET "++INCLUDE" statements encountered as part of the file being edited will appear in the work file.
Level	Defines the relative update level to be used.
	Note that the value specified for level can be a positive or negative integer. A positive value refers to a specific version. A negative value refers to a relative level, where "-1" refers to the current modification level, "-2" refers to the previous modification level, etc.
Date	Specifies the date of the update.
	Note that date must be specified in the format YYMMDD.
Time	Specifies the time of the update.
	Note that time must be specified in the form HHMMSS.

Edit Direct Call

To begin an edit session, issue a direct call using the following syntax:

```
*UEDIT object function keyword=value keyword=value ...
```

The parameters are:

object

Required.

Default

If library, file name, or volume is not specified, UEDIT will use the values currently defined in the User Profile or the system catalog (for volume only). See the section *User Profiles* later in this chapter for additional information.

If the object to be processed is still not completely identified, UEDIT will display the Full Screen Editor menu and issue a message requesting the necessary information.

Identifies the member to be processed.

Any one of the following formats can be used for object:

member	Specifies the name of the member to be processed. Note that member can be from one to eight characters long and must comply with the rules for member name construction as defined for the operating system in use. Note that member can be from one to eight characters long and must comply with the rules for member name construction as defined for the operating system in use. In VSE, the member specification has been extended to contain an eight-character sublibrary and an eight-character member-type. The format is: **`SUBLIB.MEMBER,TYPE`** Note that if a LIB-ID is used that contains a default sublibrary, the SUBLIB parameter can be omitted.
lib(member)	Specifies the library in which the member is contained followed by the name of the member to be processed. Note that lib is identified by a two-character code which must be currently defined in either the user's UL table or in the Com-plete User File Table (UEDTB1).
dsn(member)	Specifies the file name in which the member is contained followed by the name of the member to be processed.
dsn(member)/volume	Specifies the file name in which the member is contained followed by the name of the member to be processed followed by the volume serial number of the volume on which the data set is located.

function

Optional.

Default

EDIT is assumed if the object is completely identified; otherwise, MENU is assumed.

Specifies the UEDIT function to be performed.

Any one of the following functions can be specified:

EDIT	Specifies that an existing member is to be edited.
CREATE	Specifies that a new member is to be created.
RECOVER	Specifies that a member in the process of being edited when a system failure occurred is to be recovered.
SUBMIT	Specifies that a member is to be submitted to the operating system internal reader for the purpose of executing a job in batch mode.
MENU	Specifies that the Full Screen Editor menu is to be displayed. Note that a function can be specified by using the function name in its entirety or by using the first character(s) of the function name. For example, E, ED, EDI, and EDIT are all valid specifications of the EDIT function. The minimum number of letters required to specify each command is underlined above.

keyword=value

Optional.

Specifies additional information pertaining to the edit session.

Note that one or more keywords can be specified. Permissible keywords and their default values are:

PROFILE=profile-ID	Default: The Profile ID equal to the user's Com-plete LOGON user ID. Specifies the User Profile to be used during the edit session. Note that if no Profile is to be used, "PROFILE=" (with no operand) should be specified.
SDSIZE=nnnn	Default: The current size of the member if known; otherwise, the default SDSIZE defined within Com-plete. Defines the maximum number of text lines permitted for the edit session.
MODE=mode	Default: MODE=UPPER Defines the character transaction mode that is to be in effect for the session. Options available for mode are LOWER and UPPER. MODE=LOWER indicates that both lower and upper-case characters can be input during the session. Note that use of LOWER requires an input terminal that supports both lower- and upper-case characters. MODE=UPPER indicates that all lower-case input is to be translated to upper-case. **Note:** The setting for MODE must be consistent with that which is in effect for Com-plete.
LEVEL=level	Default: (Applicable only if the PANVALET or LIBRARIAN system is to be used.) Defines the relative update level to be used. The value specified for LEVEL can be a positive or negative integer. A positive value refers to a specific version. A negative value refers to a relative level, where -1 refers to the current modification level, -2 refers to the previous modification level, etc.
DATE=date	(Applicable only if the PANVALET or LIBRARIAN system is to be used.) Defines the date of the update. Note that date must be specified in the format YYMMDD.
TIME=time	(Applicable only if the PANVALET or LIBRARIAN system is to be used.) Defines the time of the update. Note that time must be specified in the format HHMMSS.

Note that a keyword can be specified by using the keyword name in its entirety or by using the first character(s) of the keyword name. For example P, PR, PRO, PROF, PROFI, and PROFILE are valid specifications of the PROFILE keyword. The minimum number of letters required to specify each keyword is underlined above.

Examples

Following are examples of UEDIT direct calls and their descriptions.

```
*UEDIT ASMLKD E
```

Edit the member ASMLKD using the User Profile equal to the user ID. The User Profile will be used to obtain library, file name, and volume information.

```
*UEDIT ASMLKD P=ASM
```

Edit the member ASMLKD, using the User Profile ASM. The User Profile will be used to obtain library, file name, and volume information.

```
*UEDIT UA(ASMLKD) R P=,S=600
```

Recover the member ASMLKD in library UA, using no User Profile, restricting the maximum number of text lines used for the session to 600.

```
*UEDIT ASMLKD SUB P=JCL
```

Submit the member ASMLKD for execution as a job in batch mode. Use the User Profile JCL to obtain library, data set, and volume information.

```
*UEDIT JL(ASMLKD) C P=JCL
```

Create the member ASMLKD in the library JL, using the User Profile JCL.

```
*UEDIT DD(DOCUMENT) EDIT,P=TEXT,M=LOWER
```

Edit the member DOCUMENT contained in library DD, using the User Profile TEXT, with lower-case mode in effect.

```
*UEDIT UA(COBOLPGM) P=COBOL
```

Edit the member COBOLPGM contained in library UA, using the User Profile COBOL.

```
*UEDIT PL(PL1PGM) CREATE,P=PL1,M=UPPER,S=5000
```

Create the member PL1PGM in library PL using the User Profile PL1, with upper-case character mode input in effect, restricting the maximum number of text lines used for the edit session to 5000.

```
*UEDIT PROG1
```

Edit the member PROG1 using the User Profile to obtain library, data set, and volume information.

```
*UEDIT SYS1.PARMLIB(IEARSV00)
```

Edit the member IEARSV00 in data set SYS1.PARMLIB, taking the volume serial number from the User Profile, or, if not contained there, from the system catalog in z/OS. Use the User Profile equal to the user’s Com-plete user ID.

```
*UEDIT VSE.SP2.LIB(CP440.ULGEX1,A)/SP2001
```

Under VSE, edit the (type A) member ULGEX1 contained in sublibrary CP440 of library VSE.SP2.LIB, which resides on volume SP2001.

Full Screen Editing

You can use UEDIT for data editing in full screen format as described below.

Screen Areas

Command/Message Line

The command/message line serves two purposes:

- You can use it to issue UEDIT commands;
- UEDIT uses it to return status messages and information to you.

Current Line

The first line after the command line (that is, the first data line on the screen) is called the "current line".

Prefix Area

The first fields of each data line make up the "prefix area"; use it to enter EDIT prefix commands. A prefix command begins in column 1 with the Escape character (the default is #) and is normally terminated with a blank. The prefix commands are discussed in detail later in this section.

PF Keys

Program Function keys (PF Keys) are used to eliminate repetitive typing of often-used commands. You can set each key to invoke a specific UEDIT command. In addition, you can modify PF Keys dynamically during an edit session by using the SET command. The PF Keys are discussed in detail later in this section.

Vertical Scrolling

UEDIT also has a vertical scrolling feature. If you issue the command STAY OFF, the editor scrolls to the line where the cursor is located when you press ENTER. For additional information, see the SET command in the section *UEDIT Commands* later in this chapter.

Note that the editor is capable of editing records with a record length greater than 80 bytes (line width > 80). The commands SET LISTCOL, LEFT, and RIGHT enable you to display and edit "windows" of the record on the screen (see the SET, LEFT, and RIGHT commands in the section *UEDIT Commands*).

Tags

A tag is a string of characters that is used to locate and position to a specific line. The tag must begin in the column defined as column one by the ZONE parameter of the SET command.

Tags are used in conjunction with the COPY, DELETE, and FIND commands. For example, when you want to copy a block of text, you would use tags to identify the block to be copied. Note that when you use tags with the COPY and DELETE commands, the tag must *always* be followed by a blank. With the FIND command, the tag must be followed by a blank only if the SCAN parameter of the SET command is

set to "TAG". For additional information, see the table below as well as the SET command in the section *UEDIT Commands* later in this chapter.

Strings

A string is similar to a tag in that it is also used to locate and position to a specific line. A string is different from a tag in the following two ways:

- A tag can occur *anywhere* between and including the columns specified as the beginning and ending columns by the ZONE parameter of the SET command.
- A tag does not need to be followed by a blank (except when the SCAN parameter of the SET command is set to "TAG", in which case it must be both preceded and followed by a blank).

Strings are used in conjunction with the CHANGE and SCAN commands.

SET ZONE Command

For the CHANGE, COPY, DELETE, FIND, and SCAN commands, a scan operation is performed as part of the function. In other words, when you specify a tag(s) or string(s) as part of the command, UEDIT actually scans for the tag/string specified. And because a scan is being performed, the settings specified using the SET ZONE command are important.

For example, if the zone was set to "15 72", the scan operation performed as part of the CHANGE or SCAN function will only apply to columns 15 through 72. If the string you entered as part of one of these commands began in columns 1 through 14, the string would not be found and the function would not be performed.

For a scan operation using the same zone setting ("15 72") performed as part of the COPY, DELETE, or FIND command, the first character of the tag must be in column 15 in order to be located during the scan.

Note that the defaults for the SET ZONE command are "1 72".

SET SCAN Command

The CHANGE, COPY, DELETE, FIND, and SCAN commands are not only affected by the SET ZONE command; they are also affected by the SET SCAN command. The two modes for SET SCAN are "ABS" and "TAG". The SET SCAN command is discussed in detail in the *SET Command* subsection in the section *UEDIT Commands* later in this chapter, and is summarized in the following table.

Command	**Operand used**	**SET SCAN ABS**	**SET SCAN TAG**
		Tag must be followed by a blank or String must be delimited by a blank if this scan mode is in effect:	
CHANGE	String	no	yes
COPY	Tag	yes	yes
DELETE	Tag	yes	yes
FIND	Tag	no	yes
SCAN	String	no	yes

Processing Priorities

The following processing priorities are in effect during an edit session:

1st priority:	A prefix command for a line
2nd priority:	Data entered for, or edited on, a line
3rd priority:	A PF Key command
4th priority:	A UEDIT command

Note that this processing priority hierarchy permits you to use PF Keys without losing any data entered since you last pressed ENTER. It also permits complex combinations of prefix commands, data entry/edit commands, and PF Keys or UEDIT commands.

Note, however, that PF Key commands take precedence over UEDIT commands. So if you issue a UEDIT command and a PF Key command at the same time, the PF Key command is executed and the UEDIT command is ignored.

Prefix Commands

Prefix commands can be divided into two distinct categories:

- Immediate Prefix commands
- Block Prefix commands

To specify these commands, enter the escape character in column one of any data line and the prefix command starting in column two.

Note:
If the data line contains information in columns three through five that could be interpreted as part of the command you are entering, you must enter a blank immediately following the prefix command.

In the tables in this section describing the types of prefix commands, the notation "< >" indicates that the arguments contained within are optional. Do not enter the "< >" symbols as part of the command input.

Immediate Prefix Commands

An immediate prefix command is a single command that applies only to a single data line. The immediate prefix commands are described in the following table.

Command	Description
C<nh>	Copies the line on which you enter the C command. **Note:** When using the C command, you must also identify the target line (the position where the line is to be copied) by using the A or B positioning command. See the section *A/B Positioning Commands* later in this chapter.
	The arguments of the C command are:
	n — Optional. Default: 1 Specifies the number of times the line is to be copied. h — Optional. Default: None Holds the line of text for later (repeated) insertion in the file.
D	Deletes the line on which you enter the D command. After you press ENTER, the cursor is positioned in column one of the previous line.
I<n>	Inserts blank line(s) immediately after the line on which you enter the I command. After you press ENTER, the cursor is positioned in column one of the first new line.
	Note that the blank lines that you do not fill with text will remain, even when you press consecutive ENTERs. If you want the blank lines to be deleted when you press ENTER, use the W command instead.
	The argument for the I command is:
	n — Optional. Default: 1 Specifies the number of blank lines to be inserted.
J	Joins the line in which you enter the J command with the following line. When you press ENTER, the cursor is positioned in column one of the resulting line.

Command	Description
JC<n>	Joins/overlays the line on which you enter the JC command with the following line, beginning at a particular position. When you press ENTER, the cursor will be positioned in column one of the resulting line.
	The argument for the JC command is:
	n — Optional. Default: The lines are joined beginning at the current cursor position. Specifies the column position at which the lines will be joined.
M	Moves the line on which you enter the M command. **Note:** When using the M command, you must also identify the target line (the position where the line is to be moved) by using the A or B positioning command. See the section *A/B Positioning Commands* later in this chapter.
P<i>	Positions to a new current line (to the line on which you enter this command).
	The argument for the P command is:
	i — Optional. Default: None
	Specifies that the editor is to enter insert mode and the entire screen is to be made available for insertions, that is the line marked will become the current line with the rest of the screen lines empty.
R<n>	Repeats the line on which you enter the R command. When you press ENTER, the cursor is positioned in column one of the (first) new line created.
	The argument for the R command is:
	n — Optional. Default: 1
	Specifies the number of times the line is to be repeated.
S	Splits the line on which you enter the S command into two lines, with the second line to begin with the current cursor position. When you press ENTER, the cursor is positioned in column one of the second (new) line.

Command	Description
SC<n>	Splits the line on which you enter the SC command into two lines, with the second line to begin at a particular position. When you press ENTER, the cursor is positioned in column one of the second (new) line.
	The argument for the SC command is:
	n Optional. Default: Line splitting will begin at the current cursor position. Specifies the column position at which the second line is to begin.
W<n>	Creates a window of blank lines after the line on which you enter the W command. When you press ENTER, the cursor is positioned in column one of the first blank line. **Note:** Each blank line you create using the W command that is still blank when you press ENTER is deleted by UEDIT. If you want the blank lines to remain when you press ENTER, use the I command instead.
	The argument for the W command is:
	n Optional. Default: 1 Specifies the number of blank lines to be inserted.

Examples

Following are examples of immediate prefix commands and explanations of how they will affect the lines of data. Note that in these examples, the escape character is "%".

For examples of the immediate prefix commands that must be used in conjunction with an A or B positioning command (i.e., C and M), see the examples in the section *A/B Positioning Commands*.

%D

Delete the single line where "%D" was entered.

%I

Insert a single line below the line where "%I" was entered.

%I5

Insert five blank lines below the line where "%I5" was entered.

%J

Join the line where the "%J" was entered with the line immediately below it.

%P

Make the line where the "%P" was entered become the new current line, i.e., move it to the top of the screen.

%R

Repeat the line where "%R" was entered a single time immediately below the marked line.

%R3

Repeat the line where "%R3" was entered three times immediately below the marked line.

%S

Split the line into two lines. The second line will begin where the cursor was positioned when ENTER was pressed.

%W5

Create a window of five blank lines immediately below where the "%W5" was entered. Any unused lines will be deleted when ENTER is pressed.

Block Prefix Commands

Block Prefix commands consist of a series of commands and involve more than one line of text (a block of text). Note that you can enter the two or three different parts of a block prefix command (e.g., "#CC", "#CC", and "#A") anywhere within the member you are editing; that is, you do not need to enter them all on a single 23-line block of text shown on your screen. In addition, with those commands that require the use of an A/B positioning command (CC and MM), you can enter the two parts of the block prefix command (e.g., "#CC" and "#CC") either before or after you enter the A or B positioning command.

The block prefix commands are described in the following table.

Command	Description
CC<nh>	Copies the block identified by entering a CC command in both the first and last lines of the block.
	Note: When using the CC command, you must also identify the target line (the position where the block is to be copied) by using the A or B positioning command. See the following section *A/B Positioning Commands*.
	The arguments for the CC command are:
	n Optional. Default: 1 Specifies the number of times the block is to be copied. h Optional. Default: None Specifies that the block of text is to be held for later (repeated) insertion in the file.
DD	Deletes the block of lines identified by entering a DD command in both the first and last lines of the block.
MM	Moves the block of lines identified by entering an MM command in both the first and last lines of the block. **Note:** When using the MM command, you must also identify the target line (the position where the block is to be moved) by using the A or B positioning command. For additional information on the A and B commands, see the following section, *A/B Positioning Commands*.
RR<n>	Repeats the block of lines identified by entering an RR command in both the first and last line of the block.
	The argument for the RR command is:
	n Optional. Default: 1 Specifies the number of times the block is to be repeated.

Command	Description
XX	Excludes a block of text from the screen display by entering an XX command in both the first and last line of the block.
	When you mark a block with XX's and press , a two-line message (telling you how many lines have been excluded) will replace the excluded text. This command is useful when you want to look at two parts of a member that are separated by many lines of text at the same time.
	To cause the excluded lines to be redisplayed, enter "RESET XX" on the command line.

Note:
You can cancel (reset) a partially submitted Prefix block command series by issuing the RESET command.

Examples

For examples of the block prefix commands, see the examples in the following section *A/B Positioning Commands*.

A/B Positioning Commands

The A and B positioning commands are used in conjunction with the C, CC, M, and MM prefix commands. Note that you can enter the A or B positioning command either before or after you enter the prefix command.

The positioning commands are described in the following table.

Command	Description
A	Specifies the target location for a move or copy operation. The line(s) being moved/copied will be placed immediately after the line in which you enter the A command.
B	Specifies the target location for a move or copy operation. The line(s) being moved/copied will be placed immediately before the line in which you enter the B command.

Examples

Following are examples of block prefix commands. The explanations describe where the commands were entered, the message displayed in the command/ message line following entry (plus pressing of ENTER) of each part of the command (shown in the far right column of the following examples), and the result. Note that in these examples, the escape character is "%".

1	`%C`	(entered on line 3, the line to be copied)	`A/B`
	`%A`	(entered on line 6, the line after which line 3 is to be copied)	
	Line 3 is copied immediately below line 6.		
2	`%CC2`	(entered on line 1, the first line to be copied)	`A/B CC`
	`%CC`	(entered on line 8, the last line to be copied)	`A/B`
	`%B`	(entered on line 20, the line before which the block of lines 1 through 8 is to be copied)	
	The block of text (lines 1 through 8) is copied twice, immediately above line 20.		
3	`%D`	(entered on line 4, the first line to be deleted)	`DD`
	`%DD`	(entered on line 62, the last line to be deleted)	
	The block of text (lines 4 through 62) is deleted.		
4	`%M`	(entered on line 7, the line to be moved)	`A/B`
	`%B`	(entered on line 9, before which line 7 is to be moved)	
	Line 7 is moved immediately preceding line 9.		
5	`%A`	(entered on line 20, the line after which the block of lines 2 through 8 is to be moved)	`C/M`
	`%MM`	(entered on line 2, the first line to be moved)	`MM`
	`%MM`	(entered on line 8, the last line to be moved)	
	The block of text (lines 2 through 8) is moved immediately below line 20.		
6	`%RR`	(entered on line 9, the first line to be repeated)	`RR`
	`%RR4`	(entered on 12, the last line to be repeated)	
	The block of text (lines 9 through 12) is repeated four times immediately below line 12.		
7	`%XX`	(entered on line 6, the first line to be excluded)	
	`%XX`	(entered on line 55, the last line to be excluded)	
	Editor source lines 6 through 55 will be excluded from screen display. A message reminding you how many lines have been excluded will be displayed on lines 6 and 7.		

Scrolling Commands

To change the position of the current line (the top text line) in the member you are editing, use vertical positioning commands. These commands enable you to change your "view" of the member vertically.

Note that you must enter vertical positioning commands beginning in column one of the command/message line.

The vertical positioning commands are described in the following table.

Command	Description
+nnn	Positions "nnn" lines forward.
-nnn	Positions "nnn" lines backward.
/tag or /+tag	Positions forward to the line containing the "tag".
/-tag	Positions backward to the line containing the "tag" . **Note:** Tag positioning is subject to the scan mode currently in effect (see the SET command in the section, *UEDIT Commands*, later in this chapter).
--	Positions to the top line of the member.
TOP	Positions to the top line of the member.
++	Positions to the bottom line of the member.
BOTTOM	Positions to the bottom line of the member.
+P	Positions one page forward.
-P	Positions one page backward.
+H	Positions a half-page forward.
-H	Positions a half-page backward.

Note that you can also specify a series of positioning commands (see the following examples).

Examples

The following are examples of UEDIT vertical positioning commands and explanations of how your screen display will be affected.

```
+P +1
```

Position forward one page plus one line.

```
B -P +1
```

Position to the bottom of the member, then backward one page, then forward one line.

User Profiles

The User Profile is a mechanism that enables you to predefine various parameters that are to be in effect during an edit session. UEDIT "executes" the commands specified in your User Profile before it starts the edit session.

Note:
Before you use the User Profile feature, you must first allocate a User Profile library that is to contain User Profiles. A Profile library can be any source statement library defined with "$$" as the short name

either by the UUTIL function UL or in the UEDTB1 table.

To establish your User Profile, invoke the "Create Profile" function on the Full Screen Editor menu. You can then define the specific parameter and PF Key settings for the session by using the SET command. Once you have entered all of the SET commands, use the SAVE command to save the User Profile.

You can edit/modify a User Profile that already exists by using the Edit Profile function.

The following figure contains examples of SET commands that can be used in a User Profile:

```
SET ESCAPE %
SET LIBID SX
SET LOWER
SET SCAN ABS
SET STAY ON
SET TABS 10 20
SET VOLUME SAG004
```

For additional information and examples concerning the definition and subsequent modification of a User Profile, see the SET command in the section *UEDIT Commands* later in this chapter.

PF Key Usage

The user can assign default PF key settings by creating the profile DEFAULT. The editor executes this profile whenever the user starts an edit session.

To change a PF Key definition, enter a new definition into the User Profile (see the SET command in the section *UEDIT Commands*, later in this chapter). Note that you can also issue the SET command during an edit session in order to change a PF Key setting.

To display the current definition of a given PF Key, use the UEDIT QUERY and GLOBALS commands. (See the section *UEDIT Commands* later in this chapter.)

Edit Session Suspend Levels

If you are using UEDIT in conjunction with COM-PASS, you can suspend the current edit session by using either the PA1 (SUSPEND) key or the COM-PASS SUSPEND commands. (See the section *COM-PASS Commands* in the chapter *COM-PASS - Parallel Transaction Utility*.) You can then use the level number identifying the suspended edit session, shown on the screen generated by COM-PASS to identify the suspended edit file you want to be included in a subsequent edit session. (See the INCLUDE command in the section *UEDIT Commands* later in this chapter.)

CommandFormat

This section describes the various UEDIT commands you can use during an edit session. Following the explanation of command syntax, a summary table of UEDIT commands is presented followed by a detailed explanation of each command.

In the following sections, the syntax elements listed below are used to indicate the command syntax of of UEDIT commands:

. (period)	Delimits strings; can be any character other than an "+", "-","=", or " " (blank). **Note:** If HEX edit mode is in effect, the only string delimiter permitted is "X".
< >	Indicates that the entire operand is optional.
n	Is a user-specified numeric integer.
A|B|C	Indicates that one and only one of the operands (here, A,B, or C) separated by a vertical bar can be selected.

You can either specify a UEDIT command in its entirety or specify only enough characters to uniquely identify it. For example, B is sufficient for the BOTTOM command since no other command begins with B. Note that SA, SC, and SE are required for SAVE, SCAN, and SET, respectively, since more than one command begins with S. The minimum characters required for each command are indicated by an underscore in the following table and command format statements.

Repeat a Command

To repeat a UEDIT command that you have just entered (after you have pressed ENTER), simply enter an "*" in the first position of the command/message line.

Redisplay a Command

To redisplay the UEDIT command that you have just entered, enter a "?" in the first position of the command/message line.

Command Overview

The following table summarizes the UEDIT commands.

Command	Function
BOTTOM	Positions to the bottom of the member.
CCHANGE	Sets up a conditional change operation.
CHANGE	Locates and changes a specified string of characters.
COPY	Copies one or more lines of text.
DELETE	Deletes one or more lines of text.
END	Leaves the current edit session.
FIND	Positions to a line containing a specified tag.
GLOBALS	Displays the current session parameters.
HELP	Invokes the online documentation facility.
HEX	Edits in hexadecimal mode.
INCLUDE	Includes text from another member.
LEAVE	Leaves (terminates) the edit session.
LEFT	Shifts the screen image to the left.
MOVE	Moves one or more lines of text.
NOTE	Sets a NOTE point.
POINT	Repositions to the NOTE point.
PRINT	Prints a hard copy of the member.
PROFILE	Executes the User Profile.
QUERY	Displays the status of a specific session parameter.
READ	Reads a new member into the edit work area.
RESET	Resets (cancels) a pending prefix block command.
RIGHT	Shifts the screen image to the right.
RSAVE	Saves the member and returns to the same location in the edit session.
SAVE	Saves the member and displays the Full Screen Editor menu.
SCAN	Positions to a line containing the specified character(s).
SET	Sets or modifies the session parameters.
STATUS	Displays the edit session data.
SUBMIT	Submits the member for execution in batch mode.
TOP	Positions to the top of the member.

Command Description

BOTTOM Command

The BOTTOM command enables you to position to the bottom of the member that you are editing.

The command format is:

```
BOTTOM
```

CCHANGE Command

The CCHANGE command enables you to set up a conditional change operation. This command identifies the string to be scanned (string1) and the string to be used to replace string 1 (string2). You can then use "SCAN =" and "CHANGE =" commands to locate and change a string, without needing to retype the string each time.

In contrast to the global CHANGE command (see the CHANGE command), CCHANGE allows you to view the string before the replacement is executed.

The command format is:

```
CCHANGE     .string1.string2.
```

string1	Required.
	Default: None
	Specifies the string of characters that is to be located when a "SCAN =" command is issued.
	Note that the beginning and ending columns to be used for scan processing will be columns 1 and 72, respectively, unless these defaults have been changed using the SET ZONE command.
string2	Required.
	Default: None
	Specifies the string of characters that is to replace string1 when a "CHANGE =" command is issued.
	Note that any line truncation caused by replacement of the string will take place only if SET TRUNC ON is in effect.

Note that the number of characters in "string1" and "string2" need not be equal.

To avoid repetitive typing of the "SCAN =" and "CHANGE =" commands, you can set a PF Key for each command (see the SET command).

Example

The following is an example of the UEDIT CCHANGE command used in conjunction with the "CHANGE =" and "SCAN =" commands.

CCHANGE.OLD.NEW.	(Specifies that the first line that contains the first occurrence of the word "OLD" is to become the current line.)
CHANGE =	(Causes the word "OLD" that appears on the current line to be changed to "NEW".)
SCAN =	(Causes the line that contains the next occurrence of the word "OLD" to become the current line.)
CHANGE =	(Causes the word "OLD" that appears on the current line to be changed to "NEW".)
(etc.)	

CHANGE Command

The CHANGE command enables you to locate and change a specified string of characters. You may also control the direction of the scan used for the change operation.

The command format is:

```
CHANGE <+/->.string1.string2. <n1> <n2> <n3>
       <+/->= <n1> <n2> <n3>
```

+/-	Optional.
	Default: Forward scanning will occur.
	Specifies the direction in which the scan operation is to go.
	The two options are:
	+ Indicates that the scan operation is to proceed forward. - Indicates that the scan operation is to proceed backward.

string1	Required.
	Default: None
	Specifies the string of characters to be located by the change operation.
	Note that the beginning and ending columns to be used for scan processing will be columns 1 and 72, respectively, unless these defaults have been changed by use of the SET ZONE command. **Note:** If TAG scan mode is in effect, the strings specified is found only if the string is delimited by blanks. If ABS scan mode is effect, the string specified will be found even if the string is not delimited by blanks. To define or change the scan mode, use the SET SCAN command.
string2	Required.
	Default: None
	Specifies the string of characters that is to replace the character string specified by string1 .
	Note that any line truncation caused by replacement of string1 will take place only if SET TRUNC ON is in effect.
=	Optional.
	Default: None
	When specified simply as "CHANGE =" (following the entry of a CHANGE command that included a string operand), specifies that the next occurrence of the same string is to be located and changed.
	Note that "=" may be used in conjunction with "+" or "-", as in "CHANGE +=" or "CHANGE -=", in order to reverse the direction of the scan operation.
n1	Optional, except when either n2 or n3 is specified.
	Default: 1
	Specifies the number of lines to be scanned.
	A value of "*" indicates that the scan operation is to proceed until either the bottom of the member (forward scan) or the top of the member (backward scan) has been reached.
n2	Optional, except when n3 is specified.
	Default: 1 (only the first occurrence on each line)
	Specifies the number of occurrences of string1 that are to be changed to string2 on each line. A value of "*" indicates that every occurrence on each line is to be changed.
	Note that if n2 is specified, then the "n1" operand must also be specified.

n3	Optional.
	Default: 1 (the first occurrence).
	Specifies which occurrence of string1 on a given line the change operation is to begin with.
	Note that if n3 is specified, then the n1 and n2 operands must also be specified. **Notes:** 1. The number of characters in "string1" and "string2" need not be equal. 2. If you do not specify "*" for n1, only the current line is scanned.

Examples

The following are examples of the UEDIT CHANGE command and explanations of how the text will be affected.

```
CHANGE .1980.1987.
```

Change the first occurrence of the value "1980" on the current line to "1987".

```
CHANGE .?.+.**
```

Change all occurrences of the character "?" to "+" in all lines including the current line.

```
CHANGE -.XXX.YYY.*
```

Perform a backward scan of all lines starting from the current line, changing the first occurrence of the string "XXX" on each line to "YYY".

```
CHANGE .AA.BB.* 2
```

Change the first two occurrences of the string "AA" on each line to "BB". If more than two occurrences of "AA" are on one line, only change the first two.

```
CHANGE .AA.BB. 10 * 2
```

For the next 10 lines, change all occurrences of the string "AA" to "BB", except for the first occurrence on each line.

```
CHANGE -=
```

Change the direction of the CHANGE function to operate in a backward direction. The change will occur for the strings specified in a previous CHANGE command.

COPY Command

The COPY command enables you to copy one or more lines of text, placing the copied text immediately after the current line.

The command format is:

```
COPY  tag1 <-tag2>
```

tag1	Required.
	Default: None
	Is a string of characters used to locate the top line of the block of text that is to be copied.
	Note that "tag1" must be followed by a blank and that the first character of the tag must be in the column defined as column one by the ZONE parameter of the SET command. The zone may be changed by using the SET ZONE command.
tag2	Optional.
	Default: Only one line of text will be copied.
	Is a string of characters used to locate the bottom line of the block of text that is to be copied.
	Note that tag2 must be followed by a blank and that the first character of the tag must be in the column defined as column 1 by the ZONE parameter of the SET command. The zone can be changed by using the SET ZONE command.

Note that you can also use the C and CC prefix commands to copy text. For additional information, see the section *Prefix Commands* earlier in this chapter.

Example

The following is an example of the UEDIT COPY command and an explanation of how the text will be affected.

```
COPY BEGIN-END
```

Copy the text beginning with the line containing the "BEGIN" tag and ending with the line containing the "END" tag; place it immediately after the current line.

Note that the first character of both the "BEGIN" and "END" tags must be in the column defined as column 1 by the ZONE parameter of the SET command.

DELETE Command

The DELETE command enables you to delete one or more lines of text.

The command format is:

```
DELETE  tag1 <-tag2>
```

tag1	Required.
	Default: None
	Is a string of characters used to locate the top line of the block of text that is to be copied.
	Note that tag1 must be followed by a blank and that the first character of the tag must be in the column defined as column one by the ZONE parameter of the SET command. The zone can be changed by using the SET ZONE command.
tag2	Optional.
	Default: Only one line of text will be deleted.
	Is used to locate the bottom line of the block of text that is to be copied.
	Note that tag2 must be followed by a blank and that the first character of the tag must be in the column defined as column one by the ZONE parameter of the SET command. The zone can be changed by using the SET ZONE command.

Note that you can also use the prefix commands D and DD to delete text. For additional information, see the section *Prefix Commands* earlier in this chapter.

Example

The following is an example of the UEDIT DELETE command and an explanation of how the text will be affected.

```
DELETE BEGIN-END
```

Delete the text beginning with the line containing the "BEGIN" tag and ending with the line containing the "END" tag.

Note that the first character of both the "BEGIN" and "END" tags must be in the column defined as column one by the ZONE parameter of the SET command.

END Command

The END command enables you to terminate the current edit session. Note that the edit work file is not saved; however, it is not deleted either.

The command format is:

```
END
```

Note that using the END command is equivalent to pressing the CLEAR key.

FIND Command

The FIND command enables you to position to a particular line of text.
The command format is:

```
FIND <+/-> tag1
     <+/-=
```

+/-	Optional.
	Default: Forward scanning will occur.
	Specifies the direction in which the scan operation is to go.
	The two options are:
	+ Indicates that the find function is to operate forward. - Indicates that the find function is to operate backward.
tag1	Required.
	Default: None
	Specifies the string of characters that is to be located by the find operation.
	Note that in the line to be located, the first character of the tag must be in the column defined as column one by the ZONE parameter of the SET command. The zone can be changed by using the SET ZONE command. **Note:** If TAG scan mode is in effect, the string specified will be found only if the string is delimited by blanks. If ABS scan mode is in effect, the string specified will be found even if the string is not delimited by blanks. To define or change the scan mode, use the SET command.
=	Optional.
	Default: None
	When specified simply as "FIND =" (following the entry of a FIND command that included a string operand), specifies that the next occurrence of the same string is to be located.
	Note that "=" can be used in conjunction with "+" or "-", as in "FIND +=" or "FIND -=", in order to reverse the direction of the find operation.

Examples

The following are examples of the UEDIT FIND command and explanations of how the text will be affected.

```
FIND ABCD
```

Make the first line that contains the tag "ABCD" the new current line.

```
FIND =
```

(entered after the command in example 1)

Make the line containing the next occurrence of "ABCD" the current line.

```
FIND -=
```

(entered after the command in example 1)

Locate the previous occurrence of "ABCD".

Note:
In each of these examples, "ABCD" will only be found when the "A" is in the column defined as column 1 by the ZONE parameter of the SET command.

GLOBALS Command

The GLOBALS command enables you to display all current session parameters on one screen and all current PF Key settings on another.

The command format is:

```
GLOBALS
```

Note:
You can not enter data on the screen generated as a result of the GLOBALS command. To edit current session parameters or PF Key settings, use the SET command.

To flip/flop between the two display screens, press ENTER.

To return to the edit session, use the CLEAR key.

HELP Command

The HELP command enables you to invoke the Help facility. The Help facility pro-vides online documentation to assist you when you’re entering UEDIT commands.

The command format is:

```
HELP      <command>
```

command	Optional.
	Default: The main HELP menu screen will be displayed by UEDIT.
	Specifies a UEDIT command for which a HELP screen is to be displayed.

Examples

The following are examples and results of the UEDIT HELP command.

```
HELP
```

Display the main Help menu screen.

```
HELP CCHANGE
```

Display the Help screen for the CCHANGE command.

HEX Command

The HEX command is only available on a 3270 model 2 terminal (80 x 24 lines). It enables you to specify that you wish to perform editing in hexadecimal mode.

The command format is:

```
HEX
```

Note:
The only string delimiter permitted in this mode is "X".

The hex command will cause each position in a line to be represented by corresponding characters occupying three lines on the screen. For example, the text line "THIS IS AN EXAMPLE 0123456789" would appear in following format:

```
THIS IS AN EXAMPLE 0123456789
ECCE4CE4CD4CECDDDC4FFFFFFFFFF
38920920150571473500123456789
```

where the bottom two lines are the hex representation of the top (text) line.

To return to the normal edit mode, re-enter the HEX command.

INCLUDE Command

The INCLUDE command enables you to include (copy) text from another location into the member you are currently editing, inserting it immediately below the current line.

The command format is:

```
INCLUDE member                  <x> <y>
        lib(member)             <x> <y>
        dsn(member)/vol         <x> <y>
        LEVEL=n                 <x> <y>
```

member	Optional.
	Default: None
	Specifies the name of the member that is to be included. **Note:** For VSE/SP1 and VSE/SP2, note the extensions for this argument as discussed in the *Edit Direct Call* subsection of the *Starting An Edit Session* section of this chapter.
lib	Optional.
	Default: The value for lib is taken from the User Profile.
	Specifies the name of the library containing the member to be included.
dsn	Optional.
	Default: The value for dsn is taken from the User Profile.
	Specifies the name of the file containing the member to be included.
vol	Optional.
	Default: The value for vol is taken from the User Profile.
	Specifies the number of the volume containing the file name to be included.
LEVEL=n	Optional.
	Default: Specifies that a member for which editing was previously suspended is to be included.
	Note that n indicates the level number of the suspended edit session containing the member to included. **Note:** This feature is available only if you are using COM-PASS. See the earlier section *Edit Session Suspend Levels* for additional information.
x	Optional.
	Default: 1 (the first record)
	Specifies the first line to be copied from the included member.
y	Optional.
	Default: All records.
	Specifies the number of records to be included.

Note that if one or more of the operands *lib*, *dsn*, or*vol* are omitted, the values for these parameters are taken from the User Profile.

Note also that you can use the P prefix command immediately prior to an INCLUDE command in order to position to the line after which the data to be included is to be placed.

Notes:

1. If the member to be included has records longer than the record length of the Edit Work File, the INCLUDE command will truncate the included records on the right. Records with a length less than that of the Edit Work File will be padded on the right with blanks.
2. If the sum of the number of text lines in the member being edited plus the number of text lines in the member being included exceeds the Work File Size of the member being edited, the message "Not Enough Space in SD-File" will appear in the command/message line. For information on how to increase the default Work File Size, see the subsection *Edit Option Settings* in the section *Starting an Edit Session* earlier in this chapter.
3. If you enter the NOTE command using the same name that was issued for an earlier NOTE command, the previous definition is overwritten.
4. If you delete a record where a NOTE command was specified, the NOTE entry is also deleted.
5. You can specify as many as eight NOTES per member.

Examples

The following are examples of the UEDIT INCLUDE command and explanations of how the text will be affected.

```
INCLUDE SUBROUT1
```

Include the member SUBROUT1, which is located in the library where you are currently editing.

```
INCLUDE LB(OPEN)
```

Include the member OPEN, which is located in library LB.

```
INCLUDE SAG.RJ.DSN011(CONV4)/SAG123
```

Include the member CONV4 contained in the data set SAG.RJ.DSN011, which is located on volume SAG123.

```
INCLUDE LEVEL=2
```

Include the edit file suspended at Level 2.

```
INCLUDE SUBROUT2 1 100
```

Include the first 100 records of member SUBROUT2.

```
INCLUDE SUBROUT3 101 200
```

Include records 101 through 200 of member SUBROUT3.

LEAVE Command

The LEAVE command enables you to terminate the current edit session without saving it. Note, however, that the edit session work file is not deleted either.

The command format is:

```
LEAVE
```

Note:
To recover the edit file you were processing when you issued the LEAVE command, use the UEDIT RECOVER function.

LEFT Command

The LEFT command enables you to shift the screen image *n* columns to the left.

The command format is:

```
LEFT n
```

MOVE Command

The MOVE command enables you to move one or more lines of text, placing it immediately below the current line.

The command format is:

```
MOVE tag1 <-tag2>
```

tag1	Required.
	Default: None
	Indicates the beginning line of text to be moved or, if a block of text is being moved, indicates the top line of the block.
	Note that tag1 must be followed by a blank and that the first character of the tag must be in the column defined as column one by the ZONE parameter of the SET command. The zone can be changed by using the SET ZONE command.
-tag2	Optional.
	Default: Only one line of text will be moved.
	Indicates the last line of the text block to be moved.
	Note that "tag2" must be followed by a blank and that the first character of the tag must be in the column defined as column one by the ZONE parameter of the SET command. The zone can be changed by the SET ZONE command.

Note that you can also use the M and MM prefix commands to move text. For additional information, see the section *Prefix Commands* earlier in this chapter.

Example

The following is an example of the UEDIT MOVE command and an explanation of how the text will be affected.

```
MOVE BEGIN-END
```

Move the block of text beginning with the line containing the "BEGIN" tag and ending with the line containing the "END" tag and place it immediately after the current line.

Note that the first character of both the "BEGIN" and "END" tags must be in the column defined as column one by the ZONE parameter of the SET command.

NOTE Command

The NOTE command enables you to assign a one- to eight-character alphanumeric name to the current line of a member. When used in conjunction with the POINT command, it allows you to rapidly page from one place in a member to another.

The NOTE command stores the current screen options (e.g., case, hex, and tab settings) as well as the SD file position.

The command format is:

```
NOTE string
```

For positioning purposes, the name specified as *string* will mark the current line.

Notes:

1. If you enter the NOTE command using the same name that was issued for an earlier NOTE command, the previous definition is overwritten.
2. If you delete a record where a NOTE command was specified, the NOTE entry is also deleted.
3. You can specify as many as eight NOTES per member.

Example

For an example of the NOTE command, see the POINT command, which follows.

POINT Command

The POINT command is used in conjunction with the NOTE command. It enables you to "point to" (i.e., find) the previously defined NOTE settings.

The command format is:

```
POINT string
```

Example

The following is an example of how you could use the NOTE and POINT commands.

NOTE remember	(You mark the current line with the name "remember".)
	(You page through the member, but then want to see the line you marked earlier.)
POINT remember	(The line marked "remember" is displayed as the current line.)

PRINT Command

The PRINT command enables you to create a hard copy printout of the member you are currently editing.

The command format is:

```
PRINT   <dest> <format>
```

dest	Optional.
	Default: The terminal ID defined in Com-plete.
	Specifies the Terminal ID or terminal name of the output device on which the member is to be printed.
format	Optional.
	Default: FORMAT
	Specifies the format in which the data is to be printed.
	Valid operand values are:
	FORMAT — The printout is to be formatted by the system. CC — The user is to provide carriage control characters. NOCC — The printout is to be spooled as is.

Note:
The PRINT command neither truncates nor pads the output record. Any length adjustment is performed by the spooling subsystem.

Examples

The following are examples of the UEDIT PRINT command and explanations of how the text will be affected.

```
PRINT 16
```

Print the member on the output device with a TID equal to 16.

```
PRINT SYSTEMS CC
```

Print the member on the output device with the TID name SYSTEMS, using user-provided carriage control characters.

```
PRINT 2 NOCC
```

Print the member on the output device with a TID equal to 2, spooling the output as is.

PROFILE Command

The PROFILE command enables you to execute the specified User Profile stored in the User Profile library.

The command format is:

```
PROFILE  n
         member
         lib(member)
         dsn(member)/vol
```

n	Optional.
	Default: If no operands are specified, the user ID is used to identify the User Profile to be executed.
	Provides a convenient means of user access to several different profiles.
	The integer n can be appended to the Com-plete LOGON user ID for the user session to define the profile member to be executed. For example, if the user ID is equal to "ST" and a value of "7" is specified for "n", the Profile member "ST7" is executed.
	Note that n must be numeric.
member	Optional.
	Default: If no operands are specified, the user ID is used to identify the User Profile to be executed.
	Specifies the name of the Profile member to be executed.
	Note that the value specified cannot exceed eight characters.
lib	Optional; is used only if the User Profile to be executed is located in a source library.
	Default: Specifies the name of the library containing the Profile to be executed.
dsn	Optional; is used only if the User Profile to be executed is located in a source library.
	Default: If no operands are specified, the user ID is used to identify the User Profile to be executed.
	Specifies the name of the file containing the Profile member to be executed.
vol	Optional.
	Default: If no operands are specified, the user ID is used to identify the User Profile to be executed.
	Specifies the number of the volume containing the file name to be used.

Examples

The following are examples of the UEDIT PROFILE command and explanations of how the text will be affected.

PROFILE

Execute the Profile identified by the user’s user ID.

PROFILE 4

Execute the Profile identified by the user’s user ID appended with value "4".

PROFILE PROF1

Execute the Profile PROF1.

PRO SP(ASM)

Execute the Profile ASM in library SP.

QUERY Command

The QUERY command enables you to display the status of current session parameter settings.

The command format is:

QUERY operand

operand	Required.	
	Default: None; a valid operand must be specified.	
	Specifies the session parameter setting for which status is to be displayed. Valid operand values and their descriptions are:	
	BLANKS	Whether columns 73 through 80 are to be set to blanks.
	CASE	Lower- or upper-case mode setting.
	DSNAME	Data set name.
	ESCAPE	Escape character.
	LIBID	Library ID.
	LISTCOL	The beginning and ending columns that are to be displayed.
	MEMBER	Member name.
	MTYPE	Member-type (VSE only).
	NUMBER	Beginning sequence number and sequence number increment.
	NULLS	Null line fill. (ON indicates that unused line positions will be filled with blanks.)
	PFK n	Setting of PF Key "n".
	SCAN	Whether scan mode is ABS or TAG.
	SEQUENCE	Line sequence numbering. (ON indicates that line sequence numbering is in effect.)
	STAY	Page skipping. (ON indicates that a page skip is to be performed when ENTER is pressed without any editing having been performed on the current screen.)
	SUBL	Sublibrary (VSE only).
	TITLE	Title line. (ON indicates that a title line is to be used.)
	TRUNC	Line truncation. (ON indicates that line truncation is permitted during a change operation.)
	VOLUME	Volume containing the member.
	ZONE	Which beginning and ending columns are to be used during CHANGE, COPY, FIND, and SCAN processing.

For additional information on the QUERY command operands, see the SET command.

Note:
The QUERY command only allows you to display the current session parameter settings. To modify the settings, use the SET command.

Examples

The following are examples of the UEDIT QUERY command and explanations of how the text will be affected.

```
QUERY ESCAPE
```

Display the escape character currently in use.

```
QUERY PFK 6
```

Display the current definition of PF Key 6.

READ Command

The READ command enables you to read a new member into the edit session work area.

Warning:
Any data currently in the edit work area will be overlaid by contents of the member read as a result of using this command.

The command format is:

```
READ     member
         lib(member)
         dsn(member)/vol
```

member	Optional.
	Default: If no operands are specified, the member currently being edited is reread (refreshed).
	Specifies the name of the member to be read. **Warning:** **Use of this operand will cause the edit work area to be overlaid.** **Note:** For VSE, note the extensions for this argument as discussed in the *Edit Direct Call* subsection of the *Starting An Edit Session* section, earlier in this chapter.
lib	Optional.
	Default: If no operands are specified, the member currently being edited is reread (refreshed).
	Specifies the name of the library containing the member to be read. **Warning:** **Use of this operand will cause the edit work area to be overlaid.**
dsn	Optional.
	Default: If no operands are specified, the member currently being edited is reread (refreshed).
	Specifies the name of the file containing the member to be read. **Warning:** **Use of this operand will cause the edit work area to be overlaid.**
vol	Optional.
	Default: If no operands are specified, the member currently being edited is reread (refreshed).
	Specifies the name of the volume containing the file to be read. **Warning:** **Use of this operand will cause the edit work area to be overlaid.**

Examples

The following are examples of the UEDIT READ command and explanations of how the text will be affected.

```
READ STAT1
```

Read the member STAT1.

```
READ LB(MATH6)
```

Read the member MATH6 in library LB.

```
READ DSN001(MATH8)/VOL123
```

Read the member MATH8 contained in data set DSN001 on volume VOL123.

```
READ VSE.SP2.LIB(SUB1.DATA,SOURCE)/VOL333
```

Under VSE, read the member DATA of type SOURCE contained in the sublib SUB1 of file VSE.SP2.LIB on volume VOL333.

RESET Command

The RESET command enables you to reset (cancel) a pending edit prefix block command.

The command format is:

```
RESET   <XX>
```

XX	Optional.
	Default: An incomplete block command (that is, a "pending" condition) will be canceled.
	Resets an outstanding exclude block.

RIGHT Command

The RIGHT command enables you to shift the screen image *n* columns to the right.

The command format is:

```
RIGHT   n
```

n	Required.
	Default: None
	Specifies the number of columns that the screen image is to be shifted to the right.

RSAVE Command

The RSAVE command enables you to save the member you are currently editing without leaving the member, that is, you will remain in the same position in the edit session.

The command format is:

```
RSAVE     member
                                  lib(member)
          dsn(member)/vol
```

member	Optional.
	Default: If no operands are specified, the member is saved using the member name, library, data set, and volume, as taken from the current settings.
	Specifies the name of the member to be saved. **Note:** For VSE, note the extensions for this argument as discussed in the *Edit Direct Call* subsection of the *Starting An Edit Session* section, earlier in this chapter.
lib	Optional.
	Default: If no operands are specified, the member is saved using the member name, library, data set, and volume, as taken from the current settings.
	Specifies the name of the library where the member is to be saved.
dsn	Optional.
	Default: If no operands are specified, the member is saved using the member name, library, data set, and volume, as taken from the current settings.
	Specifies the name of the file where the member is to be saved.
vol	Optional.
	Default: If no operands are specified, the member is saved using the member name, library, data set, and volume, as taken from the current settings.
	Specifies the name of the volume containing the file where the member is to be saved.

Note:
If you have changed any of the file description parameters (member, library, DSN, volume) during the current session, the Editor prompts you for confirmation of an RSAVE as shown in the following figure.

```
COMEBP0006 - RSAVE Parameters Are:
       Member :
       Library :
       DSN :
       Volume :
       Hit Enter to Continue, Clear to Return
       Sublib:       (VSE only)
       Memb-type:    (VSE only)
```

Note:
If truncation is allowed and the DSN has a a record length greater than that of the Edit Work File, use of the RSAVE command truncates records on the right. Records with length less than that of the target DSN are padded on the right with blanks.

Examples

The following are examples of the UEDIT RSAVE command and explanations of how the text will be affected.

```
RSAVE
```

Save the member using the member name, library, file, and volume data from the current settings, and continue with the edit session (at the same position in the member) after the member has been saved.

```
RSAVE REPORT5
```

Same as example 1, except save the member under the member name REPORT5.

```
RSAVE LB(REPORT5)
```

Same as example 1, except save the member under the member name REPORT5 in library LB.

SAVE Command

The SAVE command enables you to save the member you are currently editing and return to the Full Screen Editor menu.

The command format is:

```
SAVE   member
       lib(member)
       dsn(member)/vol
```

member	Optional.
	Default: If no operands are specified, the member is saved using the member name, library, file, and volume, as taken from the current settings.
	Specifies the name of the member to be saved. **Note:** For VSE, note the extensions for this argument as discussed in the *Edit Direct Call* subsection of the *Starting An Edit Session* section, earlier in this chapter.
lib	Optional.
	Default: if no operands are specified, the member is saved using the member name, library, file, and volume, as taken from the current settings.
	Specifies the name of the library where the member is to be saved.
dsn	Optional.
	Default: if no operands are specified, the member is saved using the member name, library, file, and volume, as taken from the current settings.
	Specifies the name of the file where the member is to be saved.
vol	Optional.
	Default: if no operands are specified, the member is saved using the member name, library, file, and volume, as taken from the current settings.
	Specifies the name of the volume containing the file where the member is to be saved.

Note:
If you have changed any of the file description parameters (e.g., member, library, DSN, volume) during the current session, the Editor will prompt you for confirmation of a SAVE as shown in the following figure.

```
COMEBP0006 - SAVE Parameters Are:
       Member :
       Library :
       DSN :
       Volume :
       HIT Enter to Continue, Clear to Return
       Sublib:       (VSE only)
       Memb-typ:     (VSE only)
```

Note:
If truncation is allowed and the DSN has a record length greater than that of the Edit Work File, use of the SAVE command truncates records on the right. Records with length less than that of the target DSN are padded on the right with blanks.

Examples

The following are examples of the UEDIT SAVE command and explanations of how the text will be affected.

```
SAVE
```

Save the member using member name, library, file, and volume data from the current settings, displaying the Full Screen Editor menu after the member has been saved.

```
SAVE REPORT5
```

Same as example 1, except save the member under the member name REPORT5.

```
SAVE LB(REPORT5)
```

Same as example 1, except save the member under the member name REPORT5 in library LB.

```
SAVE VSE.SP2.LIB(SUB1.DATA,SOURCE)/VOL333
```

Under VSE, save the member DATA of type SOURCE contained in the sublib SUB1 of file VSE.SP2.LIB on volume VOL333.

SCAN Command

The SCAN command enables you to locate a specific string of characters.

The command format is:

```
SCAN   <+/->.string
       <+/->.string.*
       <+/->=
```

+/-	Optional.
	Default: Forward scanning will occur.
	Specifies the direction in which the scan operation is to go.
	The two options are:
	+ Indicates that the scan function is to execute forward. - Indicates that the scan function is to execute backward.
string	Required.
	Default: None
	Specifies the string of characters to be located by the scan operation.
	Note that the beginning and ending columns to be used for scan processing will be columns 1 and 72, respectively, unless these defaults have been changed by using the SET ZONE command. **Note:** If TAG scan mode is in effect, the string specified will be found only if the string is delimited by blanks. If ABS scan mode is effect, the string specified will be found even if the string is not delimited by blanks. To define or change the scan mode, use the SET command.
=	Optional.
	Default: None
	When specified simply as "SCAN =" (following the entry of a SCAN command that included a string operand), specifies that the next occurrence of the same string is to be located.
	Note that "=" can be used in conjunction with "+" or "-", as in "SCAN +=" or "SCAN -=", in order to reverse the direction of the scan operation.
*	Optional.
	Default: None
	Indicates that UEDIT is to return the number of occurrences of the specified string. **Note:** When * is specified, no repositioning of the cursor is performed.

Note:
The data values entered for the SCAN command will not be destroyed by use of the READ, INCLUDE, or PROFILE commands.

Examples

The following are examples of the UEDIT SCAN command and explanations of how the text will be affected.

SCAN ABCD

Make the first line that contains the string "ABCD" the current line.

SCAN =

(when used after the command in example 1)

Locate the next occurrence of "ABCD".

SCAN -=

(when used after the command in example 2)

Locate the previous occurrence of "ABCD".

SCAN 'ABCD'*

Return the number of occurrences of the string "ABCD".

SET Command

The SET command enables you to set the edit session parameters. This command is primarily used for defining default session parameters stored in your User Profile; however, you can also use it to change session parameters during an edit session.

The command format is:

```
SET BLANKS              ON OFF
    CURSOR              STAY HOME
    DSNAME              <name>
    ESCAPE              ON OFF char
    FILE                lib(mem)/vol
    LIBID               <libid>
    LISTCOL             col1 col2
    LOWER
    MEMBER              name
    MTYPE               name                      (VSE only)
    NUMBER              n m
    NULLS               ON OFF
    PFK                 n command
    SCALE               ON OFF
    SCAN                ABS TAG
    SEQUENCE            ON OFF col1 col2
    STAY                ON OFF
    SUBL                sublibrary                 (VSE only)
    TABS                ASM FOR COB NONE PL1
                        t1 t2 t3 t4 t5 t6
    TITLE               ON OFF
    TRUNC               ON OFF n
    UPPER
    VOLUME <volume>
    ZONE                col1 col2
```

where the parameters are:

BLANKS	Optional.
	Default: OFF
	Determines whether columns 73 through 80 are to be filled with blanks immediately before a member is saved.
	The operands are:
	ON — Columns 73 through 80 are to be filled with blanks. OFF — Columns 73 through 80 are not to be filled with blanks.
CURSOR	Optional.
	Default: STAY
	Determines the position of the cursor immediately after has been pressed.
	The operands are:
	STAY — The cursor remains on the same line. HOME — The cursor is positioned to the top of the screen.
DSNAME	Optional.
	Default: The current data set name is set to blanks.
	Indicates that the current file name is to be changed to the name specified.
ESCAPE	Optional.
	Default: #
	Specifies the character to be used as the escape character. (The escape character is entered in column 1 of the editor screen and followed by a prefix command. For example, you could enter "#D" to delete a line, "#R" to repeat a line, or use "#CC"s to mark a block of text that you want to copy.)
	The operands are:
	ON — The default escape character (#) is to be used. OFF — No escape character is to be used. char — A user-specified escape character is to be used.
FILE	Optional.
	Indicates that the current library, member, and volume names are to be changed to the library, member, and volume specified.

LIBID	Optional.
	Default: The current library name is set to blanks.
	Indicates that the current library ID is to be changed to the library name specified.
LISTCOL	Optional.
	Default: 1-72
	Specifies the columns to be displayed on the screen.
LOWER	Optional.
	Indicates that both lower- and upper-case mode input are to be accepted. **Note:** If LOWER is specified, the terminal used to input data must support both upper and lower-case.
MEMBER	Optional.
	Indicates that the current member name is to be changed to the name specified.
MTYPE	Optional (VSE only).
	Indicates that the current member-type is to be changed to the name specified.
NUMBER	Optional.
	Default: 0100 0100 (i.e., start = 0100; increment = 0100)
	Specifies the beginning sequence number (n) and the sequence increment (m) to be used to assign sequence numbers.
NULLS	Optional.
	Default: ON
	Determines whether or not unused portions of a line are to be filled with the null character (X’00’).
	The operands are:
	ON Unused positions are to be filled with a null character. OFF Unused positions are not to be filled with a null character.

PFK	Optional.
	Assigns a UEDIT command to a PF Key.
	The operands are:
	n — The PF Key number (1 through 24) command — The UEDIT command to be assigned to the PF Key **Note:** If PFK n is specified with no command, the setting for the PF Key specified will be set to blanks.
SCALE	Optional
	Default: ON
	Determines whether a scale line (ON) or a blank line (OFF) is to be used as the separator line when the edit line size is set to be equal to the screen size.

SCAN	Optional.
	Default: TAG
	Determines whether a scan or find operation is to be performed in ABS or TAG mode. Note that a scan operation is performed by both the SCAN and CHANGE commands; a find operation is performed by the FIND command. **Note:** In any scan operation, the string to be located must be between and/or include the beginning and ending columns defined using the SET ZONE command.
	The operands for the scan operation (SCAN or CHANGE command) are defined as:
	ABS — The string to be located will be found even if it is not delimited by blanks. TAG — The string to be located will be found only if it is delimited by blanks. **Note:** In the find operation, in the line to be located, the first character of the tag must be in the column defined as column one using the SET ZONE command (the default is 1).
	The operands for the find operation are defined as:
	ABS — The tag to be located will be found even if it is not followed by a blank. TAG — The tag to be located will be found only if it is followed by a blank.
	For additional information on the SET SCAN command, see the section *Full Screen Editing*, earlier in this chapter.

SEQUENCE	Optional.
	Default: 73-80
	Determines the columns to be used for sequence numbering. **Note:** Sequence numbers are inserted/updated only when a member is saved.
	Valid options are:
	ON Columns 73 through 80 are to be used. OFF No sequencing is to be performed.
	The operands are:
	col1 User-specified beginning sequence column. col2 User-specified ending sequence column.
STAY	Optional.
	Default: ON
	Determines whether the current line is to be positioned to the next page when ENTER is pressed without any editing having been performed on the current screen.
	If STAY OFF has been set, the editor will scroll to the line on which the cursor has been placed.
	Valid options are:
	ON No page skip is to be performed. OFF A page skip is to be performed.
SUBL	Optional (VSE only).
	Indicates that the current sublibrary is to be changed to the name specified.

TABS	Optional.
	Default: NONE
	Defines the tab settings to be used during the edit session.
	Valid options are:
	ASM — Assembler language tab settings. FOR — Fortran language tab settings. COB — COBOL language tab settings. NONE — Columns 1 through 72 with no tabs. PL1 — PL/I language tab settings. t1 through t6 — User-specified tab settings.
TITLE	Optional.
	Default: ON
	Determines whether the top line in the editor displays UEDIT information or a scale line.
	Valid options are:
	ON — Display a title line, e.g., "Lib=UU Mem=UUP1A Edit Abs Z=1-72 C=1-72", where: Lib= — Specifies the two-character library ID defined either by ULIBID or in UEDTB1. Mem= — Specifies the membername. Edit — Specifies the scan mode in effect. The two options are Tag and Abs. (See the SCAN parameter of the SET command, earlier in this section.) Z= — Specifies the work zone. (See the ZONE parameter of the SET command, later in this section.) C= — Specifies the columns to be displayed on the screen. (See the LISTCOL parameter of the SET command, earlier in this section.) OFF — Display a scale line, i.e., "....+....1....+....2...." etc.

TRUNC	Optional.	
	Default: ON	
	Determines whether a line may be truncated as a result of a change operation.	
	Valid options are:	
	ON	Line truncation is permitted.
	OFF	No line truncation is permitted.
	n	The Editor will split/join an edit line from column 1 up to and including column n-1 while leaving columns n through end-of-line untouched.
UPPER	Optional.	
	Indicates that upper-case input character mode is in effect.	
VOLUME	Optional.	
	Default: The current volume setting will be set to blanks.	
	Indicates that the current volume number is to be changed to the specified volume number.	
ZONE	Optional.	
	Defaults:	
	col1	1
	col2	72
	Defines a work zone for the CHANGE, COPY, DELETE, FIND, and SCAN commands.	
	The operands are:	
	col1	Beginning column for CHANGE, COPY, DELETE, FIND, and SCAN command processing.
	col2	Ending column for CHANGE and SCAN command processing.

For additional information on the SET ZONE command, see the section *Full Screen Editing*, earlier in this chapter.

Examples

The following are examples of the UEDIT SET command and explanations of how the text will be affected.

SET CURSOR STAY

When ENTER is pressed, the cursor is to remain positioned at the last line being edited/changed.

SET ESCAPE %

Use the character "%" as the escape character.

SET FILE GR(EDIT3)/SAG006

Use the library "GR", the member name "EDIT3", and the volume number "SAG006" as the current library ID, member name, and volume, respectively.

SET LIBID GX

Use the library ID "GX" as the current library ID.

SET LISTCOLS 10 70

Display columns 10 through 70 on the screen.

SET LOWER

Accept lower- and upper-case mode character input.

SET MEMBER EDITFILE

Use the member "EDITFILE" as the current member name.

SET NULLS ON

Fill all unused positions of a line with the null character.

SET NUMBER 100 50

Begin sequence numbering with "100" and use an increment of "50" when sequence numbers are assigned.

SET PF 6 -3

Set PF Key 6 to cause the screen display to be positioned backward three lines.

SET SCAN ABS

Set absolute scan mode in effect.

SET SEQUENCE 78 80

Use columns 78 through 80 for sequence numbers.

SET STAY OFF

Perform a page skip when ENTER is pressed without any editing having been performed on the current screen.

SET TRUNC OFF

Do not permit line truncation during a change operation.

SET VOLUME SAG004

Use the volume "SAG004" as the current volume serial number.

SET ZONE 10 71

Use columns 10 through 71 only during CHANGE and SCAN command processing.

STATUS Command

The STATUS command enables you to display statistics about the current Edit Work File.

The command format is:

STATUS

In the command/message line, statistics such as those shown in the following figure will be displayed.

```
Size Was: 341 Now: 341 Left: 1258 Current: 47
```

SUBMIT Command

The SUBMIT command allows you to submit a member to an operating system internal reader for execution as a batch job.

The command format is:

SUBMIT S|N|R

If no operand is specified, S is the default.

S	Optional.
	Specifies that the member is to be submitted for execution as a batch job as well as saved.
N	Optional.
	Specifies that the member is to be submitted for execution as a batch job only, but not saved.
R	Optional.
	Specifies that the member is to be submitted for execution as a batch job only (not saved) and you are then to be returned to the edit session.

TOP Command

The TOP command enables you to position to the top of the member that you are editing.

The command format is:

```
TOP
```

Moving PDS Members to and from a Personal Computer (PC)

You must have ENTIRE CONNECTION installed on your personal computer in order to upload PC files to PDS members and download PDS members to PC files. Note that the upload/download utilities only work for 3270-type PCs.

Before attempting an upload or a download, you must first define a PC library using the UUTIL function UL as follows:

On the Library ID Table screen, enter a two character ID of "PC", a DSNAME (for example, MY.PC), and a library type of "M" for personal computer.

For a detailed explanation of the UL utility, see *UUTIL - Menu-driven Utility Functions*.

Downloading a PDS Member to a PC File

To download a PDS member to a PC file, perform the following steps.

Step 1

On the COM-PASS screen, enter the following command:

```
*UEDIT libname(membername
```

where:

libname	Specifies the name of the library where the member is located.
membername	Specifies the name of the member.

Note that this is the same command used to edit a member. The edit screen will be displayed, showing the first 23 lines of text of the member.

Step 2

Enter the following command on the command/message line of the editor screen:

```
SAVE pc(membername
```

where:

pc	Specifies the PC library ID set up earlier.
membername	Specifies the name of the member.

When you press ENTER, the following screen is displayed.

```
COMEBP0006  - SAVE Parameters Are :-
Member  : TESTPDS
DSN     : MY.PC
Volume  :
Hit Enter To Continue, Clear To Return
```

Step 3

Press ENTER, as prompted, to continue the procedure.

Step 4

When you press ENTER, a prompt appears at the bottom of the screen asking you for the download information (as illustrated in the following figure). Enter the name of the download data file.

```
Enter name of DOWNLOAD DATA FILE ( 1)
```

Step 5

Press ENTER, and the Full Screen Editor menu is displayed.

Step 6

To exit the Full Screen Editor menu, press CLEAR.

Example 1: Downloading to a Diskette

If you wanted to download the PDS member "TESTPDS" to a diskette, you could simply use PC-DOS syntax to specify a disk drive. First complete steps one through three (above). In order to copy the PDS member to a diskette that you have inserted in the A-drive, key in "A:TESTPDS.TXT" at the "Enter name of DOWNLOAD DATA File (1):" prompt (step four, above).

Note:
You can change the name of the file when it is being copied to the PC. Also, the name you are assigning for the download data is for a PC-DOS file, so you must use the format "filename.ext" (where filename is an eight-character filename and ext is a three-character extension) as illustrated in the preceding example.

Example 2: Downloading to a DOS Directory

If you wanted to download the PDS member to a specific DOS directory, you could also specify the directory by using PC-DOS syntax.

Note:
*By default, a member will be downloaded to the ENTIRE CONNECTION (*SAG) directory.

For example, to copy the PDS member to a directory called "FILE1", you would (after completing steps one through three) enter "FILE1\TESTPDS.TXT" at the "Enter name of DOWNLOAD DATA File (1):" prompt.

Note:
If you do not specify a drive, as in the above example, the data will be downloaded into the default drive for the PC.

When the download is completed, you will be returned to the Full Screen Editor menu.

Uploading a PC File to a PDS Member

To upload a PC file to a PDS member, perform the following steps.

Step 1

Either

- Create a member in an appropriate PDS data set (for example, one which corresponds to the logical record length you wish to upload).

  On the UEDIT command line enter the following command:

  ```
  INCLUDE pc(membername
  ```

  Or

- On the COM-PASS screen, enter the following command:

  ```
  *UEDIT pc(membername
  ```

where:

pc	Specifies the PC library ID set up earlier.
membername	Specifies the name of the member.

When you press ENTER, the "Enter name of UPLOAD DATA File (1):" prompt will be displayed at the bottom of the COM-PASS screen.

Step 2

You can specify where the data to be uploaded is to be taken from, for example, a directory other than the ENTIRE CONNECTION (that is, \SAG, the default) directory or from a specific disk drive that is different from the PC’s default drive specification.

Enter the name of the upload data file at the prompt, and press ENTER.

When the upload is completed, the UEDIT editor screen with the first 23 lines of the uploaded text will be displayed.

Step 3

You must now save the uploaded member as a PDS member. If you created a member in Step 1 above, then the DSN and membername will still be set to the values you originally typed in, and you only have to enter SAVE on the command/message line of the editor. Otherwise you have enter the following command on the command/message line of the editor screen:

```
SAVE dsn(membername
```

If you wish, you can enter the same membername as the one you specified for the upload data file, as long as it is eight characters or less.

Warning:
You must change the name of the library. At this point, when the uploaded file is displayed in UEDIT, you are in library PC. Attempting to save in library PC will invoke the Download function, that is, the "Enter name of DOWNLOAD DATA File (1):" prompt will appear.

When you enter the "SAVE DSN(MEMBERNAME" command, a screen will be displayed that shows the SAVE parameters you typed in. A sample screen is illustrated in the following figure.

```
COMEBP0006  - SAVE Parameters Are :-
Member   : TESTPC
DSN      : MYLIB
Volume   :
Hit Enter To Continue, Clear To Return
```

Step 4

If the parameters are as you want them, press ENTER.

You will be returned to the Full Screen Editor menu. A message will appear at the top of the screen confirming that the PDS member has been saved.

Step 5

To exit the Full Screen Editor menu, press the CLEAR key.

ULIB - Program Catalog Maintenance Utility

ULIB is Com-plete's library maintenance utility that allows you to:

- maintain parameters for programs intended for use under Com-plete;
- maintain Com-plete's instorage program directory.

In addition to those load module attributes defined by the program itself and by the linkage editor, many utilities and application programs for execution under Com-plete require specific parameters such as region size, thread lock number, or group lock number. The online utility program ULIB is designed to maintain these and other parameters.

Unlike previous releases of Com-plete that required all load modules to reside in a special program library, in this version all programs are loaded directly from the resident areas of Com-plete, from the resident areas of the operating system, or from Com-plete's COMPLIB load library chain.

Maps and other load modules that do not need to be assigned any Com-plete specific attributes need not to be cataloged. To avoid cataloging a large number of modules with a unique set of parameter values, defaults may be defined.

Note:
Setting applymods 79 or 80 causes Com-plete to load or invoke a program (maps are not affected) only if it has been cataloged.

To avoid searching for a load module each time it is being loaded, Com-plete keeps the BLDL / LOADLIST information for the most recently used load modules in an instorage directory. This enhances performance, but may cause a load module newly stored or replaced in a load library not to become available for use under Com-plete while an entry for this module exists in the instorage directory. Using the ULIB utility program, you can make sure a newly stored load module becomes available immediately.

This chapter covers the following topics:

- Program Parameters
- Defining a Default Parameter Set
- Command Format

Program Parameters

The following parameters can be defined for a program intended for use under Com-plete:

RG=region size	Default: If no region size is specified for a program, ULIB stores a value of zero. When a non-cataloged program is being invoked, Com-plete uses the region size specified for the dummy member $DEFAULT. If none exists, the actual load module size rounded up to the next multiple of 4K will be used.
	Specifies the region size to be used as a limiting factor for program executing GETMAINs, LOADs, etc. concerning storage below the 16MB line. This parameter is taken into account only when the program is the one initially invoked in a thread. When the program is being loaded by another program already executing in the thread, the space occupied by the program being loaded is its actual load module size.
	Note that the region size may be any integer multiple of 4K not exceeding 1008K.
	Note that the amount of storage available above the 16 MB line is the same for all programs and can be defined only by the THSIZEABOVE sysparm.
TG=Task Group	Default: If no task group is specified, the program will be serviced by the DEFAULT task group.
	Specifies the name of the task group the program is to run in.
PG=Thread Group	Default: If no thread group is specified, the program will run in the DEFAULT thread group.
	Specifies the name of the thread group the program shall run in. **Note:** (Important performance implications) Software AG recommends that all programs assigned to the same thread group always be assigned to the same task group also. In other words, it should be avoided that programs running in the same thread group are serviced by different task groups. This will minimize the overhead caused by programs competing for both tasks and threads. On the other hand, from the performance point of view, having multiple thread groups serviced by the same task group is not expected to have a negative impact.
PV	Privileged. A program having this attribute assigned can execute any privileged machine instruction and any standard operating system SVC (e.g., OPEN, CLOSE, etc.) without the performance of validity checking by Com-plete. **Warning:** **You must take great care when using this option, otherwise loss of system integrity could be the result. Note that in z/OS, FORTRAN programs must always be cataloged as PV because FORTRAN always tries to open an "FT06001" DD name for its error files.**

UR	Unrestricted. A program having the "unrestricted" attribute assigned can be invoked by any terminal user, i.e., its usage is not controlled by the COM-PASS Program Services Tables.
BL	Before-logon. (Only for sites using the accounting facility.) A program having this attribute assigned can be executed by a terminal user before logon processing has been completed.

Note:
Only users with "control" status may set or change the attributes "privileged", "unrestricted", or "before-logon".

Defining a Default Parameter Set

You can catalog a dummy module with the reserved name $DEFAULT. When loading a non-cataloged program, Com-plete uses the parameter values defined in this catalog entry. Also, for modules cataloged with a zero region size, the region size defined in $DEFAULT will be used if this value is higher than the actual physical size of the load module.

Note:
Only users with "control" status are allowed to access the catalog entry $DEFAULT.

Command Format

Initial access to ULIB is gained by entering on the COM-PASS main menu screen the call request:

```
*ULIB
```

As the result, the ULIB main menu is displayed:

```
17:15:55        TID    22          COM-5.1.          User MBE          22.04.97
                            -- Program Catalog --                          ULI0

    Function                          ID  Operand
    --------------------------------- --  ------------------------------
    Program overview ................ OV  Prefix/Name, System
    Catalog ......................... CA  Name, System
    Modify catalog entry ............ MO  Name, System
    Delete catalog entry............. DE  Name, System
    Refresh load module ............. RF  Name





               Select function .......
               Operand ...............
               System ................



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

As an alternative, the subfunctions CATALOG, DELETE, and REFRESH can be executed directly from the COM-PASS menu. In this case, no menu screen will be displayed; instead, you will be given a message informing you about completion of the command. The command formats for this mode are:

```
*ULIB CAT,program(,RG=regionsizeK)(,PG=taskgroup)
(,TG=threadgroup)(,PV)(,UR)(,BL)*ULIB DEL,program*ULIB REF,program
```

where *program* is the load module name and all other parameters specify the program attributes as described above.

The subfunctions CATALOG, DELETE, and REFRESH are also available via the computer operator command USER.

Function OV - Program Overview

If you select funtion OV from the ULIB main menu, contents of the Com-plete program catalog are displayed on your screen in alphabetical order, for example:

```
19:29:53        TID     3          TEST-510          User MBE          10.11.97
System: GLOBAL              -- Program Overview --                         ULI1



FC  PGMname   RGsize  PV UR BL AF  Thrd Grp  Task Grp                 Note
-------------------------------------------------------------------------------
.   $DEFAULT       0K
.   ADMC         768K X
.   ADMI         768K X
.   ADMP         768K X
.   ADMV         768K X
.   COBTEST1     204K
.   COBTEST2     200K
.   COMSEC       296K X
.   CSECSV        16K X
.   C450PR       148K X
.   C46CSC       148K X
.   C46MDSNS      32K X
.   C46MTGRP     252K
.   C5SDCONV     256K X        X
.   DISBGN        64K X
FC:  M odify  D elete  R efresh            Reposition to:                  MORE
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  End                           Forwd
```

You can start the display from any program name by typing a name or prefix in the *Operand field* of the ULIB main menu.

The Program Overview screen displays the program names cataloged and the attributes assigned to each of them. You can scroll the catalog forward by pressing PF8 or restart the display from any program name by entering a name or prefix in the field prompted *Reposition to*.

From this menu you can call the Modify, Delete, and Refresh functions for one or more of the catalog entries displayed. To do this, enter M, D, or R respectively in the first column of the appropriate line and press ENTER. The menu of the subfunction selected is displayed. After completion, the Program Overview menu is displayed again with a note about the subfunction executed in the appropriate line. After you press ENTER once more, attribute modifications become visible and deleted entries disappear.

If you select more than one line for subfunction calls, they are executed consecutively top-down, with the Program Overview displayed only after completion of the last subfunction call or after an error occurred.

Function CA - Catalog a Program

If you select the CA function from the ULIB main menu, the Program Catalog Information screen is displayed, initialized with default values for all parameters:

```
19:31:05        TID     3          TEST-510          User MBE          10.11.97
                       -- Program Catalog Information --                   ULI2


System: GLOBAL
Name= MBEPROG

-------------------------------------------------------------------------------

Region size:............    0 Kbyte

Privileged:............. N (Y/N)

Unrestricted use:....... N (Y/N)

Allowed before logon:... N (Y/N)

OS task affinity:....... N (Y/N)

Thread group name:......

Task group name:........

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

You can now enter the parameter values required for the program (see also the section *Program Parameters above*).

Note that you must have control status to set or change any of the attributes *privileged*, *unrestricted*, or *before-logon*.

Press PF5 to write the entry to Com-plete’s program catalog. Pressing PF5 or ENTER causes all data entered to be checked for validity; in case of an error a message is displayed and the cursor is placed on the erroneous input field.

Function MO - Modify a Program Catalog Entry

If you select the MO function from the ULIB main menu or from the Program Overview screen, the Program Catalog Information screen is displayed, indicating the current parameter values for the program.

You can now make the required changes. For details, refer to the description of the Catalog function above.

Function DE - Delete a Program Catalog Entry

If you select the DE function from the ULIB main menu or from the Program Overview screen, the Program Catalog Information screen for the selected program is displayed (see the description of the Catalog function above).

To delete the catalog entry, press PF5.

Note that only the program attributes are reset; no action against the load module takes place.

Function RF - Refresh a Program

To avoid searching for a load module each time it is loaded, Com-plete keeps the BLDL / LOADLIST information for the most recently used load modules in an instorage directory. This enhances performance, but may cause a load module newly stored or replaced in a load library not to become available for use under Com-plete as long as an entry for this module exists in the instorage directory.

The Refresh function causes Com-plete to search for a load module. During execution of the Refresh function, if an entry for the module specified exists in Com-plete's instorage directory, it will be deleted. If the module is in Com-plete's program lookaside buffer due to sysparm PGMLOOKASIDE, it will be deleted from this buffer as well.

When executing the Refresh function, ULIB does not search for the load module; the search will be done only the next time the program is requested to be loaded into a thread.

Notes:

1. Using this function does not require the selected program to be cataloged using the Catalog function of this utility.
2. No warning message is issued if no entry for the module exists in Com-plete's instorage directory or if the load module does not exist at all.
3. Execution of any ULIB Catalog, Modify, or Delete function also implies a Refresh against the appropriate program.

ULOG - LOGON/LOGOFF UTILITY

The ULOG utility program is used to perform the following functions:

- Identify the terminal user to the Com-plete system;
- Terminate the terminal user’s session;
- Alter the user’s logon password.

If Com-plete is initialized with the accounting option activated, ULOG must be used to identify the user to the Com-plete accounting and security routines. Otherwise, use of any application program or Com-plete utility will be disallowed.

A terminal session established through ULOG must also be terminated with ULOG.

Once a user is identified to Com-plete with ULOG, a so-called User Id accounting block (UAB) is created. Statistical information about every user in conversation with Com-plete is accumulated and maintained using this control block. This block is discarded from the system once a user logs off from Com-plete (using *ULOG OFF) or when Com-plete terminates.

This chapter covers the following topics:

- Logon Procedure
- External Security Systems
- ULOG Commands

Logon Procedure

When you invoke Com-plete, ULOG presents you with a logon screen:

```
19:02:07        TID     6          COM-5.1.          User               05/22/97
                                                                             ULG0
 -------------------------   Com-plete  System Logon   -------------------------


 User ID ...........:
 Password ..........:
 New password ......:
 Group (ACF2) ......:



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

Enter your User ID and your current password and press ENTER.

Note:
A user ID and password are character strings uniquely identifying you as user and are assigned to you by the system administrator. You can change your password by supplying the "New Password" in addition to your current password and pressing ENTER. Com-plete asks you to confirm the new password by typing it a second time.

If you press CLEAR or PF3 from the logon screen, the terminal is disconnected and you are returned to VTAM, the calling system (in case of ACCESS), or the logon screen is redisplayed, depending on your installation.

If the logon is successful, the broadcast message is displayed, together with a message with information concerning your last logon.

Note:
The broadcast message is defined using the SI function on the Utilities menu, see the chapter on UUTIL.

The following figure shows a typical broadcast message:

```
COMULG0003 (8) Last access at 19:29:33 0n 05/22/97 Tid(SHRDAEN) Sys(COM-5.1.)
*----------------------->  BROADCAST  <-->   COM-5.1.  <----------------------*
*                                                                              *
*                        W e l c o m e   t o   t h e                           *
*                                                                              *
*                    w o n d e r f u l   w o r l d   o f                       *
*                                                                              *
*  CCCCC   OOOOO   MM   MM          PPPPP   LL      EEEEEEE TTTTTTTT EEEEEEE *
* CCCCCCC OOOOOOO MM MMM MM         PPPPPP  LL      EEEEEEE TTTTTTTT EEEEEEE *
* CC      OO   OO MM  M  MM         PP   PP LL      EE         TT    EE      *
* CC      OO   OO MM     MM  ******  PPPPPP  LL      EEEEE      TT    EEEEE   *
* CC      OO   OO MM     MM  ******  PPPPP   LL      EEEEE      TT    EEEEE   *
* CC      OO   OO MM     MM          PP      LL      EE         TT    EE      *
* CCCCCCC OOOOOOO MM     MM          PP      LLLLLLL EEEEEEE    TT    EEEEEEE *
*  CCCCC   OOOOO  MM     MM          PP      LLLLLLL EEEEEEE    TT    EEEEEEE *
*                                                                              *
*                                                                              *
*                                                                              *
*                          >>>  Version 5.1.1  <<<                             *
*------------------------------------------------------------------------------*
```

External Security Systems

Com-plete interfaces with the most commonly used external security system packages available for z/OS. When an error occurs during logon processing, these security systems pass one or more messages to Com-plete, describing the cause of the problem.

If only one message is received, it is displayed on the logon screen, for example:

```
19:33:12        TID     6           COM-5.1.           User             05/22/97
                                                                            ULG0
 -------------------------   COM-PLETE  System Logon   -------------------------




 User ID ...........: SAGATEST
 Password ..........:
 New password ......:
 Group (ACF2) ......:





 ACF01004 LOGONID SAGATEST NOT FOUND

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

If the External Security System answers the Com-plete logon request request with more than one message, you are notified with the following message displayed on the logon screen:

```
SAF Messages waiting ...
```

Press PF2 to display the security system messages. Up to 15 messages can be displayed.

ULOG Commands

There are some commands that can be issued to ULOG to access to the Com-plete system and to release or transfer control.

ULOG ON

The ULOG ON command identifies you as a Com-plete terminal user to the Com-plete accounting and security routines. If applymod 52 is on, this command is automatically issued to Com-plete when you invoke Com-plete from VTAM.

The command format is:

```
*ULOG ON,userid,[password]
```

where *userid* is a valid user ID. The password option must be specified if your installation requires it.

If you issue the *ULOG ON command from a Com-plete session on a terminal on which another user is already logged on, the first user is automatically logged off and you are logged on.

ULOG OFF

This command logs you off from Com-plete; your terminal is disconnected from the Com-plete system.

The command format is:

```
*ULOG [OFF]
```

The OFF operand is the default and therefore not required.

ULOG STAY

This command is the equivalent of the ULOG OFF command, except that under VTAM, you are not returned to VTAM but to the Com-plete logon screen.

The command format is:

```
*ULOG STAY
```

ULOG DISC

Under VTAM, the ULOG DISC command forces disconnection from the terminal and you are returned to VTAM.

The command format is:

```
*ULOG DISC
```

ULOG PASS

This command forces a logoff and causes VTAM to establish a new session with the application specified in the command operands.

The command format is:

```
*ULOG PASS,VTAM-appl-ID<,userid><,user-data>
```

where:

PASS	Can be abbreviated to P.
VTAM-appl-ID	Required: specifies the ID of the VTAM application. The value specified can be up to eight bytes.
userid	Optional: specifies a user ID of up to eight bytes.
user-data	Optional: specifies data passed to the target application.

Note:
Errors detected during command interpretation result in a normal logoff with an additional call to ULOGX1, passing the appropriate error code.

UM - Message Switching / Printout Spooling Utility

The online utility program UM is can be used by all Com-plete users. Some UM functions are especially designed for the application-oriented users, while others are designed for the system programmer. This means that you can issue some UM functions only if you have control status. These functions are marked as such in this section.

By default, the usage of the "TID=" argument is restricted to control users. This facility can be made available to all users by specifying APPLYMOD=7 in the Com-plete startup sysparms. A full description of the APPLYMOD sysparm can be found in the Com-plete System Programming documentation.

Note that throughout this section, all references to messages or message switching should be interpreted to mean both message switching and printout spooling, unless otherwise specified.

The UM system functions are:

- Send messages to any terminal in the Com-plete terminal network;
- Retrieve, purge, delete, reroute, repeat, hold and release messages;
- Change the sending authorization class codes for any given terminal (control user);
- Change the receiving authorization class codes for any given terminal (control user)

  These functions are described in this section in the sections *Menu Operation* and *Direct Command Operation*.

This chapter covers the following topics:

- Destination Codes
- Class Codes
- Message Queue
- Message Delivery and Receipt
- Message Interruption and Recovery
- Message Text
- Message Routing
- Alternate Terminals
- Disabled Terminals
- Inoperative Terminals

- Menu Operation
- Direct Command Operation

Destination Codes

When you send a message, you must include information indicating where the message is to be received. You can select the terminal(s) that are to receive the message by specifying one or more destination codes. A destination code can be either:

- A Terminal Identification number (TID);
- A destination code that represents one or more TIDs.

  Destination codes allow an installation to form convenient groupings of TIDs. Each installation can also define procedures for changing and adding destination codes. When you use a destination code in a message switching terminal operator command, Com-plete first converts it into TID numbers and then sends the message(s) to the appropriate terminals.

Class Codes

A class code designates security restrictions and other information about a message. There are two categories of class codes:

1. Security class codes (numbered 1 through 4);
2. Other class codes (numbered 8 through 16).

For a successful message request, you must assign at least one security class code to each message you send.

Each terminal in the Com-plete network has two sets of security class codes assigned to it.

1. The first defines which message classes it can send;
2. The second defines which message classes it can receive.

If the accounting option is used, class codes are assigned through the user ID. If the accounting option is not used, class codes are assigned through the terminal definition table.

Note that messages must have security class codes assigned. Com-plete checks to determine whether a sending terminal is authorized to send a specific message by verifying that the class code(s) assigned to that message is among the sending class codes assigned to the sending terminal. Com-plete also verifies that the assigned class codes(s) are included in the list of the receiving class codes of the terminal(s) to which the message is sent. If the class codes are not compatible, the message is not sent, and in some cases, a security violation is logged to the Com-plete logging device.

Several class codes (numbered 1 through 4) cause certain special operations to be performed as a message is sent. These operations and their class codes are explained in the following table.

Class	Description
Class 1	Standard message class.
	Messages with this class assigned do not interrupt a terminal while it is in conversation with a program.
Class 2	Urgent message class.
	Overrides the MESSAGE DISABLED status of a terminal and causes the message to be displayed immediately at the receiving terminal. The receiving terminal will even be interrupted if it is in conversation with a program.
	If the receiving terminal has the audible alarm feature, the audible alarm will sound.
Class 3	Special purpose class code.
	The message will interrupt a terminal in conversation with a program. **Note:** If Com-plete is reinitialized before this type of message is successfully sent, the message is restarted from the beginning.
Class 4	Reserved for Com-plete logged messages.
	No application program or terminal can initiate this type of message. If Com-plete is reinitialized before successful message receipt, these messages are exempt from restart.

Note:
If the receiving tid is an ACCESS terminal, class codes 2 and 3 will be treated the same as class 1.

Class codes 12 through 16 cause special services to be performed as a message is sent.

Class	Description
Class 12	Causes a message to be deleted from the message queue file after thirty minutes if it cannot be sent to the receiving terminal.
	If Com-plete is reinitialized before successful message receipt, these messages are exempt from restart.
Class 13	Causes a message to be written without the standard message header that normally accompanies all messages.
	If Com-plete is reinitialized before successful message receipt, these messages are exempt from restart.
Class 14	Standard message class; no special action is taken.
	If Com-plete is reinitialized before successful message receipt, these messages are exempt from restart.
Class 15	If the receiving terminal has the audible alarm feature, causes the audible alarm to be sounded when the message is written to the terminal.
Class 16	Prevents a message from being queued to the message queue file on disk. Instead, the message remains in main storage until it is successfully sent.
	The length of a message using this class code is limited to the amount of text that can be contained in one message buffer. To calculate this amount, double the number of terminals that are to receive the specified message and subtract this amount from 240. For example, if there were 3 receiving terminals, a class 16 message could be a maximum of 234 characters long.

Message Queue

When a message is sent to a terminal, it is by default copied onto a disk file containing queues of messages for each terminal. The order of the messages queued to a specific terminal depends on the priority rating of the sending terminal. Note that you can recall messages from this file for redisplay at any time as long as they have not been successfully delivered. To accomplish this, use the UM time command options (*UM D or *UM *number*) described later in this section.

Message Delivery and Receipt

Messages sent to hard copy terminals are printed automatically if the terminal is in ready status and no other messages are queued to the terminal.

Messages sent to display devices require operator acknowledgment following display. To acknowledge a message, simply press ENTER.

Note:
If you are in the process of using an application program and receive a message, any screen formatting you are using may be destroyed. Since the specific procedures for recovering a screen format are application-dependent, you must reference the specific application you were using at the time of the interrupt for recovery procedures. Regardless of the acknowledgment procedures you use, the application you are using will not be terminated, only interrupted.

A message sent to you may be longer than the buffer size for your receiving terminal. This is indicated by an asterisk (*) as the last character displayed. To display the remaining portion of the message, press ENTER.

There are three common situations in which messages sent to a terminal are*not* automatically displayed:

1. If the terminal is MESSAGE DISABLED. Normally, when messages are sent to a terminal, they may interrupt and destroy whatever data is being entered at the time. To prevent this, set the terminal to disabled status for receipt of messages.

   Note:
   MESSAGE DISABLED status does not prevent class 2 messages from being received.

2. If a conversational program is executing at the terminal. If an active program is currently in use at the receiving terminal, messages will be placed in a special message queue for that terminal. The messages will be received when the application program terminates.

   Note:
   This feature does not apply to messages sent with class codes 2 or 3 assigned.

3. If the previous message was not acknowledged. If a previously sent message has not been acknowledged, additional messages will be queued.

   Note:
   This condition can occur only with CRT devices.

   When a terminal is enabled for receipt of messages, (e.g., a conversational program is ended, or the MESSAGE DISABLED status is removed), any messages that have been placed in the message queue will be available for display. On CRT terminals, as you acknowledge each message queued to your terminal, the next message will be displayed until all have been displayed and acknowledged.

Message Interruption and Recovery

The message switching facility of Com-plete provides automatic message restart in the event that terminal output is interrupted. Message restart is performed on a checkpoint basis. Each message sent to a terminal is initially queued to the message switching file queue residing on disk. Messages residing on the message file queue are sent or written to the destination terminal on an availability basis. As messages are written, a core queue checkpoint is taken to indicate the status of the output being written (for large messages, this checkpoint is taken at the completion of every full page of output). If output is interrupted (for example, by pressing the STOP key), output resumes at the last checkpoint when the terminal is made ready again (for example, by pressing the START key).

If a system failure occurs, all message core queue checkpoint records are destroyed. When Com-plete is again initialized, various message restart options are available, the default being recovery of message output from the beginning of the output message. For more details on the message recovery options available at your installation following a system failure, consult the system programmer responsible for Com-plete maintenance.

Message Text

The text of any given message can contain any character combination except:

- A left parenthesis "(" must not be the first character of a message;
- The phrase "TID=" must not be the first phrase of a message;
- An asterisk (*) must not be the last character of a message.

Com-plete treats all messages, regardless of content, as text when they are displayed at the receiving terminal: they are printed or displayed using the maximum line length of the receiving device. A word that will not fit at the end of a line is moved to the beginning of the next line. All blanks at the end of a message are ignored.

Message Routing

Note that, except for class 13 messages, each message received has a standard header containing the following:

- The message identification number;
- The date;
- The time the message was written;
- Terminal Identification number of the sending terminal.

The following illustrates a typical message header:

```
MSG ID:    133, SENT 10/17/97 AT 1106, FROM TID 17
SAMPLE MESSAGE HEADER DISPLAY
```

Note that if you do not want this header to appear, you must assign class code 13 to your message.

Alternate Terminals

Alternate terminals are those designated to receive message output for terminals that are either inoperative, currently in conversation with an online program, or disabled for message receipt. If a terminal has an alternate terminal assigned and cannot immediately receive a message, any message sent to it (except class 2 messages) is automatically displayed or written to the alternate terminal. Class 2 messages ignore an alternate terminal assignment and are received at the destination terminal unless it is inoperative.

Note:
Messages rerouted to an alternate terminal are no longer queued to the original receiving terminal.

You can assign alternate terminals in one of three ways:

1. Through the Terminal Definition Table (TIBTAB);
2. By using the ALT command of the UM facility.

Note that an alternate terminal assignment made via TIBTAB is a permanent assignment, whereas the setting made via the ALT command is only valid for the duration of the Com-plete session.

Disabled Terminals

When a message is sent to your terminal, it may interrupt and/or destroy whatever data you may have entered at the time the message is sent. The DISABLE terminal function prevents this by enabling you to temporarily suspend message receipt for a terminal.

If your terminal has been disabled, all messages sent to it (except those assigned message class 2) will be either routed to the alternate terminal, if assigned, or placed in the message file disk queue. Class 2 messages override a disabled status and are displayed immediately.

Inoperative Terminals

Inoperative terminals are those that are either:

- Powered off;
- Malfunctioning;
- Write-inhibited via a switch.

Com-plete handles inoperative terminals in a manner similar to its handling of disabled terminals. Messages sent to these terminals are either rerouted to a designated alternate terminal, if assigned. If no alternate terminal is assigned, the message is placed in the message queue for the inoperative terminal.

Messages sent to an inoperative terminal, if not rerouted to an alternate terminal, are automatically displayed when the terminal is once again able to receive messages.

Menu Operation

Message switching can be performed using direct commands or by selecting options from the UM menu. The following subsections describe the menu mode of work. For a description of direct command mode, see the section *Direct Command Operation*.

You can invoke the UM main menu with the command *UM. This displays the UM Message Switching Menu on the screen of your terminal:

```
COMMSG0001 (8) Please enter function.
 09:59:23        TID     9           COM-5.1.          User SAGAWW       05/23/97
                          ---  Message switching  ---                        UM00
                         Function. . . . . . Fc Parm(s)
                         ------------------- -- -------
           Send msg to.: Users, TIDs ....... SM 1,2,3
                         All active users .. SA 1,3
                         Users (from menu) . SS 1,3
                         Accounting Groups . SG 1,3
                         TIB Group(s) ...... ST 1,3



                         Select Function ...
 (1) Message:
                                                                          <More
 (2) Users, TIDs:
 (3) Class Codes: 1                                                 0 Receivers
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
             Mode  End         Send
```

You perform UM functions by selecting the appropriate option from this menu and specifying values in the other input fields as described in the following subsections.

Function SM - Send Message to User(s) or TID(s)

If you already know the individual user IDs or TID numbers that you wish to receive the message, enter up to seven destinations in the *Users*, *TIDs*field and press PF5 to send the message specified in the*Message*field.

If you want to send the message to more than seven destinations, press ENTER instead of PF5. Repeat this procedure until your destination list is complete, then press PF5 to send the message.

Note:
The "n Receivers" field on the bottom right hand side of the screen shows the number of destinations selected so far.

Press PF2 to display the destinations you have already selected., for example:

```
11:31:09        TID     9           COM-5.1.          User SAGAWW       05/24/97
 Func: SM                  ---  Message switching  ---                       UM01
 Destination      Destination      Destination      Destination      Destination
 -----------      -----------      -----------      -----------      -----------
 U=MBE            U=NG             U=SAGAWW

 (1) Message: TEST
                                                                          <More
 (2) Users, TIDs:
 (3) Class Codes: 1                                                  3 Receivers
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
             Mode  End         Send        bwd   fwd
```

Function SA - Send Message to all Logged on Users

You can select option SA to send the message to all users currently logged on to Com-plete. These messages are not saved and are lost when Com-plete terminates.

Function SS - Send Message to User(s) (from Menu):

If you select function SS and press ENTER after specifying the message text, a list of all users currently logged on to the Com-plete system is displayed. The following figure shows an example of such a display:

```
COMMSG0017 (8) Press 'PF5' to execute function.
 16:30:15        TID     9           COM-5.1.          User SAGAWW       05/24/97
                          ---  Message switching  ---                       UM20
 M   Userid..    M   Userid..    M   Userid..    M   Userid..    M   Userid..
     NG              TIMERM          SAGAWW

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

You select users by entering any character into the M column left to the user ID. press PF8 to scroll to the next page. When your selection is complete, press PF5 to send the message.

Function SG - Send Message to Accounting Groups:

If you select function SG and press ENTER after specifying the message text, a screen displaying Accounting Groups of currently logged on users appears, for example:

```
COMMSG0017 (8) Press 'PF5' to execute function.
 10:43:12        TID     7           COM-5.1.         User SAGAWW       05/27/97
                          ---  Message switching  ---                       UM20
 M Acct.Group..  M Acct.Group..  M Acct.Group..  M Acct.Group..  M Acct.Group..
   WIEN            EBERSTADT       MR. COMPLETE    GROUP           COMPLETE









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

You select accounting groups by entering any character into the "M" column left of the accounting group name. Press PF8 to scroll the list forward to the next page. Once you have selected all desired groups, press PF5 to send the message.

Function ST - Send Message to TIB Groups:

If you select function ST and press ENTER after specifying the message text, a list of the defined TIB Groups is displayed, for example:

```
COMMSG0017 (8) Press ’PF5’ to execute function.
 10:51:22        TID     7           COM-5.1.         User SAGAWW       05/27/97
                          ---  Message switching  ---                       UM20
 M   Tibgroup    M   Tibgroup    M   Tibgroup    M   Tibgroup    M   Tibgroup
     WIEN

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

Select the desired group(s) by entering any character into the "M" column left to the "Tibgroup" column. Press PF8 to scroll the list to the next page. Press PF5 to send the message to all selected groups when your selection is complete.

Direct Command Operation

Initial access to UM is gained by entering the call request:

`*UM command`

where *command* represents the desired command.

Note:
Arguments in UM command syntax must be delimited by periods (full stops)*not* commas.

If you omit the *command* parameter under COM-PASS, The UM Message Switching Menu is displayed (see the section *Menu Operation* above).

The available UM commands are summarized in the following table and described in more detail in the subsequent subsections.

Command	Function
d1.message	Sends a message to a terminal(s).
D	Displays status information on all messages in the message file.
message	Displays a specific message.
DELETE.n	Deletes a specified message from the message file in order to prevent its display.
HOLD.n	Holds a specific message in order to prevent its display.
RELEASE.n	Releases a message from hold status in order to allow its display.
PURGE	Purges all messages destined for the terminal.
D.t1	Obtains a listing of messages displayed during a specified time interval.
ENABLE	Enables a terminal that was disabled.
DISABLE	Disables a terminal in order to prevent receipt of the output.
RESET	Resets a terminal to allow terminal I/O.
ALT=tid	Assigns an alternate terminal for message output when the desired terminal is unavailable.
ALT=REMOVE	Removes the alternate status of a terminal.
RCLASS*	Change receiving authorization class codes. Format: `*UM RCLASS= (c1=YES|NO,c2=YES|NO,...,cn=YES-|NO).TID=xxx`
SCLASS*	Change sending authorization class codes. Format: `*UM CLASS= (c1=YES|NO,c2=YES|NO,...,cn=YES|NO).TID=xxx`

* Available only if you have control status.

Class codes are assigned to specific terminals by use of the terminal definition table (TIBTAB). They are also assigned to specific terminal users through the user ID, thus overriding any assignments made in TIBTAB. The class codes assigned to a specific user ID can be overridden by a control user while the user ID is in use, however, by using the RCLASS and SCLASS commands.

Send to Terminal(s):

This command enables you to send a message to one or more terminals or a set of terminals identified by a destination code. In addition, all class codes to be assigned to the message have to be given at that time.

The command format is:

```
*UM d1<,d2,...,dn><.(c1,c2,...,cn)>.message
```

The arguments are:

d1	Required. Specifies the destination code of the Terminal ID Number (TID) of the terminal to receive the message.
d2,..,dn	Optional. Specifies additional destination codes or TIDs to receive the message.
	If more than one argument is entered, each must be separated by a comma and the set of arguments must be bounded by periods.
	A value of ALL will send the message to all terminals in the network that have a receiving message class code of 2.
(c1,..,cn)	Optional. Specifies the class code(s) to be assigned to the message. If d1 is ALL, the default is 1; otherwise the default is 2. **Note:** Class codes must be enclosed in parentheses. If more than one class code is specified, each must be separated by a comma. In addition, the parentheses must be bounded by periods.
message	Required. Specifies the message text. See the sections *Message Text* above and *Message Segmentation* below.

Message Segmentation

Messages are formed in segments. Each segment, including all characters in the message command itself, may not exceed 1,000 bytes in length. (This is the equivalent of 12.5 lines of message text on a 3277 Model 2 terminal.) Note that there is no limit to the number of message segments that make up a given message; however, the size of the message queue file itself is a limiting factor. Before a segmented message is written to the receiving terminal device, the segments are linked together by Com-plete and the message is sent as a single unit.

To end a message segment, enter an asterisk as the last character of the message text. To attach subsequent text segments to the initial message text, simply enter an asterisk as the last character of each segment. Consequently, an asterisk cannot be the last character in a given message. Note that you need to invoke UM only once via the initial call. The asterisk, which is used only as a segment delimiter, does not appear in the message when it is received.

Display Message Status

Messages sent to a terminal that are waiting to be displayed are queued in the message queue. It is often desirable to be able to obtain status information about pending messages in order to decide which UM functions you want to perform on specific messages (for example, reroute or delete a specific message).

The message status display function enables you to display status information about all pending messages, including:

- The TID of the terminal that generated the status display;
- The TID of an alternate terminal, if any (see the section *Alternate Terminals* above);
- The sending and receiving class codes;

- The terminal status (for example, Message Disabled or not. See the section *Disabled Terminals* above);
- The message number, status, and other information for all messages sent to the terminal that have not been written.

The command format is:

```
*UM D<.TID=tid>
```

with the optional argument *tid* as described above. If you omit the TID number, the status display is given for the terminal at which you enter the request.

Note:
The TID operand must be separated from the remaining characters of the command by a period.

Display a Message

You can selectively display messages from the message queue. You must know the number of a specific message in order to obtain a selective display.

The command format is:

```
*UM message
```

where *message* is a message number, which will cause that message to be displayed;

Note:
If you enter a request to display a message, the receiving message class codes of the terminal you are using (or your User ID, if the accounting option is in effect) must be compatible with the sending message class codes assigned to the message or a message request violation will occur.

Delete a Message

You can selectively delete messages pending output to a terminal from the message queue in order to prevent completion of their output.

The command format is:

```
*UM DELETE.n
```

where *n* is the number of the message to be deleted.

Note that you can only delete messages pending output, and only those sent to the terminal you are currently using.

Hold a Message

This function enables you to hold specific messages destined for the terminal you are using.

The command format is:

```
*UM HOLD.n
```

where *n* is the number of the message to be held.

Note:
A message in hold status will not be displayed at your terminal. Before you can display it, you must first release it from hold using the RELEASE command. The hold will be in effect until you request a release.

Release a Message

You must release a message in hold status before you can display it.

The command format is:

```
*UM RELEASE.n
```

where *n* is the number of the message to be released from hold.

Purge all Messages

Use this function to eliminate message output to a terminal regardless of the type or number of messages being sent.

The command format is:

```
*UM PURGE
```

Enable Terminal

This function allows you to remove the disabled message status from your terminal.

The command format is:

```
*UM ENABLE
```

After you invoke this function, the following message is displayed at your terminal:

```
UMS0022 - REQUESTED OPERATION COMPLETE
```

Any messages waiting in the message queue for display at your terminal are presented immediately.

Disable Terminal

This function allows you to temporarily suspend message receipt from your terminal.

The command format is:

```
*UM DISABLE
```

After you invoke this function, the following message is displayed at your terminal:

```
UMS0022 - REQUESTED OPERATION COMPLETE
```

Any messages waiting in the message queue for display at your terminal are presented immediately after you remove the disabled status with the ENABLE command.

Reset Terminal I/O

Use this function to restart output to a terminal that has stopped because of permanent terminal I/O errors. This function enables you to cause messages destined for your terminal to be requeued to the message queue.

The command format is:

```
*UM RESET
```

Note:
If permanent I/O errors persist, consult a hardware maintenance engineer.

Set Alternate Terminal

Use this function to temporarily route messages to an alternate instead of your terminal.

The command format is:

```
*UM ALT=tid
```

where *tid* is the Terminal Identification number (TID) of the terminal designated as the alternate.

Note:
If an alternate already exists, you must first remove it before you may assign another alternate. To accomplish this, use the remove option of the ALT command (see the following sub-section).

Remove Alternate Terminal

Use this command to remove an alternate terminal assigned to your terminal.

The command format is:

```
*UM ALT=REMOVE.TID=ORIGINALTID
```

When you invoke the Remove Alternate function, all messages sent to the designated original destination terminal are either received immediately or queued in the message queue.

Change Receiving Class Codes (Control User)

The command format used to modify the receiving authorization class codes for a given terminal is:

```
*UM RCLASS=(c1=YES|NO,c2=YES|NO, .........cn=YES|NO).TID=yyy
```

This command is used to temporarily change the receiving authorization class code specifications for the terminal designated by yyy; it may be issued only from the control terminal with TID=1.

The arguments are:

c1,c2,...,=YES|NO	Each cn represents a class code number from one to seven (c1 represents class 1, c2 represents class 2, etc.) and must be replaced with the class code it represents. The class codes are fully described in the section *Class Codes* above.
	Cn=YES indicates that the designated terminal is allowed to receive messages for that particular class.
	Cn=NO indicates that the designated terminal is not allowed to receive messages for that particular class.
	If a particular class code number is not specified, its status remains as specified in the TIBTAB.
yyy	Specifies the Terminal Identification (TID) number of the terminal for which the RCLASS function is to be performed.

Note that class code modification is temporary only; class codes remain the same until changed with another RCLASS command or until Com-plete is reinitialized.

Example

Terminal 66 currently can receive messages with class codes 1 and 2. To disable receipt of class 2 messages and enable receipt of class 3 messages, issue the following command:

```
*UM RCLASS=(2=NO,3=YES).TID=66
```

Change Sending Class Codes (Control User)

The format of the command used to modify the sending authorization class codes for a given terminal is:

```
*UM SCLASS=(c1=YES|NO,c2=YES|NO, ......... cn=YES|NO).TID=yyy
```

This command is used to temporarily change the sending authorization class code specifications for the terminal designated by *yyy*.

The arguments are:

c1,c2,...,cn=YES|NO	Each cn represents a class code number from one to seven (c1 represents class 1, c2 represents class 2, etc.) and should be replaced with the class code it represents. The class codes are fully described in the section *Class Codes* above.
	Cn=YES indicates that the designated terminal is allowed to send messages for that particular class.
	Cn=NO indicates that the designated terminal is not allowed to send messages for that particular class.
	If a particular class code number is not specified, its status remains as specified in the TIBTAB.
yyy	Specifies the Terminal Identification (TID) number of the terminal for which the SCLASS function is to be performed.

Class code modification is temporary only; class codes remain the same until changed with another SCLASS command or until Com-plete is reinitialized.

Example

Terminal 66 currently may send messages with class codes 1 and 2. To disable sending class 2 messages and enable sending class 3 messages, issue the following command:

```
*UM SCLASS=(2=NO,3=YES).TID=66
```

UMAP - Terminal Mapping Utility

UMAP is a real-time terminal mapping utility that allows you to dynamically create, modify, and test maps on a 3270-type or compatible terminal. UMAP also generates Assembler, COBOL, and PL/I source statements that define the data area passed to the Com-plete mapping system at execution time.

Note:
The UMAP utility functionally replaces the current macros MAPSTART, MAPF, and MAPEND.

This chapter covers the following topics:

- Command Format
- Using UMAP
- UMAP Main Menu
- UMAP Main Menu Functions
- UMAP Modify Map Menu

Command Format

To invoke UMAP, enter the following command:

```
*UMAP
```

When you enter this command, a menu of the UMAP command functions is displayed. See the section *UMAP Main Menu* for a detailed description of this menu.

Direct Command Support

You can also access UMAP functions using direct command input.

The command format is:

```
*UMAP f name [operand] *
```

where:

f	Specifies the one-character function ID as listed on the UMAP Main Menu and the UMAP Modify Map Menu.
name	Specifies the four-character name of the map on which the function is to be performed. Note that the first character must be alphabetic.
operand	Specifies any operands associated with the function.
*	Separated by a blank, as the last character of the input string, indicates that the user is to remain in the UMAP utility instead of leaving after completion of the selected function.

Note:
When you use * as the symbol for variable fields (for example, with the SHOW function), you will remain in the UMAP utility after completion of this function.

Using UMAP

You can initiate all UMAP functions via the UMAP Main Menu, which is discussed later in this section. If you want to modify a map, choose that function on the Main Menu and you will be prompted to a (secondary) Modify Map Menu, where you can select more specific modify functions.

Map Storage

UMAP uses two storage methods.

- SD files, for the dynamic creation and modification of maps;
- The map library, for backup and cataloging the map for use by an application program.

SD Files

Each time you return to the Modify Map Menu, UMAP writes your map to the SD files, except when you are using the DELETE, TEST, SHOW, and SAVE functions. This feature enables you to create and/or modify the map within the SD file over an extended period of time. The name of the SD file is the same as the map name plus TID=SHR. This convention allows you to modify the maps from any compatible terminal.

You can monitor the use of the SD files by using the UUTIL function SD. For additional information, see the section on the UUTIL utility.

Map Library

To save a map in a designated map library, use the UMAP SAVE function or select the SAVE function on the Clean Up screen when you leave the UMAP utility. Note that a map is saved in the map library for backup and cataloging operations.

In z/OS, the load library is defined by the SYSMAP DD in the Com-plete procedure. This load library may be altered for an installation by user exits.

In VSE, the map library is the first core image library defined by LIBDEF PHASE,CATALOG=xxxxxx.

Map Retrieval

If you do not mark the input field beside "Retrieve from COMPLIB Load Library chain" on the UMAP Main Menu, UMAP retrieves the map by first searching for a copy of the map saved in an SD file and, if no map is found there, then searching the map load library.

If you do specify retrieval from COMPLIB Load Library chain on the UMAP Main Menu, UMAP will search for the map in Com-plete’s load library chain only.

Termination

UMAP Function

To terminate a UMAP function, press the CLEAR key, which backs you out one level at a time. For example, if you invoked a UMAP function from the Modify Map Menu, you would press the CLEAR key twice to return to the Main Menu. To leave the UMAP utility, press the CLEAR key a third time.

To return directly to the Main Menu, invoke the SAVE function. If you have outstanding SD files, UMAP first displays a Clean Up screen, shown in the following figure, giving you the choice to either:

- Delete the SD file;
- Save the map in the SD file into the map library and delete the SD file;
- Retain the SD file.

Direct Exit

You can leave the UMAP utility from any map by entering only "*" and pressing ENTER. If there were any SD-files for the UMAP session, you will first be prompted by the Clean Up screen (shown in the following figure) to indicate whether you wish those files to be retained, saved, or deleted.

```
13:28:56     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                             -- Clean up --                               U2MF


           Map Name                     Action









The above Maps were modified in this session and remain in the SD files.
Actions:
         R - Retain Map in the SD Files.
         S - Save the Map into the Map Library and delete from the SD Files.
         D - Delete the Map from the SD Files.
```

UMAP Main Menu

When you enter the command *UMAP, the UMAP Main Menu will be displayed, as shown in the following figure.

```
13:22:19      TID    13           COM-5.1.           User ID ADMIN    09/26/97
                           -- Map Maintenance --                          U2MA
      Function             ID     PFK          Operands
 ----------------------    --     ---   --------------------------------------
             Modify Map    M        1   Mapname
       New Map Creation    N        2   Mapname
               Copy Map    C        3   Mapname, Copy from
Delete Map from SD File    D        4   Mapname
               Show Map    S        5   Mapname, Symbol
 Terminal Control Codes    T        6   Mapname
         Edit Copy Code    E        7   Mapname, (Member,) Language
  Load GDDM Symbol Sets    L        8

                  Help     ?       12   (ID)

        Select Function:       or PFK
           and Operands
                Mapname:       F2  Retrieve from COMPLIB Load Library chain
              Copy from:
       Show with Symbol:
                 Member:                                      Field Indicators
               Language:            (Asm,Cobol,PL1)             Variable  ?
                   Help:                                         Constant  +
```

Use the Main Menu to specify the map name and select a UMAP function together with its required operands. The UMAP Main Menu contains the following input fields:

- A field for entering a UMAP function from the list on the menu;
- The first four characters of the mapname;
- A field for indicating that map retrieval is to be from the Com-plete load library chain;
- Field(s) for the operand(s) required by each UMAP function;
- The Help indicator.

Mapname

The mapname, a six-character code, is required to invoke all functions except "Help" and "Load GDDM Symbol Sets".

You only need to specify the first four characters, which identify the map. Note that the first character must be alphabetic.

UMAP specifies the last two characters, which is the terminal device code indicating the type of terminal for which the map was designed.

Note:
UMAP allows you to create, display, modify, and delete maps only from terminals for which the map was designed.

UMAP obtains the terminal code from a GETCHR call to Com-plete. The device codes are identified in the following table.

Device Code	Line Length	Number of Lines	Terminal Type
F1	40	12	3277M1
F2	80	24	3277M2/3278M2
F3	80	32	3277M3/3279
F4	80	43	3278M4
F5	80	12	3278M1
F6	132	27	3278M5

Com-plete terminal mapping allows application programs to use maps with device codes other than the terminal's device code. For an explanation of map scaling, see Terminal Mapping in *Terminal I/O Functions* of the *Com-plete Application Programming* documentation.

Map Retrieval from Com-plete's Load Library Chain

To request retrieval from Com-plete's load library chain, enter any character in the input field to the right of "Retrieve from COMPLIB Load Library chain" on the UMAP Main Menu.

Note:
This option is not valid for the CREATE, DELETE, and SAVE functions.

If you do not select program library retrieval, the default retrieval is from one of the following:

- The SD file with this map name and TID=SHR;
- The map library.

Field Indicators

On the UMAP Main Menu, you can define the characters used to distinguish between variable and constant fields. Two field indicators are used: one to define a variable field and the other to define a constant field. Note that, when a map is displayed, the indicator immediately precedes the field that it defines.

A constant field contains permanent text that resides in the map and is displayed when the map appears on the screen.

A variable field contains a series of characters obtained from a program's calling parameter list, the number of which defines the length of the field.

Note:
If the first character of a variable field is numeric, the field is defined as numeric rather than alphanumeric.

The following table shows examples of field definition using "+" as a constant field indicator and "?" as a variable field indicator.

Example	Field Type
+ADDRESS	Constant field
?xxxxxxxxxx	Alphanumeric variable field
?9999	Numeric variable field

UMAP Main Menu Functions

To select a function, either enter the single character function ID in the "Select Function" field or press the corresponding PF Key.

Note:
Except for the "Load GDDM Symbol Sets" and the "Help" functions, you must key in a map name when selecting a function.

HELP

To display the "UMAP Online HELP System" menu, enter a question mark (?) in the "Select Function" field, or press PF12. When the HELP screen appears, position the cursor at the field next to the desired function, and press ENTER to get help text on the function selected.

To get help for a particular function that is listed on the Main Menu, key in the single character function ID in the "Select Function" field; then either press PF12, or mark the "Help" field at the bottom of the screen and press ENTER.

The UMAP Main Menu functions are summarized in the following table and described in detail in the remainder of this section.

Function	Description
Modify Map	Displays the UMAP Modify Map Menu.
New Map Creation	Displays a blank screen on which you may design a map and specify the desired field types and then save the map in an SD file.
Copy Map	Creates a duplicate of the specified map in the SD files.
Delete Map from SD File	Deletes a specified map from the SD files.
Show Map	Views a mapped screen with all the specified attributes as well as the variable alphanumeric fields displayed with the designated character.
Terminal Control Codes	Modifies the current Terminal Control Codes.
Edit Copy Code	Edits the source code or the copy code using the Com-plete editor.
Load GDDMSymbol Sets	Displays the "Load Programmed Symbols" screen where you may load GDDM-generated symbol sets into a 3279 graphics terminal and assign a symbol set ID.

Modify Map

The "Modify Map" function enables you to display the Modify Map Menu, where you can select more specific modify functions. These functions are described in a later section, *UMAP Modify Map Menu*.

New Map Creation

The "New Map Creation" function enables you to design, create, and store a map in the SD Files. The procedure to create a new map is as follows:

Step 1

Enter the new map name and select the "New Map Creation" function on the UMAP Main Menu.

UMAP:

- Verifies the name, default field indicators, and default FDCs;
- Verifies that the map does not already exist in the SD Files or the map library;
- Displays a blank screen for you to use to define the map.

Step 2

Design the map on the terminal screen as you want it to appear in final form by inserting the appropriate field indicators immediately preceding each field.

Step 3

You can now either:

- Press the CLEAR key to return to the UMAP Main Menu without creating (saving) the map, or:
- Press ENTER if you are satisfied with the map.

  UMAP validates the information you entered. If there are no errors, the map is written into the SD files and you are passed to the UMAP Modify Map Menu.

Step 4

You can now test the new map, modify fields, or save the map. If errors exist, you are prompted with a HELP screen and the appropriate error message. To return to correct the previously entered map, press ENTER.

In this procedure, the map is written into the SD Files with the following default Terminal Control Codes:

- Reset modified data tags;
- Reset keyboard;
- Erase unprotected fields;
- Do not sound the audible alarm;
- Allow Com-plete to determine if the screen should be erased;
- Allow Com-plete to determine if the constant fields should be rewritten.

  Note that you can change these control codes by using the UMAP Main Menu function "Terminal Control Codes".

  Each field on the map is initially defined with the following:

- A blank field name;
- The user-defined screen location (x and y coordinates of the first character) and field length;
- The default FDCs listed on the UMAP Modify Map Menu.

  Note that the extended 3279 attributes for color and symbol set ID are null.

  Variable fields have the following definitions:

- The position in the buffer for the data for any given field is the sum of the length of all the previous fields in the buffer;

- Numeric variable fields in the map are represented in zoned format.

  You can assign field names, alter the FDCs, and/or change the data buffer offset using the Modify Map Menu "All Fields Update" function.

- To view the map as it will appear when displayed under program control, use the UMAP Main Menu "Show Map" function or the Modify Map Menu "Dynamic Test" function.

- To add new fields, modify existing fields, or perform other final edits, use the Modify Map Menu "Full Screen Modify" function.

Copy Map

The "Copy Map" function enables you to make a duplicate of a map. The copy procedure is as follows:

Step 1

Enter the new map name, select the "Copy Map" function on the UMAP Main Menu, and enter the name of the map you wish to copy in the "Mapname" input field.

UMAP:

- Verifies that the new map name does not already exist in the SD Files or the map library.

- Displays the map to be copied so that you can verify that you selected the right map.

Step 2

You can now either:

- Press the CLEAR key to terminate the function without copying and return to the UMAP Main Menu, or:

- Press ENTER to cause a duplicate map with the new name to be written to the SD Files and return to the UMAP Modify Map Menu.

Delete Map from SD File

The "Delete Map from SD File" function enables you to delete maps in the SD Files from within the UMAP utility. The delete procedure is as follows:

Step 1

Enter the map name and select the "Delete Map from SD File" function on the UMAP Main Menu.

UMAP will display the map on the screen so that you may verify that you selected the right map.

Step 2

You can now either:

- Press the CLEAR key to terminate the function without deleting the map and return to the UMAP Main Menu, or:
- Press ENTER to delete the map and return to the UMAP Modify Map Menu.

Note that the "Delete Map from SD File" function prevents the indiscriminate deletion of a map by requiring:

- The entry of a valid map name;
- That the TID=SHR;
- User verification of the map selected.

Note also that further security may be attained by using a user exit.

Show Map

The "Show Map" function enables you to view a formatted map with all the specified attributes and the variable alphanumeric fields displayed with the designated WITH character. The procedure is as follows:

Step 1

Enter the map name, select the "Show Map" function on the UMAP Main Menu, and enter the desired display character in the "Show with Symbol" input field.

UMAP displays the map with the specified symbol displayed in the alphanumeric variable fields and the numeric variable fields displayed with "9"s in the variable positions.

Your map will be written with a Write Mapped Conversational call using the MRCB fields with:

- Blanks in the cursor out field;
- No overriding Terminal Control Codes;
- A blank write option.

  The data buffer fields are written with:

- Zeros in the numeric fields;
- The specified symbol displayed in the alphanumeric variable fields.

Note that in order to reduce buffer storage, the preceding procedure recomputes the data buffer offsets and rearranges overlapping fields. No Field Control Table is used.

Warning:
The Terminal Control Codes are not overridden in the MRCB. If the "Do not erase screen before write" TCC option (W) is defined for the map, the formatted map is written over the UMAP Main Menu.

Terminal Control Codes

The "Terminal Control Codes" function allows you to select alternate Terminal Control Codes (TCCs) for a specified map. The procedure is as follows:

Step 1

Enter the map name and select the "Terminal Control Codes" function on the UMAP Main Menu.

UMAP will display the "TCC Update for Map" screen.

Step 2

Make the desired alterations.

UMAP will process the screen until there are no errors.

Step 3

You can now either:

- Press the CLEAR key to terminate the function without altering the TCCs and return to the UMAP Main Menu, or:
- Press ENTER to update the map and write the map to the SD files.

  The TCC update screen, shown in the following figure, displays the current TCCs and allows you to alter the codes. You can also alter the TCCs either via the TCC field in the MRCB at execution time within the application program or when you use the Modify Map Menu "Dynamic Test" function.

```
13:30:22     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                     -- TCC Update for Map  TESTF4 --                     U2ME
Yes/No  TCC                                          Defaults
 YES    E/N         Erase unprotected fields.............Yes.
 NO     A/Q         Sound audible alarm..................No.
 NO     P/S         Start the printer....................No.
 YES    K/M         Turn off the modified data tags......Yes.
 YES    R/L         Reset the keyboard...................Yes.

Select only one
 B      B  COM-PLETE determines if erase is needed before a write (default).
        W  Do not erase screen before the write.

 F      C  Always rewrite the constant fields of this Map.
        D  Do not rewrite the constant fields.
        F  COM-PLETE determines if constant fields are to be written
           (default).

Extended 3270 Attributes

        Default Color:  BL RE PI GR TU YE NE
        Default Highlighting: B-blink, V-video, X-underline.
        Default Symbol Set ID
```

The Terminal Control Codes (listed in the above in the "TCC" column) are described in the following table.

TCC	Description
A	Sounds the audible alarm.
B	Allows Com-plete to determine if the screen should be erased before the write.
C	Always formats the screen for this map with constant fields.
D	Does not format the screen. Even though a new map is requested, the format will not be written.
E	Erases unprotected fields.
F	Allows Com-plete to determine if the screen is to be formatted.
K	Turns off all modified data tags.
L	Does not reset the keyboard.
M	Does not turn off modified data tags.
N	Does not erase unprotected fields.
P	Starts the printer.
Q	Does not sound the audible alarm.
R	Resets the keyboard.
S	Does not start the printer.
W	Does not erase the screen before writing the format (constant fields).

The extended 3270 attributes, as shown at the bottom of the TCC Update screen, are as follows:

Default Color	Specifies the color of the default screen. This two-character color code can be used on all fields not overridden by a color code unless changed for the individual field by the Modify Map Menu "Modify with Prompt" function for the individual field. Valid codes are:
	BL blue RE red PI pink GR green TU turquoise YE yellow NE neutral or blank neutral
Default Highlighting	Specifies the character to be used on all fields not overridden by an FDC on a field. Valid characters are:
	B blink V video X underline or blank null
Default Symbol Set ID	Specifies a one-character symbol set ID to be used on all fields not overridden by a symbol set ID specified on a field. See the section Extended Graphics Support in *Terminal I/O Functions* in the *Com-plete Application Programming* documentation for a discussion of symbol sets.

Edit Copy Code

The "Edit Copy Code" function allows you to either produce both a copy code for the map’s data buffer and a model Field Control Table (FCT) or produce the source code of the map. The procedure is as follows:

Step 1

Enter the name of the map, select the "Edit Copy Code" function on the UMAP Main Menu and enter the member name (optional) and programming language (ASM, COBOL, or PL/I) desired for copy code/FCT, or MACRO for the source code.

UMAP displays the map on the screen so that you can verify that you selected the right map.

Step 2

You can now either:

- Press the CLEAR key to terminate the function without producing copy code/FCT or source code and return to the UMAP Main Menu, or:
- Press ENTER to verify the map.

  If there are any maps remaining in the SD Files from this session, UMAP displays the Clean Up screen.

Step 3

Enter the desired Action and press ENTER.

UMAP:

- Stores the copy code/FCT or source code in the editor work file format.
- Fetches the editor with the recover option using the library and member name specified. Note that in VSE, the SUB-LIBRARY CODE is set to A for ASM, C for COBOL, P for PL/I, and M for MACRO.

Step 4

Make any desired changes and then use the UEDIT "EDIT SAVE" command to place the text into a source library to be specified.

Notes:

1. At this point, you are operating in the UEDIT utility; that is, UEDIT has replaced UMAP as the active program in the thread. This function generates copy code for Assembler, COBOL, or PL/I. The copy code may appear very complex because any overlapping fields or fields with negative data offsets require overlayed buffer definitions.
2. Once you are back in UMAP, to "normalize" the data buffer, use the Modify Map Menu "Rearrange Data Buffer" function.

Construction of Source Variable Names

If COBOL copy code is generated, the data buffer created has a group name of the mapname concatenated with "data-buffer". The symbol used in the Write or Read Mapped Call statement is a concatenation of the mapname and MCALL-PARM. PL/I is similar to COBOL but the Assembler names are, of necessity, very short. Note that the Assembler copy code requires that you modify the first two characters of the symbolic names.

Assembler copy code is created with the assumption that the calling parameter is aligned on a fullword boundary. All filler bytes are included in the generated code with this assumption.

For COBOL and PL/I, data buffer variables are a concatenation of the map name and field names. Names of FCT entries are a concatenation of the map name, the FCT entry field, and the map's field name. The copy code generated is skeletal and may not compile because:

- Different levels of PL/I and COBOL compilers handle overlayed data storage differently.
- UMAP creates variable names from a concatenation of the map name and the field name. Invalid variable names will result if a map has variable fields without field names.
- All definitions of filler bytes use the same variable name FILLER, which is invalid in many PL/I compilers.
- Assembler names are restricted to eight characters, the labels generated by a concatenation of ( ) and the field name. No attempt is made to label the FCT table entries.
- The FCT generated is for an "E" format FCTE, that is, 13 characters each.

EDIT SD File Creation

The source is placed in an SD file that is retrievable via the editor. If your installation is one that requires users to log on, the SD file created is named with the mapname and the TID='SHR'. In installations not requiring logon, the SD file is named UEDIT and the TID is the actual Terminal ID. Note that the above naming conventions are for user information only.

Load GDDM Symbol Sets

The "Load GDDM Symbol Sets" function enables you to load the storage planes of a 3279 graphic terminal with GDDM-generated symbol sets.

Note:
This function can only be performed from 3279 graphics terminals.

If you select this function, the following screen is displayed.

```
13:31:14     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                       -- Load Programmed Symbols --

Enter:                                          Result:
- Programmed Symbol Set Name                    - Test Data
- Symbol Set Identification Character           - Storage ID Used
- Storage ID to be Loaded (optional)
- Test Data

  Symbol Set       Storage        Test                    Storage Plane
   Name        ID   Plane         Text                       Used
1.                            .......................
2.                            .......................
3.                            .......................
4.                            .......................
5.                            .......................
6.                            .......................
```

The procedure for loading the storage planes is as follows:

Step 1

Select the "Load GDDM Symbol Sets" function on the UMAP Main Menu.

UMAP displays the "Load Programmed Symbols" screen.

Step 2

Specify the symbol set names, a symbol set ID, and (optionally) the storage plane ID.

UMAP accesses the symbol sets by searching the STEPLIB libraries and then the VSAM file with DD name of ADMF in the Com-plete JCL. The same "Load Programmed Symbols" screen is rewritten using an extended FCT specifying the symbol set ID for each "Test Data" field.

Step 3

To see the symbols as defined in the symbol set, enter any keyboard characters into the "Test Data" input field.

The symbol sets loaded are now available for application programs testing.

For more information describing the use of programmed symbols and their creation, see the *IBM User's Guide for the Graphical Data Display Manager*.

UMAP Modify Map Menu

When you select the "Modify Map" function on the UMAP Main Menu and enter a mapname, the Modify Map Menu is displayed as shown in the following figure.

```
13:26:16     TID    13            COM-5.1.          User ID ADMIN     09/26/97
                          -- Map Maintenance --                           U2MH
                             -- Modify Map --
       Function            ID    PFK          Operands
------------------------   --    ---   ---------------------------------------
      Full Screen Modify   F       1   Mapname (preset)
      Modify with Prompt   M       2   Mapname (preset)
       All Fields Update   A       3   Mapname (preset)
  Variable Fields Update   V       4   Mapname (preset)
   Rearrange Data Buffer   R       5   Mapname (preset)
   Save Map into Library   S       6   Mapname (preset)
            Dynamic Test   D       7   Mapname (preset)

                     Help   ?      12   (ID)

         Select Function:      or PFK
                 Mapname:  TEST F4
                    Help:

Field Indicators     New Field Default FDC   <DNHBLVX.UTPS..RO..KM..YZ..E >
  Variable  ?          Variable Alphanumeric     D     U    O   K   Y
  Constant  +          Variable Numeric          D     U    O   K   Y
                       Constant                  D     S        K   Y
```

Use the Modify Map Menu to select a UMAP modify function (the map name will already be supplied with the input from the UMAP Main Menu) and modify the field indicators and field description codes. The UMAP Modify Map Menu contains the following input fields (indicated in the preceding figure by underscores):

- A field for entering a UMAP modify function from the list on the menu;
- The help indicator;
- Fields to define the two field indicators;
- Fields to define the default Field Description Codes (FDCs).

Field Indicators

On the Modify Map Menu, you may define the characters used to distinguish between variable and constant fields. Two field indicators are used: one to define a variable field and the other to define a constant field. Note that, when a map is displayed, the indicator immediately precedes the field that it defines.

For additional information, see the subsection *Field Indicators* in the section *UMAP Main Menu*, earlier in this section.

New Field Default Field Description Codes

The Modify Map Menu displays the default Field Description Codes (FDCs) used when a field is added to a map by use of either the UMAP Main Menu "New Map Creation" function or the Modify Map Menu "Full Screen Modify" function. Note that these codes are intended only to be a starting point for initialization.

You may change the FDCs on the menu; however, the codes are only verified when you request one of the two functions mentioned above.

Valid FDCs are listed in Field Descriptor Codes of the *Com-plete Application Programming* documentation.

UMAP Modify Map Menu Functions

You can select a function on this menu in the same way as is described for the UMAP Main Menu functions (see the earlier section, *UMAP Main Menu Functions*).

The UMAP modify functions are summarized in the following table and described in detail in the remainder of this section.

Function	Description
Full Screen Modify	Modifies the fields within a map.
Modify with Prompt	Modifies the fields within a map and updates the field attributes after each modification.
All Fields Update	Modifies the name and attributes for each field on a map.
Variable Fields Update	Modifies the name and attributes for each variable field.
Rearrange Data Buffer	Rearranges the data fields in the buffer.
Save Map into Library	Saves the map into the designated load library and deletes the SD file.
Dynamic Test	Tests the user-designed map.

Full Screen Modify

The "Full Screen Modify" function enables you to add, move, delete, copy, size, or alter the attributes of the fields within a map. The procedure to modify a map is as follows:

Step 1

Enter the name of the map to be modified and select the "Modify Map" function on the UMAP Main Menu.

UMAP prompts you with the Modify Map Menu.

Step 2

Select the "Full Screen Modify" function.

UMAP displays the selected map on the screen.

Step 3

Edit the map using the valid modify procedures described below. Note that each time you modify a field, you must press ENTER.

After each modification, UMAP displays the updated map.

Step 4

To write the map to the SD Files and return to the UMAP Modify Map Menu, press the CLEAR key once.

Step 5

To return to the UMAP Main Menu, press the CLEAR key a second time.

Note that in this procedure, you can only modify one field each time you press ENTER. Each time you press ENTER, UMAP immediately shows you the results of the modification. The cursor is moved to the indicator position of the modified field after processing. If an error occurs, UMAP displays the error message and the Full Screen Modify HELP screen.

Note that the field indicators and cursor location control the action of this function.

The valid full screen modify operations are described below.

Note:
Pressing ENTER without making a modification to a field is considered to be an error.

Adding a Field

To add a field, enter a field indicator, the desired text, and press ENTER.

Note:
UMAP assumes that you are adding a field if you haven't altered any existing field indicators.

UMAP will backscan from the cursor location until an indicator is found. If the indicator belongs to an existing field, you will be prompted with an error message and a HELP screen. If the indicator was one that you entered, UMAP will attempt to add the new field.

Note that a field with zero length or one that would cause overlapping screen fields causes an error message and displays the HELP screen.

The new field will be assigned:

- A blank field name;
- Screen location and length as defined by you;
- The default FDCs currently listed on the menu;
- The field type as defined by the indicator and the first character of the field;
- No color attribute code;
- No symbol set ID.

  Variable fields will be assigned:

- The data buffer location starting at the end of the current data buffer;
- If the first character of the field is numeric, the field is defined as zoned.

Deleting a Field

To delete a field, replace the field's indicator with a blank and press ENTER. The map will then be displayed without the field.

Copying a Field

To copy an existing field, replace the field's indicator with a "C", move the cursor to the location on the screen where you wish the new field's indicator to be, and press ENTER.

Note that if the copy operation would result in a map with overlapping screen fields, the copy is not performed and you are prompted with an error message and the HELP screen.

The copy operation adds a new field identical to the original, except the new field has:

- A blank field name;
- A different screen location.

Note that variable fields have the data buffer offset starting at the end of the current data buffer.

Moving a Field

To move a field, replace the field's indicator with an "M", move the cursor to the location on the screen where you want the field to be moved, and press ENTER.

Note that if the move operation you attempt would result in overlapping screen fields, the move is not performed and you are prompted with an error message and the HELP screen.

Note:
The move operation only alters the screen location of a field. UMAP will still be able to find the fields in the data buffer.

Sizing a Field

To alter the size of an existing field, replace the field's indicator with an "S", move the cursor to the position immediately following the new end of the field, and press ENTER.

Note that if the sizing operation you attempt would result in a map with overlapping screen fields or a zero-length field, you are prompted with an error message and the HELP screen.

Be sure to consider the following when performing a sizing operation:

- The text associated with a constant field will be the characters located between the field indicator and the cursor location with trailing blanks removed.
- For alphanumeric and zoned fields only, you must change the length of the field in the data buffer if you change the length of the field on the screen. The sizing operation does not affect the data buffer locations of other fields. Thus, lengthening a field will cause it to overlap the next field's buffer position and shortening a field means a field of filler will exist before the next field's buffer position.

Updating Attributes

To access the Attribute Update screen for a field, replace the field's indicator with a "U" and press ENTER. The Attribute Update screen for that field will then be displayed; it is described in detail later in this section.

Note that each attribute update only affects the field you're currently altering; that is, a change in one field's data buffer offset does not affect the buffer offsets of any other field.

When you press ENTER, the modifications are made and you are returned to the map display.

Modify with Prompt

The "Modify with Prompt" function is identical to the Modify Map Menu "Full Screen Modify" function except that UMAP automatically displays the Attribute Update screen after each field modification. This option is useful for updating the attributes of modified fields and adding new fields. The Attribute Update screen is described later in this section.

All Fields Update

The "All Fields Update" function enables you to verify and/or modify the attributes for every field on the map using the Attribute Update screen (described later in this section). The procedure is as follows:

Step 1

Enter the name of the map and select the "Modify Map" function on the UMAP Main Menu.

UMAP prompts you with the UMAP Modify Map Menu.

Step 2

Select the "All Fields Update" function on the Modify Map Menu.

UMAP displays the Attribute Update screen, shown in the following figure, for the first field on the map.

Step 3

Modify the attributes as desired.

Step 4

To display the attributes for the next field on the map, press ENTER.

Step 5

To write the map to the SD Files and return to the UMAP Modify Menu, press the CLEAR key.

Step 6

To return to the UMAP Main Menu, press the CLEAR key a second time.

Note that when you update attributes, it only affects the field you're currently altering, that is, changing one field's data buffer offset does not affect the buffer offset of any other field.

Attribute Update Screen

The appearance of the Attribute Update screen depends on the type of field being updated. The following figure shows the Attribute Update screen for a constant field.

```
13:32:20     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                 -- Attribute Update in Map TESTF4 --                     U2MB
                                                      Field Name .....: M04010
FDC    (select one from each group)                   Location ........( 3  2 )
 D D=display, N=nondisplay, H=high, B=blink,          Screen Length ..:    13
      V=video, L=light pen, X=underline;
 S U=unprotected, P=protected, S=skip,
      T=tab (variable only);
   O=optional, R=required (variable only);
 K M=set MDT on (variable only), K=set MDT off;       Color ..........:
 Y Y=skip may end field, Z=skip may not end field;    ( bl gr ne pi re tu ye )
   ' '=no erase, E=erase (variable only).             Symbol Set ID ..:


+JKLFSDAJKLFDS
...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....

+KAJHSDFKJHASDF
```

Common input fields on the preceding screen for constant and variable fields are:

Field Name	Displays the field name.
	Note that the field name must be either blank or a field name that is unique for this map.
Location	Displays the row and column location of the first character in the field.
Screen Length	Displays the number of characters used by the field.
FDC	Displays the Field Description Codes (FDCs) currently applied to the field.
(select one from each group)	Lists the valid FDC(s) and their meaning(s).
bottom half	Displays a window of the selected field within the map. The row of screen/tab/containing the selected field will be shown above the row scale.

The Attribute Update screen for variable fields also displays the following data elements:

Type	Displays the type of data. Valid data types are:
	A alphanumeric P numeric (packed) Z numeric (zoned) F numeric (fullword) H numeric (halfword)
	Note that alphanumeric data types can not be modified. Numeric data types, however, can be modified to other numeric data types.
Buffer Offset	Displays the data buffer offset for the field.
	Note that the offset must be a positive or negative integer. Altering a field's data buffer offset has no affect on any other field's data buffer offset. **Note:** Buffer offset modifications may cause overlapping fields and fields with negative buffer offsets.
Color	Displays the color code of this field. Valid codes are:
	bl blue re red pi pink gr green tu turquoise ye yellow ne neutral or blank neutral
Symbol Set ID	Displays the one-character symbol set ID. Valid codes are:
	blank none nonblank character greater than x'40' value

The Attribute Update screen for numeric variable fields also displays the following data elements:

Decimal Places	Displays the number of decimal places for fullword and halfword fields only.
Packed Length	Displays the internal length for packed numeric fields only.

Variable Fields Update

The "Variable Fields Update" function is identical to the previous function ("All Fields Update") except that the sequence of Attribute Update screens is limited to the variable fields for a selected map.

Rearrange Data Buffer

The "Rearrange Data Buffer" function enables you to manipulate the data buffer of a map while remaining totally independent of the screen locations and without the need to compute the data offsets. The procedure to arrange the data buffer is as follows:

Step 1

Enter the name of the desired map and select the "Modify Map" function on the UMAP Main Menu.

UMAP prompts you with the UMAP Modify Map Menu.

Step 2

Select the "Rearrange Data Buffer" function on the Modify Map Menu.

Note:
If the map has overlapping data fields or fields with negative data offsets, UMAP recomputes the data offsets and retains the previous ordering.

UMAP displays the Arrange Data Buffer screen which lists the current buffer arrangement and the arrange data buffer operations.

Step 3

You can perform any of the listed operations.

Step 4

To write the map to the SD Files and return to the Modify Map Menu, press the CLEAR key.

Step 5

To return to the UMAP Main Menu, press the CLEAR key a second time.

Note that in this procedure, if there are fields overlapping or fields that have negative data offsets as a result from modifying the map, UMAP will recompute the data buffer offsets. If overlapping fields and/or fields with negative data offsets are required, you can modify the data offsets on a field-by-field basis using the Attribute Update screen. This function enforces the normal practice of placing all data buffer locations after the calling buffer parameter.

The Arrange Data Buffer screen lists the variable data fields as they currently exist in the data buffer, giving the field name, field type, and field length. Note that the gaps between the fields of the data buffer are displayed as implicit pad fields.

The Arrange Data Buffer screen, shown in the following figure, enables you to perform six data buffer operations. To select an operation, enter any character in the field preceding the desired function. Note that an underscore in the following figure indicates a data entry field.

Note:
Most operations require the fields to have field names.

```
13:33:32     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                 -- Arrange Data Buffer for Map TESTF4 --                 U2MD

  Move        before        .                    Pad   0 bytes before       .
  Move        after         .                    Pad   0 bytes after        .
  Align binary fields.                           Remove all padding.

       A 0007
```

Using the Arrange Data Buffer screen causes UMAP to process one operation at a time. After each operation, the data buffer screen is redisplayed.

The arrange data buffer operations are described in the following table.

Operation	Instructions	Result
Move xxxxxx before yyyyyy	Enter the names of the fields to be moved.	All existing filler bytes (i.e., padding) will be retained in their respective places.
Move xxxxxx after yyyyyy	Enter the names of the fields to be moved.	All existing filler bytes(i.e., padding) will be retained in their respective places.
Align binary fields	Enter any character in the input field.	The data buffer fields will be shifted in order to ensure the alignment of binary fullword and halfword fields. Note that the number of filler bytes between fields is not reduced.
Pad nnn bytes before xxxxxx	Enter the number of filler bytes to be added to the data buffer immediately preceding the field xxxxxx.	A negative nnn will reduce an existing pad by nnn bytes. Note that other filler bytes are not affected.
Pad nnn bytes after xxxxxx	Enter the number of filler bytes to be added to the data buffer immediately following the field xxxxxx.	A negative nnn will reduce an existing pad by nnn bytes. Note that other filler bytes are not affected.
Remove all padding	Enter any character in the input field.	The data buffer offsets are recomputed so that no filler bytes remain between the data buffer fields. Note that non-aligned binary halfwords and fullwords may result.

Note:
Because there is no explicit definition of filler bytes within a map, the addition of padding will cause an error if you attempt to add filler to the end of a data buffer.

Alternative Method

There is another method to reorder the data fields in a large map in an arbitrary sequence. The procedure is listed below.

Step 1

Assign the data buffer offset for each field (all or just variable) on the Attribute Update Screen.

Step 2

Then use the Modify Map Menu "Rearrange Data Buffer" function to detect overlapping fields and recompute the data buffer offsets while retaining the desired sequence of the fields.

Save Map into Library

The "Save Map into Library" function enables you to save the map in a map library in order to:

- Enable the use of the map by an application program;
- Create a backup.

The procedure for this function is as follows:

Step 1

Enter the mapname on the UMAP Main Menu and select the "Modify Map" function.

UMAP prompts you with the Modify Map Menu.

Step 2

Select the "Save Map into Library" function on the UMAP Modify Map Menu.

UMAP displays the map on the screen so that you can verify that you selected the correct map.

Step 3

You can now either:

- Press the CLEAR key to abort the function and return to the UMAP Modify Map menu, or:
- Press ENTER to save the map in the map library.

  UMAP saves your map in the designated map library, delete the associated SD file, and return you to the UMAP Main Menu.

Dynamic Test

The "Dynamic Test" function enables you to experiment with a selected map before using the map under application program control. You can enter an MRCB, FCT, and DATA BUFFER and then see how the map would react using the WRTMC and READM functions.

The test procedure is as follows:

Step 1

Enter the map name on the UMAP Main Menu and select the "Modify Map" function.

UMAP prompts you with the Modify Map Menu.

Step 2

On the UMAP Modify Map Menu, select the "Dynamic Test" function.

UMAP retrieves the selected map and displays a sequence of the following (press ENTER repeatedly):

- The MRCB/FCT screen;
- One or more data buffer screens;
- The selected map.

Step 4

To return to the UMAP Modify Map Menu, press the CLEAR key once.

Step 5

To return to the UMAP Main Menu, press the CLEAR key a second time.

With Com-plete mapping support, your map is formatted by a Write Mapped Conversational (WRTMC) call using information from:

- The map;
- The MRCB you specified on the MRCB/FCT screen;
- The data buffer you specified on the DATA BUFFER screen(s);
- The Field Control Table you specified on the MRCB/FCT screen.

  The function reads the contents of the screen using a Mapped Read (READM), which modifies information in:

- The data buffer fields;
- The MRCB;
- The Field Control Table (FCT).

  Each screen in the test cycle is described below.

MRCB/FCT Screen

The MRCB/FCT screen, shown in the following figure, displays the MRCB and FCT to be used during a Write Mapped Conversational call and a Read Mapped call.

One field that is not part of the MRCB or FCT is the "Use FCT?" field. Entering a character in this field causes the FCT to be included in the parameter lists for the Write Mapped Conversational call and the Read Mapped call of your map. Note that a character entered in this field also causes the "FCTE Count" of the MRCB to be set to 20.

```
13:34:24     TID    13           COM-5.1.          User ID ADMIN     09/26/97
                             -- Test Map --                                U2MC
           MRCB AREA                                  FCT AREA
Mapname .............:     TEST                 Use FCT?
Map Count ...........:        0                 FCT Entries:
FCTE Count ..........:        0                 nnnnnnifffccs nnnnnnifffccs
FCTE Format..........:        E
Write Option ........:
Read Option .........:
TCC Codes ...........:
Cursor Out ..........:   MAPNAM
Cursor In ...........:
Return Code .........:        0
Enter Code ..........:       00
Fields Read .........:        1
Field Errors ........:        0
Feedback Length .....:      154                      (nnnnnn=field-name
                                                      i     =input-flag
                                                      fff   =fdc
                                                      cc    =color
Feed Back Area                                        s     =symbol-set)
```

The "FCTE Format" field allows you to specify the FCT format. The valid choices are described in the following table:

Abbreviation	Format	FCTE Length
blank or "S"	short	6
"L"	long	10
"E"	extended	13

Note that the Modify Map Menu "Dynamic Test" function does not edit the FCT format field. The contents of this field are edited by Com-plete mapping at the time of your map’s WRTMC call.

Note:
If you enter invalid data, UMAP abends the same way that any application program would abend.

The format of the FCT entries is described in Field Control Table (FCT) of the *Com-plete Application Programming* documentation.

Data Buffer Screen

The Data Buffer screen lists each variable field defined on your map in the format shown in the following figure.

```
13:35:17     TID    13           COM-5.1.          User ID ADMIN     09/26/97
DATA  RR/CC FIELD  T         VALUE                                       HEX?
OFFS         NAME  Y DISPLAY  AND  HEX
 0000 09 02        A
                   Z
```

The Data Buffer screen displays:

- The hex option field;
- The field data buffer offsets;
- The location of the field on the screen, in row and column format;
- The field name;
- The field type;
- The field data buffer areas;
- The hex representation of numeric fields.

You may enter data into the data buffer.

Note that numeric fields allow you to enter the largest value according to the internal data length.

Alphanumeric fields are truncated to forty characters. Note that, in order to reduce buffer storage requirements and remove overlapping fields, the data fields will appear in buffer sequence even though the data offsets have actually been altered.

By default, the contents of a numeric field are used in the display of the hex representation of the numeric field. When you specify the hex option, the contents of the hex representation are translated into the numeric data buffer locations for display on your mapped screen.

Warning:
The above translation is not edited for valid hex codes.

User Map

You are then presented with your formatted map written with a Write Mapped Conversational call, using the MRCB from the MRCB/FCT screen, data from the data buffer screen, and (optionally) the FCT from the MRCB/FCT screen.

You can now enter data on this screen that, in turn, will be reflected in the MRCB, data buffer, and FCT.

The cycle is repeated with the MRCB/FCT screen and the data buffer screens displaying the results of the Read Mapped call.

To terminate the Modify Map Menu "Dynamic Test" function and return to the UMAP Modify Map Menu, press the CLEAR key.

Note:
The "Dynamic Test" function does very little editing. A faulty map or MRCB that would abend a user routine also abends the "Dynamic Test" function.

UMATH - Mathematics Utility

The UMATH utility enables you to perform mathematical computations.

UMATH has two modes:

- Math (MTH) mode
- Function (FCN) mode

This chapter covers the following topics:

- Command Format
- Math Mode
- Function Mode

Command Format

The primary method for invoking UMATH is to enter the command:

```
*UMATH
```

The next screen to appear contains several lines of brief instructions on using UMATH. At this time, you are already in Math Mode, which is the default.

Note that you can switch between the two modes by entering "MTH" or "FCN" and pressing ENTER.

Math Mode

In MTH mode (the default mode), you enter an arithmetic expression using the actual numbers to be used in the calculation combined with one or more arithmetic operators. (The operators "*", "/", "+", and "-" represent multiplication, division, addition, and subtraction, respectively.) Use parentheses to specify the order of operations. Note that decimal fractions are permitted for decimal format only.

Note also that you must enter Hex numbers (always integers) as X1CD or ABC (if the number starts with a character A through F).

Example

To divide the sum of 140 + 16 + 3,228 by 4 and then subtract 7, enter the following calculation while in Math (MTH) mode:

```
((140+16+3228)/4)-7=
```

When you press ENTER, UMATH returns the answers in both decimal (DEC) and hexadecimal (HEX) format, as shown below:

```
DEC=839
HEX=0347
```

Note:
When a calculation results in a fraction, only a decimal format answer is returned.

Function Mode

To use Function Mode, enter FCN and press ENTER. The next screen will prompt you to "ENTER EXPRESSION". At this time, you can enter up to 10 one-character variables, for example, A + B/(C - D)=. When you press ENTER, UMATH instructs you to enter a value for each of the variables in the format A=xx, B=xx, C=xx, etc., where *xx* represents the actual value(s). Press ENTER again after entering the required values, and UMATH displays the result of the calculation using those values in both decimal (DEC) and hexadecimal (HEX) format (if the result is an integer value), or decimal format only (if the result is a fraction).

Example

Let’s say you’re a teacher calculating final grades for the semester for a class of 30 students. You have five grades to average: three exercises which count a total of 40%, a midterm test which counts 20%, and a final exam which counts 40%. While in Function (FCN) mode, enter:

```
(((a+b+c)/3)*.4)+(d*.2)+(e*.4)=
```

After pressing ENTER, at UMATH’s prompt, enter the first student’s grades in the sequence listed above in the format:

```
a=79,b=84,c=90,d=82,e=94
```

When you press ENTER, UMATH will return the answer:

```
DEC=87.733,333,333
```

Note that since the above calculation results in a fraction, only a decimal format answer is returned.

Press ENTER again, and UMATH prompts you again; enter the second student’s grades in the same format as for the first student.

UP - Terminal Paging Utility

The Com-plete terminal paging utility UP enables you to view data in a terminal paging data set created by an online program.

Overview

Terminal paging data sets are temporary disk-resident SD data sets, created when needed by an application program. Data sets created by an application program for this purpose are automatically deleted by Com-plete when the application program terminates and another program (or the same program) is invoked.

The application program arranges the data in a paging data set for convenient viewing at a terminal using the UP terminal paging utility. The data is organized into sections called pages. Note that with UP, you can only access this data and display the individual pages at the terminal from which the paging data set was created.

Command Format

To invoke UP, either use a terminal command or PF keys. The command format is:

```
*UP/option
```

where *option* is a one-character code indicating the UP display request to be performed. The available options are:

C	When you initially access the terminal paging file, displays the current page defined by the application program that created the file.
Â	After you initially access the terminal paging file, displays the most recently displayed page.
n	Displays the page whose page number is n. Note that n must be an integer equal to or larger than one.
N	Displays the next page.
H	Displays the highest page in the terminal paging file.
P	Displays the page before the currently displayed page.

The PF key assignments for UP are defined in the following table.

Key	Display Function	Command
PF1	Display next page	*UP/N
PF2	Display previous page	*UP/P
PF3	Display page one	*UP/1
PF4	Display last page	*UP/H
PF5	Display current page	*UP/C

Note:
The PF key assignments are valid only if the application program currently in use has not issued the SETEID function (which intercepts usage of the PF keys).

Functional Considerations

The pages in a terminal paging file are numbered sequentially beginning with one. The application program that created the file may have defined a particular page as the current page. This page is displayed at the terminal when you initially access the paging file with a current page request. Note that once another page is displayed, however, it becomes the current page.

To request display of individual pages from the paging file created by an application program, use the paging commands summarized in the table above.

Note:
These commands allow you to access only the file that the program has created for the terminal you are using. You must access page files created for other terminals from those terminals.

Note that using the UP paging command functions will not cause the application program in use to be terminated. All paging requests are serviced by the Com-plete terminal paging subtask which executes independently of the application program task. After you have made one or more paging requests, press ENTER to cause control to be returned to the application program.

While you are using the paging commands, Com-plete may issue an error message stating that you have made an invalid page request. This message usually indicates that you have requested a non-existent page number for display.

UPDS - Partitioned Data Set Maintenance Utility (z/OS only)

The Com-plete 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 Com-plete 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
ALIAS	Assigns an alias for a specified member name.
COPY	Prints the list of member names (the directory) in a partitioned data set.
DISPLAY	Displays a specific member of a partitioned data set.
EDIT	Transfers a PDS member in the Com-plete Full Screen Editor while displaying it in UPDS.
FILES	Lists the two-character library identification codes and their respective assignments as defined by the module UEDTB1.
HELP	Displays the UPDS Menu.
KEYWORDS	Displays the keywords and their current values.
LIST	Lists the member names in a partitioned data set.
PRINT	Prints the statements in a member of a partitioned data set.
RENAME	Renames a member of a partitioned data set.
SCRATCH	Scratches (deletes) a member from a partitioned data set.
SUBMIT	Submits a member from a partitioned data set to RJE.
ZAP	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 DSNAME 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 section, 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 section, 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 subsections of this section.

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 section, 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 DSNAME 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:

```
DSNAME='APPLE PIE'
```

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

```
DSNAME='JOE''S DATA SET'
```

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

```
DSNAME=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
CVOL	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 section *UDS - Data Set Maintenance Utility (z/OS Only)* for cataloging procedures.
DESTCODE	Specifies the screen-to-hardcopy device to be used for printout spooling requests.
DSNAME	Specifies the data set name of the partitioned data set to be accessed.
FORMAT	Specifies a character, hexadecimal, or interpreted dump format for the DISPLAY and PRINT commands.
LIBRARY	Specifies a two-character library identification code.
MEMBER	Specifies the member name to be used when accessing the library.
NOTE	Specifies a one- to eight-character alphanumeric tag to be used for later reference.
POINT	Specifies a tag previously defined using the NOTE keyword.
RECORD	Specifies the desired position in the displayed data set.
SCAN	Specifies the value for a scan request.
USERDATA	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.
VOLSER	Specifies the volume identification number of the disk volume to be accessed.
ZONE	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* below.

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, Com-plete 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.

<table>
<tr><td rowspan="2">FORMAT=format</td><td>Specifies the display format for the DISPLAY and PRINT commands. Valid formats are:</td></tr>
<tr><td>
<table>
<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>
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:
<table>
<tr><th>Format</th><th>Mm. Charts/Line</th><th>Bytes/Line Shown</th></tr>
<tr><td>F=HEX</td><td>125</td><td>48</td></tr>
<tr><td></td><td>77</td><td>32</td></tr>
<tr><td></td><td>40</td><td>8</td></tr>
<tr><td>F=INT</td><td>112</td><td>32</td></tr>
<tr><td></td><td>59</td><td>16</td></tr>
<tr><td></td><td>33</td><td>8</td></tr>
</table>
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.
</td></tr>
<tr><td rowspan="2">LIBRARY=libcode</td><td>Specifies the two-character library code to be used when a command for which a library code has not been entered is processed.</td></tr>
<tr><td>Note that entry of libcode initializes the keywords LIBRARY, DSNAME, 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.</td></tr>
</table>

MEMBER=member	Specifies the member name to be used when a command for which no member name has been entered is processed.
	In order to facilitate initialization of the appropriate keywords, member can also be specified in one of the following formats:
	a. dsn(member)/volume b. dsn(member)volume c. libcode(member)
NOTE=name	Specifies a one- to eight-character alphanumeric tag that allows rapid paging from one place in a member to another while displaying.
	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.
	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.
	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.
POINT=name	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.

RECORD=position	Is used to position the display of a data set. Position can be specified as one of the following:
	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.
	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.
	You can perform the positioning function in one of two ways:
	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:
	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, Com-plete 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 section.

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 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 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 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 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 Com-plete Full Screen Editor (UEDIT) 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:

```
 F
ID...DSNAME...(ON...VOLSER)
```

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*.

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 DSNAME 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 Idenfication number (TID) or a message switching destination code that identifies the terminal(s) to which a hard copy listing will be queued. **Notes:** 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.
	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.
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. **Note:** This function will not work properly if UPDS has been invoked under a user program.
CC	Specifies that ASA carriage control characters in position one of data records are to be used. **Note:** With the CC option, no other formatting is performed and no headings are provided by UPDS.
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 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 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 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 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 partioned 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 DSNAME 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 inplace.

UQ - System Job Queue Display Utility

The Com-plete online utility UQ enables you to obtain system status information.

This chapter covers the following topics:

- Overview
- Security
- Command Format
- UQ Commands
- UQ Keywords
- Entering Operator Commands (Control User)

Overview

The specific functions available with UQ are grouped into two categories:

- Display operations;
- Control operations.

Display Operations

Display operations include combinations of command functions and keyword arguments that permit you to inquire about the status of pending, active, or completed jobs. These operations include:

- Displaying job names and status information for active batch jobs;
- Displaying paging statistics if the system is a pageable system;
- Displaying job names and status information for jobs in the input/output queues;
- Displaying jobs in the input queue;
- Displaying information for jobs in the output queue;
- Displaying status information for the tape units and disk units defined;
- Displaying operator messages including those that require a reply;
- Obtaining a display of the available disk space for all available volumes.

Control Operations

Control operations include combinations of command functions and keyword arguments that enable you to control the disposition of pending, active, or completed jobs. These operations include:

- Placing jobs in hold status in order to prevent execution or printing;
- Releasing jobs from hold status;
- Canceling job output from the output queue;
- Changing the output class of specific jobs;
- Selectively routing all or portions of the output for a job to a hard copy terminal.

Note:
Because UQ can include a user-written exit routine for establishing usage conventions, some of the features described in this section may not function as described or may function in a restricted manner.

Security

Because the function of UQ enables you to both view and control system job information and job flow, four levels of security have been established in order to guarantee both user control and authorization at the terminal user level. These security levels are:

- Control;
- Com-plete initialization;
- Comment control statements;
- User-written exit.

Control

Certain functions of the UQ utility are restricted to users with control status. These functions are marked as such in this section.

Com-plete Initialization

Any user can, by default, normally view and control job output for any job submitted to the operating system for execution. This control is not restricted by terminal, user ID, or control status. Note, however, that this control can be initially restricted at the time Com-plete is initialized.

When Com-plete is initialized, the UQDEFAULT sysparm enables an installation to either:

- Permit any user to view and control all input and output information;
- Disallow all users from viewing and controlling any input or output information.

If this security restriction option is selected, users cannot view any SYSIN or SYSOUT information from the job queue unless the job stream contains UQ comment control statements.

Note:
The UQDEFAULT sysparm is described in the *Com-plete System Programming* documentation.

Comment Control Statements

UQ recognizes six comment control statements that can be used to restrict the display and control of job input and output to designated users. These comment control statements are summarized in the following figure.

To restrict job access to the specified user ID(s):

```
//*UQ USERID userid(,userid,...,userid)                  (z/OS)
* *UQ                                                   (VSE)
```

To restrict job access to user IDs having the indicated accounting information:

```
//*UQ ACCOUNT account(,account,...,account)              (z/OS)
* *UQ                                                   (VSE)
```

To restrict job access to user IDs having the indicated auhorization levels:

```
//*UQ AUTHORIZE auth#(,auth#,...,auth#)                  (z/OS)
* *UQ                                                   (VSE)
```

For no restrictions on job access:

```
//*UQ ALLOW                                              (z/OS)
* *UQ                                                   (VSE)
```

For restricting any terminal user to access the job:

```
//*UQ DISALLOW                                           (z/OS)
* *UQ                                                   (VSE)
```

Passes data to the user-written security exit and determines job access security from this data:

```
//*UQ USER ... user determined data ...                  (z/OS)
* *UQ                                                   (VSE)
```

Note that in order to be recognized by UQ, all comment control statements must be placed in the job stream after the job statement and before the first EXEC statement. In addition, UQ comment control statements will be ignored if the accounting option is not in effect and if the user does not log on to Com-plete.

The UQ keyword arguments affected by usage of the comment control statements are H, R, C, DE, and S.

If one of these keyword arguments has been initialized and the S command is in effect, the request against a job that contains no comment control statements is allowed or disallowed depending on the setting of the Com-plete sysparm UQDEFAULT. If the selected job contains one or more comment control statements, the job can be accessed by any terminal user that passes at least one condition specified in the control statements.

User-Written Exit

Some of the features discussed in this section may generate security violation messages. Except for the conditions discussed in this section, security violation messages are caused by implementation of the UQ user-written security exit. For more details on the use and implementation of the UQ security exit, consult the system programmer responsible for Com-plete maintenance.

Command Format

UQ is a fully conversational program. This means that you do not need to invoke UQ each time you request a new display. You can request a new display conversationally simply by entering a new command. For example, to display the status of the tape drives once you have invoked UQ, you need to enter only the display command "T". You can request functions either at the time you initially invoke UQ, or after invocation.

In addition to the various UQ functions, a set of keyword arguments is available that enable you to tailor the output of the various functions. You can also request these keyword arguments conversationally, either at the time you invoke UQ or after invocation.

Note:
Since each keyword argument is designed to augment the use of a specific command, the use of a keyword argument without the appropriate command has no effect on execution of other command functions.

The command format for invoking UQ is:

```
*UQ [function][,keyword=value,keyword=value,...]
```

Note that you can only request one function at any one time. You must enter the function before entering any keyword options, and separate each keyword option from the preceding option (or function) by a comma. If you request a function at the time you invoke UQ, you must separate it from the characters "UQ" by a single space.

UQ Menu

You can also invoke UQ functions via the UQ menu. To display the UQ menu, simply enter the UQ commands with no operands, as shown below:

```
*UQ
```

The following figure illustrates the UQ menu.

```
13:20:06       TID    11           COMPLETE          User SAG          03.07.03
                                                                           UQY2
                       --- System Job Queue display ---

Function ....................  Fc           Function .....................Fc
--------------------------------  --         ------------------------------  --
Active task display ...........  A           Disk Unit status display .....  D
Job queue display .............  Q           Tape allocation display ......  T
Selective Job display .........  S           Free space data (OS) .......  V
Console message display .......  M           List JES datasets (OS) .........  L
Console action display ........  O




         Please select function

         and keywords for Job/Job queue functions:

JB: COMTEST2  Job name                       SI:     ..... type of queue info
DS: 01   .... logical data set nbr           LL: 132  .... line length display
PP:      .... start display position
```

In addition to the default functions, COM-PASS allows you to set the JB, DS, LL, SI, and PP keywords as defaults. Note that, except for JB, all keywords set by default will override any user-supplied keywords. Note also that the CANCEL command is not available from the UQ menu.

UQ Commands

UQ commands are one-character function identifiers. You can execute a command either at the time you invoke UQ, or any time thereafter.

After you enter a command, the last two lines of the display will contain the status of all keyword values including OP. The keyword OP always identifies the current command that is being executed.

For example, the expression "OP=M" indicates that the currently executed command is the M command function (a request for display of all operator messages).

The commands available with UQ and the display that results from their usage are summarized in the following table. The UQ commands are described in detail in the remainder of this section.

Command	Description
A	Displays active tasks (default)
D	Displays disk allocation status
L	List JES datasets for a job (OS only)
M	Displays operator messages
O	Displays operator messages requiring a reply
Q	Displays the input/output job queue
S	Enables the selective display of input or output
T	Displays tape allocation status
V	Displays a volume free space summary (OS only)

Note that if you do not issue a command when you invoke UQ, the A command function is the default.

In addition to the commands listed, there are three control operations that you can request using the Q command function. They are summarized in the following table.

Operation	Description
C	Cancels or purges a job from the input or the output queue
H	Holds a job that is either in the input or the output queue
R	Releases a job that is in hold status

Note:
These control operations are available only after execution of the Q command and only if a unique job has been selected by using the JB keyword argument.

A Command

The A command enables you to display the active tasks in the system.

The command format is:

```
A
```

The default display generated by this command is a display of the active user partitions. Note that the display format varies slightly in each operating system environment.

z/OS/XA (JES2)

A typical display for an z/OS/XA (JES2) operating system is illustrated in the following figure.

```
ID       JOBNAME  STEPNAME PROCSTEP ST DP STORE       CPU  EXCP  ASID DM RP

S0030318 DAEFCI06 DAEFCI06 CICS     NS E3   888K   169.18  6419  00B9 00
S0030323 DAEFCIT1 DAEFCIT1 CICS     NS ED   796K   214.01  6059  00BB 00
S0030406 DAEFCQS  DAEFCQS  CQSPROC  NS FE   480K   367.62   697  00C5 00
S0030650 DAEFCI17 DAEFCI17 CIQESMF  NS ED  1068K   301.02 16437  00D7 00
S0022269 DAEFCI09 DAEFCI09 CICS     NS ED  1180K    17.82  6647  00DA 00
S0041203 DAEFCOAD DAEFCOAD IEFPROC  NS F5   584K     8.49  7892  017C 00
S0043699 DAEFCI10 DAEFCI10 CICS     NS E3  1536K   232.35 16987  01B3 00
S0052293 DAEFCI01 DAEFCI01 CICS     NS E3  1148K    88.94 14125  01E3 00
S0030892 DAEFCODE DAEFCODE SCO##034 NS F5  5072K    18.17 12827  02C3 00
S0022979 DAEFCI15 DAEFCI15 CICS     NS E3  6384K   393.90 18334  0300 00
S0030894 DAEFCO   DAEFCO   IEFPROC  NS F5 28592K   680.40   292K 032B 00
S0023507 DAEFCIA1 DAEFCIA1 CICS     NS E3  1156K   236.13 13872  0332 00
S0030896 DAEFCOTS DAEFCOTS IEFPROC  NS F5   976K    19.70  4762  033E 00
S0031260 DAEFCIA2 DAEFCIA2 CICS     NS E3  1124K     5.21  5808  0352 00
S0032261 DAEFCIUK DAEFCIUK CICSESA  NS E3 13488K     1.09  3224  0376 00
S0031699 DAEFCI11 DAEFCI11 CICS     NS E3  7900K     3.58  4953  0379 00
S0032194 DAEFCOT  DAEFCOT  IEFPROC  NS F5 12184K     6.54 16140  0390 00

--------------------- CPU %   95.25  PAGE RATE/SEC   16.50  INTERVAL  14.55 S
OP=A CL=ALL IN=ALL JB=DAEFC DS=1 SI=CC LR=1 LL=132 PP=0 DC= RL=999999
AD=S DE= SC= PT= SQ= RR=0 OC= SD=1500  20.02.2002, 14:25:17
```

Various display formats are available using the A command. The type of display information is determined by the AD keyword (see the section on keywords below).

Note that the column headings remain the same regardless of the display option selected. The column headings are explained in the following table:

Heading/Field	Description
ID	The identification number of each initiator. ID is blank for system tasks and TSO tasks.
JOBNAME	The jobname of the active task.
STEPNAME	The job step name being executed, either within the procedure or within the job if no procedure is being executed.
PROCSTEP	The name of the procedure being executed, if any.
ST	The number of subtasks associated with the indicated task or job. This count identifies the number of unique z/OS subtasks created.
DP	The dispatching priority for a job.
STORE	The amount of virtual storage assigned to the job.
CPU	The number of CPU seconds consumed.
EXCP	The I/O the job has generated.
DM	The performance group domain.
AS	The address space.
RP	The reply ID.
CPU %	The percent of time the CPU was active of the time period displayed in the INTERVAL field (see below).
PAGE RATE/SEC	The average paging rate.
INTERVAL	The interval of time, in minutes and seconds, over which the paging statistics have been accumulated.

z/OS/XA (JES3)

In z/OS/XA (JES3) environments, the active display has a slightly different format. The header format for the A display in an z/OS/XA (JES3) environment is illustrated in the following figure.

```
ID  JOBNAME  STEPNAME  PROCSTEP  STATUS  GROUP  DP  RT  LINES  STORE  CPU  RP
```

Note that in z/OS/XA (JES3) environments, the active display includes the JES3 job class group and the JES3 job number. In addition, the SYSOUT line count provided does not include any output in a "held" SYSOUT class.

The following table describes the column headings that appear on the UQ A display for the z/OS/XA (JES3) environment.

Heading	Description
ID	The identification number of each initiator. ID is blank for system tasks and TSO tasks.
JOBNAME	The jobname of the active task.
STEPNAME	The job step name being executed, either within the procedure or within the job if no procedure is being executed.
PROCSTEP	The name of the procedure being executed, if any.
STATUS	The number of subtasks associated with the indicated task or job. This count identifies the number of unique z/OS subtasks created.
GROUP	Performance group.
DP	The displatching priority for a job.
RT	Ready TCB count.
LINES	Output spool lines produced by job.
STORE	The amount of virtual storage assigned to the job.
CPU	The number of CPU seconds consumed.
RP	The reply ID.

D Command

The D command enables you to display the current status of all disk units defined to the operating system.

The command format is:

```
D
```

The display generated by the UQ D command is illustrated in the following figure.

```
UNIT   SERIES  STATUS    VOLUME  DCB   UNIT   SERIES  STATUS    VOLUME  DCB

 124    3390   OFFLINE                  125    3390   OFFLINE
 126    3390   OFFLINE                  127    3390   OFFLINE
 128    3390   OFFLINE                  129    3390   OFFLINE
 12A    3390   OFFLINE                  12B    3390   OFFLINE
 12C    3390   OFFLINE                  12D    3390   OFFLINE
 12E    3390   OFFLINE                  12F    3390   OFFLINE
 200 A  3380 RSIDNT/PRIV DBDC02  000    201 A  3380 RSIDNT/PRIV DBDC06  039
 202 A  3380 RSIDNT/PRIV DBDC07  056    203 A  3380 RSIDNT/PRIV DBDC01  089
 204 A  3380 RSIDNT/PRIV EUP001  009    205 A  3380 RSIDNT/PRIV EUP002  005
 206 A  3380 RSIDNT/PRIV EUP003  006    207 A  3380 RSIDNT/PRIV GSALL1  003
 208 A  3380 RSIDNT/PRIV GSMUE1  003    209 A  3380 RSIDNT/PRIV GSSTU1  007
 20A A  3380 RSIDNT/PRIV RSC001  002    20B A  3380 RSIDNT/PRIV ANW001  002
 20C A  3380 RSIDNT/PRIV AER001  004    20D A  3380 RSIDNT/STOR USR006  006
 20E A  3380 RSIDNT/STOR USR007  017    20F O  3380 RSIDNT/PRIV SMIG01
 210 A  3380 RSIDNT/PRIV DCE001  003    211 A  3380 RSIDNT/PRIV DCE002  010
 212 A  3380 RSIDNT/PRIV XCOL01  001    213 A  3380 RSIDNT/PRIV XKGSD1  022
 214 A  3380 RSIDNT/PRIV ADA003  028    215 A  3380 RSIDNT/PRIV ADA004  007
 216 A  3380 RSIDNT/PRIV ADAE01  010    217 A  3380 RSIDNT/PRIV ADAE02  005

--------------------------------------------------------------------------------
OP=D CL=ALL IN=ALL JB= DS=1 SI=CC LR=LAST LL=80 PP=1 DC= RL=999999 AD=J
DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 14:48:13
```

Note:
The last column of the display ("DCB") is only shown in z/OS environments.

Note that there are no options available with this command function. If, however, the number of disk units defined is large enough, the display may extend, for a 3270 display terminal, to multiple displays. In this situation, press ENTER to scroll to the next screen page.

The column headings remain the same regardless of the length of the display.

The following table describes the column headings that appear on the UQ D display for all environments.

Heading	Description	
UNIT	The CUU of the indicated disk unit.	
	The indicator in the column immediately to the right of the UNIT column specifies the current usage of the defined unit. It can have one of the following four values:	
	O	The unit is online, but not in use.
	A	The unit is online and allocated.
	B	The unit is online, allocated, and busy performing I/O.
	blank	The unit is offline.

Heading	Description	
SERIES	The type of disk unit defined. The codes used to designate unit types are:	
	2305A	2305-1
	2305B	2305-2
	2314	2314
	3310	3310
	3330	3330 model 1
	3331	3330 model 11
	3340	3340
	3350	3350
	3370	3370
	3375	3375
	3380	3380
	FBA	3310-3370 (DOS)
	none	Unknown unit type

Heading	Description	
STATUS	The mount status for the indicated unit and the disk volume on that unit. The status indicators are:	
	ONLINE	The unit is online (VSE).
	MOUNT RQD	There is a mount pending.
	OFFLINE	The unit is offline.
	NOT READY	The unit is inoperative.
	CHNG STATUS	The mount status is transient.
	RESERVD	The volume is dismountable only upon request.
	RSIDNT	The volume is non-removable.
	REMOVE	The volume can be removed dynamically.
	PRIV	It is a private volume.
	PUBL	Temporary data sets will be allocated.
	STOR	Permanent data sets with nonspecific volume references will be allocated.
	SYSRES	It is a system residence volume(VSE).
VOLUME	The volume identification name of the disk volume mounted on this unit. This entry is blank if the unit is offline.	
DCB (OS)	The number of z/OS data sets currently allocated and in use on the indicated volume.	
OWN (DOS)	The number of VSE partitions owning the indicated volume.	

L Command

The L command enables you to display all JES datasets for a given job. If the result of a previous job selection gave more than one job, the datasets for the first one found are listed. By selecting one of the datasets with the cursor, you can display this dataset.

The command format is:

L

M Command

The M command enables you to display the console operator messages.

The command format is:

M

Note that no arguments are given with this command.

z/OS/XA (JES2)

The following figure is a typical display that might result from entry of the M command in an z/OS/XA (JES2) environment. Note that console operator messages requiring a reply, if any, are included in this display.

```
*05 CAT2291D REPLY WITH VALID INQUIRY COMMAND

 $HASP100 SAGRP10C ON INTRDR      PARKINSON
 $HASP100 SAGRPB9B ON INTRDR      PARKINSON
 $HASP100 SAGRPB9C ON INTRDR      PARKINSON
 $HASP100 SAJGSNV2 ON INTRDR      STOUT
 $HASP373 STARTED - INIT  A - CLASS J - SYS IPO1
 IEF453I SAJGSNV2 - JOB FAILED - JCL ERROR
 $HASP395 ENDED
 $HASP309    INIT  A INACTIVE ******** C=AJ
 $HASP608 SAJGSNV2 PRT LOCAL      PUN LOCAL      PRIO  1 PURGE ANY
 $HASP317 SAJGSNV2 0003 DATA SETS CANCELLED
 $HASP250 SAJGSNV2 IS PURGED
 ZSF0004 - LOGON: SAGSC TID=1 ACCT=DALL AUTH=0 RMC=1,2,3,4,5,6,7
 SMC=1,2,3,4,5,6,7 NONCONTROL
 SAGNA2 - LOGON DEMO     USERID=SAGSC  TID=0001 TIBNAME=NPLT0001
 $HASP100 SAJGSNV2 ON INTRDR      STOUT
 $HASP373 STARTED - INIT  A - CLASS J - SYS IPO1
 $HASP395 ENDED
 $HASP309    INIT  a INACTIVE ******** C=AJ
-------------------------------------------------------------------------
OP=M CL=ALL IN=ALL JB=ADMIN DS=1 SI=CC LR=1 LL=80 PP=0 DC= RL=999999
AD=S DE= SC= PT= SQ= RR=0 OC= SD=1500    9/26/87, 13:46:53
```

z/OS/XA (JES3)

In z/OS/XA (JES3) environments, the M command displays the system log for the processor on which Com-plete is executing. You can, however, examine active SYSLOG jobs on any processor in the JES3 complex by using the S command.

O Command

The O command enables you to display the console operator reply messages requiring a reply.

The command format is:

O

Note that no arguments are given with this command.

The following figure is a typical display that might result from entry of the O command in an z/OS/XA (JES2) environment.

```
*05 CAT2291D REPLY WITH VALID INQUIRY COMMAND









--------------------------------------------------------------------------------
OP=O CL=ALL IN=ALL JB=ADMIN DS=1 SI=CC LR=1 LL=80 PP=0 DC= RL=999999
AD=S DE= SC= PT= SQ= RR=0 OC= SD=1500    9/26/87, 13:47:14
```

Extended Console Support (z/OS 5.1 and above, VSE/ESA 2.1 and above)

UQ provides support for Extended Console in conjunction with the CONSOLE server. This allows console display (M or O) even if the system has no defined console. For details how to install the console server see the documentation Installation and Migration section *Installing the Extended Console Server*.

The Extended Console Interface can display the most recent messages up to the number of entries specified in the SERVER sysparm. Data is scrolled using PF7/PF8. Outstanding WTORs are displayed highlighted at the bottom of the last page. Use PF10/PF11 to display the left, right or center part of the message.

Notes:

1. The Console messages and WTORs are stored in the incore table as they arrive. This means that only messages and outstanding replies that arrived after the server was activated can be displayed.
2. Function =’ ’ (blanks) has the same effect as Function=’K ’ if the Command field is not empty.
3. Operator commands *must* be entered in the *command*field. The *function* field may contain only blanks, "K" followed by a blank or "/" followed by a blank.

Notes:

1. Function =’ ’ (blanks) has the same effect as Function=’K ’ if the Command field is not empty.
2. Operator commands *must* be entered in the *command*field. The *function* field may contain only blanks, "K" followed by a blank or "/" followed by a blank.

Sample UQ M Screen (Model 2):

```
16:25:42        TID    23          COM-5.1.          User RSF1         30.08.97
                         ---   System Messages   ---                       UQM1

JOB01502  IEF404I KAS - ENDED - TIME=16.25.27
JOB01502  -KAS      ENDED.  NAME-                     TOTAL TCB CPU TIME=   .00
JOB01504  IEF403I KAS - STARTED - TIME=16.25.28
JOB01504  -                                                --TIMINGS (MINS.)--
JOB01504  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
JOB01504  -KAS      TEST                 00     29     38    .00    .00     .0
STC01303  .C      J01502
JOB01502  .HASP634 KAS      CAN NOT BE CANCELED
STC01303  .P      J01499
JOB01499  .HASP608 KAS      AWAITING PURGE            PRIO  1 PURGE ANY
JOB01503  -                                                --TIMINGS (MINS.)--
JOB01503  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
JOB01503  -ESI      ESI                  00     95    254    .06    .00     .1
JOB01503  IEF404I ESI - ENDED - TIME=16.25.35
JOB01503  -ESI      ENDED.  NAME-                     TOTAL TCB CPU TIME=   .06
STC06632 *23 DFS996I *IMS READY*  IMS4
******************************** BOTTOM OF DATA ******************************
Select Function:                                                       CONSOLE
Operand/Command:
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                                  Retrv  up    down        left  right
```

rcInput Fields/Keys:

Function: any valid UQ function

Operand/Command: *operand* for the above UQ Function or *operator command* to be sent to the console.

Special Functions/PFKeys:

ENTER	Performs the entered Function or Command . If both are empty the last messages are displayed.
CLEAR	Terminate UQ
PF6	Retrieve the (10) last entered operator commands. The command is displayed and can be modified before execution.
PF7	Scroll backward/forward within the limits of the table.
PF10/PF11	Move displayed window to left or right. For all terminal models except 5 the default part is the center. On model 5 terminals the default is the left part of the line.

All other PFKeys will honor the definitions made for UQ in UUTIL - user and global PFKey definitions. Note that some definitions may not be applicable thus not causing any action on the UQ M screen.

Notes:

1. Function =’ ’ (blanks) has the same effect as Function=’K ’ if the Command field is not empty.
2. Operator commands *must* be entered in the *command*field. The *function* field may contain only blanks, "K" followed by a blank or "/" followed by a blank.

Q Command

The Q command enables you to request a display of the jobs residing in the input/output queue.

The command format is:

Q

The default display generated by this command is a display of the jobs in the input/output queue. The following figure illustrates a typical Q display for an z/OS/XA (JES2) or VS1 system.

```
 JOB     NUMBER   Q CLS DEST    PRI  ST      LINES    FORM    FCB  FLASH
CHECKSTC S.3551  *X  -           15              0
CSCNUC   S.4443  *X  -           15              0
CSCCOM46 S.4444  *X  -           15              0
COMTEST  S.4497  *X  -           15              0
COMISE   S.2289  *X  -           15              0
COMTEST1 S.3373  *X  -           15              0
COK463   S.3503  *X  -           15              0
CSCNET54 S.3549  *X  -           15              0
COKSCAN  J.1837   O  X            9            111    STD
COKSCAN  J.1843   O  X            9             97    STD
CNEALOD  J.1916   O  X            9            159    STD
CNEALOD  J.1922   O  X            9            229    STD
CSISRH   J.2017   O  X            9            413    STD
COKCOPY  J.1754   O  X            9             95    STD
CNEEXP   J.2870   O  X            9            128    STD
COKASMA  J.3346   O  X            9            597    STD
COK463   S.3496   O  X            9            416    STD
CSISRH   J.2019   O  X            8            620    STD
CSISRH   J.3269   O  X            8            614    STD
------------------------------------ JOBS      28  TOTAL RECORDS       14,547
OP=Q CL=ALL IN=ALL JB=C DS=1 SI=CC LR=LAST LL=80 PP=100 DC= RL=999999
AD=J DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 15:05:04
```

Various display formats are available with the Q command. The display format is determined by both the operating system environment and the keywords CL, DE, IN, JB, JC, JO, OC, and SQ. The display format can also be determined by the UQ user-written exit.

Column headings vary depending on the operating system environment.

z/OS/XA (JES2)

The following table describes the column headings that appear on the UQ Q display for the z/OS/XA (JES2) environment.

Heading	Description
JOB	The names of the jobs on the queue.
NUMBER	The job numbers assigned to specific jobs by the spooling system (HASP, or JES).
Q	The type of queue in which a job resides.
	For non-HASP systems, one of the following two values will appear:
	I The job is in the input queue. O The job is in the output queue. For HASP systems, one of the following five values will appear: * The task is active. I The job is in the input queue. X The job is executing. O The job is in the output queue. P A purge request is pending for the job.
CLS	The class of queue input or output in which the job resides.
DEST	The destination code of the job. The codes displayed are those codes defined to the spooling system and normally indicate that the job either originated from or was destined for an RJE station.
	In HASP systems, if the job task is active on an I/O device, the device name is also displayed.
PRI	The priority of the indicated job within the indicated job class queue.
ST	The status of the indicated job. One of the following two values may appear:
	H The job is in hold status. blank The job is not in hold status.
LINES	Number of lines generated by the job (not including lines in output datasets, which are logically deleted).
FORM	Name of the JES2 form associated with the output.
FCB	Name of the JES2 forms control buffer associated with the output.
FLASH	Name of the JES2 flash associated with the output.

z/OS/XA (JES3)

The header format for a Q display in an z/OS/XA (JES3) environment is illustrated in the following figure.

```
OB-NAME JOB# Q CLASS PRI ST ORIGIN
```

The following table describes the column headings that appear on the UQ Q display for z/OS/XA (JES3).

Heading	Description
JOB-NAME	The name(s) of the job(s) on the queue.
JOB#	The JES3 job number.
Q	The JES3 function that the selected job is currently executing. One of the following three functions may be indicated:
	I Input service M Main service O Output service
CLASS	The JES3 job class.
PRI	The JES3 priority.
	Note that PRI can have a value ranging from 0 to 15.
ST	The JES3 job class status. ST can be one of the following:
	H The job is in hold. A The job is currently active on the function indicated by Q. Q The job is currently available for the functionindicated by Q. W The job is currently waiting for the function indicated by Q.
ORIGIN	The JES3 job origin (ORG).

VSE (POWER)

The header format for a Q display in a VSE environment is illustrated in the following figure.

```
JOB NUMBER Q CL PR DI RECORDS DEST SID
```

The following table describes the column headings that appear on the UQ Q display for VE:.

Heading	Description
JOB	The name(s) of the job(s) on the queue.
NUMBER	The POWER job number.
Q	The type of queue in which a job resides:
	R The job is in the reader queue. L The job is in the list queue. P The job is in the punch queue.
CL	The POWER job class.
PR	The POWER priority.
DI	The POWER job disposition. DI may be one of the following:
	L Disposition "Leave". H Disposition "Hold". D Disposition "Delete". * Job is marked active.
RECORDS	The number of data records contained in a job.
DEST	The POWER remote output destination.
SID	The POWER system ID (only applicable if shared spooling is used.

S Command

The S command enables you to request a display of a specific job as it exists either in the input or output queues. It is used in conjunction with the JB keyword.

The following table summarizes the z/OS and VSE equivalents of the information that may be displayed using the S command.

z/OS	VSE
Input queue jobs:	Reader queue jobs:
a. JCL	a. Job control
b. SYSIN data	b. SYSIPT data
Output queue jobs:	List or Punch queue jobs:
a. SMB information	a. SYSLST information
b. Job step information	b. SYSPUN information
c. SYSOUT information	

The information to be displayed is determined by use of the keyword argument SI; it can be displayed in one of several formats. The format is determined by use of the keyword arguments DS, JB, LR, LL, PP, RL, SC, and SI. Other keywords used in conjunction with the S command are DC, DE, PT, and RR.

The command format for the S command is:

```
S,JB=identifier
```

JB=identifier	Identifies the specific job to be selected for display.
	Note that identifier can be either the jobname or the job number (as determined by HASP, POWER, or JES).
	Jobnames are not necessarily unique but job numbers are. identifier , however, must uniquely identify the job to be selected.

The default display generated by this command is determined by the initialized value of the SI keyword argument. If you do not initialize this keyword argument, it defaults to SI=SM for z/OS and LS for VSE.

The output displayed is the SMB/SYSLST information.

The following figure illustrates a typical display that might occur as a result of using the default values of the S command in an z/OS/XA (JES2) environment.

In a VSE environment it is not possible to display jobs that are marked active (DISP = * in Q-display).

```
       2 //STARTING EXEC XCOM148
       4 XXXCOM148  EXEC PGM=NPRINIT,REGION=8M,TIME=1440
IEF287I   MM.SYSF.LISTING                              NOT CATLGD  2
IEF287I   VOL SER NOS= ADAE01.
IEF287I   MM.SYSF.LISTING                              NOT CATLGD  2
IEF287I   VOL SER NOS= ADAE02.
IEF287I   ADABAS.SYSF.QAS.RESULT.KM02.V620             NOT CATLGD  2
IEF287I   VOL SER NOS= WRK001.
IEF287I   ADABAS.SYSF.QAS.RESULT.KM02.V620             NOT CATLGD  2
IEF287I   VOL SER NOS= WRK001.
IEF287I   ADABAS.SYSF.QAS.RESULT.KM02.V620             NOT CATLGD  2
IEF287I   VOL SER NOS= WRK001.
IEF287I   ADABAS.SYSF.QAS.RESULT.KM02.V620             NOT CATLGD  2
IEF287I   VOL SER NOS= WRK001.
IEF287I   ADABAS.SYSF.QAS.RESULT.KM02.V620             NOT CATLGD  2
IEF287I   VOL SER NOS= WRK001.
IEF287I   PUX.UNXTRANS.ALF150                          NOT CATLGD  2




-------------------------------------------------------------------------------
OP=S CL=ALL IN=ALL JB=XCOM148,S.3374 DS=1 SI=CC LR=1 LL=80 PP=1 DC= RL=999999
AD=J DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 15:13:23
```

Note that the output displayed is arranged in data sets, each of which is displayed by use of the DS keyword.

T Command

The T command enables you to request a display of the status of the tape units defined to the operating system.

The command format is:

T

No keyword arguments affect the output displayed by this command.

The default display generated by this command is a display of all the tape units defined to the operating system and the mount status associated with them. For a display type terminal, if the number of units exceeds the capacity of one screen display, you can display the additional tape units by pressing ENTER.

The following figure illustrates a typical display that might occur when the T command is entered in an z/OS/XA (JES2) environment.

```
UNIT SERIES BPI       STATUS    VOLUME

 810  3480  38000     OFFLINE
 811  3480  38000     ALLOCATED 004988
 812  3480  38000     OFFLINE
 813  3480  38000     OFFLINE
 814  3480  38000     ONLINE
 815  3480  38000     OFFLINE
 816  3480  38000     OFFLINE
 817  3480  38000     OFFLINE
 818  3480  38000     OFFLINE
 819  3480  38000     OFFLINE
 81A  3480  38000     OFFLINE
 81B  3480  38000     OFFLINE
 81C  3480  38000     OFFLINE
 81D  3480  38000     OFFLINE
 81E  3480  38000     OFFLINE
 81F  3480  38000     OFFLINE
 880  3400  1600/6250 ONLINE
 890  3490  38000     OFFLINE
 8A0  3400  1600/6250 OFFLINE
-------------------------------------------------------------------------------
OP=T CL=ALL IN=ALL JB= DS=1 SI=CC LR=1 LL=80 PP=1 DC= RL=999999 AD=J
DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 15:15:05
```

V Command (z/OS Only)

The V command enables you to request a display that summarizes the free space available on all direct access volumes currently mounted and online to the operating system.

The command format is:

V

No keyword arguments affect the output of the resulting display.

The display generated by this command is a display of direct access devices online to the operating system and a summary of the free space for each volume as shown in the following figure. Note that a maximum of 50 volumes is displayed.

The following figure illustrates a typical display that might occur when the V command is entered in an z/OS/XA (JES2) environment.

```
                                TOTAL    TOTAL    TOTAL        LARGEST CONTIG
                                FREE     FREE     FREE             FREE
VOLUME  CUU DEVTYP  STATUS      CYLS     TRACKS   EXTNTS       CYLS, TRACKS

EDU107  10B 3390    ONLINE      1951        5        3         1528       0
DBDC02  200 3380    ONLINE       907        9        4          907       0
DBDC06  201 3380    ONLINE         0        0        0            0       0
DBDC07  202 3380    ONLINE        49      173       68           13       0
DBDC01  203 3380    ONLINE        19      306       65            2       0
EUP001  204 3380    ONLINE       786      165       89           81       0
EUP002  205 3380    ONLINE       529      294       57          208       3
EUP003  206 3380    ONLINE       326      575       97           34       0
GSALL1  207 3380    ONLINE       514      447      140          132       0
GSMUE1  208 3380    ONLINE      1206       51       14          878       0
GSSTU1  209 3380    ONLINE       290       86       25          140       0
RSC001  20A 3380    ONLINE      1692       47       11          735       0
ANW001  20B 3380    ONLINE       279        5        4          144       0
AER001  20C 3380    ONLINE        51       66       23           11       0
USR006  20D 3380    ONLINE        31     1296      320            1      21
USR007  20E 3380    ONLINE        87     1134      249            6       3

--------------------------------------------------------------------------------
OP=V CL=ALL IN=ALL JB= DS=1 SI=CC LR=1 LL=80 PP=1 DC= RL=999999 AD=J
DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 15:16:26
```

If the number of volumes to be displayed exceeds the capacity of one display for a display type terminal, press ENTER to continue the display.

The column headings illustrated in the UQ V display are described in the following table.

Heading	Description
VOLUME	The volume identification name of the DASD volume mounted on this unit.
CUU	The CUU of the unit.
DEVTYP	The device type of the unit.
STATUS	The mounted status of the DASD volume. Normally, this entry is online.
TOTAL FREE CYLS	The sum of the count of all free cylinders on the indicated volume.
TOTAL FREE TRACKS	The sum of the count of all free track extents on the indicated volume.
TOTAL FREE EXTNTS	The sum of the count of all free extents for the indicated volume.
LARGEST CYLS/TRKS	The largest free extent, per volume, given for both cylinders and tracks.

Note that the volume display is obtained for a five-minute interval only. The beginning of this interval is given at the bottom right of the display.

Note:
If you repeat the V command within this five-minute interval, the display is repeated. It will not reflect any data set allocations that may have occurred since the start of this interval.

UQ Keywords

The UQ keyword is used to tailor the output of a specific UQ command function or to direct hard copy routing of portions of a display.

Although some command functions are unaffected by keyword usage, you can enter any keyword at any time while UQ is active, and it will subsequently be in effect when a command is executed that uses one or more of them.

This section gives a detailed description of all keywords and the effect they have on specific command functions. Sample displays are given to illustrate the effects of specific keywords on output displays.

Keyword Values

You can enter keyword values, either when you invoke UQ or after invocation; however, using a specific keyword will have no apparent effect if the command function with which you enter it is not applicable to that keyword. Keyword initialization occurs at the point that a function is issued that uses that keyword. In addition, the use of some keyword values may cause other keyword values to be modified. When you use the keywords later, their output results will reflect the most recent modifications made, either by you or other keyword values.

The keywords and their general application are summarized in the following table. The keywords are designed to be used with specific commands. The commands associated with each keyword are also indicated. Keywords that have related effects when used with other keywords are also listed.

Keyword	Description	Used With Command	Related Keywords
AD	Sets the display option.	A	
CL	Sets which job classes are displayed.	Q	all systems except z/OS JES3
		S	(z/OS JES3 only)
DC	Sets the destination code for hard copy output.	S	DS, RL, PP, LL, LR, SD
DE	Sets the destination code for job output being sent to a RJE station.	S,C	JB
DS	Sets which logical data set within a unique job is to be displayed.	S	DC, RL, SI
IN	Sets whether input or output jobs will be displayed.	Q	

Keyword	Description	Used With Command	Related Keywords
JB	Sets which job(s) will be displayed.	S, L, Q	
JC (z/OS/XA[JES3] only)	Sets a JES3 job class.	Q	
JO (z/OS/XA [JES3] only)	Sets a JES3 job origin.	Q	
LL	Sets the line length to be used for output display of a specific queued job.	S	
LR	Sets the line number to be used for output display of a specific queued job.	S	DS
NO	[note]		
OC (z/OS)	Sets the output class of the job being displayed to the specific output class.	Q	JB
OP		All (used for information purposes only)	
PO	[point]		
PP	Sets the print position to be used for output of a specific queued job.	S	
PT	Sets the destination code for hard copy output.	S	
RL	Sets the maximum number of statements to be sent to another terminal.	S	DC
RR	Specifies the frequency with which UQ will write displayed data to the terminal.	all	
SC	Specifies a character string to be searched for using the S command.	S	
SD	Sets the number of lines that will be written to a UEDIT work file.	S	DC
SI	Specifies the type of queue information to be displayed.	S	

Keyword	Description	Used With Command	Related Keywords
SQ	Specifies the sorting sequence for the job queue display.	Q	
WR (z/OS)	Changes the WRITER name of the job being displayed to the specific name.	Q	JB

The status of all keyword values is maintained by UQ; their initialized values are displayed in the bottom two lines of every command function display.

Default values are established for most keywords. To view the default values for any keyword at any time, enter the keyword, followed by an equal sign, but do not specify a value. The SC keyword is an exception; details about its use are given later in this section. A more detailed description of UQ keywords follows below.

AD Keyword

The AD keyword affects the output display of the A command.

The format is:

`AD=[value]`

value	Specifies the display option to be used when the A command function is executed. Note that the value specified is dependent upon the operating system environment.	
	In z/OS environments, one or more of the following three values can be specified:	
	J	The display is to be restricted to user tasks.
	T	The display is to be restricted to TSO users.
	S	The display is to be restricted to system tasks.
		z/OS Default: AD=J
	In VSE, the only option is:	
	U	The display is to be restricted to user tasks.

Example:

The following figure illustrates a typical display that might occur when the the AD keyword is used with a value of S (in a z/OS environment) and the A command function is subsequently requested.

```
ID     JOBNAME  STEPNAME PROCSTEP ST DP STORE       CPU  EXCP  LINES  ASID DM RP

S03032 TMON8DLS TMON8DLS TMON8DLS NS EF   748K   575.13 25550    N/A  003E 00
S03042 TMDBDLS  TMDBDLS  TMDBDLS  NS EF   356K    38.28  9984    N/A  003F 00
S03120 NUC177   NUC177   NUC177   NS F3  2152K  6818.66  1938K   N/A  0040 00
S03077 TIMER    TIMER    TIMER    OT FF   176K    79.20  2549    N/A  0041 00
S03124 NATEDPM2 NATEDPM2 GLOBAL   OT FF   156K    48.64     3    N/A  0043 00
S03122 NUC11177 NUC11177 NUC11177 NS F3   368K   107.78 31064    N/A  0044 00
S03528 NUC010   NUC010   NUC010   NS F3  4416K 10614.39  3696K   N/A  0045 00
S03343 DB23DIST DB23DIST IEFPROC  NS EF   224K     4.28   340    N/A  0046 00
S03130 NATEDPM3 NATEDPM3 GLOBAL   OT FF   144K    46.37     8    N/A  0047 00
S03286 IRL4PROC IRL4PROC          NS EF   220K    80.72    90    N/A  0048 00
S03128 NATGBPM  NATGBPM  GLOBAL   OT FF   140K    46.43     2    N/A  0049 00
S03132 NATGBPM2 NATGBPM2 GLOBAL   OT FF   148K    48.57     3    N/A  004A 00
S03167 NATGBPM3 NATGBPM3 GLOBAL   OT FF   144K    46.49     7    N/A  004B 00
S03169 NATSWPMH NATSWPMH GLOBAL   OT FF   152K    48.60     3    N/A  004C 00
S03196 TSO      TSO      TSO      OT FF   332K     9.63  1075    N/A  004D 00
S03190 XCOM145  XCOM145  XCOM145  NS F3  4384K 17866.57  1537K   N/A  004E 00
S03292 DB23DBM1 DB23DBM1 IEFPROC  NS EF  3652K  6523.80  2123    N/A  004F 00
S03231 DB23MSTR DB23MSTR IEFPROC  NS EF   340K   548.11  6141    N/A  0050 00
--------------------- CPU %   46.00  PAGE RATE/SEC    0.71  INTERVAL   1.42 S
OP=A CL=ALL IN=ALL JB= DS=1 SI=CC LR=1 LL=80 PP=1 DC= RL=999999 AD=S
DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 15:19:29
```

CL Keyword

In all systems except z/OS/XA (JES3), the CL keyword affects the output display of the Q command. This keyword determines which job classes are displayed.

In z/OS/XA (JES3) systems, the CL keyword is used in conjunction with the Q command when SI=SO in order to limit the display to specific SYSOUT class(es).

The format is:

`CL=[value]`

value	Specifies the job classes for which the display is to be restricted. One of two formats can be specified: ALL — The display is to include all job classes. ab...n — The display is to be restricted to one or more job classes. For example, CL=A restricts the display to jobs in class A; CL=ABC restricts the display to jobs in job queue classes A, B, and C. Default: If the Q command dor S command in z/OS/XA (JES3) systemsf is requested and no value is specified for the CL keyword argument, the default is ALL.

The following figure illustrates a typical display that might occur when the CL keyword is used with a value of A and the Q command is subsequently requested.

```
 JOB     NUMBER   Q CLS DEST    PRI  ST      LINES    FORM    FCB  FLASH
OPFWDMS2 J.3151  *X  A           10              0
SSFDEV   J.8752   O  A  AHU       9              2    STAN
USARMU   J.8882   O  A  USARMU    9              2    STAN
 ... 45 PERCENT SPOOL UTILIZATION.


------------------------------------ JOBS       3  TOTAL RECORDS           4
OP=Q CL=A IN=ALL JB= DS=1 SI=SO LR=1 LL=80 PP=1 DC= RL=999999 AD=S
DE= SC= PT= SQ= RR=0 OC= SD=1500  11.11.1997, 15:24:05
```

DC Keyword

The DC keyword is used only in conjunction with the S command. It sets the destination code of the destination terminal(s) to be used when data is sent to another terminal. The data sent is from the currently displayed SYSIN/SYSIPT or SYSOUT/SYSLST for a specific job.

The format is:

`DC=destination`

where destination is either any valid Terminal Identification number (TID) or any valid message switching code.

When you enter destination as an asterisk (DC=*), the output is directed to a work file. You can then use the RECOVER function in order to process the work file under UEDIT.

The DC keyword is used primarily to obtain hard copy output of data from a job queued either in the input or output queue. The amount of data to be sent is determined by the settings of the DS, RL, PP, LL, LR, and SD keywords.

The DS keyword determines which logical output data set is to be included. Since job queue data, whether on the input or output queue, is arranged in logical data sets by Com-plete, the only data that will be sent by using the DC keyword is data in the logical data set determined by the DS keyword. For example, if DS=5 is in effect, only data from logical data set 5 will be sent.

The RL keyword determines the number of statements to be sent. For example, if RL=28 is in effect, only 28 lines of data will be sent.

The PP keyword determines the beginning position of each statement from which the data will be generated. The PP keyword determines the beginning column or print position within each statement being displayed. Note that this same print position will be used in all DC output.

Note:
If the spool data set includes carriage control characters, they will be transmitted and used only if PP=0.

The LL keyword determines the length of each line to be displayed. When the DC keyword is used to send the data to another location, the same line length will be used.

The LR keyword determines the logical record or statement number of the first line to be included in the output data.

The SD keyword determines the size (number of lines) of the UEDIT work file that will be used when "DC=*" is specified.

DE Keyword

The DE keyword is used with either the S or Q command when a unique job is being displayed. It sets the destination code of the destination RJE station to be used when job output is sent to an RJE station. The data being sent is data from the currently displayed SYSOUT/SYSLST for a specific job.

The format is:

```
DE=destination
```

or:

```
DE=node.destination
```

where *destination* is any valid RJE destination code as defined to the spooling system (HASP, JES, or POWER), possibly on a different node.

The DE keyword is used primarily to route all output to a line printer, either central or remote. Note that a Com-plete terminal cannot be designated as an RJE terminal.

The amount of data to be routed is the entire job. Use of the DE keyword submits a standard operating system route statement to the operating system. Any and all standards in effect for a given spooling system (HASP, JES, or POWER) are in effect when the route statement is executed. For example, if the operating system in use is z/OS, a route statement routes all jobs with the same name. These jobs are then executed on a priority basis (that is, a job in execution is routed upon completion; the jobs in the output queue is not routed).

Note that even though multiple jobs with the same job name may exist in the output queue, a unique display must be obtained by using the JB keyword before a request using the DE keyword is accepted.

DS Keyword

The DS keyword affects the output display of the S command. This keyword determines which logical data set within a unique job (whether on the input or output queue) is to be displayed. Note that the DC and RL keywords are restricted by the use of the DS keyword.

The format is:

```
DS=[number]
```

number	Identifies the relative number of the logical data set to be displayed.
	Note that number can be specified as a "-" or "+" integer. If "+" or "-" is specified immediately before the integer, the data set displayed is the logical data set plus or minus that amount relative to the data set currently displayed.

z/OS

In z/OS, when you use UQ to view job statement information from the input or output queue, Com-plete arranges the data in logical data sets. These data sets are determined as follows:

Input queue logical data sets:

- Job stream JCL
- Each SYSIN data set

VSE

In VSE, when you use UQ to view job statement information from the List or Punch queue, Com-plete arranges the data from duplicate job names/job numbers in logical files. These files are determined by the order in the List or Punch queue.

Output queue logical data sets:

- JOB statement SMB messages
- EXEC statement SMB JCL
- Each SYSOUT data set

The end of a logical data set forces the DS keyword to be incremented by one, thus enabling the continued display of data from one data set to the next. In order to force the display to switch from one data set to the next, however, you must initialize the DS keyword. If the DS keyword is not initialized at the time the S command is requested or if the DS keyword is initialized with a null value, the default is DS=1. Note that the DS keyword can be initialized at any time.

The SI keyword determines which logical data set can be displayed. Note that using SI without specifying the DS keyword forces the DS keyword value to be initialized to one.

IN Keyword

The IN keyword affects the output display of the Q command. This keyword determines whether input or output jobs are displayed.

The format is:

```
IN=[value]
```

value	Specifies the restrictions to be imposed upon the display given as a result of execution of the Q command. One of the following nine formats may be specified:	
	ALL	The display is to include all jobs, whether in the input queue or the output queue.
	I	The display is to be restricted to jobs in the input queue.
	O	The display is to be restricted to jobs in the output queue.
	H	The display is to be restricted to jobs that are in hold status and includes jobs from both the input and output queues.
	R	The display is to be restricted to jobs that are not in hold status and includes jobs from both the input and output queues.
	IH	The display is to be restricted to jobs both in the input queue and in hold status.
	OH	The display is to be restricted to jobs both in the output queue and in hold status.
	IR	The display is to be restricted to jobs both in the input queue and not in hold status.
	OR	The display is to be restricted to jobs both in the output queue and not in hold status.

Note that the display resulting from executing the Q command with the IN keyword specified is the same type of display as that given when the IN keyword is not initialized. The only difference is the restrictions imposed by the specification of IN.

If you specify the IN keyword with a null value, the default is IN=ALL.

JB Keyword

The JB keyword is used with both the S command and the Q command. This keyword determines which job(s) are displayed.

The format is:

`JB=[value]`

value	Specifies the jobname prefix, jobname, or job number to which the display is to be restricted. The acceptable formats for value are:
	a. A one- to eight-character value to be used as a jobname prefix. Note that all jobs with this character string as a prefix to their jobnames will be included in the display. **Note:** This option cannot be used in conjunction with the S command. b. A one- to eight-character value to be used as a jobname. Note that use of this option assumes that only one job having this unique name is in the queue. If more than one job exists with the same name, this option is treated the same as the prefix option (see above). c. The unique job number as assigned by the spooling system (HASP, JES, or POWER). Since all job numbers are unique, this option guarantees a unique job selection. The job selected will have a job number relative to the current job number established for JB.

If you enter the Q command without specifying a value for the JB keyword, the default display is all jobs in the queue. If you request the S command without specifying a value for the JB keyword, an error will result.

You can reset the default display of queued jobs produced by the Q command to display all jobs in the queue by entering the JB keyword with a null value.

JC Keyword z/OS/XA (JES3) Only

The JC keyword, valid for z/OS/XA (JES3) systems only, is used in conjunction with the Q command.

The format is:

```
JC=[number]
```

number	Default: All jobs in the queue will be displayed.
	Specifies a JES3 job class. All jobs assigned to the specified job class will be included in the display.
	Note that number is a one- to eight-character value.

JO Keyword z/OS/XA (JES3) Only

The JO keyword, valid for z/OS/XA (JES3) systems only, is used in conjunction with the Q command.

The format is:

JO=[number]

number	Default: All jobs in the queue will be displayed.
	Specifies a JES3 job origin. All jobs originating at the specified origin will be included in the display.
	Note that number is a one- to eight-character value.

LL Keyword

The LL keyword is used only in conjunction with the S command. This keyword determines the line length to be used when information about a specific job either on the input or the output queue is on display.

The format is:

LL=[number]

number	Specifies the line length to be used when statements are displayed from either SYSIN/SYSIPT or SYSOUT/SYSLST. All displays begin with the column position indicated by the PP keyword argument.
	Note that number can optionally be entered with a "+" or "-" sign. In this form, the value of number is added or subtracted, respectively, from the initialized value of LL in order to obtain a new LL value.

If you enter the S command without specifying a value for the LL keyword, the default is LL=133. If you enter the LL keyword with a null value, the default is LL=80.

If you select a line length for a display that is longer than the line length of the terminal you are using, the line length continues on the following line. If you select a line length that is shorter than the line length of the terminal, the line length is displayed in its entirety on one line.

LR Keyword

The LR keyword is used only in conjunction with the S command. This keyword determines the line number to be used when you’re displaying information about a specific job either on the input or the output queue.

The format is:

LR=[number]

number	Specifies the relative line number or displacement from the current line number to be used when statements are displayed from either SYSIN/SYSIPT or SYSOUT/SYSLST.
	Note that number can optionally be entered with a "+" or "-" sign. In this situation, the statement to be displayed is the statement relative to the current statement by a value of +n or -n, respectively.

Note:
If the value you request exceeds the maximum line count for the logical data set being displayed, an error will result. To cause the display to resume, reenter a correct value.

You can optionally specify the LR keyword in the format:

```
LR=LAST[-n]
```

This format specifies that the last line in the logical data set is to be displayed. The optional -n indicates that the statement that is *n*lines before the last line is to be displayed.

If you enter the S command without specifying a value for the LR keyword, the default is LR=1. If you enter the LR keyword with a null value, the default is also LR=1. Note that whenever you change or initialize the DS keyword, LR=1 becomes effective.

OC Keyword

The OC keyword is used only in conjunction with the Q command after a unique job has been displayed. It sets the output class of the job being displayed to the specified output class.

The format is:

```
OC=class
```

where *class* is any valid output class code as defined to the spooling system (HASP, JES, or POWER).

The OC keyword is used primarily to change all output for a unique job to a designated output class. For example, if class D is reserved for a dummy output writer, entering OC=D will route output to this class and effectively result in deletion of the output from the queue.

Note:
In z/OS/XA (JES3) systems, you can only use the OC keyword to change the SYSOUT class from the JES3 "HOLD-Q" to the "WRITER-Q".

Even though multiple jobs with the same job name exist in the output queue, you must obtain a unique display using the JB keyword before the OC keyword is accepted.

OP Keyword

The use of OP is for information purposes only. Each time a UQ command is issued, the OP keyword is initialized. The initialized value always contains the UQ command code for the command currently in effect.

If you attempt to initialize this keyword, the following error message will appear:

```
UQI0002 - INVALID COMMAND
```

PP Keyword

The PP keyword is used only in conjunction with the S command. This keyword determines the print position to be used when you’re displaying information about a specific job either on the input or the output queue.

The format is:

PP=[number]

number	Specifies the print position or relative print position to be used when statements are displayed from either SYSIN/SYSIPT or SYSOUT/SYSLST.
	Note that number can optionally be entered with a "+" or "-" sign. The display then begins in the column plus or minus, respectively, the number of positions from the current value contained in the PP keyword.

If you enter the S command without specifying a value for the PP keyword argument, the default is PP=0. If you enter the PP keyword with a null value, the default is also PP=0.

Note:
You must use the PP=0 value when printout/spooling data sets with embedded carriage control characters (when you want carriage control to occur).

PT Keyword

The PT keyword is used only in conjunction with the S command. It sets the destination code of the destination terminal(s) to be used when you ere sending data to another terminal. The data that is sent is the entire output of the currently displayed job. Carriage control characters in output data sets are executed.

The format is:

PT=destination

where destination is any valid Terminal Identification number (TID) or any valid message switching code. The entire SYSOUT/SYSLST data set including SMB information is sent to the destination terminal.

Note that other initialized keyword values such as PP, LR, LL, etc., have no effect on the format of the output generated by the PT keyword.

RL Keyword

The RL keyword is used only with the S command, in conjunction with the DC keyword. It enables you to limit the number of statements to be sent to the destination terminal(s).

The format is:

RL=[number]

number	Specifies the maximum number of statements to be sent to another terminal(s).
	Note that number can optionally be entered with a "+" or "-" sign. In this format, the maximum number of statements to be sent is determined by adding or subtracting, respectively, n to the current value of RL.

If you specify a number that is less than the number of statements in the logical data set, the number of statements sent is determined by number. If number is more than the number of statements in the logical data set, the number of statements sent is all statements beginning with the statement determined by the

LR keyword.

If you do not initialize the RL keyword argument, the default is RL=999999. If you initialize the RL keyword value to a null value, the default is RL=1000.

Note:
For DC=*, the number of lines written to the SD file is limited not only by the value of RL, but also by the value of the SD keyword.

RR Keyword

Although the RR keyword is not restricted to any use with one UQ command, it is restricted to use with local and hard copy terminals. This keyword determines the refresh rate or write rate with which UQ display information will be written to the terminal.

The format is:

RR=[number]

number	Default: 0 seconds
	Specifies the frequency, in seconds, with which UQ will write displayed data to the terminal.
	Note that number must be an integer from 0 to 32,767.

If you enter a value for the RR keyword argument, the data displayed will be the same as that which you could obtain by pressing ENTER. If an active task display is the current display at the time you initialize RR, the active task display will automatically be refreshed when the specified number of seconds has elapsed. If you enter an S command before initializing RR, the output of the S command will continue from page to page.

The RR keyword invokes a timed rollout function. To terminate the output of RR, cause an attention interrupt from the terminal and then reset the RR keyword to 0.

SC Keyword

The SC keyword is used only in conjunction with the S command. This keyword enables you to specify a character string for which the S command is to search.

The format is:

```
SC=[string] [,cond]
   [,direction]
   [,begin]
   [,end]
```

string	Default: The initialized value of SC is used.
	Specifies the character string for which the search is to proceed. Note that string may be from 1 to 16 characters. **Note:** If the specified string contains embedded blanks or commas, it must be enclosed in single quotation marks. Embedded quotation marks (apostrophies) must be entered as two single quotation marks only if the character string is enclosed in quotation marks.
cond	Default: EQ
	Specifies the condition under which the search is to be performed. One of six values can be specified:
	EQ — An equal condition. NE — A not equal condition. LT — A less than condition. GT — A greater than condition. LE — A less than or equal condition. GE — A greater than or equal condition.
direction	Default: +
	Specifies the direction in which the search is to proceed. One of two values may be specified:
	+ — A forward search. - — A backward search.
begin	Default: Zero
	Specifies the location within the record, relative to zero, at which the search is to begin.
	If no ending value is specified (see below), a beginning value is required.
end	Default: The value specified for the beginning location.
	Specifies the ending location within the record, relative to zero, at which the search is to terminate. **Note:** Any ending location specified must be greater than the specified beginning location.
	Note that if both the beginning and ending locations are omitted, the value 240 is used for the end location.

When you use the SC keyword, a SCAN function is invoked. The scan proceeds, statement by statement, according to the conditions specified with the keyword values. If a match is found in a statement, that statement is displayed as the first output statement. If no match is found, an end-of-data set condition will occur.

SD Keyword

The SD keyword is used only in conjunction with the S command and when the DC keyword is specified as an asterisk (i.e., DC=*). This keyword determines the number of lines that will be written to a UEDIT work file.

The format is:

`SD=number`

number	Default: The system programmer’s setting of the SDSIZE sysparm.
	Specifies the number of lines in the UEDIT work file.
	Note that number must be an integer from 0 to 32767.

SI Keyword

The SI keyword is used only in conjunction with the S command. This keyword determines the type of job queue information to be displayed.

The format is:

`SI=value`

value	Specifies the type of information to be displayed from the input/output job queue. One of the following values can be specified:	
	For z/OS:	
	JL	The job to be displayed is on the input queue and a display of the JCL is requested.
	SM	The job to be displayed is on the output queue and a display of the SMB information is requested.
	SO	The job to be displayed is on the output queue and a display of the SYSOUT is requested.
	SI	The job to be displayed is on the input queue and a display of the SYSIN is requested.
	CC	The job to be displayed is on the output queue and a display of the condition codes is requested.
	For VSE:	
	LS	The job to be displayed is on the power List queue.
	PU	The job to be displayed is on the power Punch queue.
	RD	The job to be displayed is on the Reader queue.

If you omit the SI keyword argument when you enter the S command, the default is SI=SM in z/OS, or SI=LS in VSE.

SQ Keyword

The SQ keyword is used only in conjunction with the Q command and is valid for all systems except VS1. This keyword argument is used to sort the job queue display into a desired sequence.

The format is:

```
SQ=[option]
```

option	Specifies the sorting sequence into which the job queue display is to be sorted. One of five values can be specified:	
	T	The display is sorted by the type of job queue entry (input, output).
	C	The display is sorted by the class of the job queue entry (for example, job class, SYSOUT/SYSLST class).
	P	The display is sorted by the priority of the job queue entry.
	N	The display is sorted by the jobname of the job queue entry.
	#	The display is sorted by the job number of the job queue entry.

If you do not initialize the SQ keyword, the default job queue display will have no sorting in effect.

WR Keyword

The WR keyword is used only in conjunction with the Q command after a unique job has been displayed. It sets the writer name of the job being displayed to the specified name.

The format is:

```
WR=name
```

where *name* is any valid writer name (JES).

Note:
Even though multiple jobs with the same job name exist in the output queue, you must obtain a unique display using the JB keyword before the WR keyword is accepted.

Entering Operator Commands (Control User)

The UQ K command allows the entry of operator commands.

Any valid operator command may be issued from a Com-plete terminal if the terminal user has control status. The command format for performing this function is:

```
*UQ K command
```

where command is any valid operator command.

Note:
In VSE systems, the K command is only valid if asynchronous operator communications are in effect (that is, if ASYNOC=YES was specified in the VSE FOPT supervisor macro).

Example

The operator system's DISPLAY ACTIVE command can be invoked by entering the command while invoking UQ:

```
*UQ K D A
```

Or the command may be entered after invoking UQ:

```
K D A
```

After the entry of an operator command, the UQ utility displays the operator WTO messages at the Com-plete terminal.

USERV - Library Service Utility (VSE Only)

The Com-plete online utility USERV enables you to display library directories and members, labels, and VTOCs.

This chapter covers the following topics:

- Overview
- Command Format
- Using USERV
- Commands

Overview

Specifically, USERV allows you to:

- List the directory of a library that is managed by the VSE Librarian (either in VSAM space or of SD format);
- Display a module contained in a library managed by the VSE Librarian that is of standard type (i.e., Phase, Obj, Dump, Proc, Source);
- Display the two-character library identification codes and the associated VSE file names defined by the UEDTB1 module;
- List the VTOC of a specified disk CUU;
- jList the free space on a specified disk CUU;
- List the partition, temporary, and standard labels defined on the VSE label cylinder.

Note:
There is a user-written exit routine (UUSVX1) provided that enables an installation to restrict usage of certain functions of USERV.

Command Format

USERV is a fully conversational program. This means that you can select the various functions available either when you invoke USERV, or any time thereafter:

- By displaying the USERV command function menu, or:
- By entering a command function at the time of invocation

The basic command format is:

***USERV**

This displays the USERV menu, which summarizes the commands recognized by USERV. It is shown in the following figure.

```
 10:38:28     TID     9          COMTEST2         User ID COK          02/12/02
                      -- Data Set Maintenance --                           USRV
     Function            ID     PFK                 Operands
 ----------------------  --     ---   ---------------------------------------
   List Library Members   L       1   LIB or DSN , SUBLIB, and VOLUME
 Display Library Members  D       2   LIB,MEM,MTYPE or DSN,SUB,MEM,MTYPE,VOL
   Print Library Member   P       3   LIB or DSN,SUB,MEM(,VOL)(,DEST)(,CC)(,A)
                   Vtoc   V       4   VOLUME
                  Space   S       5   VOLUME
   File ID's (LIB ID's)   F       6
         Help Functions   H       7
        Keyword Display   K       8
     Label Area Display   B       9   Type and Partition/Class (unless STD)

        Select Function:      or PFK     and Operands:
                LIBrary:                SUBLIBrary:
                   DSN
            MEMBER name:                Member TYPE:
                 VOLUME:                Enter Volume Serial Number
        Partition/Class:                Enter Partition ID or Class
                   Type:                Enter STD, PARSTD, TEMP or CLASS
           Destination :                Enter Print Destination
      CC and/or Attach :                Enter 'CC', 'A' OR 'CC,A'
```

These menus display the USERV 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.

The valid USERV commands are summarized in the following table. The shortest possible abbreviations are indicated by underlining.

Command	Description
DISPLAY	Displays a specific module, book, or procedure of a library.
FILES	Lists the two-character library identification codes and their respective assignments as defined by module UEDTB1.
HELP	Displays the USERV menu.
KEYWORDS	Displays the keywords and their current values.
LABEL	Lists the temporary, partition standard, and standard labels on the VSE label cylinder.
LIST	Lists the directory of a specified library.
PRINT	Printout spools the statements in a module, book, or procedure of a library.
SPACE	Lists the free space of a specified CUU.
VTOC	Lists the volume table of contents of a specified CUU.

You can also enter a command when you invoke USERV using a direct call and so bypass the USERV Menu.

The direct command format is:

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

where:

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

This feature allows you to enter a series of commands that use common keyword values, thus eliminating your needing to reinitialize keywords as you enter each command.

The details of using the keyword parameters are given in the remaining text of this section.

Using USERV

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

For example, you could use the LIST command to list the books in a specific library by entering the library ID as a positional argument with the LIST command. Once you have obtained the display, entering another library name for the FILENAME keyword argument causes the LIST command to be executed for the new library.

Positional Arguments

Positional arguments, if used, must always immediately follow the command function and precede any keyword arguments. You can separate the command and the first positional argument with either a blank or a comma. Note that in all illustrations in the USERV menu and in this section, a blank will be used as a separator.

If you enter more than one positional argument, you can separate them with either a comma or a blank. Note that in all illustrations in the USERV menu and in this section, a comma is used as a separator.

Entering a positional argument always causes one or more keyword arguments to be initialized. If you enter a command that includes positional arguments, USERV 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.

Keyword Arguments

You must specify keyword arguments in one of three ways:

- The keyword argument only, with no command;
- As the only argument given with a command;
- After all positional arguments.

Keyword arguments must be separated from commands, positional arguments, and other keyword arguments by a comma or a blank. Note that in all illustrations in this section, a comma is used as the separator.

The keywords defined for USERV are summarized in the following table. The shortest possible abbreviations are indicated by underlining.

Keyword	Description
DESTCODE	Specifies the destination routing code or TID to be used for all USERV PRINT commands.
DSNAME	Specifies the file name of the library to be accessed. (VSE file ID up to 44 characters).
FORMAT	Specifies a character, hexadecimal, or interpreted dump format for the DISPLAY and PRINT commands.
LIBRARY	Specifies a two-character library identification code.
MEMBER	Specifies the member (module, book, or procedure) name to be used when the library is accessed.
MTYPE	Specifies the member-type (i.e., Phase, Obj, Proc, Dump, etc.) to be used when the library is accessed.
PART	Specifies the partition (BG, F1 through FB) desired when either the temporary, partition or class standard labels are displayed.
RECORD	Specifies the desired position in the displayed library.
SUBLIB	Specifies the sublibrary name to be used in conjunction with the LIST or DISPLAY command.
TYPE	Specifies the type of VSE labels to be displayed.
UNIT	Specifies the CUU of the disk volume to be used for the VTOC and SPACE commands.
USERDATA	Specifies whether or not the optional user data or stow data is to be displayed after execution of the LIST command. Also indicates the format of the resulting display.
VOLSER	Specifies the VOLSER to be used by the VTOC and space functions of USERV.

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 section.

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

1. USERV command processing always occurs in the following sequence:
   - Initializes keyword arguments;
   - Reinitializes the keyword arguments based upon the positional arguments given;
   - Executes the command.
2. Once you have successfully or unsuccessfully executed a command, you can use the KEYWORDS command to display all 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 last-executed command is then re-executed using the new arguments 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.
4. Initialized keyword values will remain in effect from command to command as long as:
   - USERV 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.

If you enter more than one keyword in a single command, they are processed from left to right. Consequently, if two or more keywords conflict in that they force initialization of a common keyword argument, the *last keyword entered* determines the initialized value that will be in effect (unless it is overridden by a positional argument).

Keyword initialization considerations are described in detail in the following table:

DESTCODE=destcode	Specifies the destination routing code or TID to be used for all USERV PRINT commands.
	Default: The screen-to-hardcopy terminal defined for the terminal in use in the TIBTAB module, if any.
DSNAME=filename	Specifies the VSE file name to be used when processing all commands for which a 44-character file name is not supplied.
	Note: Entry of this keyword argument or entry of a unique file name causes the LIBRARY keyword to be initialized to blanks.

<table>
<tr><td>FORMAT=format</td><td>Specifies the dump format for the DISPLAY and PRINT commands. Valid dump formats are:

CHAR — Character
HEX — Hexadecimal
INT — Interpreted

Note that for both the HEX and INT format options, one of three line formats are selected depending on line size. The formats have been designed around the standard line sizes of 40, 80, or 132 characters. The selection algorithm, however, uses the minimum line sizes needed for each format, as shown in the following table:

<table>
<tr><th>Format</th><th>Mm. Charts/Line</th><th>Bytes/LIne Shown</th></tr>
<tr><td>F=HEX</td><td>125</td><td>48</td></tr>
<tr><td></td><td>77</td><td>32</td></tr>
<tr><td></td><td>40</td><td>8</td></tr>
<tr><td>F=INT</td><td>112</td><td>32</td></tr>
<tr><td></td><td>59</td><td>16</td></tr>
<tr><td></td><td>33</td><td>8</td></tr>
</table>

Note:
If less than 33 or 40 characters per line are available, the respective option cannot be performed and the keyword value is reset to CHAR.</td></tr>
<tr><td>LIBRARY=libcode</td><td>Specifies the two-character library code to be used when all commands for which a library code has not been entered are processed.</td></tr>
</table>

MEMBER=member	Specifies the member name to be used when commands for which no member name is entered are processed.
	member must be an eight-character module name.
	Note that in order to facilitate initialization of the appropriate keywords, member can also be specified with filename or libcode in the following formats:
	a. filename(member) b. libcode(member)
	A sublibrary name and member-type may also be necessary.
MTYPE=type	Specifies the type of the member to be selected.
	Note that type can be from one to eight characters.
PART=partition	Specifies the partition ID desired (BG, F1 through FB) when TEMP (User), PARSTD (Partition Standard) or CLASS (Class Standard) labels are displayed. This keyword is associated with the LABEL command.

RECORD=position	Is used to position the display of a book or procedure, but not the relocatable or core image library module/phases.

Note that position can be specified as one of the following:

n	Positions to the n th record of the module/book/procedure.
+n	Positions forward by n records.
-n	Positions backward by n records.
++	Positions to the last record of the module/book/procedure.
--	Positions to the first record of the module/book/procedure.
*	Redisplays the current record from the beginning.

The number of the current record is displayed in the heading of the DISPLAY display and/or in the KEYWORDS display. Note that if the request causes the record number to go below 1 or above 9999999, these values are substituted respectively. In addition, the substitution takes place as soon as the discovery is made that the requested record number is larger than the number of records in the book or procedure.

The positioning function may be accomplished in one of three ways:

a. By including the RECORD keyword in the command string.

   Example:

   Note://If a member other than the currently open one is accessed, the RECORD value is reset to 1 unless it is specified in the same command line as the display request (as shown above).

b. By entering the RECORD keyword values without the preceding keyword, if no operation or positional operand is to be entered. Examples:

```
+10
-10
++
--
*
```

c. By using PF keys to enter positioning requests. (See the section *PF Key Assignments* later in this section for a description of the functions assigned to the PF keys by the USERV utility.)

SUBLIB=name	Specifies the name of the sublibrary to be used in conjunction with the LIST or DISPLAY commands.
	Note that name can be from one to eight characters.
TYPE=type	Specifies the type of VSE labels that are to be displayed. This keyword is associated with the LABEL command.
	STD — Standard label track TEMP — User label track PARSTD — Partition Standard track CLASS — Class Standard track
UNIT=cuu	Specifies the CUU of the disk volume to be used for the VTOC and SPACE commands.
USERDATA=indicator	Default: USERDATA=NO
	Specifies whether or not user stow information is to be displayed when directory information is requested.
	Note that the value specified can be either YES or NO.
	USERDATA=YES indicates that directory information is included in the display. If no directory information is present, only member names are displayed.
	USERDATA=NO indicates that directory information is not displayed.
VOLSER=vvvvvv	Specifies the VOLSER to be used for the VTOC and SPACE functions of USERV.

Paging Requests

If you have requested a large amount of information to be output from a hard copy terminal, and you want to cause the output to be interrupted, use an Attention Interrupt. At this point, to cause one more line of output to be generated, simply press ENTER, and you will be prompted to enter a new command.

Note:
If you are using a 3270-type terminal or compatible device, to continue the display of output that cannot be contained in one display, press ENTER.

PF Key Assignments

The PF1 through PF8 keys have been assigned scrolling functions that enable you to adjust the current display. When you press one of these keys, the equivalent operand is displayed on the command line, and the function is performed. Note that you can also position within a module/book/procedure by using the RECORD keyword.

In addition to the scrolling functions, other frequently-used functions have been assigned to PF keys 9 through 12. A 40-byte command area is assigned to each of these PF keys so that functions other than those assigned can be zapped in.

The following table lists the functions assigned to the PF keys by the USERV utility.

Key	Equivalen Command	Function
PF1	-20	Scroll one full screen backward, if recsize < 81 bytes.
PF2	1	Position to the start of the member.
PF3	++	Position to the last line of the member.
PF4	+1	Advance the display one line.
PF5	-1	Move the display backward one line.
PF6	+10	Scroll one-half screen forward.
PF7	-10	Scroll one-half screen backward.
PF8	*	Position back to the beginning of the current record.
PF9	*	Position back to the beginning of the current record.
PF10	-	*UEDIT xx (mmmmmmmmmm)1
PF11	-	*UEDIT xx (mmmmmmmmmm)1
PF12	-	*UEDIT xx (mmmmmmmmmm)1

1 where: *xx* =library ID and *mmmmmmmmmm*=sublibrary.book

Note that UEDIT requests are ignored unless you specify both library ID and member.

Commands

The USERV commands are discussed in the remainder of this section.

DISPLAY Command

The DISPLAY command allows you to obtain a listing of a module of any standard type that is contained in a library.

The command format is:

```
DISPLAY [member]
```

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

(.member,type)	Specifies the one- to eight-character member name to be used in conjunction with the one- to eight-character member type to be displayed. **Note:** With this format, the SUBLIB and either the LIBRARY or DSNAME keywords must be initialized.
(sublib.member,type)	Specifies the one- to eight-character sublibrary name and the one- to eight-character member name to be used in conjunction with the one- to eight-character member type to be displayed. **Note:** With this format, either the LIBRARY or DSNAME keyword must be initialized.
library (sublib.member,type)	Specifies either a fully qualified library name or a two-character library code defined in UEDTB1 or in UUTIL. Here, member is the one- to ten-character member name to be displayed.
	If member is omitted, the enclosing parentheses can also be included, or the member format is assumed.

Note that in each of the formats described, if you omit *member*, the keyword argument MEMBER determines the member displayed.

The following figure is an example of a USERV display of a core image library member.

```
 F=I
SUBLIB=SP2LIB   MEMBER=TLSRESTA                                    REC=00000001
DSN=VSE.SP.COMPLETE.SEP24.LIB                     LIB=CC VOL=      UNIT=00C

0000 47F0F010 E3D3E2D9 C5E2E3C1 F4F4F040 *.00.TLSRESTA440 *
0010 18CF5820 C0B05810 00144800 102E4A00 *.....0.. .. ..  *
0020 105A5000 41285810 0080D200 4128105B *.!& .... .K ...$*
0030 411040C0 1B000A28 4110C0AC 91801002 *.. .. ......J...*
0040 4710C046 0A075800 C0B41814 4B10C0BC *....... .4......*

0050 BE07C05D 5010C060 4510C064 40000000 *...)&..-....    *
0060 00000000 18010A25 91040038 4710C074 *    ....J. .....*
0070 94FB4000 D3004001 22AAD207 40704000 *M.  L  ...K. .  *
0080 D23F4158 40084110 C0BE0A35 44002030 *K... ........ ..*
0090 10000001 00061F00 5810C0B8 41F0000F *.  . .. ...8.0 .*

00A0 0A6B980F 41588200 40700000 00000000 *.,Q...B  .      *
00B0 00000000 00000000 00000001 00500000 *           . &  *
00C0 00000000 00000000 00000000 00000000 *                *
00D0 00000000 00000000 00000000 00000000 *                *
00E0 00000000 00000000 00000000 00000000 *                *
```

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 entered must be keyword arguments. Entering them will cause only initialization of the appropriate keyword functions; it has no effect on the output of the FILES display.

The resulting display consists of a listing of the various libraries defined by UEDTB1. For each library listed, the two-character library identification code is given, along with the VSE file name and the library type in the following format:

The following figure shows a USERV files display for VSE/SP2.

```
 FILES
ID...DSNAME.....................................SUBLIB   TYP
CC - VSE.SP.COMPLETE.SEP15                     - SP2LIB   NV
$$ - VSE.SP.COMPLETE.SEP15                     - SP2LIB   NV
SL - ???????????                               - SYSLIB   NV
P1 - ???????????                               - BASE     NV
P2 - ???????????                               - PROD     NV
```

On this screen, "TYP" can be either:

VS	indicating that the library is in VSAM space.
NV	indicating a non-VSAM library.

The "SUBLIB" column contains the default sublibrary name that is used if this argument is omitted.

HELP Command

The HELP command enables you to display a description of each of the functions available from the USERV menu along with a listing of its associated operands.

The command format is:

HELP

You can optionally enter the HELP command as:

?

Entry of either format causes the following screen to appear.

```
 HELP
OP.........ARGUMENTS...................MEANING.............
L IST      FILENAME(MEMBER)            LIST LIBRARY MEMBER NAMES
D ISPLAY   FILENAME(MEMBER)            DISPLAY LIBRARY MEMBER
V TOC      VOLSER/UNIT                 DISPLAY DISK VOLUME CONTENTS
S PACE     VOLSER/UNIT                 SHOW DISK VOLUME AVAILABLE SPACE
LA BEL     TYPE,PART                   DISPLAY DOS/VSE LABEL AREA
P RINT     DESTCODE,'ATTACH'           HARDCOPY OF CURRENT MEMBER
F ILES                                 LIST LIBRARY ID DEFINITIONS
H ELP                                  DISPLAY (THIS) COMMAND SUMMARY
K EYWORDS                              DISPLAY KEYWORDS & THEIR CURRENT VALUES
A LIBRARY ID MAY BE SUBSTITUTED FOR A FILENAME
ENTER KEYWORD PARAMETERS AFTER COMMANDS OR SEPARATELY
ENTER 'L ' TO RESTART LIB MEMBER LIST AT BEGINNING
'ATTACH' WILL SPOOL PRINTOUT ASYNCHRONOUSLY
PRINTOUT SPOOLING MAY BE ABORTED BY AN ATTENTION INTERRUPT
FUNCTION KEYS: PF1..8 = PAGING                          PA2 = UCOPY
               PF10.= SUBMIT .. PF11 AND 12 = UEDIT
```

Any additional arguments entered with the HELP command must be keyword arguments. If you include keyword arguments with the HELP command, it will cause only the initialization of the appropriate keyword entries, which you can subsequently use by entering a command function with no arguments. To view the new initialized keyword values, use the keyword display (discussed in the following section).

KEYWORDS Command

The KEYWORDS command allows 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.

```
 KEYWORDS
KEYWORD PARAMETER CURRENT VALUES:
L IBRARY  =
D SNAME   =
S UBLIB   = (VSE/SP2 ONLY)
M EMBER   =
MT YPE    = (VSE/SP2 ONLY)
R ECORD   = 00000001
F ORMAT   = CHAR   (CHAR,HEX,INT)
U SERDATA = NO
UN IT     =
V OLSER   =
T YPE     = STD
P ART     =
DE STCODE = 10
```

For additional information on the KEYWORDS command, see the section *Using USERV*, earlier in this section.

LABEL Command

The LABEL command enables you to display the temporary, partition standard, or standard labels on the VSE label cylinder.

The command format is:

```
LABEL [type][,partition]
```

The arguments are defined below:

type	Specifies the type of VSE label desired. The type argument can be one of the following:	
	STD	Standard Label track
	TEMP	User Label track
	PARSTD	Partition Standard label track
	CLASS	Class Standard label track
partition	Specifies the partition desired when temporary or partition standard labels are displayed.	
	Note that partition can be one of the following:	
	Fn	The desired partition number as generated in the VSE supervisor, where n must be a hexadecimal value from 1 to b.
	BG	The background partition or a dynamic class (if type=CLASS) or the identifier of an active dynamic partition

The following figure is an example of a display of the standard labels on a VSE label cylinder.

```
 LABEL STD,BG
                     SYSRES VOL=DOSRES        UNIT=150
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
// DLBL IJSYSRS,'PROD.CORE.IMAGE.LIBRARY.A',99/365,SD
// EXTENT SYSRES,DOSRES,1,0,00001,01139                      CCCHH=00001,03729

// DLBL PRDCLB,'PROD.CORE.IMAGE.LIBRARY.B',99/365,SD
// EXTENT ,DOSLIB

// DLBL PRDRLA,'PROD.RELO.LIBRARY.A',99/365,SD
// EXTENT ,DOSLIB

// DLBL PRDRLB,'PROD.RELO.LIBRARY.B',99/365,SD
// EXTENT ,DOSLIB

// DLBL PRDSLA,'PROD.SOURCE.LIBRARY.A',99/365,SD
// EXTENT ,DOSLIB

// DLBL PRDSLB,'PROD.SOURCE.LIBRARY.B',99/365,SD
// EXTENT ,DOSLIB

// DLBL IJQFILE,'POWER.QUEUE.FILE',99/365,DA
// EXTENT SYS001,DOSPWR,1,0,00001,00059                      CCCHH=00001,00129
```

LIST Command

The LIST command allows you to obtain a listing of the library directory of a specified library.

The command format is:

`LIST [member]`

member can be in one of the following formats:

library(*)	Specifies the 2-character library code or the 1- to 44-character field as defined for the desired library in UEDTB1 or via UUTIL.
	(*) indicates that a list of all sublibraries that make up that library is to be displayed.
	Note that the parentheses surrounding "*" are required with this format.
library	Specifies the 2-character library code or the 1- to 44-character field as defined for the desired library in UEDTB1 or via UUTIL.
	All members in all sublibraries are to be listed (if no default sublibrary is associated with the specified library); otherwise, specifies that only members of the default sublibrary are to be listed.
library1-(sublib.member,type)	library specifies the 2-character library code or the to 44-character field as defined for the desired library in UEDTB1 or via UUTIL.
	sublib specifies the required one- to eight-character sublibrary name.
	member specifies the optional one- to eight-character member name.
	type (optional) specifies any valid one- to eight-character member type.
	Only the members with names and types having the prefixes member and type, respectively, are to be listed.

If *member* is omitted, the keyword argument MEMBER determines the first member to be displayed.

Note:
In each of the VSE formats described above, the two-character library code can be substituted by a fully qualified library name (file-ID) and a volume name in order for this command to execute.

The following two figures show examples of the USERV LIST command.

```
 L 62
DSN=COM.RLSE.IV462.VESA13.LIBRARY                LIB=62 VOL=ESARM3  UNIT=A40
SUBLIB=COM462          1 SUBLIBS
*A       *     *A       *     *A       *     *A       *     *A       *
  ACSDEFLT       CCFBPT         CCTR3270       CMFBPM         CMPRFPFE
  ACSSTART       CCFLAGS        CCUAB          CMFLAGS        CMRSRM
  ACSTAB         CCGLOBS        CCUIT          CMGROUP        CMSACB
  ACSTABLE       CCGRP          CCUPCB         CMIREG         CMSEC
  ACSTAB46       CCLIB          CM$CALL        CMIVAL         CMSETOP
  ADABAS         CCLOADP        CM$ERMAC       CMIVALM        CMSPCB
  ADALNK4C       CCMRCB         CM$INNRA       CMLAB          CMSPPM
  CAPLAB         CCOSEQU        CM$WTO         CMLDRVW        CMSRVD
  CAPTUR         CCREGS         CM$WTOR        CMLHEX         CMSVC
  CAPTURE        CCSCA          CMAMODE        CMLIB          CMTIB
  CCAC           CCSPSA         CMBTAB         CMMSGCB        CMTITLE
  CCCAPT         CCSTCK         CMCPRNT        CMNAME         CMTRACE
  CCCGLOB        CCTCB          CMDEVS         CMNOTE         CMTRACE1
  CCCGSET        CCTHCB         CMDROP         CMOSIM         CMTRACE2
  CCCOMREG       CCTHTAB        CMEDCB         CMOSTYPE       CMTST
  CCCOMSEC       CCTIB          CMEDDEFT       CMPAR          CMTYPE
  CCCSCB         CCTIOT         CMEDLDIR       CMPDSD         CMULST
  CCCSMF         CCTOP          CMEDTB1        CMPPGMT        CMUPFK
  CCDLM          CCTRACE        CMENTRY        CMPRF          CMUSE
  CCEDGBL        CCTR2741       CMEXIT         CMPRFALL       CMUSTK
```

```
 U=Y
DSN=COM.RLSE.IV462.VESA13.LIBRARY                LIB=62 VOL=ESARM3  UNIT=A40
SUBLIB=COM462          1 SUBLIBS
NAME      TYP       CREATED   UPDATED   BLOCKS  RECORDS/BYTES A-MOD R-MOD
ACSDEFLT  A         95-09-27  95-09-27  6         132      R
ACSSTART  A         95-09-05    -  -    1         18       R
ACSTAB    A         95-09-05    -  -    1         13       R
ACSTABLE  A         95-09-05    -  -    5         103      R
ACSTAB46  A         95-09-27    -  -    1         25       R
ADABAS    A         95-09-05    -  -    1         5        R
ADALNK4C  A         95-09-05    -  -    15        349      R
CAPLAB    A         95-09-05    -  -    2         45       R
CAPTUR    A         95-09-05    -  -    1         10       R
CAPTURE   A         95-09-05    -  -    3         55       R
CCAC      A         95-09-05    -  -    17        283      R
CCCAPT    A         95-09-05    -  -    6         103      R
CCCGLOB   A         95-09-05    -  -    4         81       R
CCCGSET   A         95-09-05    -  -    3         64       R
CCCOMREG  A         95-09-05    -  -    1         12       R
CCCOMSEC  A         95-09-05    -  -    16        388      R
CCCSCB    A         95-09-05    -  -    1         20       R
CCCSMF    A         95-09-05    -  -    8         149      R
CCDLM     A         95-09-05    -  -    1         15       R
CCEDGBL   A         95-09-05    -  -    1         14       R
```

The following are examples of use of the LIST command:

1. *USERV L P2(*)

   Display all sublibraries in the VSE.PRD2 library.

2. *USERV L VSE.COM.LIBRARY(,A)/SP2RES

   Display all type A members in the default sublibrary in the Com-plete distribution library.

PRINT Command

The PRINT command enables you to obtain a hard copy listing of a member in a library.

The command format is:

```
PRINT [destcode][ATTACH][CC]
```

The optional arguments are defined below:

destcode	Specifies either a Terminal Idenfication number (TID) or a message switching destination code that identifies the terminal(s) to which a hard copy listing will be queued. **Note:** destcode must always be the first positional operand.
	Default: The value determined by the keyword function DESTCODE will determine 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 (i.e., no default hard copy is assigned), the calling terminal is assumed as default.
ATTACH	Specifies that asynchronous spooling of long printouts is allowed. The program attached is an identical copy of USERV with the name taken from TMGETNAM.
CC	Specifies that usage of ASA carriage control characters in position one of data records is allowed. Note that no headings are provided by USERV. **Note:** If CC is specified, the destcode parameter must also be indicated.

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

Wherever permitted by the terminal access methods, you can use an attention interrupt to abort print requests. On spooled printouts, a message is added to indicate this condition. If you interrupt output by using the BREAK key or equivalent on hard copy terminals, the following message is displayed and new input will be accepted:

```
USV0000 - ENTER USERV COMMAND
```

SPACE Command

The SPACE command enables you to obtain a display of all free space on the specified disk.

The command format is:

SPACE [unit|VOLSER]

where *unit* is the three-character hexadecimal CUU address of the disk device for which a summary of the free space is to be displayed. Note that when VOLSER is the volume serial number of the disk device for which a summary of the free space is to be displayed, VOLSER must be a four- to six-character value.

The following two figures illustrate a typical display generated by the SPACE command.

```
 S COMTST
 VTOC ADDRESS 000-01 000-04          VOL=COMTST      UNIT=14E
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
AVAILABLE SPACE               ACTUAL            RELATIVE
                          000-05 000-14      000005,000010
                          019-00 023-29      000570,000150
                          075-00 084-29      002250,000300
                          094-20 096-29      002840,000070
                          097-20 098-19      002930,000030
                          112-00 199-29      003360,002640
                          201-00 554-29      006030,010620
END OF VTOC  002830 OF 016650 TRACKS USED, 013820  FREE
 ****** USV0400 - END OF DATA ******
```

```
 S FBA001
 VTOC ADDRESS 000002 000017          VOL=FBA001      UNIT=170
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
AVAILABLE SPACE               ACTUAL            RELATIVE
                          000018 000351      000018,000334
                          046113 046463      046113,000351
                          048864 061599      048864,012736
                          149600 153899      149600,004300
END OF VTOC  136178 OF 153899 BLOCKS USED, 017721  FREE
 ****** USV0400 - END OF DATA ******
```

VTOC Command

The VTOC command enables you to obtain a display of the volume table of contents of a specified disk.

The command format is:

VTOC [unit|VOLSER]

where unit is the three-character hexadecimal CUU address of the disk device for which a summary of the volume table of contents is to be displayed. Note that when VOLSER is the volume serial number of the disk device for which a summary of the volume table of contents is to be displayed, VOLSER must be a four- to six-character value.

The following two figures illustrate a typical display generated by the VTOC command.

```
 VTOC COMTST
 VTOC ADDRESS 000-01 000-04          VOL=COMTST      UNIT=14E
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
FILE ID                                TYPE     ACTUAL       RELATIVE    DATE
COMPLETE.PROCS                          SD PD 001-00 003-29  00030,00090  84202
TOTAL.CL                                UN PD 064-00 066-29  01920,00090  87322
COM.V440.AUG.MSGLIB                     SD PD 085-00 088-29  02550,00120  87217
COMINSTL.PUNCH                         *SD PD 000-15 000-29  00015,00015  84201
TOTAL.RL                                UN PD 067-00 069-29  02010,00090  87322
CERT.COMV44.PCL                         UN PD 004-00 018-29  00120,00450  87267
COM.V440.AUG.ROLL1                     *SD PD 089-00 091-29  02670,00090  87217
CERT.COMV44.PRL                         UN PD 024-00 063-29  00720,01200  87267
BG.WORK.PUNCH2                         *SD PD 097-00 097-19  02910,00020  87288
REP.WORK1                              *SD PD 111-10 111-29  03340,00020  87266
TEMP1                                  *SD PD 092-00 092-19  02760,00020  87314
TEMP2                                  *SD PD 092-20 093-09  02780,00020  87314
TEMP3                                  *SD PD 093-10 093-29  02800,00020  87314
TEMP4                                  *SD PD 094-00 094-19  02820,00020  87314
TEST.SOURCE                            *SD PD 200-00 200-29  06000,00030  87320
TOTAL.SL                                UN PD 070-00 074-29  02100,00150  87322
NETWORK.V10.RELO                        SD PD 098-20 104-29  02960,00190  85128
NETWORK.V10.CORE                        SD PD 105-00 111-09  03150,00190  85128
END OF VTOC  002830 OF 016650 TRACKS USED, 013820  FREE
```

Note that an asterisk under the "TYPE" column in this figure indicates that the file has expired.

```
 VTOC COMTST
 VTOC ADDRESS 000-01 000-04          VOL=COMTST      UNIT=14E
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
FILE ID                                TYPE     ACTUAL       RELATIVE    DATE
COM.V440.ROLL1                          SD PD 114400-118271 114400,003872 87219
COM.V44.SEP23.PRL                       UN PD 014432-046111 014432,031680 87314
COM.V44.SEP23.PCL                       UN PD 000352-014431 000352,014080 87314
COM.V44.SEP23.PSL                       UN PD 061600-069343 061600,007744 87314
COM.V440.PGMLIB                         SD PD 069344-083423 069344,014080 87219
COM.V44.SEP.SD                          SD PD 088352-114399 088352,026048 87265
COM.V440.MSGLIB                         SD PD 083424-088351 083424,004928 87219
COMPLETE.JCLINCAT                      *SD PD 118272-121791 118272,003520 87219
GDDM.R14.PCL                            SD PD 121792-132351 121792,010560 87224
GDDM.R14.PRL                            SD PD 132352-142559 132352,010208 87224
GDDM.R14.PSL                            SD PD 142560-149599 142560,007040 87224
TEMP.WORK.DELETE.ANYTIME               *SD PD 046112-046112 046112,000001 87224
REP.WORK1                              *SD PD 046464-047663 046464,001200 87275
REP.WORK2                              *SD PD 047664-048863 047664,001200 87275
END OF VTOC  136178 OF 153899 BLOCKS USED, 017721  FREE
 ****** USV0400 - END OF DATA ******
```

The information displayed for each file is described in the following text.

VOL	Specifies the six-character volume name of the displayed volume.
UNIT	Specifies the CUU of the displayed volume.
FILE ID	Specifies the file identification name.
TYPE	Specifies the two-character file organization type followed by the two-character extent type.
	File organization types are:
	DA Direct Access IS ISAM LB VSE Library SD Sequential Disk VS VSAM UN Unknown or unidentified Extent types are: NX Cylinder Index or Master Index (ISAM) OV Overflow Area (ISAM) PD Prime Data SC Split Cylinder (SEQ.) U Unknown
ACTUAL	Specifies the starting and ending cylinder-head or PBN on which the file resides.
RELATIVE	Specifies the starting and ending track/block of the space on which the file resides.
DATE	Specifies the date the file was created.

USPOOL - Printout Spooling Utility

The Com-plete printout spooling facility (USPOOL) permits the flexible management and distribution of output to any online printer in the TP network.

Many applications require output to be printed on special forms. A problem often arises when different applications require different forms to be mounted on the same printer at the same time. With USPOOL, you can create printouts on virtual printers that are not currently active in the system and subsequently route the printouts through the TP network to the physical printer where the appropriate forms have been mounted.

In addition, USPOOL can be used to provide relevant information pertaining to each printout in the system, thus providing a comprehensive overview of all queues for all printers. You can also display the contents of a printout before requesting a print operation.

USPOOL also supports the operation of online printers using commands such as "DISPLAY STATUS", "HALT", and "RESET". In addition, you can route any output from the online queues to the system spool by using the special destination "SYSOUT".

USPOOL is completely menu-driven and provides the capability of full screen data entry. An online HELP facility is also available to assist you.

Note:
The system programmer for your installation has the option of restricting access to any particular USPOOL function.

This chapter covers the following topics:

- General PF Key Assignments
- USPOOL Functions

General PF Key Assignments

You can use the CLEAR or PF3 key to return to the previous menu. Note that entering one of these keys on the Main Menu terminates USPOOL.

Use PF1 to invoke the appropriate help display.

USPOOL Functions

To reach the USPOOL Main Menu, enter the following command:

```
*USPOOL
```

The following screen appears:

```
10:00:37     TID    84           COM-4.5.            User ID LBL        03/18/97
                         -- Printout Spooling --                           USP0


       Function                          ID    PFK     Operand
 ---------------------------------------------------------------------------

     List Queue                          LQ     1      (Listname Form)
     Printer Overview                    PO     2      (Printer Name)
     HELP

                         Function ....         or PFK
                         Operand .....

 Spoolsystem Status

     7  Printouts in system

  4048  Diskblocks available
    48  Diskblocks currently is use          2 percent
  2412  Read I/Os     1357  Write I/Os
 ---------------------------------------------------------------------------
```

The USPOOL Main Menu provides the following:

- Access to the major spooling management functions on the top half of the screen;
- An overview of the status of the spooling system (number of printouts and spool file usage) on the bottom half of the screen.

You can execute either of the two listed functions by either entering the corresponding function ID or pressing the appropriate PF Key.

The spooling management functions available from the USPOOL Main Menu are summarized in the following table and discussed in the remainder of this section.

Function	Explanation
LQ	Allows spooling display and manipulation specified by printout.
PO	Allows spooling display and manipulation specified by printer.

Note that you can leave the Listname (printout name) operand blank in order to produce the entire list. In addition, you can limit the Listname specification to all items with the same prefix by using "*" as the final character, for example, "EXEC*".

List Queue

To display and/or modify one or more printouts, select the List Queue ("LQ" or PF1) function from the USPOOL Main Menu. The following screen appear:

If you did not specify a name operand, a list of all queued printouts for all printers is displayed in printout number sequence. If you did specify a name/prefix operand, only printouts matching this criteria are displayed.

Note that each line displayed represents one printout.

```
10:02:32     TID    84           COM-4.5.           User ID LBL        03/18/97
                         -- Printout Spooling --                           USP1
-- List Queue --
LQ                                  Page    0
FC  Listname List-No Form Disp Lines  Copy Pri  Userid     log-drv    Printer
-------------------------------------------------------------------------------
.   NDMWEIT       9        R      64     0   0  HV                    DAE9809E
.                12        R       1     0   0  JTE                   ALA354D7
.               394        R      33     0   0  GS                    DAESC132
.   INSTALL     399        R     271     0   0  EF                    DAE9809E
.   HARDCOPY    409        R      33     0   0  ULK                   DUGU1108
.   HARDCOPY    410        R      33     0   0  ULK                   DUGU1108
.   HARDCOPY    411        R      33     0   0  ULK                   DUGU1107
         0           0                   0   0
         0           0                   0   0
         0           0                   0   0
         0           0                   0   0
         0           0                   0   0
         0           0                   0   0
         0           0                   0   0
         0           0                   0   0
-------------------------------------------------------------------------------
```

The following operations are available to view the entire queue:

- If the available amount of data exceeds the screen size, press PF8 to reach the next page;
- If the available amount of data does not exceed the screen size, press PF8 to restart the display from the beginning;
- To modify the printouts according to the user requests, press ENTER.

The following table describes the column headings on the USPOOL List Queue screen.

Field	Usage
FC	Specifies the Function Code. Type the value directly over the "." in the FC column. The following values, listed in the bottom line of the screen, are permitted:
	S — SHOW function: Displays the contents of the specified printout at your terminal. (This invokes the "Printout Queue Display".) See the section *Display Printout on Screen* later in this section. M — MOVE function: Moves the printout to another printer, that is, queues it for another printer and deletes it from the queue of the original printer. Note that this value must be accompanied by a new name in the "Printer" field. The values for "Pri " (priority) and "Copy" (number of copies) can also be modified at the same time. C — COPY function: Same as M (above) except that the printout is copied, not moved; that is, it is not deleted from the queue of the original printer. P — PURGE function: Purges the printout from the printer queue. U — UPDATE function: Updates the specifications of the printout. All fields marked as modifiable can be changed.
Listname	Specifies name of the printout as specified by the originator.
List-No	Specifies the Com-plete identification number assigned to the printout.
Form	Specifies the printout form specification (is modifiable).
Disp	Specifies the disposition of this printout:
	E — Logic error detected during MSG/PO restart. Before you decide to change this status to R or L, use the show function to check the content. I — Is inputting, that is, the printout has not finished. R — Is ready; is waiting for printer to get ready. H — Is ready but will be held until Disp is changed to "R." L — Is ready and printed, but is still left in the spooling system. O — Is outputting, that is, the printout is currently being printed.
	Note that "R" Disp status can be modified to "H" or a request to leave the printout in the spool (Disp "L") can be made. In addition, "H" can be changed to "R" or a request to leave the printout in the spool ("L") can be made.
Lines	Specifies the number of text lines for this printout.

Field	Usage
Copy	Specifies the number of additional copies requested for this printout (modifiable).
Pri	Allows the priority of the printer to be specified. Note that the highest priority is 1.
Userid	Specifies the User ID of the printout originator.
Log-Drv	Specifies the name of the logical output driver routine, which can perform additional output formatting during printing.
Printer	Specifies the name of the destination for this printout (modifiable with the COPY or MOVE function).

Route to System Printer

You can route printouts within the Com-plete TP spooling system to the operating spooling system by using the COPY and MOVE functions (see the preceding table) and defining SYSOUT as the new printer name. The output is then be transferred to the output class "A" as default. To select another output class, specify SYSOUT=x.

Printer Overview

To display and/or operate one or more printers, select the Printer Overview ("PO" or PF2) function from the USPOOL Main Menu. The following screen appears:

```
10:04:06     TID    84           COM-4.5.           User ID LBL        03/18/97
                         -- Printout Spooling --                           USP2
-- Printer Overview --
PO                                  Page
FC  Printer    Tid     Dev-Typ    Status    Form   Q-Num
-------------------------------------------------------------------------------
.   ALA274DF    265   3288  L    WAIT              0
.   ALA354BE    261   3288  L    WAIT              0
.   ALA354BF    262   3288  L    WAIT              0
.   ALA354DE    264   3288  L    WAIT              0
.   ALA354D7    263   3288  L    ERROR             1
.   ALB364BD    277   3288  L    WAIT              0
.   ALB364BE    275   3288  L    WAIT              0
.   ALB364BF    276   3288  L    WAIT              0
.   ALC254DF    289   3288  L    WAIT              0
.   ALC364BE    286   3288  L    WAIT              0
.   ALC364BF    287   3288  L    WAIT              0
.   ALC394DE    288   3288  L    WAIT              0
.   AMSU1116    233   3288  L    WAIT              0
.   BRUU1110    232   3288  L    WAIT              0
.   DAERZ993    102   3288  L    WAIT              0
-------------------------------------------------------------------------------
```

The printer name can be associated with either a real or a virtual printer. Note that you can leave the printer name operand blank in order to produce the entire list. In addition, you can limit the printer name specification to all printers with the same prefix by using "*" as the final character, for example, "EXEC*".

The following operations are available to view the entire printer list:

- If the available amount of data exceeds the screen size, press PF8 to reach the next page;
- If you're at the end of the printer list, press PF8 to restart the display from the beginning;
- Press ENTER to switch to the requested function.

The following table describes the column headings on the USPOOL Printer Overview screen.

Field	Usage
FC	Specifies the Function Code. Type the value directly over the "." in the FC column. The following values, listed in the bottom line of the screen, are permitted:
	O — OPERATE function: Modifies the printer operation of the specified printer. Q — QUEUE function: Displays the queue for the printer, i.e., switches to the List Queue of Printouts display. (This function is the same as selecting the USPOOL Main Menu "List Queue" function, described earlier in this section.)
	The OPERATE and QUEUE functions are discussed in detail in the subsections below.
Printer	Specifies the logical name of the printer.
Tid	Specifies the unique terminal ID as specified in the TIBTAB.
Dev-Typ	Specifies the device type of the printer.
Status	Specifies the current status of the printer:
	WAIT — Is waiting for work. RUN — Is currently active. ERROR — Is held due to physical I/O error during output. UNDEFIN — Is currently undefined in the spooling system. This status will be indicated if output was scheduled for this printout but Com-plete has no corresponding active session. INTVREQ — Is currently in "intervention required" status (for example, paper is out, hold, etc). SIMLOGON — Waiting for VTAM simlogon request to be completed.

Operate Printer (OPERATE Function)

The OPERATE function enables you to modify the operation of a specific printer. Note that you must specify a logical printer name or TID when you invoke this function. The "Operate Printer" screen, shown in the following figure, is then displayed for the specified printer.

```
10:06:11     TID    84           COM-4.5.           User ID LBL        03/18/97
                         -- Printout Spooling --                            USP4
-- Printer Operation--
PO
         Function                 ID       Operands
-------------------------------------------------------------------------------
      Start                        S
      Reset                        R
      Halt                         H
      Position current printout    P       mode / number of pages
      Cancel current printout      C
      Flush all queue entries      F
      Mount new form               M       form

                       Function.. .
                       Mode/num..   /     0
                       Form......
Printer   ALA274DF
Status    WAIT         Listame
Form                 Records              0
Q-num            0       printed          0
-------------------------------------------------------------------------------
USP0004  NO PRINTOUT QUEUED
```

Note that the OPERATE function supports the operation of all printers of the TP network that are defined to Com-plete.

The function IDs in the above figure are explained in the following table.

Function ID	Description
S	Starts a printer that has been previously halted.
R	Resets a printer on which an error has occurred.
H	Halts the printer after the current printout has been completed.
C	Cancels the current printout. Printing will resume with the next printout.
F	Cancels all queue entries for this printer. Note that the current printout is not affected by this function.
P	Positions the current printout, that is, stops printing and resumes at any page within the printout. Note that this function requires that the positioning mode and number of pages be specified as operands. The positioning modes are:
	T Top: Printing will resume nnn pages from the top of the printout. B Bottom: Printing will resume nnn pages from the bottom of the printout. R Relative: Printing will resume nnn pages from the current page. A Absolute: Printing will resume at page nnn .

Note that you can abbreviate the values in the preceding table by using only the first two characters.

Display Printout on Screen (SHOW or QUEUE Function)

The SHOW or QUEUE function enables you to display the contents of a specific printout at your terminal. You can request the display from either the List Queue screen (Function Code "S") or from the Printer Overview screen (Function Code "Q").

The following figure shows a sample of the output produced by selecting this function.

```
16:13:54     TID     6          COM-5.1.            User ID MBE        11.11.97
                        -- Printout Spooling --                            USP3
-- Printout-display --                                            Line       1
Find string:                                                        of      72
 1        11        21        31        41        51        61        71
!---------!---------!---------!---------!---------!---------!---------!--------
                       J E S 2  J O B  L O G  --  S Y S T E M  D A E F  --  N O

15:15:10 JOB03628 ---- TUESDAY,   11 NOV 1997 ----
15.15.10 JOB03628  $HASP373 COKBAT   STARTED - INIT    3 - CLASS K - SYS DAEF
15.15.11 JOB03628  ACF9CCCD USERID COK      IS ASSIGNED TO THIS JOB - COKBAT
15.15.11 JOB03628  IEF403I COKBAT - STARTED - TIME=15.15.11
15.15.11 JOB03628  -                                              --TIMINGS (
15.15.11 JOB03628  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SR
15.15.11 JOB03628  -COKBAT   BATCH                00     14     54    .00    .0
15.15.11 JOB03628  IEF404I COKBAT - ENDED - TIME=15.15.11
15.15.11 JOB03628  -COKBAT   ENDED.  NAME-                     TOTAL TCB CPU TI
15.15.11 JOB03628  $HASP395 COKBAT   ENDED
------ JES2 JOB STATISTICS ------
  11 NOV 1997 JOB EXECUTION DATE
           16 CARDS READ
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  End         Find        Up    Down        Left  Right
```

There are the following options for navigating in the printout:

- scroll up (PF7) and down (PF8) by one screen;
- position to any distinct line in the printout by overtyping the line number displayed in the right upper corner of the screen;
- shift left (PF10) and right (PF11) by 10 positions;
- position to any distinct column by overtyping the column number in the left upper corner of the screen;
- search for a character string starting from the currently displayed position. To use this search function, enter the search string in the field prompted*Find string* and press PF5. Note that the search string can contain any characters, must not be enclosed in apostrophes, and must not contain more than two consequent spaces.

You can modify carriage control for the display in the "Eject:" field in the top right-center of the screen. Possible values:

O	"original carriage control" (that is, all line and page ejects are to be performed as defined in the printout).
L	"line eject" (that is, line ejects, but not page ejects, are to be executed at the terminal.
<blank>	"no carriage control" (i.e., that all line and page ejects are to be suppressed).

To return to the USPOOL Main Menu, press the CLEAR key.

USTOR - Storage Display Utility (Control User)

Using the USTOR utility program, you can:

- Display virtual storage;
- Alter contents of virtual storage;
- Monitor storage modifications by system tasks;
- Scan storage for a given character string;
- Display Com-plete internal buffer usage;

Note:
For z/OS, XA, and VSE systems, access to virtual storage is limited to the common storage area and the address space of Com-plete.

This section describes the techniques for using the USTOR utility program to perform the functions outlined above.

This chapter covers the following topics:

- Command Format
- Main Storage Display

Command Format

Initial access to USTOR is gained by entering the call request:

```
*USTOR
```

Entry of this call request establishes communication with the USTOR utility program. Once communication is established, the USTOR commands and functions are available for use.

Note that USTOR is fully conversational, that is, the functions can be entered any time after USTOR has been initiated. The USTOR functions may also be invoked at the time USTOR is initially invoked with the command format:

```
*USTOR function
```

If more than one function is applicable to the display desired, multiple functions can be entered when USTOR is invoked. In this situation, the function codes must be separated by commas.

Main Storage Display

If you initially invoke the USTOR utility without arguments, a display of the contents of main storage in hexadecimal format is generated, beginning at location zero. The following figure illustrates a typical display of the contents of main storage.

```
KEY=0 FETCH REF CHG PAGE-PROTECTED
00000000  040C0000  81099500  00000000  00000000  *    a n        *
00000010  00FCD590  00000000  070C1000  8100F89E  *  N-       a 8-*
00000020  070C0000  000B272C  040C0000  8109F470  *           a 4 *
00000030  00000000  00000000  070E0000  00000000  *               *
00000040  00000000  00000000  00000000  00FCD590  *             N-*
00000050  00000000  00000000  040C0000  81096D18  *           a - *
00000060  040C0000  00BE9F60  000C0000  817A9680  *      --   a:0-*
00000070  00080000  817A9040  040C0000  80FE2080  *    a:-    -  -*
00000080  00000000  00021202  00040101  00040016  *               *
00000090  008CD000  00000000  00000000  00000000  * -             *
000000A0  00000000  00FFF8D8  00000000  00000000  *      8Q       *
000000B0  00000000  00000000  00010026  00FCF2E8  *             2Y*
000000C0  00000000  00000000  00000000  00000000  *               *
000000D0  00000000  00000000  00000000  00000000  *               *
000000E0  00000000  00000000  00000000  00000000  *               *
000000F0  00000000  00000000  00000000  00000000  *               *
00000100  00000000  00000000  00000000  00000000  *               *
00000110  00000000  00000000  00000000  00000000  *               *
00000120  00000000  00000000  00000000  00000000  *               *
00000130  00000000  00000000  00000000  00000000  *               *
00000140  00000000  00000000  00000000  00000000  *               *
00000150  00000000  00000000  00000000  00000000  *               *
```

Note the following:

- The default display is in interpretive mode;
- The default display is the number of lines defined in the TIBTAB for this terminal less one. Normally, the default display is 23 lines for a 3277 model 2;
- Hexadecimal main storage addresses are given in the left-most margin;
- The key of the storage location addressed is given in the first data line display.

Main Storage Addressability Features

Once the main storage display is obtained, there are several USTOR features that enable you to address main storage locations. The following table summarizes the main storage addressability features and lists the corresponding commands, where applicable:

Feature	Command	Description
Contents	C	Displays by using the contents of a 24-bit address from storage.
Contents (XA only)	X	Displays by using the contents of a 31-bit address in storage (XA only).
Cursor Addressing	(ENTER key)	Moves the indicated byte to the top of the screen.
Displacement		Displays storage relative to a given location, that is:
	+H	Increments the current address by a half-page.
	H	Decrements the current address by a half-page.
	+P	Increments the current address by one page.
	-P	Decrements the current address by one page.
Mode	L,HEX,INT	Determines the type of display.
Null Entry	(null)	Continues the display of main storage in hexadecimal format at the next logical address.
Paging	(ENTER key)	Scans virtual storage.
Register	R	Saves a location address for future reference.
Specified Address	xxxxxx	Displays main storage at the specified hexadecimal location.
Specified Address (XA only)	xxxxxxxx	Displays main storage at the specified hexadecimal location (XA only).

Contents Feature

Indirect addressing is accomplished by use of the contents feature. This feature refers to a location in storage and uses the 24-bit contents at that location as the main storage address to be displayed. The format is:

```
C(nnn)
```

In XA, the contents of a 31-bit address in storage is displayed. The format is:

```
X(nnn)
```

where the character C must be entered and the parentheses must be entered as illustrated. For XA system users who wish to access a 31-bit address indirectly, note that X must be entered instead of C in the following examples. The variable within the parentheses must be one of three items:

- A main storage address - C(2A0) uses the fullword found at location X’2A0’ as the address of main storage to be displayed. In the XA environment, X(2A0) indicates that the address found at location X’2A0’ will be treated as a 31-bit address.

- An address relative to the current location - C(+16) uses the fullword found at the current location plus 16 (decimal) as the address of main storage to be displayed. In the XA environment, if the fullword found at current location plus 16 (decimal) is to be treated as a 31-bit address, X(+16) should be used.

- An asterisk - C(*) uses the fullword found at the current location as the main storage address to be displayed. In XA, X(*) must be used if the fullword at the current location is to be treated as a 31-bit address.

Cursor Addressing Feature

By positioning the cursor to any byte within the main storage display and pressing ENTER, the display is scrolled forward, moving the indicated byte to the top of the screen.

Displacement Feature

Displacement addressing means displaying main storage at an address relative to the currently displayed location. This is accomplished by entering a plus (+) for positive displacement or a minus (-) for negative displacement along with one of the following additional features with each display request:

- A number indicating the desired displacement. Note that this number must be preceded by an "X" if it represents a hexadecimal number; otherwise, it is treated as a decimal number.

- A "P" or "H" to increment or decrement the current display by a full page or half a page, respectively.

For example, entering +234 would display main storage at relative location +234 from the currently displayed address. Entering -X2C8 would display main storage at relative location -X2C8 from the currently displayed address.

Mode Feature

The mode feature determines the display characteristics to be generated by USTOR. Three primary display characteristics are available. Each is determined by entering the mode command function to obtain the desired characteristic.

The mode command functions are:

L	Determines the number of lines to be written for each display.
HEX	Gives the display in machine internal hexadecimal mode only.
INT	Gives the display both in hexadecimal and character translation modes.

- *The L mode function:*
  is convenient for restricting the number of lines displayed at the terminal. If the terminal in use is not a 3277-type terminal, it may be preferable to change the number of lines in the display. The function may be entered either when USTOR is invoked or any time thereafter.

- *The HEX function:*
  generates a display in machine internal hexadecimal format only. The hexadecimal addresses that appear in the left margin are in increments of X’20’ (decimal 32).

- *The INT mode function:*
  generates a display in both internal machine hexadecimal format and external translated format. The output is the same basic display as that obtained with the HEX function except that the storage addresses are in increments of X’10’ (decimal 16) and the character mode is displayed, line for line, to the right of the hexadecimal display. Note that non-displayable characters are translated to blanks.

Note that the INT display mode is the default when USTOR is invoked.

Null Entry Feature

After USTOR is invoked, the initial display is the default hexadecimal display of main storage beginning with location 0. Null entries or pressing ENTER causes a continuation of the display with the next logical address. The first page of a typical main storage display is illustrated previously in this section.

Since the type of terminal determines the number of lines to be displayed, the next logical address is in increments based upon the last address in a given display.

Paging FeatureRegister Feature

Once a main storage display has been obtained, contiguous storage locations can be displayed by pressing ENTER. The number of lines or addresses displayed depends upon the line mode used (L, HEX, or INT). Specific locations within main storage can be addressed directly or indirectly. Register addressability is the ability to save an address in a USTOR storage area called a register and subsequently display the memory located at that address by referring to the contents of the register. USTOR provides 10 registers (0 through 9) to store addresses.

Specified Address Display Feature

A specific address can be displayed by entering the desired address as a one- to six-digit character string:

```
xxxxxx
```

where xxxxxx is the desired address.

In XA, a one- to eight-character string can be entered:

```
xxxxxxxx
```

where xxxxxxxx is the desired address.

Note that the address represented by xxxxxx or xxxxxxxx is interpreted as a hexadecimal address. Entry of value 80 means an address equivalent to X’80’, not decimal 80.

The desired address can be entered either when USTOR is invoked or any time thereafter.

USTOR Commands

In addition to the functions mentioned earlier in this section relating to main storage addressability, USTOR performs other types of functions.

The USTOR functions are invoked by entering the appropriate command. The following table summarizes the available commands.

Command	Description
=	Re-executes the last command line entered.
?	Displays the last command line entered.
AM	Alters main storage.
ASU	Displays data about utilization of the Com-plete address space
BFS	Displays the usage of internal buffers.
BUMP	Increments addresses upon paging.
NOBUMP	Does not increment addresses upon paging.
COMREG	Displays the COMREG control block.
CVT	Directly displays the CVT (z/OS only).
GETVIS	Displays the storage usage in partition (VSE only).
HEX	Causes a display in hexadecimal format.
INT	Causes a display in interpretive mode.
L	Sets the number of lines to be displayed.
LOOP	Monitors storage modifications by system tasks.
NOLOOP	Terminates the LOOP function.
LUBS	Displays storage beginning at the LUBs in the partition in which Com-plete is running (VSE only).
PARTCR	Displays the VSE partition COMREG for Com-plete (VSE only).
PIBS	Displays storage beginning at the PIB table (VSE only).
PUBS	Displays storage beginning at the PUB table (VSE only).
R	Remembers an address in a register.
S	Scans storage for a given character string.
SYSCOM	Displays the VSE system COMREG (VSE only).
TCBS	Displays the system TCB chain (z/OS only).
TCWS	Displays the transaction control words.
TIB	Displays a specified terminal information block.
TRACE	Displays the Com-plete internal trace table.
UCB	Directly displays a given UCB (z/OS only).

You can also enter these commands as part of the USTOR invocation sequence or any time thereafter.

Detailed descriptions of each command, its function, format, and features follow below.

AM Command

One to eight bytes of virtual storage acn be altered via the USTOR utility program with the command:

```
AM address data
```

where the arguments are:

address	Specifies a one- to eight-digit hexadecimal number representing the address of the memory location to be modified. Note that the address of the currently displayed memory location must be the same as this address.
data	Specifies the hexadecimal character string to be placed in the memory location specified by the address argument.

For example, the command:

```
AM 0002C5 1A2A
```

alters the two bytes of storage located at 0002C5 to 1A2A. Note that the AM operation is executed in this case only if the currently displayed address is 0002C5.

ASU Command

The ASU command displays the amount and percentage of used and free virtual storage in the Com-plete address space, separately for storage below and above the 16MB line.

BFS Command

The BFS command displays the usage of internal buffers. The display provides location, size, and usage rate. The command format is:

```
BFS
```

The following figure illustrates a typical display resulting from this command.

```
BUFFER NAME       LOCATION     SIZE                FREE           %FREE    DIFF
RESIDENT PGMS      94578BC0 006B3D10 ( 6864K)  00000000 (    0K)    0.0     0.0
TIBTAB             12A46000 00003C00 (   15K)  000039C0 (   14K)   96.2  + 96.2
NUCLEUS            00008000 000672B8 (  413K)  00000000 (    0K)    0.0     0.0
```

BUMP and NOBUMP Commands

The BUMP command causes the currently displayed address to be bumped to the next logical address in memory whenever either the ENTER key is pressed or the LOOP command is in effect. Note that the invocation of USTOR automatically sets the BUMP mode of operation. The command format is:

```
BUMP
```

The NOBUMP command causes the currently displayed address to remain the same whenever either ENTER is pressed or the LOOP command is in effect. The command format is:

```
NOBUMP
```

COMREG Command

The COMREG command causes a display of the Com-plete Communications Region. The command format is:

```
COMREG
```

CVT Command (z/OS only)

CVT stands for the operating system Communications Vector Table. This table is the center of control for operating system functions and is frequently referenced by system maintenance personnel. The CVT command causes a memory display beginning with the memory location of the CVT. The command format is:

```
CVT
```

GETVIS Command (VSE only)

The GETVIS command causes a display of free storage in the Com-plete partition. The command format is:

```
GETVIS
```

HEX Command

The HEX command causes the current display to be given in full, internal, hexadecimal format, 32 characters to a line. The address in the left margin, also in hexadecimal, is in increments of X’20’ (decimal 32). The command format is:

```
HEX
```

INT Command

The INT command causes a memory display where each line of the display consists of the hexadecimal address of the data in the line, 16 characters in internal hexadecimal format, and the translated external format of the same 16 characters. Note that all non-displayable characters are translated to blanks. The command format is:

```
INT
```

L Command

The L command sets the number of lines to be displayed after a USTOR command is entered. For 3277 model 2 terminals or their logical equivalent, the default number of lines is 23. For hard copy terminals, the default number of lines is one. The L command format is:

```
L=nn
```

where nn is any two-digit integer that specifies the number of lines to be displayed.

LOOP and NOLOOP Commands

The LOOP and NOLOOP commands monitor storage modifications being made by system tasks or application programs. The LOOP command causes USTOR to display memory dynamically; NOLOOP terminates the LOOP facility.

LOOP and NOLOOP are normally used in conjunction with the BUMP and NOBUMP commands. If NOBUMP and LOOP are entered, the same memory location is displayed repeatedly. The format of the LOOP command is:

```
LOOP
```

Note that the display is at I/O response time intervals; no timer facility is provided.

Termination of the LOOP command is caused by an external interrupt from the terminal. To cause this interrupt, press ENTER or power off the terminal.

Note:
If you press ENTER to cause an external interrupt to terminate the LOOP command, you must also enter the NOLOOP command; if not, pressing ENTER a second time causes the LOOP command to become effective again.

The NOLOOP command causes the LOOP command to be ignored. This command is normally entered immediately after termination of the LOOP command (by an external interrupt). The format of the NOLOOP command is:

```
NOLOOP
```

Note that this command has no effect if the LOOP command has not been entered.

LUBS Command (VSE only)

The LUBS command displays storage beginning at the LUBs (Logical Unit Blocks) in the partition in which Com-plete is running. The command format is:

```
LUBS
```

This command may be entered either when USTOR is invoked or at any time thereafter.

PARTCR Command (VSE only)

The PARTCR command enables the display of the Com-plete VSE Partition Communication Region. The command format is:

```
PARTCR
```

PIBS Command (VSE only)

The PIBS command displays storage beginning at the PIB (Partition Information Block) table. The command format is:

```
PIBS
```

PUBS Command (VSE only)

The PUBS command displays storage beginning at the PUB (Physical Unit Block) table. The command format is:

```
PUBS
```

R Command

USTOR maintains a set of internal storage areas called registers. These registers (0 to 9) are available for the storage of addresses and can be referred to later for recall of the same addresses. The R command permits you to store an address in a given register and, if USTOR does not terminate, recall that address for display purposes. The command format is:

```
Rn=*
```

where the variable *n*, an integer from 0 to 9, designates in which register to store the desired address. Note that the address to be stored in the designated register is the address currently being displayed. The items R, =, and * must be entered as shown.

If USTOR is not terminated, an address stored in a register can be recalled by entering the command:

```
Rn
```

where *n* is the integer (0 to 9) that designates the register containing the desired address. The memory location with the address specified in register *n* is displayed.

S Command

The S command enables the scanning of virtual storage for a specified hexadecimal value. The command format is:

```
S xxxx
```

where *xxxx* is a one- to four-byte hexadecimal character string (two- to eight-hexadecimal digits). You can specify one of the following formats:

S xxxx	Scan forward, byte by byte, for a value "equal" to the specified character string.
S Xxxxx	Scan forward, byte by byte, for a value "not equal" to the specified character string.
S xxxx(+n)	Scan forward, n bytes at a time, for a value "equal" to the specified character string.
S xxxx(-n)	Scan backward, n bytes at a time, for a value "equal" to the specified character string.
S Nxxxx(+n)	Scan forward, n bytes at a time, for a value "not equal" to the specified character string.
S Nxxxx(-n)	Scan backward, n bytes at a time, for a value "not equal" to the specified character string.
S	Repeat the previous scan request beginning with the current address.

Note that the optional *n* argument represents any decimal integer.

SYSCOM Command (VSE only)

The SYSCOM is the system control program’s communication area for VSE. This area controls the operating system functions and is frequently referenced by system maintenance personnel.

The SYSCOM command causes a memory display beginning with the memory location of the system COMREG. The command format is:

```
SYSCOM
```

TCBS Command (z/OS only)

The TCBS command obtains a formatted display of the operating system TCB chain. The command format is:

```
TCBS nnn
```

where the variable *nnn* is used in one of the following options:

null	Display all TCBs.
n	Display all TCBs beginning with the n th TCB. Note that n is any integer from 0 to 99.
P	Display all user (partition) TCBs.
Pn	Display all user (partition) TCBs beginning with the n th user TCB. Note that n is any integer from 0 to 99.

The number of lines in each display is determined by the L command; therefore, if all system TCBs are requested in a display, a maximum of 24 are displayed with a 3277 model 2 terminal.

In z/OS and XA, only the TCBs that are within the Com-plete address space are displayed.

The following figure illustrates a typical display of the first page of an operating system TCB chain.

```
 TCBAD PKEY MOTHER TCBRBP FREE   JOB      STEP     PROCSTEP LOPART HIPART  #K
8FDE40      000000 8FDAD8        MSTJCLJ2          JES2
8FD080      8FDE40 8FFCC8        MSTJCLJ2          JES2
8FFD70      8FDE40 8FFAB8        COM440            COM440
8E9E88      8FFD70 8FF038        COM440            COM440
8E9CF0      8E9E88 8FF7A0        COM440            COM440
8E9AC0      8E9E88 8E9A00        COM440            COM440
8E9868      8E9E88 8E97A8        COM440            COM440
8E9610      8E9E88 8E9550        COM440            COM440
8E92A0      8E9E88 8FF130        COM440            COM440
8E90F0      8E9E88 8E9030        COM440            COM440
8DFE88      8E9E88 8DFD88        COM440            COM440
```

TCWS Command

For each system task created by Com-plete at initialization time, a Transaction Control Word (TCW) is maintained in memory to record the status of each task.

The TCWS command displays the TCW control blocks at the terminal. The command format is:

TCWS

TIB Command

The TIB command displays a designated terminal information control block. The command format is:

TIBn

where *n* is an integer designating the Terminal Identification number of the terminal whose information control block is to be displayed.

TRACE Command

The TRACE command displays the Com-plete internal trace table. The command format is:

TRACE

UCB Command (z/OS only)

The UCB command displays a designated Unit Control Block (UCB). The command format is:

UCB cuu

where *cuu* is the channel number and unit number (CUU) for the UCB to be displayed.

UTIMER - Com-plete Timer Services Utility

The Com-plete menu-driven utility UTIMER is a timer that allows you to schedule various functions at a predefined time. You can schedule the following items with UTIMER:

- Message switching (Send Message);
- Job handling (submit, release, cancel);
- Your own logoff from Com-plete;
- Installation functions.

This chapter covers the following topics:

- Command Format
- Using UTIMER
- UTIMER Functions
- Timer Exits

Command Format

To invoke UTIMER, enter the command:

***UTIMER**

This displays the UTIMER Main Menu:

```
17:24:25     TID    15          COM-5.1.            User ID MBE        09.12.97
                       -- Com-plete Timer Services --                       UTMR
        Function          ID   PFK               Operands
 ----------------------   --   ---   -------------------------------------------
           Send Message   S      1   Time etc., Message, Userid (,Urgent)
             Handle Job   H      2   Time etc., Job Name, Job Action
             Log Me Off   OF     3   Time etc.
 Installation Functions   I      4   Time etc.
 List Personal Services   LP     5
      List All Services   LA     6
      Restore Last Menu   R     12


        Select Function:      or PFK
           and Operands

           Request Time:   0 :  0 on  9 / 12 / 94
   Repeat If Impossible:
                  Up to:  24 :  0 on  9 / 12 / 94
       Request Interval:                           (D Day W Week M Month Y Year)
           Message Text:
                 Userid:  MBE      Urgent:
               Job Name:           Job Action:     (S Submit R Release C Cancel)
```

Using UTIMER

UTIMER is menu-driven. This means that you can access its functions directly from the UTIMER Main Menu.

Entering Operands

Request Time

Note that the first four functions on the UTIMER Main Menu ("Send Message", "Handle Job", "Log Me Off", and "Installation Functions") require that the "Request Time" operand be entered.

Note that when you add a new timer request, you must enter a valid time and date that is later than the present time. If the action could not be performed on the first attempt, you can also request that the timer attempt to repeat the action "Up to" a certain time and day.

Request Interval

By using the optional operand "Request Interval", you can also repeat the request in increments of a day ("D"), a week ("W"), a month ("M"), or a year ("Y").

Serving Timer Requests

The timer monitor UTIMRM is loaded into an attached TIB at Com-plete startup time, storing all timer requests in a timer SD file. The Com-plete nucleus consults UTIMRM each minute in order to determine what is to be performed at the present time.

UTIMER Functions

The UTIMER functions are discussed in the remainder of this section.

Send Message

To send a message either to yourself or or to another user ID, select function ID "S" or press PF1 on the UTIMER Main Menu. Also enter the required operands in the input fields next to "Request Time", "Message Text", and "Userid" (note that "Userid" defaults to your user ID). In addition, if it is an urgent message, mark "Urgent" with any character on the UTIMER Main Menu.

When you press ENTER, Com-plete displays the following message at the top of the UTIMER Main Menu:

```
YOUR REQUEST HAS BEEN ADDED
```

Handle Job (OS systems only)

To submit, release, or cancel a job, select function ID "H" or press PF2 on the UTIMER Main Menu. Also enter the required operands in the input fields next to "Request Time" and "Job Name", and enter "S", "R", or "C" as "Job Action".

When you press ENTER, Com-plete displays the following message at the top of the UTIMER Main Menu:

YOUR REQUEST HAS BEEN ADDED

The jobs to be submitted are stored in a data set referenced by a "SYSJOBS" DD statement in the Com-plete startup procedure.

Log Me Off

To request that UTIMER log you off the Com-plete system at a certain time, select function ID "OF" or press PF3 on the UTIMER Main Menu. Also enter the required operand "Request Time".

When you press ENTER, Com-plete will display the following message at the top of the UTIMER Main Menu:

YOUR REQUEST HAS BEEN ADDED

List Personal Services

To display all of the scheduled UTIMER requests for the user ID displayed at the bottom of the UTIMER Main Menu (the default is your user ID), select function ID "LP" or press PF5 on the UTIMER Main Menu.

When you press ENTER, the UTIMER Services Display (for the user ID displayed at the bottom of the UTIMER Main Menu) appears:

```
14:03:59     TID    13           COM-5.1.           User ID ADMIN     09/26/97
                                                                         UTML
USERID |  DATE   |  TIME |  FREQUENCY |SEND MSG  TO| USER |L|R|C|S|  JOBNAME || F
==========================================================================
SAJXS   09/24/87   09.05                                        X  SAJXSREP
SAJXS   09/24/87   15.00               REMINDE... SAJXS            SAJXSCNF
SAJXS   09/24/87   19.00                                   X        SAJXSCNF



==============================================================================
L = LOGOFF USERID ; R = RELEASE JOB   ; C = CANCEL  JOB   ; S = SUBMIT  JOB
F = FLAGS ::   L = LONG MSG ;  D = DELETE   ;  F = FREEZE   ;  A = ACTIVATE
```

List All Services

Note:
The "List All Services" function is available for control users only.

To display all scheduled requests for *all* users, select function ID "LA" or press PF6 on the UTIMER Main Menu.

When you press ENTER, a UTIMER Services Display screen like the one shown above appears, except it will show all users' requests.

Restore Last Menu

The "Restore Last Menu" function is useful when you want to add several similar requests.

To cause the last-entered input data (i.e., function ID and operands) to reappear on the UTIMER Main Menu, select function ID "R" or press PF12 on the UTIMER Main Menu.

Timer Exits

The following table describes the three UTIMER timer exits.

Exit	Description
UTMEX1	Is called by UTIMER in order to decide whether a request is to be logged in the timer SD file or not.
UTMEX2	Is consulted each minute by UTIMRM in order to decide whether a request is to be served at the present time or not. This exit can be used to add new "artificial" requests.
UTMEX3	Is an RJE submission exit called by UTIMRM; is similar to UXEEX3.

For detailed exit interface descriptions, see the Com-plete System Programming documentation.

Note:
In order for an installation to be able to use UTIMER, the Com-plete systems programmer must add the following statement to the Com-plete sysparms:

```
STARTUPPGM
sysparm
STARTUPPGM=UTIMRM
```

With the SUBMIT function, you are advised to catalog UTIMRM in ULIB with AF, as UTMOO3 errors may occur otherwise.

UZAP - Load Module Modification Utility

The UZAP utility enables you to define user-restricted data replacements (zaps) for load modules without affecting any other user sessions.

This utility is applicable to load modules residing in the COMPLIB load library chain or in Com-plete's program lookaside buffer and executing in the Com-plete threads. No modification of the original load module itself takes place; instead, data is replaced when the module is loaded into Com-plete thread storage under control of your session.

The UZAP utility enables you to:

- Add a zap defined by verification data and replacement data;
- Display data from a load module with the ability to define modifications;
- Display and modify existing zaps;
- Remove a zap;
- Switch a zap off and on temporarily;
- Export a zap and edit it using the Com-plete editor;
- Recover your zaps after relogon.

Note:
Defining a zap for Com-plete nucleus modules or other resident programs has no effect, as far as these modules are never loaded into a thread.

This chapter covers the following topics:

- Command Format
- The UZAP Menu
- Reestablish Zaps from Previous Session

Command Format

To add a zap, enter one of the following commands:

```
*UZAP module,address,verifydata,replacedata
```

or

```
*UZAP module,address,L=length
```

where:

module	Specifies the one-to eight-character name of the load module to be zapped. The module must reside in the COMPLIB load library chain or in Com-plete's program lookaside buffer.
address	Specifies the hexadecimal address, relative to the beginning of the load module, of the data to be verified / replaced.
verifydata	Specifies the data character string to be verified. verifydata must be a character string consisting of an even number from 2 to 120 of hexadecimal digits.
replacedata	Specifies the replacement character string. The number of verification and replacement characters must be the same.
length	Specifies the length of data you want to zap. If you specify this parameter, verifydata and replacedata must not be specified. UZAP will create a zap of the length indicated, with both verification data and replacement data equal to the data existing in the load module with any already existing zaps applied. This zap will be displayed using the UZAP menu screen (see below), and you can overtype replacement data. Note that length must be a decimal number from 1 to 120 or a hexadecimal number from X1 to X78.

Once you have defined a zap, it will be applied each time the appropriate module is loaded into thread under control of your session. No other sessions are affected.

Note that attached programs also execute in separate logical sessions and therefore will *not* be affected by a zap defined using UZAP.

The UZAP Menu

You can display all zaps existing for your session by entering

```
*UZAP
```

The UZAP menu screen is displayed, listing your zaps in ascending order by module name and displacement:

```
12:37:05        TID    19          COM-5.1.          User MBE          25.04.97
                           -- Individual User Zaps --                       UZT1

 FC  Module     Offset   Verify-data        Replace-data       Note
 -----------------------0-1-2-3-4-5-6-7----0-1-2-3-4-5-6-7--------------------
 .   DEMO       000000   47F0F060E9C5D4D6   47F0F060E9C5D4D6
                         40404040F4F6F040   40404040F4F6F040









 FC:  S witch  D elete  E xport              Reposition to:                  END
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End                     Top   Forwd                   Edit
```

From this menu, you can perform any of the following:

- Scroll the listing of zaps forward by pressing PF8
- List the zaps starting from the top by pressing PF7, or from the module name entered in the input field after the prompt Reposition to
- Modify replacement data of any zap by overtyping the data and pressing ENTER
- Delete a zap by entering the line command "D" in the FC column.
- Switch a zap on or off by entering the line command "S" in the FC column. This causes a zap not to be applied temporarily, without deleting it. UZAP marks inactive zaps with an appropriate note in the last column of the screen.
- Export one or more zaps in character format to an UEDIT work file by entering the line command "E" in the FC column.
- Fetch to Com-plete’s editor UEDIT to edit and/or save exported zaps by pressing PF12. This causes UEDIT to be invoked, to recover, and to display the exported zaps. You can now edit zap data and save it to the destination you desire. After you terminate UEDIT, the UZAP menu will be displayed again.

Reestablish Zaps from Previous Session

All zaps defined for your session using UZAP are automatically saved in Com-plete’s system data set and associated to your user ID. After you terminate your session and log on again, you can reestablish your zaps. To achieve this, enter any UZAP command (with or without parameters).

Note that the LOGON procedure does not reestablish any UZAP zaps. Note also that all zaps existing for your user ID from a previous session are reestablished when you invoke any UZAP command.

UUTIL - Menu-Driven Utility Functions

The UUTIL utility is a menu-driven facility for all users of Com-plete. With the UUTIL utility you can set up your personal Com-plete environment, and authorized users can also perform system monitoring and maintenance functions.

This chapter covers the following topics:

- Command Format
- Overview of Functions

Command Format

To invoke the UUTIL utility main menu, enter one of the following commands:

*UUTIL	Presents a menu with a list of utilities you are authorized to use, depending on your control status. From this menu, you can press PF10 to invoke the password prompt. Having entered the correct password, the menu with administrator utilities is displayed (Super User).
*UPROF	(supported for compatibility reasons)
*UUTIL <maintenance-password>	Presents the menu with administrator functions.

Alternatively, you can access UUTIL functions directly by specifying the corresponding two-character function code in the *UUTIL call using the format:

```
<*UUTIL call>/<function-code>
```

Examples:

*UUTIL	Displays general user menu.
*UUTIL <password>	Displays the administrator menu (Super User).
*UUTIL/UL	Displays the personal library ID subfunction of UUTIL.
*UUTIL <password> /UM	Displays the User ID maintenance screen.

Overview of Functions

The following table lists each UUTIL function together with the utility it replaces (if any) and a brief description of its use:

Function	Authorization	Replaces	Description
AI	General	UACCT	Display accounting information about your session.
FK	General	UPF	Display/set your personal utility PF keys.
FM	Administrator		Maintain file (DDN) definitions.
HC	General	USCHC	Define a destination for screen-to-hardcopy.
HE	General	UHELP	Provide online documentation for Com-plete utilities.
HM	Administrator	UHELPM	Maintain Com-plete HELP data.
GL	General	UGLIB	Display global library IDs.
MM	Administrator	UMSGF	Maintain Com-plete System and User messages.
MO	Control	UCTRL	Monitor and manipulate the status of the running Com-plete session.
NQ	Administrator	UQSCAN	Monitor and manipulate System Enqueues.
PW	General	UPWD	Change your logon password.
SD	Administrator	USDLIB	Maintain SD files online.
SI	General	UINFO	Display information pertaining to your installation.
TO	Administrator		Set Com-plete trace options.
TT	Administrator	UDTIB	TIBTAB maintenance.
TU	Administrator		Terminal / User ID Group maintenance.
UD	General	UDEF	Define your personal COM-PASS default settings.
UM	Administrator	ULOGM	Maintain Com-plete User ID definitions.
UL	General	ULIBID	Define your personal two-character library short-IDs. UP Administrator Maintain Com-plete default utility PF keys.
UP	Administrator		Allows you to assign commands to PF keys in the various Com-plete utilities
ZA	Administrator	UZAPS	Display information about currently implemented fixes. Display / modify applymod settings for the current Com-plete session.

Function AI - Accounting Display

If you select option AI from the User Definitions, your current Com-plete accounting information is displayed on your screen, for example:

```
14:50:45     TID     5          COM-5.1.                User MBE        12.11.97
                     -- Com-plete Accounting Display --                      UACT


 Userid Characteristics:  USERID: MBE                   Logon Date: 1997/11/12
                         Account: Com-plete             Logon Time: 14:50:27
                   Authorization: 0                            SMC: 1,2,3,4,5,6,7
                         Control: YES                          RMC: 1,2,3,5,6,7

 Terminal Characteristics:   TID: 5                    Device Type: 3270 VS
                         LU Name: SHRDAEN                     Case: UPPER
                       HC Device: DUMMY             Message Queue: 0

 Accounting Data:            CPU: 0.07          Data sent via MSG: 0
                     Thread Time: 14.85     Data sent to Terminal: 7538
                    Transactions: 5                          SIOS: 4
```

Meaning of the items:

Item	Meaning
Userid Characteristics: As defined using the UM utility.	
Userid	User ID to which the information applies.
Account	Account or group code or name assigned to the user ID.
Authorization	Authorization code assigned to the user ID. This code can be used if your installation runs its own security exit.
Control	Indicates whether the user is a control user YES or NO.
Logon Date/Time	Date and time of the user’s last logon.
SMC	Class code assigned to the user ID when sending messages and printout spooling. Note that code 4 is automatically assigned. Code 0 means the user is not allowed to send messages or printouts.
RMC	Class code assigned to the user ID when receiving messages and printout spooling. If this code is 0, the user is not allowed to receive messages or printouts.
For more information on these user items, see *Function UM - User Maintenance*.	
Terminal Characteristics:	
TID	ID of terminal at which the user is currently logged on.
LU Name	Terminal name; if Com-plete is running in a VTAM network, this is the name of the VTAM device.

Item	Meaning
HC Device	Name of the destination hardcopy device defined using the HC function of the UUTIL utility.
Device Type	Terminal classification as defined by Com-plete.
Case	Default case: modifiable using the UP and LOW direct commands.
Message Queue	Indicates the number of messages pending delivery to the user's terminal.
Accounting Data:	
CPU	Amount of CPU time elapsed since logon time, in seconds and hundredths of a second.
Thread time	Amount of time the thread was occupied, in seconds and hundredths of a second.
Transactions	Number of of transactions since logon time.
Data sent via MSG	Amount of data sent via message switching and printout spooling since logon time.
Data sent to terminal	Amount of data sent to and received by the terminal since logon time.
SIOS	In z/OS systems, number of SIOs issued since logon time. In none-z/OS systems, this is the number of EXCPs issued.

Function FK - PF Key Definition

This function allows you to assign commands to PF keys in the various Com-plete utilities. Pressing a PF key in the utility has the same effect as if you had entered the assigned command in the command line. If you select the FK option from the utility menu, the list of utilities for which you are authorized is displayed, for example:

```
16:38:51        TID     6           COM-5.1.         User SAGAWW       05/07/97
 System: Lcl 8            ---  PF-Key Maintenance  ---                      UPF0
 Nbr Utility     Nbr Utility     Nbr Utility     Nbr Utility     Nbr Utility
 --- --------    --- --------    --- --------    --- --------    --- --------
   1 COM-PASS     13              25              37              49
   2 UQ           14              26              38              50
   3 UPDS         15              27              39              51
   4 USTOR        16              28              40              52
   5 UDUMP        17              29              41              53
   6              18              30              42              54
   7              19              31              43              55
   8              20              32              44              56
   9              21              33              45              57
  10              22              34              46              58
  11              23              35              47              59
  12              24              36              48              60
 Please select Utility number..:   0
           and System Id.......:
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Updat
```

Select the utility for which you wish to define PF Keys by typing the number in the *Utility Number* input field.

If you have multiple Com-pletes installed, and you do not enter data into the *System ID* field, your Global PF Key definitions for the specified utility are defined. If you require special PF Key definitions for one Com-plete system only, enter its ID in this field (in the above example, the current Com-plete system has an ID of 8).

When you press ENTER, the PF definition screen is displayed, for example for UQ (no system ID specified):

```
14:19:49        TID    11          COM-5.1           User SAGAWW       23.11.97
System: Global                                                            UPF1
                            Dynamic PF-Key Table
                             UQ       (Private)                        Dis-
PFnr Command                                                          play   >
---- ---------------------------------------------------------------  ----   -
PF01
PF02
PF03
PF04
PF05
PF06
PF07
PF08
PF09
PF10
PF11
PF12
---- ----------------------------------------------------------------    -
     Override undefined keys with system default keys: N

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
            Mode  End         Updat       bwd   fwd         Purge       Quit
```

You can enter or modify data in the input fields. Meaning of the fields according to column header:

Column	Meaning
Command	The command to be executed when you press the PF Key. Valid commands are: Com-plete control functions. Com-plete commands, optionally with parameters, to execute a utility program, for example: **`*UQ Q,JB=name,RR=105=UEDIT PC(COMSM1)`** Program-specific commands. For example, for UQ: **`A,RR=3`** If the required command string exceeds 64 characters, press PF2. This displays the screen in which you can enter up to 79 characters for each command string.
Display	Enter any character here to cause the assigned command to be displayed in the command line when the PF key is pressed. You can then modify the displayed command before you press ENTER to execute it. If you use underscore characters in the assigned command string, the cursor is placed on the first underscore when it is displayed.
>	An asterisk (*) is automatically displayed in this column if you have entered more than 64 characters in the command string using PF2.

Press PF5 to save the definitions. Press PF3 to leave this function.

Function FM - File (DDN) Catalog Maintenance (Administrator)

The FM function enables you to maintain the Com-plete file catalog established in the COMSYS Data Set.

Application programs refer to files by DD/DLBL names. To establish the link between these names and the corresponding data sets, all DD/DLBL names referenced by application programs must be declared ("cataloged") to Com-plete using this subfunction.

The declaration includes the data set name, disposition (z/OS only), the name of the VSAM user catalog (VSE only), and other information (see below). The data set is allocated to Com-plete dynamically when an OPEN request is issued against the appropriate DD/DLBL name and deallocated when Com-plete is stopped, or when the file is closed explicitly using the CLOSE or BATCH subfunctions of UUTIL FM.

In comparison with the permanent data set allocation by JCL DD/DLBL statements used with previous versions of Com-plete, this mechanism provides maximum flexibility of data set access by BATCH jobs and for data set maintenance (backup, restore, reallocation, rename, etc.) without the need of restarting Com-plete.

The file declaration defines all parameters necessary for Com-plete to open the file. Options specified for a file in an application program are ignored. Instead, Com-plete uses the parameters defined via UUTIL FM to build control blocks and buffers once per DD/DLBL name. However, the parameters specified here for a file must be consistent with the file processing techniques used by application programs.

All online programs referring to a given DD/DLBL name share the same control blocks and buffers. This allows efficient use of resources in the system, but can also significantly influence performance. Therefore, careful choice of parameter values is recommended for files referenced frequently and by a large number of terminal users.

Access Method Options

The File Catalog Maintenance function records the status and processing options of online files to be manipulated using one of the following access methods:

- Indexed-Sequential Access Method (ISAM, z/OS only);
- Basic Direct Access Method (BDAM, z/OS only);
- Virtual storage Access Method (VSAM).

ISAM Considerations (z/OS only)

The addition of records to ISAM files in the online environment is not recommended because of the high overhead operation involving use of ISAM overflow areas. This, in turn, requires periodic reorganization of the ISAM file.

The use of the ADD option imposes two restrictions on programs accessing ISAM files:

1. All programs that add records must be thread-locked to the thread in which the ADD option is allowed.
2. The file must be allocated with DISP=OLD in Com-plete. Note that sharing of an ISAM file between Com-plete and another job step is not supported when records are added.

   You can use the following alternatives to circumvent these restrictions:

3. Instead of the ADD option, keep enough dummy records in the data set and update those records.
4. If the ADD option is required, split the ADD processing of the file into a separate program and use a FETCH or ATTACH function to invoke it. This allows most of the processing to take place in any thread, thus reducing the overhead of thread- locking all processing to a single thread.

BDAM Considerations (z/OS only)

The ADD option has no meaning for BDAM files because BDAM files are preformatted.

Com-plete extends BDAM support by allowing blocked BDAM files. This support is governed by the LRECL keyword in the DD statement or the data set attributes as recorded by the operating system. Com-plete computes the block number, reads it, and passes back the requested record from the block.

Note:
Only relative record access is supported with this technique.

VSAM Considerations

The way a VSAM cluster is created (by Access Method Services) and defined to Com-plete (File Catalog utility) can have a significant impact on performance and storage utilization within Com-plete.

It is recommended that you select a small control interval size when establishing the cluster in order to minimize buffer requirements and the impact of control intervals being held for update requests.

Com-plete provides optional support of establishing a VSAM local shared resource pool during system initialization. This support allows many VSAM clusters to share buffer pool and control block resources, thus minimizing storage requirements and the chance that storage may not be available to open a new cluster at first access. This resource pool is configured by the system programmer using the VSAMBUFFERS, VSAMHIPERSPACE (z/OS only), VSAMRPL and VSAMFIX sysparms.

Using the shared resource pool is recommended when it is not necessary - for performance reasons - to dedicate buffers and control blocks to the file. To use the shared resource pool, be sure that the file to be accessed can use the buffer pool defined by the VSAMBUFFERS sysparm and that all programs using the file conform to the restrictions for programs using shared resources.
Note that files using shared resources cannot be loaded online.

The default options for cataloging a VSAM DDN (file) include LSR, which assumes use of shared resources. If you do not wish shared resources with other VSAM files, specify MACRF=NSR.

There are a few restrictions on Com-plete's support of VSAM: neither the locate option (OPTCD=LOC in the RPL), nor ISAM emulation (for example, the use of a VSAM cluster trough ISAM) are supported.

Using the File Catalog Maintenance Function

If you select the function FM from the UUTIL menu, the File Catalog Maintenance menu listing the various utility subfunctions is displayed on your screen:

```
15:40:01        TID    15          COM-5.1.          User MBE          20.04.97
                             -- File Maintenance --                          UFI0

     Function                           ID  Operands
     ---------------------------------- --  ------------------------------
     File overview ................... FO  Prefix/Name, System
     Display catalog entry ........... DI  Name, System
     Catalog ......................... CA  Name, System , Access method
     Modify catalog entry ............ MO  Name, System
     Remove catalog entry............. RE  Name, System
     Set to BATCH status ............. BT  Name
     Set to ONLINE status ............ ON  Name
     Open (VSAM only) ................ OP  Name
     Close ........................... CL  Name
     SHOWCB ACB info (VSAM only) ..... SH  Name

               Select function .......
               Operand ...............
               System ................
               Access method .........    ( VS / DA / IS )

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

Meaning of the input fields:

Select function	Type in a subfunction from the list under the heading ID.
Operand	Type in a DD/DLBL name (optional for subfunction FO).
System	Default: Global.
	If you are not sharing COMSYS data sets among multiple Com-pletes, the easiest way is to use this default all the time.
	If you do share COMSYS data sets among multiple Com-pletes, all files used from a certain Com-plete should be defined as local objects for this Com-plete. Enter the patch character identifying the appropriate Com-plete. Maintenance on the "local" file definitions should take place only from the Com-plete they are assigned for, except when this Com-plete is not active. **Notes:** 1. Maximum care is recommended with global filedefinitions in a shared COMSYS; changing such a definition while the file is open in another Com-plete may cause unpredictible results. 2. If you set a file defined as global to BATCH status, it will appear to have BATCH status for any other Com-plete which tries to open this file. On the other hand, setting a file to BATCH status from one Com-plete does not close it in the other one.
Access method	Specify the access method used if different from the current default, which is indicated by highlighting (for CATALOG function only).

File Overview

The File Overview function enables you to display the files cataloged to Com-plete. Use the *operand* field to specify a name or prefix as a starting point of the file list.

The following figure illustrates the display given by the File Overview function (result of function *FO* with operand *):

```
FC:  D=Display  M=Modify  R=Remove  S=SHOWCB  B=BTCH  O=ONLN  P=OPEN  C=CLOSE
 10:03:06        TID    18          COM-5.1.          User MBE          21.04.97
 System: Global                -- File Overview --                          UFI1

 FC DDName     Type   Status Attributes   Users   Last Access           Note
 ------------------------------------------------------------------------------
 .  ADMF       VSAM   ONLN   R U A                closed
 .  DFHTSD     VSAM   ONLN   R U A                closed
 .  QAAUT      VSAM   ONLN   R U A                closed
 .  QAAUTX     VSAM   ONLN   R U A                closed
 .  QAAUTY     VSAM   ONLN   R U A                closed
 .  QAFINC     VSAM   ONLN   R U A                closed
 .  QAPER      VSAM   ONLN   R U A                closed
 .  QAPERX     VSAM   ONLN   R U A                closed
 .  QAPERY     VSAM   ONLN   R U A                closed
 .  SKUVSAM    VSAM   ONLN   R U A   S            closed





 Reposition to:                                                            END
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
 Refresh           End   Mode                    Forwd
```

Meaning of the fields by column heading:

Column	Description
FC	Input field to call a function for the appropriate file. Two of the functions available on the File Catalog Maintenance menu can be called by entering the appropriate function code in this field:
	The first line of the screen lists valid function codes. You can enter function codes in more than one line. When you press ENTER, these function requests are satisfied successively.
DDName	DD/DLBL-name of the file.
Type	Access method type of the file (VSAM, BDAM or ISAM).
Status	Indicates whether the file is currently in ONLINE or in BATCH status.
Attrib	Types of access allowed for this file in Com-plete: Retrieval (R), Update (U), Add (A), as well as additional options: Capture (C) and Serialization (S).
Users	For a VSAM file currently open: The number of ACBs in user programs currently open for this file.
Last access	For files currently opened in Com-plete: date and time of last access. Files not currently opened in this Com-plete are marked "closed". If you are dealing with "local" file definitions assigned to a different Com-plete (see main menu parameter System), this field containes the string "other system" indicating that no information is available from this Com-plete.
Note	Shows the result after a subfunction call.

You can scroll the catalog forward PF8 or restart the display from any file name by entering a name or prefix in the field prompted *Reposition to*. You can switch the mode (press PF4) to display data set names and disposition (z/OS) or user catalog name (VSE):

```
FC:  D=Display  M=Modify  R=Remove  S=SHOWCB  B=BTCH  O=ONLN  P=OPEN  C=CLOSE
 10:27:25        TID    18          COM-5.1.          User MBE          21.04.97
 System: Global                -- File Overview --                          UFI1

 FC DDName     T   Data Set Name                                DISP    Note
 ---------------------------------------------------------------------------
 .  ADMF       V
 .  DFHTSD     V
 .  QAAUT      V   SKU.SYSF.AUTO                               SHR
 .  QAAUTX     V   SKU.SYSF.AUTOX.AIX                          SHR
 .  QAAUTY     V   SKU.SYSF.AUTOY.AIX                          SHR
 .  QAFINC     V   SKU.SYSF.FINANC                             SHR
 .  QAPER      V   SKU.SYSF.PERSON                             SHR
 .  QAPERX     V   SKU.SYSF.PERSX.AIX                          SHR
 .  QAPERY     V   SKU.SYSF.PERSY.AIX                          SHR
 .  SKUVSAM    V   SKU.TEST.VSAM                               SHR




 Reposition to:                                                          END
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
 Refresh           End   Mode                    Forwd
```

Display Catalog Entry

This function enables you to display all attribute information cataloged for a file.

When the Display Catalog Entry function is invoked from the File Catalog Maintenance Menu, the DD/DLBL name must be specified in the*operand*field. Alternatively, you can use the line command D from a list of catalog entries (see the *File Overview* section).

The display of catalog entry information is the same as for the CATALOG function. The difference is that the DISPLAY function is "read" only (no modification possible). For a description of the information fields, see the section *Catalog a File Entry* below.

Catalog a File Entry

The CATALOG function enables you to add entries to the Com-plete file catalog. This function is not available from the File Overview menu. The DD/DLBL name to be cataloged must be specified in the operand field of the File Catalog Maintenance menu.

One of the valid access methods shown on the File Catalog Maintenance menu is always highlighted, indicating the current default. You can overwrite the default by specifying the required value in the Access Method field (z/OS only).

The access method dependent map is displayed, in which you can specify the file parameters to be included in the catalog entry. All parameters on the map are initialized with default values. Every time a DDN is cataloged, displayed, or removed, its values are saved and taken as current default for the corresponding access method. This information is lost when the UUTIL FM function is terminated.

After specifying the parameters, press ENTER to run a check for any invalid values or PF5 to check and save. In the latter case, if no errors are found, the entry is added to the Com-plete file catalog and control is returned to the File Maintenance menu. If an error is detected, an error message is displayed and the invalid value is indicated by the cursor position. Correct the value and press PF5 again.

A description of the access method dependent cataloging screen follows below.

Cataloging a VSAM File

The following map is displayed when cataloging a VSAM file:

```
10:47:10        TID    18          COM-5.1.          User MBE          21.04.97
                         -- File Catalog Information --                     UFI3
 System: Global
 DDName= MBETEST                      VSAM
 ------------------------------------------------------------------------------
 DSName:....

 DISP:......          (OLD/SHR)                   MACRF=
 Password:..          (if protected)                  ,     ,      (KEY,ADR,CNV)
 Retrieval:.   (Y/N)                                  ,     ,      (SEQ,SKP,DIR)
 Update:....   (Y/N)
 Add:.......   (Y/N)                                                (NUB/UBF)
 Capture:...   (Y/N)                                                (NRM/AIX)
 Serialize:.   (Y/N)                                                (NSR/LSR)
 BUFSP=      0 Kbyte                                                (NDF/DFR)
 BUFND=      0                                                      (NRS/RST)
 BUFNI=      0                                                      (DDN/DSN)
 STRNO=      0                                                      (NCI/ICI)
 BSTRNO=     0                                                      (NIS/SIS)
 SHRPOOL=    0                                                      (NFX/CFX)

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

Meaning of the parameters:

Parameter	Description
DSName	Specifies the one-to-fourtyfour character data set name of the data set to be allocated to this DDN at OPEN time.
DISP	(z/OS only.) Specifies the disposition, OLD or SHR, to be used for data set allocation.
UCAT	(VSE only.) Specifies the one-to-seven character file name of the appropriate VSAM user catalog.
Password	For password protected VSAM clusters only. Specifies the password to be used by Com-plete when an OPEN for the VSAM cluster is issued, and that is required of each application program requesting access to the file.
Retrieval	Y Yes retrieval of records is to be performed. N No retrieval is not allowed.

Parameter	Description
Update	Y Yes records can be updated and deleted. N No records cannot be updated or deleted.
Add	Y Yes: records can be added to the file. N No record additions are not allowed.
	Note: At least one of the access types retrieval, update or add must be allowed.
Capture	Y indicates that 'after' images of any updates, deletions or additions to this file are to be written to the Com-plete capture data set before control is returned to the application program. Note that the 'after' image for a record deletion consists of the Relative Byte Address (RBA) of the deleted record. Default: N Capture images will not be taken.
Serialize	Y Com-plete performs serialization of requests against this file. N no serialization is to be performed by Com-plete. Applications issuing terminal I/O functions while having uncompleted VSAM request(s) active are not cancelled; instead, only a warning message is written to the console.
	Y must be specified for files used by NATURAL applications, but not for CMEDIT.
	Com-plete uses an internal resource with compare-and-swap logic where applicaple to allow only one output-type request at any one time to be active against this file. For example, a GET-for-update request causes any requests issued by other users against the same file to wait until the first request is completed by an appropriate PUT, ERASE, or ENDREQ. Multiple read-only requests can run in parallel with each other, but will be serialized with output-type requests. Applications issuing terminal I/O functions while having uncompleted VSAM request(s) active are cancelled with a thread dump being produced. Setting this option avoids requests from failing due to the "buffer / controlinterval in use" condition (VSAM reason code X'14'). On the other hand, it requires all application programs accessing the file to observe certain rules to avoid running into deadlocks (A holds file X and waits for file Y, B holds Y and waits for X) or being cancelled when issuing terminal I/O.

The following are the ACB parameters to be used for the central ACB Com-plete builds for the given DD/DLBL name. Note that Com-plete uses this one ACB to pass to VSAM all requests issued from any application program against any ACB with the same DD/DLBL name. Careful choice of all parameter values is recommended, taking into account the maximum possible number of users simultaneously accessing the file. For more detailed information about these parameters, their impact on resource usage and performance, and about VSAM tuning, please refer to the description of the ACB macro instruction in the*VSAM Macro Instruction Reference* and to the *VSAM Administration Guide*.

Parameter	Description
BUFSP	Specifies the maximum amount of virtual storage in Kbytes to be used for data and index I/O buffers.
BUFND	Specifies the number of I/O buffers VSAM is to use for transmitting data.
BUFNI	Specifies the number of I/O buffers VSAM is to use for transmitting the contents of index entries.
STRNO	Specifies the maximum number of simultaneous positioning requests that will be made from all terminals for this VSAM cluster.
BSTRNO	Specifies the initial number of strings to be allocated for this VSAM cluster when it is first opened.
SHRPOOL	For future use. Specifies the number of the local shared resource pool to be connected to the ACB. With the current version of Com-plete, only SHRPOOL 0 is supported.
MACRF	Specifies the processing options to be used with the file. The specified values must include all options used by any program accessing the file. The IN and OUT options cannot be specified explicitly; they are defined by the retrieval/update/add parameters. Note that there are groups of options, which can be specified in any combination (shown on the map separated by commas) and groups of alternative options (separated by a slash ’/’) where only one option of each group can be specified.

Note that the RMODE31 parameter cannot be specified, as it has no influence on application program processing. Com-plete forces RMODE31=ALL where applicable, thus causing buffers and internal VSAM control blocks to reside above the line.

Cataloging an ISAM file (z/OS only)

The following map is displayed when cataloging an ISAM file.

```
15:06:13        TID     6          COM-5.1.          User MBE          21.04.97
                        -- File Catalog Information --                     UFI4
System: Global
DDName= MBETEST                      ISAM
-------------------------------------------------------------------------------
DSName:......
VOLSER:......        (if not cataloged)
DISP:........        (OLD/SHR)




Retrieval:...      (Y/N)
Update:......      (Y/N)
Add:.........      (Y/N)
Capture:.....      (Y/N)

Thread lock for adding records:  0



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

Meaning of the parameters:

Parameter	Description
DSName	Specifies the one-to-fourtyfour character data set name of the data set to be allocated to this DDN at OPEN time.
VOLSER	Specifies the six-character serial number of the volume containing this data set. Should be specified only if the data set is not cataloged.
DISP	Specifies the disposition, OLD or SHR, to be used for data set allocation.
Retrieval	Y Yes retrieval of records is to be performed. N No retrieval is not allowed.
Update	Y Yes records can be updated. N No records cannot be updated.
Add	Y Yes records can be added to the file. N No record additions are not allowed.
	Note: At least one of the access types retrieval, update or add must be allowed.
Capture	Y indicates that ’after’ images of any updates, deletions or additions to this file are to be written to the Com-plete capture data set before control is returned to the application program. Note that the ’after’ image for a record deletion consists of the Relative Byte Address (RBA) of the deleted record. Default: N Capture images will not be taken.
Thread lock for add	Number of the thread to which addition of records will be restricted. See the ISAM considerations above in this section.
	Default: 0

Cataloging a BDAM file (z/OS only)

The following map is displayed when cataloging an BDAM file.

```
15:14:55        TID     6          COM-5.1.          User MBE          21.04.97
                         -- File Catalog Information --                     UFI5
 System: Global
 DDName= MBETEST                      BDAM
 ------------------------------------------------------------------------------
 DSName:......
 VOLSER:......        (if not cataloged)
 DISP:........        (OLD/SHR)
 OPTCD:.......        (R/A)
 RECFM:.......        (F/FB)


 Retrieval:...        (Y/N)
 Update:......        (Y/N)
 Add:.........        (Y/N)
 Capture:.....        (Y/N)





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

Meaning of the parameters:

Parameter	Description
DSName	Specifies the one-to-fourtyfour character data set name of the data set to be allocated to this DDN at OPEN time.
VOLSER	Specifies the six-character serial number of the volume containing this data set. Should be specified only if the data set is not cataloged.
DISP	Specifies the disposition, OLD or SHR, to be used for data set allocation.
OPTCD	Specifies the value, R or A, to be used for DCB parameter OPTCD.
RECFM	Specifies the value, F or FB, to be used for DCB parameter RECFM. See the BDAM considerations above in this section.
Retrieval	Y Yes retrieval of records is to be performed. N No retrieval is not allowed.
Update	Y Yes records can be updated. N No records cannot be updated.
Add	Y Yes records can be added to the file. N No record additions are not allowed.
	Note: At least one of the access types retrieval, update or add must be allowed.
Capture	Y indicates that ’after’ images of any updates, deletions or additions to this file are to be written to the Com-plete capture data set before control is returned to the application program. Default N Capture images will not be taken.

Modify a Catalog Entry

The MODIFY function enables you to change any parameters of a file definition.

You are not recommended to modify parameters of a file while it is open online. Once you have changed the catalog entry, no more information is available about the parameter values currently in effect for the open file. Therefore, you are given a warning notification each time you use this function against an open file. See also the notes above concerning "global" and "local" definitions if you share COMSYS data sets among multiple Com-pletes.

Contents of the catalog entry are displayed in the same format as for the CATALOG function (see above). Press PF5 to confirm your changes or PF3 to cancel the request.

Remove a Catalog Entry

The REMOVE function enables you to delete an entry from the Com-plete file catalog.

Contents of the catalog entry are displayed in the same format as for the CATALOG function (see above). Press PF5 to confirm removal or PF3 to cancel the request.

Set File To BATCH Status

This function allows you to disable online access to application files allocated to Com-plete.

If the file is currently opened by Com-plete, it is now closed and deallocated. Unlike the CLOSE function described below, the BATCH function sets a flag causing subsequent OPEN requests issued against this file from application programs to be rejected until it is set back to ONLINE status. Batch jobs accessing the data set are free to use the required DISP parameter (SHR or OLD).

To ensure that the file is currently not in online use, use the FO function (File Overview) described above. This function tells you whether the file is currently open, the current number of users and the time of the last access.

Set File To ONLINE Status

This function enables you to allow online access to application files which are set to BATCH status.

The ONLINE function does not allocate the data set to Com-plete, and no availability check is performed.

You can use the OPEN function described below to check if the data set is available for online processing.

Open a VSAM File

The OPEN function enables you to open VSAM files.

Note:
Opening a file using this function does not release application programs from opening the file.

You can use this function for test purposes, for example, to ensure you have specified a valid combination of ACB parameters using the Catalog or Modify functions, or to ensure the data set is available for online processing. If the OPEN failes, the appropriate reason code is displayed.

Hint:

In most cases when an OPEN request fails, additional information is available on the operator console.

Close an Open File

The CLOSE function enables you to close and deallocate files currently opened in Com-plete.

A file is allocated to Com-plete and opened when it is first accessed by a Com-plete program and remains open until Com-plete is terminated. Consequently, storage used for I/O buffers and control blocks is not freed as long as Com-plete is active. This may lead to a high requirement of virtual storage, especially when VSAM files do not use shared resources.

In contrast to a CLOSE macro instruction used in an application program, the CLOSE function of this utility really closes the file, frees all storage reserved for I/O buffers and control blocks, and deallocates the data set from Com-plete.

The CLOSE function may also be useful when you want to change file parameters in the Com-plete file catalog and the file is currently open.

To ensure that the file is currently not in online use, use the FO function (File Overview) described above. This function tells you whether the file is currently open, the current number of users and the time of the last access.

SHOWCB ACB Information

When a VSAM data set is currently opened by Com-plete, information from its Access Control Block (ACB) can be displayed using the SHOWCB Information function.

The following figure illustrates an example display given by the SHOWCB Information function (the fields speak for themselves):

```
16:41:23        TID     6          COM-5.1.          User MBE          21.04.97
                        -- VSAM SHOWCB Information --                      UFI2

ACB= QAAUT       RMODE31=ALL      KSDS      DATA      INDEX
-------------------------------------------------------------------------------
Available space (in K) ...............        32         32
Number of extents ....................         1          1
Number of index levels ...............                    1
Control interval size ................      8192       8192
(Max) Record length ..................       800       8185
Key length and offset in record ......   8     0

Total records at OPEN time ...........        40          1
Total records now ....................        40          1
Since OPEN: Records retrieved ........         0
            Records updated ..........         0          0
            Records deleted ..........         0
            EXCP count ...............         0          0

I/O buffers requested ................                       LSR
I/O buffers allocated ................
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Refresh           End
```

For more information about the values returned by the SHOWCB macro instruction, please refer to the description of the SHOWCB macro instruction in the *VSAM Macro Instruction Reference* documentation.

Function HC - Hardcopy Device Definition

This function enables you to change the setting of your terminal's screen-to-hardcopy target definition. This definition is used as the default for the UCOPY utility and for the PRINT functions of many other Com-plete utilities such as UEDIT and UPDS.

If you select the HC function from the UUTIL menu, a screen is displayed showing the current screen-to-hardcopy definition for your terminal, and lists all of the hardcopy devices currently defined to Com-plete at your installation. For example:

```
17:16:50     TID     4           COM-5.1.            User ID MBE        21.04.97
                                                                            USCH
                              Select Hardcopy device

  Device name..:
  TID..........:(     0 )
  Type.........:
  Node.........:     0                    Save screen in SD-file.: N (y/n)

 Currently defined hardcopy devices
 00001 SYSOUT   BATCHC









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12-
                   Exit        Updt              Next
```

The current hardcopy device is indicated by TID number and name (for example, VTAM LU name, ACC-ESS name). The list of available hardcopy devices shows, from left to right, the TID number, name, device type, node ID, and status.

Possible values for status are:

ACT	ACTIVE: currently printing or available for printing.
ERR	ERROR status: not available until error is eliminated.
DEL	The DELETED option in the TIB entry is set to YES. This device is not available for hardcopy generation. It can be used, for example, as a dummy printer (see the Com-plete System Programming documentation). If you need this printer for your hardcopies, see your system programmer.

If there are more defined hardcopy devices than fit on one screen, press ENTER to scroll the list.

Press the CLEAR key to return to the UUTIL menu.

Setting the Hardcopy Definition

To define a target hardcopy device, you can use the input fields at the top of the hardcopy definition screen as follows:

- *Specify any Com-plete terminal:*
  Enter the required TID number in the field labeled *Specify Hard Copy TID* and press ENTER. The specification is then indicated as the current hardcopy device.

- *Delete a hardcopy definition:*
  Enter zero in the field labeled *Specify Hard Copy TID* and press ENTER. The current hardcopy specification is deleted.

- *Create a UEDIT workfile:*
  If you specify Y in the Save screen for edit field and press ENTER, the current target hardcopy device is specified as an edit workfile.

Note:
You can only use the UEDIT work file option with the UCOPY utility, which you can invoke by either entering *UCOPY or pressing the hardcopy key. You can then retrieve the screen image placed in a editor work file by invoking UEDIT with the RECOVER option. COM-PASS users must invoke UEDIT on the same level as the program that was active when they invoked UCOPY.

Function HE - Help System

Com-plete provides an online help facility that allows you to display information about the main Com-plete functions. Help texts can consist of Com-plete-supplied help data or installation-supplied help information.

If you select the HE function from the UUTIL menu, the help selection menu is displayed:

```
09:48:08        TID     2           COM-5.1.          User MBE          22.04.97
 System: Lcl *                ---  HELP  menu  ---                          UHE0

 Subject..........................  Fc     Subject..........................  Fc
 ---------------------------------  --     ---------------------------------  --
 Com-plete Utilities .............   1     COM-PASS Main Menu ..............  13
                                     2     COM-PASS utility menus ..........  14
                                     3     COM-PASS security system ........  15
                                     4     COM-PASS parallel transactions ..  16
                                     5     COM-PASS recovering edit sessions  17
                                     6     COM-PASS message switching ......  18
                                     7                                        19
                                     8                                        20
                                     9                                        21
                                    10                                        22
                                    11                                        23
                                    12                                        24

                            Please select an item:


 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End                     bwd   fwd                     Quit
```

To select a topic on which you wish to see help, type its number in the input field labeled *Please select an item* and press ENTER.

The option Com-plete Utilities displays a list of utilities on which there is help available:

```
13:36:04        TID     6            COM-5.1.         User SAGAWW       05/21/97
 System: Lcl 8                ---  HELP  menu  ---                          UHE0
 Subject..........................  Fc     Subject..........................  Fc
 ---------------------------------  --     ---------------------------------  --
 UDS - Disk Space handling (OS) ..   1     ULIB - Online Pgmlib handling ...  13
 UPDS - Dataset Maintenance (OS) .   2     UM - Msg switching (direct cmnds)  14
 UEDIT - The fullscreen Editor ...   3                                        15
 UED - Line Editor ...............   4                                        16
 UMAP - Online map maintenance ...   5                                        17
 UCTLUT - System Control Utilities   6                                        18
                                     7                                        19
                                     8                                        20
                                     9                                        21
                                    10                                        22
                                    11                                        23
                                    12                                        24
                            Please select an item:  0
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End                     bwd   fwd                     Quit
```

You can also display help for most of the listed utilities by calling the help function or entering a question mark (?) from within that utility.

In addition to multiple screens summarizing each utility, a glossary may be provided that defines the terms used on the help screens and gives a description of the syntax.

To return to the previous help level, press PF3 from any screen. To leave the help facility from any help screen, press PF12.

Function HM - Help Maintenance (Administrator)

The help maintenance allows modification of the Com-plete Online Help system to your installation's requirements. When you select the HM option from the UUTIL menu, the help maintenance submenu is displayed:

```
18:03:28        TID     4           COM-5.1.         User SAGAWW       05/06/97
System: Lcl 8             ---  HELP Maintenance  ---                       UHM0
Fc Description..............  Operands     Fc Description..............  Operands
-- ------------------------- --------     -- ------------------------- --------
AH Add help item ...........              LH List help items .........
DH Display help item .......
MH Modify help item ........
PH Purge help item .........



Fc:           (1) Item....            Sub-item   1   Itemtype   /
              (2) Language   0
              (3) System..





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

Meaning of the input fields:

Field	Meaning
FC	Enter one of the available function codes from the column headed FC .
Item	The eight-character name identifying the help item.
Sub-Item	A number further identifying the sequence within the Item .
Item Type	Type of help. Two characters are required. Possible options for the first character:
	M Menu. Help menu from which other help items can be selected. T Text Help text. G Glossary. Defines terms used in the help text with the same name. When the user presses PF10 from the help text, the glossary for that help screen is displayed.
	Possible options for the second character:
	S System item Helps of this type are supplied by SOFTWARE AG and should not be modified. U User item. Use this type for your own help data.
Language	Help data for up to 255 languages (1 to 255) can be defined.
System	This is the Com-plete patch character of the system under consideration. You are recommended to keep help data ’global’ to all Com-pletes at your site, but it is possible to build different help structures for one or more Com-pletes.

Listing Help Items

Select function LH on the main menu, specifying additional criteria for a more selective display:

- A prefix followed by an asterisk (*), for example, ED*, to view all items with names starting ED.
- The type of help item you want to see (M, T and/or G). An asterisk selects all types.

The following shows a sample output of the LH function (item U*, all types):

```
14:10:04        TID    10           COM-5.1.          User SAGAWW       30.11.97
 System..: Lcl *            ---  HELP Maintenance  ---                       UHM0
 Language:   1 / System            List Items

 Sel Item.... Type.... Sub  Sel Item.... Type.... Sub  Sel Item.... Type.... Sub
 --- -------- -------- ---  --- -------- -------- ---  --- -------- -------- ---
     UCTLUT   Menu       1      UMAPCREA Text       1                          0
     UCTRL    Menu       1      UMAPDELE Text       1                          0
     UDS      Glossary   3      UMAPDYNT Text       1                          0
     UED      Glossary   3      UMAPEDCO Text       1                          0
     UEDIT    Menu       1      UMAPLPSS Text       1                          0
     UEDITCAL Text       4      UMAPMODI Text       1                          0
     UEDITPFK Text       1      UMAPSAVE Text       1                          0
     UEDITPRF Text       1      UMAPTCCS Text       1                          0
     UEDITSYN Text       1      UPDS     Glossary   1                          0
     ULIB     Glossary   4      UQ       Text       1                          0
     UM       Glossary   2      USPOOL   Text       1                          0
     UMAP     Menu       1      UTILMENU Menu       1                          0
     UMAPARRD Text       1                          0                          0
     UMAPCOPY Text       1                          0                          0
```

Enter D(isplay), M(odify), P(urge) in the *Sel* column to manipulate specific items.

Help Menus

If you specify an M-type help, the help menu definition screen is displayed, for example:

```
18:07:53        TID     4            COM-5.1.          User SAGAWW       05/06/97
 System: Lcl 8              ---  HELP Maintenance  ---                       UHM1

                              Edit Menu MAINMENU   1


                 Item description................. Itemname Type
                 --------------------------------- -------- ----
                 COM-PLETE Utilities ............. UTILMENU M S









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Updat       bwd   fwd                     Quit
```

You can enter up to twelve help items in the menu, each representing one half of a help menu. This means that two menu items, one with an even*Sub-item* number and one with the following odd*Sub-item*number, are used to build one help menu page.

The input columns on the edit menu screen have the following meaning:

Column	**Meaning**
Item description	Short comment indicating the topic covered by this help item.
Itemname	The name of an existing help item. The hierarchy within the help structure must be clearly defined. This means that an item which is to be used at a higher level than another must also have been defined first.
Type	Two characters must be specified. The first character defines the type of help. Valid types are M (Menu), T (Text) and G (Glossary) (see above). Another type is:
	P Program. Can be used only in menus. When a user selects an item of this type, the program with the corresponding Itemname is called.
	The second character, after the slash, defines the item as being a S (System) or U (User) help (see above).

Help Text / Glossary

The following is an example of a T-type help. It shows the help text on the SET command, entered as free text. G-type helps are defined in the same way.

```
 18:10:42        TID     4           COM-5.1.          User SAGAWW       05/06/97
 System: Lcl 8             --- Edit HELP Text (T) ---      EDCMSET    1     UHM2


§Command  Parameter  Argument....  Command Description / Argument meaning.......
§-------  ---------  ------------  ---------------------------------------------
§SET§                              Set the parameter to the new status.
         §BLANKS     ON/OFF§       Fill columns 73-80 with blanks on save.
         §CURSOR     STAY/HOME§    Define positioning of cursor after input.
         §DSNAME     DSNAME§       Change current dataset name.
         §ESCAPE     char/ON/OFF§  Set escape character, ON (#), or OFF.
         §FILE       LIB(MEM)/VOL§ Define LIB, DSN, MEM, VOL with new values.
         §LIBID      LIBID§        Change current library id.
         §LISTCOL    N1 N2§        Set screen list columns.
         §LOWER§                   Switch to lower case mode.
         §MEMBER     MEMBER§       Change current member name.
         §NUMBER     N M§          Defines SEQ start (N) and increment (M).
         §NULLS      ON/OFF§       Fill line up with nulls (ON) or blanks (OFF).
         §PFK        N 'STRING'§   Assign string 'STRING' to PF Key 'N'.
         §SCAN       ABS/TAG§      Set scan mode to absolute or tag.


                     Default is always 'ON' when possible.                 §==>

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   Test              bwd   fwd                     Quit
```

Within the help structure, a help text is at the end of a branch of the help tree (the lowest level of help screens). The maximum number of levels within the help structure is 16. This means that a tree branch can consist of up to 15 menu levels before a help text.

Up to 256 pages of help text are possible for one subject. The online help facility allows the user to page help texts using PF7 and PF8.

A Glossary-type help text accompanies a Text-type help. It must have the same name as the Text-type help and is displayed when the user presses PF10 from the help text.

Function GL - Display Global Lib IDs

This function allows you to display the contents of the global library ID table (UEDTB1) as defined by the Com-plete system administrator. This is an expanded version of the information available with the FILES command under the UPDS user utility.

The following figure shows an example of a typical display invoked by selecting the GL option from the UUTIL menu:

```
16:21:59     TID     6            COM-5.1.           User ID SAGAWW     05/21/97
                                                                           UGLB
                  ----   Global Library ID Table   ----
 ID                Data Set Name                    Volume  Type
 --  ---------------------------------------------  ------  ---- ------- --------
 $$  PUB.SYSF.PROFILES                                      PO
 CP  COM.SYSF.DEV.PROJECTS                                  PO
 CS  COM.SYSF.DEV.SYSTEM                                    PO
 CW  COM.SYSF.DEV.WORK                                      PO
 XL  COM.SYSF.IV451.XA.LISTING                              PO
 ML  COM.SYSF.IV451.z/OS.LISTING                             PO
 SL  COM.SYSF.IV451.SP.LISTING                              PO
 NL  COM.SYSF.IV450.XA.LISTING                              PO
 LO  OPS.SYSF.COMLOAD                                       PO
 MG  OPS.SYSF.COMMGEN                                       PO
 PL  OPS.SYSF.PROD.LOAD                                     PO
 P2  SYSM.PROCLIB                                           PO
 SG  IPO1.GENLIB                                            PO
 SI  IPO1.INSTLIB                                           PO
 SM  SYS1.MACLIB                                            PO
 SP  SYS1.PROCLIB                                           PO
```

The libraries are listed by short ID, name, volume and type. Modifications are not possible in this screen. Libraries can be of the following type:

PO	partitioned dataset
PS	sequential file
PAN	PANVALET library
LIB	LIBRARIAN library

Function MM - Message Maintenance (Administrator)

This function allows modification of Com-plete messags definitions.

If you select the MM option from the UUTIL menu, the Message Maintenance menu is displayed:

```
15:14:40        TID     9          TEST-611          User COK          20.02.02
                         --- Message Maintenance ---                       UTM7

Fc Function..........................   Fc Function..........................
-- ----------------------------------   -- ----------------------------------
DI Display message                      LS List messages
ED Edit (Update) message                TR Translate to and other language
-- ----------------------------------   -- ----------------------------------

Function ....:

Group .......:
Msg number ..:    0     to: 9999
Language ....:    1     to:    1









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

Meaning of the input fields:

Field	Meaning	
Function	Available functions are:	
	DI	Display messages
	ED	Edit (update) message
	LS	List messages
	TR	Translate to another language
Group	Group to which the message belongs. This is usually the utility name or a code for internal Com-plete functions (for example, UPF, ZSR,ULG,ABS, ...).	
Msg Number	Message number to be displayed (single message), or a range to restrict the message display. Valid range is 0 to 9999.	

```
16:20:00        Tid    12          TEST-611          User MBE          13.11.00
System:                  --- Message Maintenance ---                       UTM8

MsgId.: ULG0001                 Language:   1

....*....1....*....2....*....3....*....4....*....5....*....6....*....7.
Logon failed for TID $1 (LU=$2): Terminal I/O error in U2QUERY









z/OS WTO Routing codes:  1234567890123456    Descriptor codes:  1234567890123456
                        ................                        ................
Send to terminal in conversation.: .     Abend the program ................: .
Retain in the message logfile....: X     Snap (take a dump) and continue...: .
Send to the system console ......: X     Do not take a dump ...............: .
CAPTURE the message .............: .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  End         Updat       Prev  Next        Purge       Quit
```

Update Message

To update a message, enter appropriate parameters in the input fields and press ENTER. If you are updating an existing message, the message definition is displayed.

You can modify various options concerning each message. The available options in the display speak for themselves. On the left, specify the message destination. On the right, specify system behavior with the messsage. An option is selected by marking it with an "X" and pressing PF5.

Below is an example of an existing message:

```
16:20:00        Tid    12          Com-plete          User MBE          13.11.00
 System:                   --- Message Maintenance ---                       UTM8

 MsgId.: ULG0001                 Language:   1

 ....*....1....*....2....*....3....*....4....*....5....*....6....*....7.
 Logon failed for TID $1 (LU=$2): Terminal I/O error in U2QUERY









 z/OS WTO Routing codes:  1234567890123456    Descriptor codes:  1234567890123456
                          ................                       ................
 Send to terminal in conversation.: .     Abend the program ................: .
 Retain in the message logfile....: X     Snap (take a dump) and continue...: .
 Send to the system console ......: X     Do not take a dump ...............: .
 CAPTURE the message .............: .

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Updat       Prev  Next        Purge       Quit
```

You can perform functions on messages by pressing one of the following PF keys:

PF Key	Function
PF3	Back to the previous screen. This can be the main menu (if you have selected function ED) or the selection list (if you have selected function SL). Any modifications not confirmed with PF5 are lost.
PF4	Modify additional data available for this message (for example, explanation, system action). See the section *Additional Text Data* below.
PF5	Saves modifications made to a message to disk.
PF7	Displays the message with the next lower number in the current group.
PF8	Displays the message with the next higher number in the current group.
PF10	Purges (deletes) any existing updates.
PF12	Leaves the MT subfunction.

Select Messages from a List

If you select function code LS from the message menu, a list of messages is displayed.

```
17:14:39        TID    15          COM-5.1.          User MBE          18.04.97
 System: Global          ---  Message Maintenance  ---                      UTM5

 S Grp/Msg Text
 - ------- ---------------------------------------------------------------------
   OPC0000 &CMON IS INITIALISED
   OPC0001 $1 $2 COMPLETED.
   OPC0002 $1: $2 IS NOT LOGGED ON
   OPC0003 UNRECOGNIZED COMMAND: $1
   OPC0004 $1 $2 $3ID=$4 - DEVICE NOT SUPPORTED.
   OPC0005 INVALID EOJ COMMAND FORMAT.
   OPC0006 $1 $2 - TID, LID, OR GROUP INVALID OR NOT DEFINED.
   OPC0007 $1 $2 - ALREADY ACTIVE.
   OPC0008 $1 $2 COMMAND INVALID FOR REMOTE TID.
   OPC0009 $1 $2 ALREADY ALLOCATED TO Com-plete.
   OPC0010 $1 $2 MULTIPLE TERMINAL CANCEL NOT ALLOWED.
   OPC0011 $1 $2 COMMAND INVALID FOR LID.
   OPC0012 $1 $2 - CANCEL DELAYED; IN Com-plete OR PV PROGRAM.
   OPC0013 $1 $2 TID/LID MUST BE IGNORED PRIOR TO DELETING.
   OPC0014 $1 $2 NOT ALLOCATED TO Com-plete.

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

You can enter any character in the S column preceding the desired message to display the message.

Function MO - Monitor Control (Control User)

The Monitor Control function allows you to monitor and manipulate the status of the running Com-plete session. It provides information about Com-plete subtasks, buffers, terminals, users, etc. In addition, Com-plete operator commands can be entered directly.

When you invoke the MO function, the following menu is displayed:

```
17:21:28        TID     8          COM-5.1.          USER MBE          07.10.98
                      --- Com-plete CONTROL FACILITY ---                   UCT1

Function             FC     Function             FC     Function             FC
-------------------- --     -------------------- --     -------------------- --
ADABAS interface ... AI     Program lookaside .. IP     Task groups ........ PG
ADABAS statistics .. AS     Program in-stor dir. OP     Task list .......... PL
Buffer activities .. BA     Resident programs .. RP     Thread groups ...... TG
Common storage ..... CS     Find program ....... FP     Thread subgroups ... SG
DB2 thread status .. D2     Queue overview ..... QO     Thread list ........ TL
FBPM statistics .... FB     Roll activities .... RA     User activities .... UA
Message log ........ CM     Server overview .... SO     User status .........US
                            Terminal overview .. TO     VSAM statistics .... VS
                            Terminal status .... TS
                                                        Print statistics ... PR

Select function ....
Operand ............

Operator command:

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

You can select any of the MO subfunctions by entering the appropriate function code in the Select Function field and, optionally, an operand in the Operand field.

You can also enter Com-plete operator commands in the Operator Command field of this and any other screen of the MO function. The Monitor function then directly switches to the CM subfunction. For available operator commands, see the Com-plete Operator Commands documentation.

All subfunction screens have the same PF key assignments:

PF key	Function
PF3/CLEAR	Leave the Monitor Function
PF7	Scroll up
PF8	Scroll back
PF10	Scroll left
PF11	Scroll right
ENTER	Refresh screen

A direct switch to other subfunction displays is invoked by entering the appropriate function code.

The following table summarizes the Monitor information subfunctions:

Subfunction	Meaning
AI	Displays ADABAS interface data.
CM	Displays the last Com-plete console messages.
QO	Displays Queue Overview.
RA	Displays roll activities.
TO	Displays an overview of Com-plete terminals.
TS	Displays the status of a specific terminal.
TG	Displays the status of all Com-plete Thread groups.
SG	Displays the status of all Com-plete Thread subgroups.
TL	Displays Com-plete Thread list.
PG	Displays the status of all Com-plete processor groups.
PL	Displays Com-plete processor list.
UA	Displays user activities and accounting data.
US	Displays user status.
BA	Displays buffer activities.
CS	Displays an overview of Common Storage usage.
FB	Lists active fixed buffer pools and statistics.
FP	Displays location of load modules
IP	Displays usage of programs in lookaside buffer.
OP	Displays the instorage program directory.
PR	Print information item.
RP	Displays resident program usage.
SO	Displays Com-plete Server overview.
D2	Displays Com-plete DB2 Server data.
VS	Displays VSAM usage statistics.
AS	Displays ADABAS statistics.
TU	Displays threadstorage information.

The subfunctions are explained below in alphabetical order.

AI Subfunction

The AI subfunction displays information saved in the Com-plete/ADABAS interface.

For every data base ID, the AI information screen contains data such as:

- Data base number;
- Parameter setting for this data base (ADAROLL, as set in the sysparms);
- Total number of ADABAS calls;
- Total number of ROLLOUTs;
- Percentage of ROLLOUTs compared to the total number of calls;
- Average Responsetime per Database call.

```
14:59:35        TID   124          -DAEFCO-          User COK       20.02.02
                            --- ADABAS Interface ---                    UCT5

  DBid.  SVC   ADAROLL ADACAL ADALIM    TotalCalls  TotRoll (in %)  AvgRTime
 *** TOP OF DATA ***
      0  249   .ALWAYS      0      0             0
      1  249      0.10     20   4096             7        1  14.29     0.000
      9  249      0.10     20   6000        319970    15214   4.75     0.001
     10  249      0.10     20   4096         72461     3471   4.79     0.003
     12  248      0.10     20   4096        422624    20125   4.76     0.002
     14  248      0.10     20   4096            20        1   5.00     0.052
     18  248      0.10     20   4096             6        1  16.67     0.016
     22  248   .ALWAYS     20   4096             0
     24  248      0.10     20   4096             6        1  16.67     0.015
     26  249   .ALWAYS     20   4096             0
     29  249      0.10     20   4096          1854       85   4.58     0.043
     31  249      0.10     20   4096             4        1  25.00     0.172
     32  241   .ALWAYS     20   6000             0
     34  249      0.10     20   4096            58        6  10.34     0.032
     38  249      0.10     20   4096            11        1   9.09    13.714
 ------------------------------------------------------- Active Users: 00123
 Select function:      Operand:
         Command:
```

AS Subfunction

The AS subfunction displays statistics about a specific ADABAS database. In order to interpret these statistics, consult the section on ADABAS in *Software Interfaces* in the Com-plete System Programming documentation.

```
13:19:41        TID    13          COM-5.1.          User MBE          12.04.97
                             - Database Statistics -                        UCTB

                              Database ID ...   177

  Adabas SVC Nr....   249
  ADAROLL .........    -1 (0=Always, -1=No)
  ADACALL .........    10
  ADALIMIT ........  4096


  Rolls for ADAROLL ..        0            ADAROLL exceeded ...        0
  Rolls for ADACALL ..        1            ADACALL exceeded ...        1
  Rolls for Priority..        0            Response before roll       18
  ----------------------------             ----------------------------
  Total Rolls ........        1            Total Calls ........       19

  Thread Posts .......        0            Average elapsed time...     8
  Commands without ...        2                (Millisec.)
     'R' option
 ------------------------------------------------------------------------------
 Select function:      Database ID:  177
         Command:
```

The information items are explained below:

Field	Meaning
Database Id	The ADABAS database ID (0 - 65535).
Adabas SVC Nr	The ADABAS SVC (Router) used for communication with this database.
ADAROLL	The value specified for the ADAROLL parameter for this database.
ADACALL	The value specified for the ADACALL parameter for this database.
ADALIMIT	The value specified for the ADALIMIT parameter for this database.
Rolls for ADAROLL	The number of times an application was rolled out due to ADAROLL being exceeded and the thread was required by another program.
Rolls for ADACALL	The number of times an application was rolled out due to ADACALL being exceeded.
Rolls for priority	The number of times an application was rolled out while waiting for an ADABAS call to complete and the thread was required by a task with a higher dispatching priority.
Total rolls	The total number of rollouts performed by applications waiting for an ADABAS call to complete.
ADAROLL exceeded	The number of times that ADAROLL was exceeded by an application waiting for an ADABAS call to complete.
ADACALL exceeded	The number of times that ADACALL was exceeded by an application waiting for an ADABAS call to complete.
Response before roll	The number of times that the ADABAS database responded before a rollout operation had to be performed.
Total calls	The total number of ADABAS calls issued against this database.
Average elapsed time	The average elapsed time (in milliseconds) of a call issued against this database.
Thread posts	The number of times the thread ECB was posted while an application was waiting for an ADABAS call.
Commands without ’R’ option	The number calls issued which could have caused the application to wait for a held record but the command did not have the "return on held" option set.

BA Subfunction

The BA subfunction displays information about various Com-plete buffers such as:

- Buffer name;
- Buffer size;

- Buffer usage;
- GET requests;
- GET fails.

```
13:36:13        TID    28          COM-5.1.          User MBE          30.08.97
                           --- Buffer Activities ---                       UCTA

Buffer name       Size         Used        Max used    Get requests   Get fails
*** TOP OF DATA ***
TIBTAB             19K       6K ( 30%)       6K ( 30%)         44            0
ROLL             4000K    1617K ( 40%)    1645K ( 41%)       1052            0 ‌‌‌
```

The number of messages available for display is set via the WTOBUFFERS sysparm.

CS Subfunction

This subfunction displays statistics about the usage of Com-plete Common Storage (COMSTOR).

```
15:04:01        TID     5          COM-5.1.           User MBE          12.11.97
                              - Common Storage -                            UCTI

 Name....  Creator. Address. Length..  LastUser  lReq/Flag  lastReq Date + Time
 *** TOP OF DATA ***
 NAT22     COK      0A07CF68 00000010  COK        PUT FREE  1997/11/12 15:04:00









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select Function:      Operand:                                  PF7=up PF8=down
         Command:
```

Meaning of the information items according to column heading:

Heading	Meaning
Name	The name assigned to this specific COMSTOR area by the creator.
Creator	The User ID that caused the COMSTOR area to be allocated.
Address	The address of the Common storage area.
Length	The length (in hexadecimal) as requested by the COMSTOR GEN function.
LastUser	The User ID which caused the last access to the COMSTOR area.
lReq/Flag	The last request type (GET, PUT, ...).
	Status Flag:
	F free H held by Roll-for-event
LastReq Date + Time	Date and time of last access to the COMSTOR area.

D2 Subfunction

This function displays the status of all DB2 Threads.

```
17:34:31        TID     5          TEST-511          User MBE          07.10.98
                             -- DB2 connections --                         UCTI

No   TaskGrp   Status    Userid    SSID  Plan      Request  DB2calls  AvgTime
*** TOP OF DATA ***
001  DEFAULT   WAIT USR  MBE       DB2   C510PLAN  SQL-CALL      52     0.021 ‡ ‡ ‡
```

Heading	Meaning
No.	The sequence number of the thread, up to the number of threads as defined in the SERVER sysparm.
TaskGrp	Name of the Task Group.
Status	Status of the thread. Possible values: WAIT WRK: Thread free, not allocated to a user WAIT USR: Thread allocated to user, awaiting action. ACT. DB2: Currently processing a DB2 request.
Userid	User ID owning the DB2 thread.
SSID	Subsystem Id this thread is connected to.
Plan	Current plan name.
Request	Last processed request. Possible values: SQL - CALL: A SQL call has been issued CAF request code: A CAF request has been issued
DB2 calls	The number of DB2 calls issued from this thread.
AVG time	The average elapsed time for one DB2 request.

FB Subfunction

The FB subfunction allows you to list the currently active fixed buffer pools and display statistics about those buffer pools. If no parameter is provided in the *Operand* field, the currently active buffer pools are listed in a display similar to the following:

```
09:37:06        TID    13          COM-5.1.          User MBE          12.04.97
                     --- Fixed Buffer Pool Statistics(*) ---                UCTQ
               ---Number of Subpools/Storage by Location---  --Subpools Totals--
 No. FBP-Name  <ANY...> <BELOW.> <DS....> <ECSA..> <CSA...>  Gets  Fails Exp Cnt
 *** TOP OF DATA ***
  1: INIT-BP   (Deleted)
  2: WORKPOOL   8  172K  9  104K  0        0        0          20K    0    0   0
  3: GRSRPOOL   2    5K  0        0        0        0          10K    0    0   0
  4: DEBUG      3  299K  0        0        0        0        3023     0    0   0
  5: COM-STOR   3   84K  0        0        0        0           0     0    0   0
  6: VSAM-CB    1    1K  1    1K  0        0        0           0     0    0   0
  7: VSAM-RSR   2    1K  0        0        0        0           1     0    0   0
  8: SERVER     0        4   24K  0        0        0          12     0    0   0
  9: VTAM       1    6K  0        0        0        0         129     0    0   0





 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:      Operand:             PF7=up PF8=down
         Command:
```

The line display for a buffer pool only contains information if the buffer pool is active. If it is not possible to display the information, a message indicating the reason is displayed in the line, for example: *Deleted*, as in the above example.

Meaning of the information items according to column heading:

Heading	Meaning
No.	Number of the buffer pool relative to the time of creation.
FBP-Name	Name of the buffer pool as specified at creation time via the DESC= parameter of the CMFBPM BPCREATE macro.
ANY*	Storage acquired above (if supported) or below the 16 MB line.
BELOW*	Storage acquired below the 16 MB line.
DS*	Storage acquired from a Data Space (if supported).
ECSA*	Storage acquired from the Extended z/OS Common Storage area (above the line).
CSA*	Storage acquired from the z/OS Common Storage area (below the line).
Gets	Total GETs for all subpools in this buffer pool.
Fails	Total FAILs for all subpools in this buffer pool.
Exp	Total Expansions for all subpools in this buffer pool.
Cnt	Total Contractions for all subpools in this buffer pool.

* Each of these columns may contain 2 values: the total number of subpools created for this buffer pool, and the total amount of storage used by all subpools in this buffer pool. These columns represent the data areas from where the stoirage is acquired for a subpool. This is a direct correlation of the LOC= parameter as specified in the CMFBPM SPCREATE macro when creating a subpool.

You can select a specific buffer pool to display a breakdown of subpool statistics: enter the buffer pool name or its relative number in the*Operand* field and press ENTER.

If a specific buffer pool is selected using the *Operand* field, there are two forms in which the buffer pool statistics can be viewed. You can toggle between the two forms using PF10 and PF11, while PF7 and PF8 can be used to scroll through the displays when there is more than one screen page of information. Where a numeric value exceeds the length of the area available to display it, it will be presented in K (value/1024) or M (value/1024/1024).

If you select a specific buffer pool, the following screen is displayed:

```
13:05:41        TID    12          COM-5.1.          User XJPO         10.03.97
                    --- Fixed Buffer Pool Statistics(1) ---                UCTJ
                          <.......Number of Elements.......>
Subpool  No Loc   Esize  Used(%)   Max Used  Base Curr High Gets  Fails Exp Cnt
*** TOP OF DATA ***
Buffer Pool Name - WORKPOOL
GENERAL   1 ANY     64     44( 69)   44( 69)   64   64   64   46     0    0   0
GENERAL   2 BELOW   64      3(  5)    7( 11)   64   64   64  271     0    0   0
GENERAL   3 ANY    128      6(  9)    8( 12)   64   64   64   25     0    0   0
GENERAL   4 BELOW  128      0(  0)    1(  6)   16   16   16    1     0    0   0
GENERAL   5 ANY    256      4(  6)    7( 11)   64   64   64   15K    0    0   0
GENERAL   6 BELOW  256      1(  3)    3(  9)   32   32   32  259     0    0   0
GENERAL   7 ANY    512      4( 12)    6( 19)   32   32   32   60     0    0   0
GENERAL   8 BELOW  512      0(  0)    4( 25)   16   16   16  208     0    0   0
GENERAL   9 ANY   1024      4( 12)    5( 16)   32   32   32    7     0    0   0
GENERAL  10 BELOW 1024      0(  0)    1(  6)   16   16   16    4     0    0   0
GENERAL  11 ANY   2048      2( 12)    2( 12)   16   16   16    2     0    0   0
GENERAL  12 BELOW 2048      0(  0)    1(  6)   16   16   16   17     0    0   0
GENERAL  13 ANY   4096      1( 25)    2( 50)    4    4    4    7     0    0   0
GENERAL  14 BELOW 4096      0(  0)    1( 25)    4    4    4    3     0    0   0
------------------------------------------------------------ Active Users: 00002
Select function:      Operand: 2           PF7=up PF8=down PF10=left PF11=right
        Command:
```

Meanings of the various column headings:

Heading	Meaning
Buffer Pool Name	This is the name of the buffer pool for which statistics are currently being displayed. This will only occur on the first screen display for a buffer pool if more than one screen is required to display all subpools in the buffer pool.
Subpool	This is the name of the subpool, which is an eight character indicator as to what the subpool is for. In the example above, the subpools are for general use and therefore have the same subpool name. In other allocated buffer pools, you will find differing names.
No	This is the number of the subpool in question. When subpools are allocated, they are given a sequential number which is one greater than the previously allocated subpool.

Heading	Meaning
Loc	This indicates where the buffer elements are allocated. There are three possible values that can be displayed here: BELOW storage is allocated below the 16 MB line. ANY Storage is allocated above the 16 meg line (XA Capable systems only) DS Storage is allocated in an ESA Data Space(ESA Capable systems only)
Esize	This is a number indicating the size of the elements allocated in the subpool.
Used(%)	This shows the number of elements in use in the subpool along with the percentage of the currently allocated elements thta this figure represents.
Max Used	This shows the maximum number of elements ever used in the subpool along with the percentage of the base allocation for the subpool. Note that if the subpool has ever expanded, these values will not be displayed.
Base	This is the initial number of elements allocated for subpool when it was built.
Curr	This indicates the current number of elements allocated for the subpool. Note that this will equal the number given for ’Base’ if there are no current expansions of a subpool.
High	This contains the highest number of elements ever allocated for the subpool.
Gets	This is the number of get requests that have been made against the subpool.
Fails	This is the number of get requests made against the subpool which have failed. A failure can occur for the following reasons: 1 The base allocation for the subpool is in use and the subpool does not allow expansions. 2 The current allocation for the subpool is in use and the subpool has expanded as often as is allowed. 3 The current allocation for the subpool is in use and an attempted expansion of the subpool failed. This could occur due to a shortage of storage in the region or partition.
Exp	This is the number of times that the subpool has expanded. If it contains ’***’, it indicates that the subpool has expanded more than 999 times. While expansion can be a normal part of the day, it is expensive and therefore subpools should be defined such that expansions are kept to a minimum.

Heading	Meaning
Cnt	This is the number of times the subpool has contracted. This may also contain '***' indicating that contraction has occurred more than 999 times. Again, this is an indication that this subpool is trashing, as expansions and contractions are occurring regularly. In this case, the parameters causing the subpool to be built should be reviewed. Also, if expansions are occurring and no contractions are subsequently occurring, it indicates that the initial allocation for the subpool is not sufficient and should be changed or that elements of the subpool are not being freed for some reason.

If you press PF11 (scroll right) on a display for a specific buffer pool, the following screen is displayed.

```
13:05:13        TID    12          COM-5.1.          User XJPO         10.03.97
                    --- Fixed Buffer Pool Statistics(2) ---                UCTK
                         ---------Storage in Kbytes--------  --Buffer % Used--
Subpool  No Loc   Esize  Used(%)   Max Used  Base Curr High  R1 R2 R3 R4 R5 R6
*** TOP OF DATA ***
Buffer Pool Name - WORKPOOL
GENERAL   1 ANY     64      3( 75)    3( 75)    4    4    4   4  4  0 89  2  0
GENERAL   2 BELOW   64      0(  0)    0(  0)    4    4    4  99  0  0  1  0  0
GENERAL   3 ANY    128      1( 12)    1( 12)    8    8    8   0 16 68  8  0  0
GENERAL   4 BELOW  128      0(  0)    0(  0)    2    2    2   0 **  0  0  0  0
GENERAL   5 ANY    256      1(  6)    2( 12)   16   16   16   0  2  0  1  0 98
GENERAL   6 BELOW  256      0(  0)    1( 12)    8    8    8   0 46 29  0  5 18
GENERAL   7 ANY    512      2( 12)    3( 19)   16   16   16   0  8  2  0  0 83
GENERAL   8 BELOW  512      0(  0)    2( 25)    8    8    8   0  3 24  0  0 71
GENERAL   9 ANY   1024      4( 12)    5( 16)   32   32   32   0 43 29  0  0  0
GENERAL  10 BELOW 1024      0(  0)    1(  6)   16   16   16   0  0 **  0  0  0
GENERAL  11 ANY   2048      4( 12)    4( 12)   32   32   32   0 **  0  0  0  0
GENERAL  12 BELOW 2048      0(  0)    2(  6)   32   32   32   0 12 44  0 31 12
GENERAL  13 ANY   4096      1( 25)    2( 50)   16   16   16   0  0  0 **  0  0
GENERAL  14 BELOW 4096      0(  0)    4( 25)   16   16   16   0 **  0  0  0  0
------------------------------------------------------------ Active Users: 00002
Select function:      Operand: 2           PF7=up PF8=down PF10=left PF11=right
        Command:
```

The various sub headings on this display, which are the same as those on the first display, have the same meaning. The difference here is that the figure presented is in Kbytes. The values provided only related to the actual buffer subpool storage and do not include the storage required to maintain the buffer pool and it's subpools. For more information on this, please refer to *Resource Usage and Estimates* in the System Programming documentation.

Buffer % Used

The values in these columns provide an overview as to how much of the actual buffer is being used when it is obtaines. That is to say, even though a request may be satisfied by a buffer with a length of 64 bytes, it may only require 32 bytes; however, it will still have 64 bytes reserved for it. The intention is that the buffer subpools can be tuned based on the usage of the buffer pools and perhaps additional subpools defined and/or current allocations changed.

6 ranges are presented (R1 to R6). The figures below these ranges indicate the percentage of overall requests which only used that amount of the buffer. Where 100% of requests fall into a specific range, '**' will be seen as the range. The various ranges represent the following usages of the buffer. Note that if the buffer is utilised 100% it will not be reflected in any of the ranges.

R1	less than 50% of the buffer used.
R2	50% to 59% of the buffer used.
R3	60% to 69% of the buffer used.
R4	70% to 79% of the buffer used.
R5	80% to 89% of the buffer used.
R6	90% to 99% of the buffer used.

The figures should be used to eliminate major wastage of storage. For example, if a buffer subpool consistently has 100% (ie '**') in the first range, it indicates that all requests satisfied by this subpool could actually be satisfied with an elements size which is half the size or possibly less, of the current element size defined for the subpool. As all requests in this category are satisified in this way, the element size of the subpool should simply be halved. In a case where 50% of the requests satisified are in the 'R1' category, another buffer subpool with an element size half the currently defined size should be allocated. The number of elements for the new subpool should be half the number allocated for the current subpool while the scurrent subpools allocation should also be halved.

While this method could be used to fine tune the allocation of subpools and the sizes allocated, the storage savings must be weighed against the time spent fune tuning the system. Also, a change in system load can "undo" the newly tuned subpools and would require that the process be repeated.

FP Subfunction

The FP subfunction can be used to find out where a given program is loaded from. The name of the program must be entered in the operand field.

The search for the load module is performed in the same sequence Com-plete uses when loading a program requested by the terminal operator or by an application:

- resident programs
- LPA/ELPA (OS systems) / SVA and partition GETVIS area (VSE systems)
- program lookaside buffer
- COMPLIB load library chain.

Depending on where the program is first found, the FP subfunction results in the screen of the RP, IP, or OP subfunction being displayed. When the program is first found in the LPA/ELPA (OS) or SVA/partition GETVIS area (VSE), this fact is indicated by an appropriate message in the top line of the screen.

IP Subfunction

The IP subfunction displays statistics on the programs in the lookaside buffer. Its contents can be specified using the PGMLOOKASIDE sysparm.

For every program in the buffer, the display contains the name and size of the program and the number of loads:

```
13:23:35        TID    13           COM-5.1.          User MBE          05.10.97
                        --- Program Lookaside Buffer ---                    UCTG

    Name        Size     Loads                  Name        Size     Loads
 *** TOP OF DATA ***
   USTOR          10K       1
   USTACK         20K       4
   USTSF2          4K       4
   UEBP           45K
   UEDIT          13K
   UEPROF         35K
   UETABS          1K
   UXEEX2          1K
   UXEEX3          2K
   UPDS           53K





 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00002
 Select function:      Operand:                                 PF7=up PF8=down
         Command:
```

OP Subfunction

The OP subfunction displays the contents of the instorage program directory (ISD)

To avoid searching for a load module each time it is being loaded, Com-plete keeps the BLDL / LOADLIST information for the most recently used load modules in an instorage directory.

The size (number of entries) of the ISD can be specified using the PROGRAMISD sysparm.

For each load module in the ISD, the display contains the name and number of loads. If the number of loads is not indicated for a module, the appropriate load module has been requested, but does not exist.

```
10:37:14        TID    16          COM-5.1.          User MBE          05.05.97
                       --- Program Instorage Directory ---                  UCTF

  Name         Loads         Name         Loads         Name         Loads
*** TOP OF DATA ***
  UCT1F2            3        UPDSF2            1        USTRE1            1
  UCTRL             4        UPF1F2            2        UCT6F2            1
  UUT1F2           16        UPF0F2            2        KLSCSP            8
  USTKX1                     UPF               3        CSCI              8
  UCT3F2            2        ULOGX1                     ETSSRV            8
  UTM5F2            2        ULOG             29        UTM2F2            2
  UTM0F2            4        ULG0F2            3        UPRTF2            1
  UTMSG             3        ULM1F2            4        UEPDMN           64
  UUTEX1                     ULM3F2            4        UXEEX1
  UUTIL            13        ULMAF2            1        UEBP             22
  UXEEX5                     ULOGM             5
  UUPDX1                     UPF2F2            1
  UPDS             26        UHM0F2            1
  UDEFF2            3        UHELPM            1
  UDEF              4        UUT0F2            3
*** BOTTOM OF DATA *** ------------------------------------ Active Users: 00006
Select function:      Operand:                                  PF7=up PF8=down
        Command:
```

PG (Task ("Processor") Groups) Subfunction

This function will display a list of all task groups in the system. The following display is a sample of what will be displayed.

```
15:46:10        TID     5          COM-5.1.          User MBE          12.11.97
                           --- Processor Groups ---                        UCTN
                          Use counts  Processors   Q Counts
Grp name  Status     Pri  Curr  High  Act Hig All  Curr High
*** TOP OF DATA ***
DEFAULT   Active     248    13    18    2   2   2     0    3
NATURAL   Active     248     2     3    2   2   2     0    1









*** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
Select function:      Operand:                                  PF7=up PF8=down
        Command:
```

Grp Name

This is the name of the group for which the information is being displayed.

Status

This is the status of the task. The task group may be Active, Quiescing or Dormant. When a task group is active, Com-plete can allocate and deallocate tasks from that group at will. When the group is quiescing, any tasks that have previously been allocated will remain available until they are deallocated by all users using them. When the last task is deallocated, the task group will have deemed to be quiesced and the control block will be marked as dormant.

Pri

This is the priority of the tasks associated with the task group. Under z/OS and Hitachi systems, this reflects the actual operating system priority assigned to the tasks.

Use Counts - Cur

This is the number of users with the tasks currently allocated from the task group.

Use Counts - High

This reflects the highest number of users which had tasks allocated in this task group at any one time since the group was initialised.

Tasks - Act

This is the number of tasks which the installation wishes to have defined within the task group.

Tasks - Hig

This is the highest number of tasks active at any one time since the task group was initialised

Tasks - All

This is the number of tasks actually allocated for the task group. This may differ from the number of tasks active as when the number of tasks is to be reduced, the tasks must first be quiesced in order to let current users of the tasks to be deleted, to finish their work. These tasks and their associated resources will only be physically deallocated and disappear from this count when the last user deallocates the task and it becomes dormant. Tasks can be added or removed from task groups using the ’TASKS’ operator command.

Q Counts

Each task group has four work queues associated with it, each queue representing a priority from 0 to 3. The total values are provided for all queues in this display; if values for each individual display are required use the QO function. Because these values are changing as they are being collected, there is a *very* slight possibility that inconsistencies may appear in the figures displayed. For this reason, these figures must be taken as a good indication rather than absolute values.

Q Counts - Curr

This is the number of users currently on queues associated with this task group.

Queue Counts - High

This is the sum total of the highest number of users which were on each queue associated with this task group at any one time.

PL (Task ("Processor") List) Subfunction

This function will display a list of all tasks in the requested task group and their current status. If no task group operand is provided, the tasks for all task groups are displayed. The following display is a sample of what will be displayed.

```
15:50:28        TID     5          COM-5.1.          User MBE          12.11.97
                            --- Processor List ---                          UCTT
                    Use counts  Waits..           ...Current/Last Active...
 Grp name  Status   Curr  High  Cur Hig  LastOp   User id  Program  Tid.. L Time
 *** TOP OF DATA ***
 DEFAULT   A-Run       8    11    2   3  Readm    MBE      UUTIL        5 1
 DEFAULT   A-Wait      1     4    2   4  Coexit   COK      UPDS         3 6
 NATURAL   A-Wait      0     3    2   4  Eoj      COK      NAT22        3 6
 NATURAL   A-Wait      0     2    2   4  Wrts     COK      NAT22        3 6









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:      Operand:                                 PF7=up PF8=down
         Command:
```

Grp Name

This is the name of the task group of which the task in question is a member.

Status

This reflects the current status of the task. The status is a combination of two state indicators separated by a dash (’-’). The primary state indicator, which is the letter preceding the dash, indicates whether the task is Active, Quiescing or Dormant by the letters A, Q and D respectively. Active in this sense indicates that the task is available to do work. When it is quiescing, it will remain active long enough to finish any work which has been started by the task while dormant tasks cannot be used and will have no secondary state associated with them. The secondary states which may occur are as follows:

’Wait’ Status

This indicates that the task is waiting. In this state, the task is waiting on new work or on events requested by programs running in threads associated with it.

’Run’ Status

This indicates that the task is currently running a user program.

’Disp’ Status

This indicates that the task is going through it’s dispatching cycle either finishing off old work or looking for new work.

Use Counts - Curr

This is the current use count for the task. The use count includes the current user of the task, any users for whom a wait was issued on the task and any users with an affinity for this task.

Use Counts - High

This reflects the highest ever use count experienced for the task since it was initialised.

Wait Counts - Curr

This is the current wait count for the task. This reflects the number of events upon which the task is waiting and includes two standard events, those being that work has been queued to the task group work queues or to the task’s own work queue.

Wait Counts - High

This reflects the highest ever wait count experienced for the task since it was initialised.

Last Op

This is the last Com-plete op which was issued under control of the task.

Current/Last Active Userid

This is the userid of the current user active under control of the task or the last user to be active under control of the task if it has a secondary status of ’wait’. This may be blank in systems where logon is not forced and will generally be blank for ULOG sessions where the user is logging on.

Current/Last Active Program

This is the name of the program currently active under control of the task, or the last program to be active under control of the task if it has a secondary status of ’wait’. If the task has never been used, this will be blank, however, once it has been used, this will always contain a value.

Current/Last Active Tid..

This is the tid of the current TIB active under control of the task, or the last TIB to be active under control of the task if it has a secondary status of ’wait’. If the task has never been used, this will be blank, however, once it has been used, this will always contain a value.

Current/Last Active L

This is the level number on which the user currently active under control of the task, or the last user to be active under control of the task if it has a secondary status of ’wait’, is running. If the task has never been used, this will be blank, however, once it has been used, this will always contain a value. Level ’0’ will be displayed for a non-COM-PASS user, for a COM-PASS user who is running a program while something is stacked on all possible levels and for certain programs which must run on level 0 such as ULOG.

Time

When the task has a secondary status of ’Run’, this will reflect the time in seconds that this user has spent under control of the task.

PR Subfunction

The PR subfunction allows you to print the information items provided by most of the other subfunctions via a hardcopy device. Having specified a destination printer in the first screen, you can define from which subfunction you wish to print:

```
14:33:09        TID    13          COM-5.1.          User MBE          12.04.97
                   --- Print Com-plete Control Information ---              UCTP

                        Function                    Sel
                        --------------------------- ---
                        ADABAS interface data .....
                        Program instorage directory  _
                        Program lookaside .........  _
                        Resident program usage ....  _
                        Roll activities ...........  _
                        Buffer activities .........  _
                        FBPM statistics............  _
                        User activities ...........  _
                        Server overview ...........  _
                        DB2 thread status .........  _
                        Spooling system ...........  _
                        Savepool usage.............  _
                        VSAM statistics............  _

                        Select (mark) for printout

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   Quit  All   Confm
```

Select the item(s) you want to be printed by typing any character in the appropriate input fields, and press PF5 to confirm your selection. Pressing PF4 causes all items to be printed.

If you select "User Activities", an additional screen is displayed, allowing you to design the printout for this subfunction:

```
14:37:47        TID    13          COM-5.1.          User MBE          12.04.97
                   --- Printout Design For User Activities ---              UCTU

 Field                                 S    Field                                 S
 ------------------------------------- -    ------------------------------------- -
 User ID ........................... A    Total number of transactions ...... M
 LU-name ........................... B    Total number of EXCPs ............. N
 TID number ........................ C    Amount data sent to/from terminal . O
 Authorization code ................ D    Message switching/printout spooling P
 Account number .................... E    Number of ADABAS transactions ..... Q
 Logon date ........................ F    Average ADABAS response time ...... R
 Logon time ........................ G    Com-plete monitor calls ........... S
 Time of the last transaction ...... H    Average monitor calls / transaction T
 Current thread .................... I    Total number of ADABAS calls ...... U
 Current Program ................... J    Average ADABAS calls / transaction  V
 Total elapsed time in thread ...... K    Average ADABAS time / transaction . W
 CPU time used ..................... L


 Select columns in desired order:  A

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   Std3p Confm
```

Press PF4 to create the standard printout on three pages, equivalent to the three screens described for the UA subfunction in this section.

You may wish to print some of the information items, by default distributed among different pages, side by side on the same page. To do so, type in the characters associated with the appropriate items in the order you want the columns to be located in the printout. When you enter this screen, it may already contain a default selection set up by user exit U2EXIT1 of your installation (in the above example, it is "A".

If the line length of the printout you design exceeds the maximum of 121, you are informed about the overflow and have to resign one or more items. The printout is created when you confirm your selection by pressing PF5.

QO (Queue Overview) Subfunction

This function displays a list of all queues currently defined in the system and their current status. The following is a sample of what will be displayed.

```
15:54:47        TID     5          COM-5.1.          User MBE          12.11.97
                            --- Queue Overview ---                          UCTR
                  -FIFO--  LIFO
Queue name        Cur Max  Curr  High  Total Enqs     Avg Time
*** TOP OF DATA ***
Stage Queue         0  16     0     0           0        0.000
Output Queue        0  16     0     1       4,501        0.000
Input Queue         0   8     0     1           1        0.000
Completion Queue    0  16     0     1       9,363        0.001
Message Queue       0  16     0     0           0        0.000
Paging Queue        0   8     0     0           0        0.000
DEFAULT Prty-3      0  16     0     0           0        0.000
DEFAULT Prty-2      0  16     0     0           0        0.000
DEFAULT Prty-1      0  16     0     2       5,954        0.000
DEFAULT Prty-0      0  16     0     1         626        0.001
DEFAULT TASK-Q      0   8     0     2       9,385        0.001
DEFAULT TASK-Q      0   8     0     1         371        0.000
NATURAL Prty-3      0  16     0     0           0        0.000
NATURAL Prty-2      0  16     0     0           0        0.000
NATURAL Prty-1      0  16     0     1         629        0.000
------------------------------------------------------------ Active Users: 00003
Select function:      Operand:                                PF7=up PF8=down
        Command:
```

Queue Name

This is the 16 character name of the queue for which details are provided. The first six queues are the standard system queues used by Com-plete. Following this, each of the task group's queues will follow. The first eight characters of these queue names will contain the task group name while the second eight characters indicates their purpose within the task group.

Each task group will have four queues with names Prty-0 to Prty-3. These queues represent the work queues upon which work is queued depending on the priority of the work. In addition, each task within the group will have its own work queue with the identifier 'TASK-Q' associated with it. These queues are used for work which is associated with a particular task.

FIFO and LIFO queues

Each queue that is built in the system will have a First In First Out (FIFO) queue which will be used under normal circumstances. These FIFO queues can only contain a maximum number of users as indicated in the display; once these queue positions have been used, the queue's Last In First Out (LIFO) queue will be used. Under normal circumstances the FIFO queue will always be used which is what would be desired as this insures work is processed in the order in which it is presented to the system. The LIFO queue can accept any number of entries and is there in the event that some problem in the system causes too many users to land on the queue. In this case, work on the LIFO queue will not be processed in the order it arrived. This is preferable to a situation whereby a TIB could not be put on a queue at all due to the FIFO queue being full and the program running on that TIB being abnormally terminated.

FIFO Cur

This is the number of users currently on the First In First Out (FIFO) queue.

FIFO Hig

This is the maximum number of users that can be handled at any one time on the First In First Out (FIFO) queue.

LIFO Curr

This is the number of users currently on the Last In First Out (LIFO) queue.

High

This is the highest number of users ever on all queues at any one time. If this is higher than the FIFO Hig count, it indicates that at some point in the run, there was not enough space on the FIFO queue and so the LIFO queue had to be used.

Total Enqs

This is the total number of enqs to the queue.

Avg Time

This is average amount of time each TIB spent on the queue in thousandths of seconds.

RA Subfunction

This function displays performance critical statistics from the Com-plete Roll Subsystem. These statistics should be monitored on a regular basis to check if changes ara necessary. The Roll Activities display is as follows:

```
15:32:30        TID    12          COM-610          User MBE          20.04.00
                           --- Roll Activities(1) ---                       UCTE



           Average Amount                                Rollout/in

        Avg Rsize    1183 K                           Out Total     418

        Avg Csize      62 K                           In  Total     400 ‡ ‡ ‡
```

Item	Meaning
Avg Rsize	Average size of thread image that has been rolled up to this point.
Avg Csize	Average size of the thread image after Com-plete compression has been performed and is the average of the actual amount of data which the roll subsystem will have rolled.
Out Total	Total number of rollout requests.
In Total	Total number of rollin requests.

An additional display is available by pressing PF11 from the first RA screen, which presents an overview of the size of the images being rolled out, the number of such images and the percentage of the total. Each rollout size is rounded to the nearest 4k, therefore for every 4k slot (8K, 12K, 16K.....1024K) a count is kept of the number of images of each size. To keep the display simple, where a size has not been used, it is not displayed. Therefore, the display consists of an entry for each size which has actually been rolled, the number of times that size was rolled and the percentage of the total rollouts in the system. This display enables you to determine what the load on the roll subsystem is and to determine the subpool sizes and number of elements required for a Fixed Roll Buffer Pool. The following is an example of this display:

```
13:07:06        TID    12          COM-5.1.          User XJPO         10.03.97
                          --- Roll Activities(2) ---                       UCTL
  Size  Number   %    Size  Number   %    Size  Number   %    Size  Number   %
*** TOP OF DATA ***
   16K      50   1     100K    500  10     256K    250   5     320K   1950  39
  500K     2250 45


*** BOTTOM OF DATA *** ------------------------------------ Active Users: 00002
Select function:      Operand:             PF7=up PF8=down PF10=left PF11=right
        Command:
```

RP Subfunction

The RP function displays statistics on the resident programs.

For every resident program, information displayed includes the name and size of the program, the number of loads, as well as the load-point, entry-point and linkage editor options. Note that RESIDENTPAGE programs are marked with an asterisk (*).

```
11:03:09        TID     6           COM-5.1.          User SAGAWW       30/08/94
                            --- Resident Programs ---                        UCTC
    Name        Size     Loads     Entry-point       Load-point       Attr
 *** TOP OF DATA ***
   NATBPMG        7K               0012C6B0          0012C6B0
   NATCOM21    1580K        1      002B7A98          0012E358
   NATURAL      706K               84490B00          84490B00
   UF220RE       34K               002B98C0          002B98C0         RN RU
   ULPGMTAB       1K       24      000440E8          000440E8
   UEPDMN        34K       30      002C2AB8          002C2AB8         RN RU





 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:       Operand:                                PF7=up PF8=down
         Command:
```

SG (Thread Subgroups) Subfunction

This function will display a list of all thread subgroups in a requested thread group and their current status. If no thread group name is provided as an operand, the subgroups of all thread groups will be displayed. The following display is a sample of what will be displayed.

```
15:13:55        TID     5           COM-5.1.          User MBE          12.11.97
                            --- Thread Subgroups ---                         UCTY
                        <-Threads->         UseCounts  WaitCount
 Name      Above  Below Def'd Alloc  Status Curr High  Curr High    CPU   Real K
 *** TOP OF DATA ***
 UTILITY1  1024K   132K     3     3  Active   13   14     0    0  10.00  20.00 M
 UTILITY2  1024K   516K     2     2  Active    2    9     0    0  10.00  20.00 M
 SMALLNAT  1024K    52K     2     2  Active    0    2     0    0  10.00  20.00 M
 BIGNAT    1024K   516K     2     2  Active    0    1     0    0  10.00  20.00 M









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:      Operand:                                  PF7=up PF8=down
         Command:
```

Name

This is the name of the subgroup to which the line of statistics relates.

Above

This is the amount of storage each thread in the subgroup has available above the line.

Below

This is the amount of storage each thread in the subgroup has available below the line.

Threads - Def'd

This is the number of threads which the installation wishes to have defined within the thread subgroup.

Threads - Alloc

This is the number of threads actually allocated for the thread subgroup. This may differ from the number of threads defined as when the number of threads is to be reduced, the threads must first be quiesced in order to let current users of the threads to be deleted, finished their work. These threads will only be physically deallocated and disappear from this count when the last user deallocates the thread and the thread becomes dormant.

Note: Com-plete does not currently provide a mechanism to change the number of threads defined within a thread subgroup dynamically, therefore the values given for 'Threads' should always be identical.

Status

This is the status of the thread subgroup. The thread subgroup may be Active, Quiescing or Dormant. When a thread subgroup is active, Com-plete can allocate and deallocate threads from that subgroup at will. When the subgroup is quiescing, any threads that have previously been allocated will remain available until they are deallocated by all users using them. When the last thread is deallocated, the thread subgroup will have deemed to be quiesced and the control block will be marked as dormant.

Use Counts - Curr

This is the number of users with the threads currently allocated from the thread subgroup.

Note that users which are rolled out and relocatible are not included in any subgroup

Use Counts - High

This reflects the highest number of users which had threads allocated in this subgroup at any one time since the subgroup was initialised.

Wait Counts - Curr

This is the number of users currently waiting for a thread of the thread subgroup to become available. Users waiting for a distinct thread due to an affinity are not included in this number.

Wait Counts - High

This reflects the highest number of users which were waiting for a thread of the thread subgroup to become available at any one time since the subgroup was initialised.

CPU

This is the number of CPU seconds which a user program using the thread subgroup in question can use between roll outs before being terminated abnormally by Com-plete. If the word ’none’ appears for this field, it indicates that no CPU time was requested in the Sysparms. Note that Com-plete still sets a nominal time of 24 CPU hours when no CPU timing is requested by the user. This is required for the calculation of CPU time statistics.

Real

This is wall clock time which is set for each thread in the subgroup. If a user program occupies the thread without doing a rollout for longer than this time, a message is issued to the operator console. This message will be repeated every 30 seconds after the first message if the user program does not give up the thread. If the word ’none’ appears in this field, it indicates that Com-plete is not calculating the length of time user programs are in a thread for threads in this subgroup.

K

This is the key in which threads in the subgroup are set when a new program starts in one of the threads. If an absolute key number is displayed, it indicates that all threads in the subgroup get that key. If the letter ’M’ is displayed, it indicates that the keys for threads within the subgroup are mixed within the range of user protect keys available from the operating system.

SO Subfunction

This subfunction displays an overview of Servers active in Com-plete.

```
13:47:45        TID    28          COM-5.1.          User MBE          30.08.97
                             --- Server Overview ---                        UCTH

 Name         CB addr     INIT module      INIT addr      Req. Handler
 *** TOP OF DATA ***
  COM-5.1.    00261370    TSVRADMN         868B0138       00000000
  COM-CNT     00261DE8    TSVRADMN         868B0138       00000000
  BROKTST     00261BD0    TSVRADMN         868B0138       00000000
  APPCTST     002619B8    TSVRADMN         868B0138       00000000
  DB2         068A8000    TLINDB2          868A7440       800260A0
  NATBPS22    0002B030    NATCBPS2         0006E060       0006E356
  JES2        05E61D00    TLINJES2         00264040       00000000









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00008
 Select Function:      Operand:                                 PF7=up PF8=down
         Command:
```

Meaning of the information items according to column heading:

Heading	Meaning
Name	The name given to the server. This is the first subparameter of the SERVER sysparm.
CB addr	The address Server Control Block where every server can keep relevant data.
INIT Module	The name of the Server Initialization module.
INIT addr	Load address of the Server Initialization module.
Req. Handler	Load address of the Server Request handler.

TL (Thread List) Subfunction

This function will display a list of all threads in the request thread group and their current status. If no thread group is provided as an operand, the threads for all thread groups will be displayed. The following display is a sample of what will be displayed.

```
15:20:26        TID     5          COM-5.1.          User MBE          12.11.97
                                --- Thread List ---                          UCTZ
                 Use counts  Wait Cnts            ...Current/Last Active...
 Subgrp   Status Curr  High  Curr High K Last Op  User ID  Program  Tid.. L Time
 *** TOP OF DATA ***
 UTILITY1 A-Occ     5     5     0    1 C Wrt      COK      UPDS         3 9
 UTILITY1 A-Run     4     4     0    0 B Readm    MBE      UUTIL        5 1
 UTILITY1 A-Occ     4     4     0    0 A Wrt      SKU      USTOR        4 5
 UTILITY2 A-Occ     1     5     0    1 E Coexit   COK      UEDIT        3 4
 UTILITY2 A-Occ     1     4     0    1 D Wrtse    SKU      UEDIT        4 2
 SMALLNAT A-Free    0     2     0    2 A Eoj      COK      NAT22        3 9
 SMALLNAT A-Free    0     1     0    1 F Wrts     COK      NAT22        3 9
 BIGNAT   A-Free    0     0     0    0 C
 BIGNAT   A-Free    0     0     0    0 B





 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:      Operand:                                  PF7=up PF8=down
         Command:
```

Subgrp

This is the name of the thread subgroup of which the thread in question is a member.

Status

This reflects the current status of the thread. The status is a combination of two state indicators separated by a dash ('-'). The primary state indicator is the letter preceding the dash and indicates whether the thread is Active, Quiescing or Dormant by the letters A, Q and D respectively. Active in this sense indicates that the thread is available to do work. When it is quiescing, it will remain active long enough to finish any work which has been started in the thread while dormant thread cannot be used and will have no secondary state associated with them. The secondary states which may occur are as follows:

'Free' Status

This indicates that the thread is free to run other work. If there was a previous user of the thread, this state indicates that this user's program ended or has been rolled out.

'Occ' Status

The 'occupied' status indicates that the thread is available to do work, however, the user program currently occupying the thread must first be rolled out prior to starting any other new work in the thread.

'Disp' Status

This indicates that the thread is reserved and the dispatcher is currently in the process of either starting a new user program or rolling in a user program which was previously rolled out.

'Run' Status

This indicates that the user program in the thread is currently running.

'Susp' Status

This indicates that the user program has been temporarily suspended as a wait was issued either directly by the user program or indirectly by a function used by the program. In this state, the user program may not be rolled out. Internally it indicates that the operating system task associated with the work is active elsewhere. Once the condition for the wait is satisfied, the task will continue processing this work.

Use Counts - Curr

This is the current use count for the thread. The use count includes the current user of the thread plus any other non relocatible users previously rolled out from this thread.

Use Counts - High

This reflects the highest ever use count experienced for the thread since the thread was initialised.

Wait Counts - Curr

This is the current wait count for the thread. This reflects the number of users waiting to run in the thread at the present time.

Wait Counts - High

This reflects the highest ever wait count experienced for the thread since the thread was initialised.

K

This is the key in which the user area of the storage is currently in when the thread is active, or is the key in which the last user of the thread left it. Note that the user area of the storage depends on the catalogue size of the current or last user of the thread.

Last Op

This is the last Com-plete operation which was issued in the thread.

Current/Last Active Userid

This is the userid of the current user active in the thread or the last user to be active in the thread if the thread has a status of ’free’ or ’occ’. This may be blank in systems where logon is not forced and will generally be blank for ULOG sessions where the user is logging on.

Current/Last Active Program

This is the name of the program currently active in the thread, or the last program to be active in the thread if the thread has a status of ’free’ or ’occ’. If the thread has never been used, this will be blank, however, once the thread has been used, this will always contain a value.

Current/Last Active Tid..

This is the tid of the current TIB active in the thread, or the last TIB to be active in the thread if the thread has a status of ’free’ or ’occ’. If the thread has never been used, this will be blank, however, once the thread has been used, this will always contain a value.

Current/Last Active L

This is the level number on which the user currently active in the thread, or the last user to be active in the thread if the thread has a status of ’free’ or ’occ’, is running. If the thread has never been used, this will be blank, however, once the thread has been used, this will always contain a value. Level ’0’ will be displayed for a non-COM-PASS user, for a COM-PASS user who is running a program while something is stacked on all possible levels or for specific programs which must run on level 0 such as ULOG.

Time

When the thread has a secondary status of ’Susp’ or ’Run’, this will reflect the time in seconds that this user has spent in the thread.

TO Subfunction

The TO subfunction displays information about terminals and printer status. Information items include:

- The name of the TIB onto which the user is logged;

- TID number;
- Device type;
- User ID (if logged on);
- Priority ;
- Screen size;
- Terminal status (ACTIVE, ERROR, etc.).

```
13:49:06        TID    28          COM-5.1.          User MBE          30.08.97
                           --- Terminal Overview ---                       UCTD
LU-Name   TermID  Device    Type    UserID  Prty Lines Columns Status
 *** TOP OF DATA ***
 SYSOUT         1  BATCH                      1    10    133   ACTIVE
 P014           2  3288  L   ACCESS           1    14    132   ACTIVE
 P015           3  3287  L   ACCESS           1    32    132   ACTIVE
 DAEPRT14       4  3288  L   VTAM             1    14    132   ACTIVE
 DAEPRT15       5  3288  L   VTAM             1    14    132   ACTIVE
 VIRTP1         6  3288  L   VTAM             1    14    132   DELETED
 DAESV16        7  3270  L   VTAM             1    24     80   ACTIVE
 DAELC516       8  3270  L   VTAM             1    24     80   ACTIVE
 COKPRT         9  3288  L   VTAM             1    14    132   ACTIVE
               10  ATTACHED          SYSUSR    0    24     80   ACTIVE
               11  ATTACHED          SYSUSR    0    24     80   ACTIVE
               12  ATTACHED          SYSUSR    0    24     80   ACTIVE
               13  ATTACHED          SYSUSR    0    24     80   ACTIVE
 DAEEC306      14  3279  L   VTAM             1    27    132   ACTIVE
               15  ATTACHED          TIMERM    0    24     80   ACTIVE
------------------------------------------------------------ Active Users: 00008
Select function:      Terminal:                                PF7=up PF8=down
         Command:
```

As in the case of the UA subfunction, you can restrict the display to specific terminals by entering their prefix. For example, entering "ABC" causes all terminals starting with "ABC" to be displayed. "**C" causes all terminals with a "C" as their third character to be displayed.

TG (Thread Groups) Subfunction

This function will display a list of all thread groups in the system. The following display is a sample of what will be displayed.

```
15:11:28        TID     5          COM-5.1.          User MBE          12.11.97
                             --- Thread Groups ---                          UCTX
                      Use counts         <-Threads->
 Grp name  Status     Curr  High  SGrps  Def'd Alloc
 *** TOP OF DATA ***
 DEFAULT   Active       15    16     2       5     5
 NATURAL   Active        1     2     2       4     4









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:      Operand:                                  PF7=up PF8=down
         Command:
```

Grp Name

This is the name of the group for which the information is being displayed.

Status

This is the status of the thread group. The thread group may be Active, Quiescing or Dormant. When a thread group is active, Com-plete can allocate and deallocate threads from that group at will. When the group is quiescing, any threads that have previously been allocated will remain available until they are deallocated by all users using them. When the last thread is deallocated, the thread group will have deemed to be quiesced and the control block will be marked as dormant.

Use Counts - Curr

This is the number of users with threads currently allocated from the thread group.

Use Counts - High

This reflects the highest number of users which had threads allocated in this thread group at any one time since the group was initialised.

Sgrps

This is the number of subgroups contained within the thread group.

Threads - Def'd

This is the number of threads which the installation wishes to have defined within the thread group.

Threads - Alloc

This is the number of threads actually allocated for the thread group. This may differ from the number of threads defined as when the number of threads is to be reduced, the threads must first be quiesced in order to let current users of the threads to be deleted, to finish their work. These threads will only be physically deallocated and disappear from this count when the last user deallocates the thread and the thread becomes dormant.

Note: Com-plete does not currently provide a mechanism to change the number of threads defined within a thread group dynamically, therefore the values given for ’Threads’ should always remain constant throughout a given run of Com-plete.

TS Subfunction

The TS subfunction displays detailed information about a specific terminal. Select a terminal by entering either its name or its TIB number in the Terminal ID field of the Terminal Status display. Information items on this display include:

- The hex display of the TIB control block;
- Current user ID;
- Device type and access method;
- VTAM session ID;
- Status;
- Important status flags.

```
10:48:53        TID     6           COM-5.1.          User SAGAWW       30/08/94
                             --- Terminal Status ---                         UCT8
 0000 00000000 10C00001 00000000 00001083       Tid number ....     6
 0010 01002010 00001FFE D201091B 00000000       Tid name ...... DAESS17
 0020 01200006 80058C00 C4C1C5E2 E2F1F740       Current user .. SAGAWW
 0030 010001D7 728E833A 00501850 18C0000C       Hardcopy Tid ..     0
 0040 00000000 FF000100 00058E50 00010000
 0050 20810000 00000000 00000000 00000000       Terminal type . 3270  L
 0060 00000000 00000000 01000000 00000000       Access method . VTAM
 0070 00000000 E0000000 00000000 00000000       Session ID .... 010001D7
 0080 FFFFFFFF 00000000 17101218 07092418
 0090 17101218 00000000 00000000 00000001       Status ........ ACTIVE

 ---------------------------------- Tib flags ----------------------------------
 Input inhibited
 SNA device
 Chaining
 Bracket protocol
 -------------------------------------------------------------------------------
 Select function:      Terminal ID:  6                     PF10=left  PF11=right
         Command:
```

In the case of VTAM terminals, an additional screen is available (press PF11. It displays:

- The hex display of the last RPL;
- Sense data;
- Trace information.

```
10:51:22        TID     6           COM-5.1.          User SAGAWW       30/08/94
                              --- Terminal Status ---                        UCT9
       ----------- RPL display -----------            Trace entries
 0000  00000000 00000000 00000000 00000000                  COMPL
 0010  00000000 00000000 00000000 00000000                  ENIN
 0020  00000000 00000000 00000000 00000000                  RDSYN
 0030  00000000 00000000 00000000 00000000                  RECEIVE
 0040  00000000 00000000 00000000 00000000                  COMPL
 0050  00000000 00000000 00000000 00000000                  COMPL
 0060  00000000 00000000 00000000                           COMPLDM
                                                            WRITE
                                                            SEND
 Sense Data: 0000 0000                                      RPLEX
                                                            COMPL
                                                            ENIN



 ------------------------------------------------------------------------------
 Select function:      Operand: 8                       PF10=left  PF11=right
         Command:
```

Use PF10 and PF11 to switch between the two screens.

TU Subfunction

The TU subfunction displays information about the amount of threadstorage used by an application. An example is shown below:

```
09:59:55        TID     9          Dev--621          User COK          16.10.03
                              Thread utilization                           UCTI

 Group    Subgroup Program  Below: Size Region Size Used  Max Above:   Used Max
 *** TOP OF DATA ***
 SYSTEM   $SYSTEM  TLAMMAIN 497000   16 498000   16    0    0 16A3A000   27   27
 SYSTEM   $SYSTEM  TLMSMESG 492000   16 493000   16    8    8 169F4000    1    1
 SYSTEM   $SYSTEM  READCONS 48D000   16 48E000   16    0    0 169AE000    5    5
 SYSTEM   $SYSTEM  TLOSOPER 0F7000   16 0F8000   16    0    0 15CC2000    1    1
 SPECIAL  SPECIAL1 UCTRL    0FC000  512 173000   40    0    0 15D08000   49   49
 NATURAL  NATURAL1
 DEFAULT  UTILITY1 PAENSTRT 2B4000  256 2B5000  256    0    0 1632C000   24   24
 DEFAULT  UTILITY1 PAENSTRT 273000  256 2AE000   24    0    0 16226000   42   42
 DEFAULT  UTILITY1 UTIMRM   232000  256 255000  120   44   44 16120000    1    1
 DEFAULT  UTILITY1 USTACK   1F1000  256 231000    4    0    0 1601A000   18   26
 DEFAULT  UTILITY1 USTACK   1B0000  256 1F0000    4    0    0 15F14000   18   29
 DEFAULT  UTILITY2
 DEFAULT  UTILITY2
 DEFAULT  UTILITY2

 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00002
 Select Function:      Operand:                                 PF7=up PF8=down
        Command:
```

Meaning of the information items according to column heading:

Heading	Meaning
Group	Threadgroup as defined in the sysparms
Subgroup	Thread subgroup as defined in the sysparms
Program	Program Initially loaded application
Below	Start address of threadarea below
Size	Threadsize below as defined in the sysparms
Region	Start of threadarea below avaliable for application usage
Size	Region size defined for the application in ULIB
Used	Currently used amount of application thread storage below (in K bytes)
Max	Maximum used amount of application thread storage below (in K bytes)
Above	Start address of threadarea above
Used	Currently used amount of application thread storage above (in K bytes)
Max	Maximum used amount of application thread storage above (in K bytes)

UA Subfunction

The UA subfunction displays activity information about user IDs currently logged on to Com-plete.

The first of the three screens, shown below, shows miscellaneous information, such as:

- LU-name (the name of the TIB onto which the user is logged);
- TID number;
- Authorization code and account number;
- Logon date and time;
- Time of the last transaction .

```
15:58:48        TID     5          COM-5.1.          User MBE          12.11.97
                             --- User Activities ---                        UCT6

 User Id. LU-name. TermId  Auth.   Account No..  Logon Date   Time on.  Last tx.
 *** TOP OF DATA ***
 TIMERM   ATTACHED      2      0                 11.11.1997   13.39:48  00:03:00
 COK      SHRDAEN       3      0   COMPLETE      12.11.1997   07:49:22  15:55:11
 SKU      SHRDAEN       4      0   Com-plete     12.11.1997   09:59:43  15:30:55
 MBE      SHRDAEN       5      0   Com-plete     12.11.1997   14:50:27  15:58:49









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00003
 Select function:      User ID:             PF7=up PF8=down PF10=left PF11=right
         Command:
```

Press PF11 to display the second information screen. This gives additional information about the user ID’s activities, such as:

- Current program;
- Total elapsed time spent in thread (seconds);
- CPU time used (seconds);
- Total number of transactions (times in thread);
- Total number of EXCPs;
- Amount of data sent to and received from terminal;

- Amount of data sent via message switching/printout spooling.

```
16:00:28        TID     5          COM-5.1.          User MBE          12.11.97
                             --- User Activities ---                        UCT7

 UserID         Program  Thrd time  CPU time  Trans   EXCPs   Term data Spl data
 *** TOP OF DATA ***
 TIMERM         UTIMRM       23.99       .53    625       8        0         0
 COK            UPDS       1014.13     29.07   1972     249  1709314         0
 SKU            UPDS       3763.59    114.46   8230     465  2060050         0
 MBE            UUTIL        45.54       .30     35      60    53929         0 ‌‌‌
```

```
14:01:10        TID    28          COM-5.1.          User MBE          30.08.97
                            --- User Activities ---                        UCT0

UserID    Enters  Avg RT Comp OPs     (Avg)     ADAcalls     (Avg)    Avg time
*** TOP OF DATA ***
SYSUSR         0    0.00       23        0.0          29      29.00     0.011
SYSUSR         0    0.00       25        0.0          27      27.00     0.011
SYSUSR         0    0.00       25        0.0          27      27.00     0.005
SYSUSR         0    0.00       25        0.0          27      27.00     0.006
TIMERM         0    0.00     2994        0.0           0       0.00     0.000
SAGARM         2    0.12       21       10.5           0       0.00     0.000
RSF2         228    0.07     1186        5.2           0       0.00     0.000
RSF          554    0.09     7520       13.5        1025      78.84     0.012
SAGAWW        20    0.82      201       10.0         629      48.38     0.012
COK         1729    0.24    11669        6.7        4138      30.65     0.054
SKU         1361    0.07     8595        6.3         629      78.62     0.011
PCKLS          5    0.19       41        8.2           0       0.00     0.000
MBE           27    0.14      142        5.2           0       0.00     0.000
RSF1          38    0.48      841       22.1         716      44.75     0.012
PCBSU          3    0.03       25        8.3           0       0.00     0.000
------------------------------------------------------------ Active Users: 00008
Select function:      User ID:             PF7=up PF8=down PF10=left PF11=right
        Command:
```

Use PF10 and PF11 to switch between the three screens.

The UserID input field is handled in the following way:

- If data is entered with one or more non-numeric bytes, the data is treated as a user ID prefix. You can use the asterisk (*) as placeholder for any character
- If a number within the range if TID numbers is entered, the number is taken as TID number. If the number entered falls outside the TID number range, it is taken as a user ID prefix.
- To force any data to be taken as user ID prefix, precede the data with "U=".

US Subfunction

The US subfunction displays status information about user IDs currently logged on to Com-plete.

This function will display a list of users IDs and the current status of the programs they are running. As an operand, you can specify a user ID, or the first characters to be matched by userIDs. You can use asterisk "*" as a wildcard. If no operand is provided, a list of all users currently logged on will be displayed. The following display is a sample of what will be displayed.

```
13:24:54        TID     7           COM-5.1           User MBE          03.11.98
                              User Program Status                           UCTI

 UserID      TID  Program   Thrd Grp  Task Grp  Status
 *** TOP OF DATA ***
 SKU           3  UPDS      DEFAULT   DEFAULT   rolled out
 TIMERM        4  UTIMRM    DEFAULT   DEFAULT   ready for rollout
 RM01          5  UPDS      DEFAULT   DEFAULT   rolled out
 RM            6  UPDS      DEFAULT   DEFAULT   rolled out
 MBE           7  UUTIL     DEFAULT   DEFAULT   running
 FHAL          8  UUTIL     DEFAULT   DEFAULT   rolled out









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00005
 Select Function:      Operand:                                  PF7=up PF8=down
        Command:
```

Along with userID, terminal ID (TID), the name of the current program, the thread group and task group this program is running in, the current status of this program is indicated.

This status can be one of the following:

no program active

This is a non-Compass user which is not running any program at the moment.

running

This program is currently being executed by one of the subtasks.

subtask WAIT

This program is occupying a thread and a subtask, and is waiting for some event to occur.

Example: WAIT for an ENQ

program WAIT

This program is occupying a thread, but is waiting for some event to occur. The subtask which serviced the program before the WAIT is available to service other threads while this one is waiting.

Examples: I/O, WAIT for a Complete internal resource

ready to be rolled out

This program is in a state where it has issued a terminal write or ROLOUT, and can be rolled out of the thread when some other program needs one.

rolled out

This program is in a state where it has issued a terminal write or ROLOUT, and it has been rolled out to a roll buffer to allow some other program to run in the thread in the meantime.

queued: <queue name>

The program is on one of Complete's internal queues, indicated by the name of this queue. Refer to subfunction QO for an overview of the different queues. Most likely, the program is waiting for a certain subtask or for any subtask out of a task group to become available. If this state seems to last too long, use subfunction PL to find out who is occupying the subtask(s).

thread SG queue <subgroup name>

The program is waiting for any thread in the thread sub-group, indicated by the sub-group name, to become available. The program can run in any of the threads in this sub-group, but all of them are occupied. If this state seems to last too long, use subfunction TL to find out who is occupying all the threads in the sub-group.

thread queue in <subgroup name>

The program is waiting for a certain thread in the thread sub-group, indicated by the sub-group name, to become available. The program has been rolled out of this thread before, and needs to be rolled back into the same thread again, however, the thread is currently occupied by some other program. If this state seems to last too long, use subfunction TL to find out who is occupying the thread.

eoj in progress

This program has terminated, and Complete is cleaning up the thread behind it.

being dispatched

Sometimes, when switching from one of the above states to another, it can happen that the program has left the previous state, but is not in the next state yet.

Note:
For internal serialization reasons, it is not possible to terminate a program while it is on any of the queues. None of the operator commands IGNORE, CANCEL, LOGOFF, or FORCE will work in this situation. Instead, you should find out who is using the resource (thread or subtask) where a queue is building up, and ask him/her to free it (or CANCEL him/her if necessary).

VS Subfunction

The VS subfunction displays statistics about user VSAM files currently open, and about the local shared resource (LSR) pool if one is being used.

Irrespective of the number of ACBs in application programs referring to a given DD / DLBL name, Com-plete opens a single ACB for each DD / DLBL name and passes all requests through it. Statistics displayed about each ACB relate to the period of time since the ACB has been opened.

```
15:15:37        TID    13          COM-5.1.          User MBE          12.04.97
                             --- VSAM statistics ---                        UCTO

 DDname   STRN  MAX  noSTR  Requests  Avg.Time  Users (max)  OPEN Time  LostRPLs
 *** TOP OF DATA ***
 SKUVSAM  ---LSR---      0         5      .079      1     1  12.04.15:12       0
 TESTVSAM    0    0      0         0                0     1  12.04.15:14       0
 ------------------------------------------------------------------------------
 LSR-POOL   16    1      0        22 READs required       2 CI-s found in buffer









 *** BOTTOM OF DATA *** ------------------------------------ Active Users: 00004
 Select function:      Operand:                                PF7=top PF8=down
         Command:
```

If at least one of the ACBs currently open uses local shared resources, the last line on the screen containes statistics about the local shared resource pool.

Meaning of the information items according to column heading:

Heading	Meaning
DDname	The DD (z/OS) or DLBL (VSE) name.
STRN	The value specified for the ACB parameter STRNO.
MAX	The maximum number of requests concurrently active at any one time. (In VSE systems, available for the LSR pool only.)
noSTR	The number of requests that failed due to a "maximum number of strings exceeded" condition.
Requests	The number of requests issued by application programs.
Avg.Time	The average response time, in seconds, for a request.
Users	The number of user ACBs currently open in application programs.
(max)	The maximum number of user ACBs open at any one time.
OPEN Time	Date and time when the ACB was opened.
LostRPLs	The number of request parameter lists (RPLs) pointing to this ACB and related to a request that could not be completed due to an abnormal termination.The most appropriate reason for a non-zero number in this field is application programs cancelled while waiting for completion of a VSAM I/O request. Uncompleted requests may result in other requests failing due to locked resources. This situation may be cleaned up by explicitly closing the "main" ACB using UUTIL subfunction FM (file maintenance) described in this section.

NQ Function - Monitor ENQs (z/OS and VSE only) (Administrator)

This function allows you to monitor z/OS ENQs and VSE LOCKs issued by user programs running in thread, and ENQs as a result of user program requests. It also enables you to monitor outstanding ENQs throughout the whole z/OS system.

After selecting the function code NQ from the UUTIL main menu, the Online Enq Display menu is displayed:

```
14:18:15        TID     4           COM-5.1.          User JPO          05/06/97
                         --- Online Enq Display Menu ---                    UEN0

                       Function ....................... Fc
                       -------------------------------- --
                       Show System Enqs ............... SE
                       Outstanding Thread Enqs ........ TE
                       Please select a function ....... SE


      Major Name ....:
      Minor Name ....:



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

Meaning of the options and parameters:

Option	Meaning
SE	Monitor ENQs in the system (System ENQs).
TE	Monitor ENQs issued by/on behalf of user programs (thread ENQs).
Major Name	This is the major name of the ENQ which you wish to monitor.
Minor Name	This is the minor name of the ENQ which you wish to monitor.

Thread ENQs / LOCKs

Depending on the specified major/minor parameter, selecting the TE subfunction results in a display of thread ENQs similar to the following:

```
14:19:39        TID     4           COM-5.1.          User JPO          05/06/97
                         --- Outstanding Thread Enqs ---                   UEN1

 Function ...:           ( List / Deq / User / Term / Thrd )
 Token 1 ....:           ( Major Name / Userid / Tid no / LU Name / Thread no. )
 Token 2 ....:                                                ( Minor Name )


User-ID  LU-Name  Task-ID  S/T Major    Minor
-------- -------- -------- --- -------- ---------------------------------------
SYSUSR            THREAD1  P/E SAGCTIMO TIMON
SKU      DAES313  SYSTEM   M/S SYSDSN   SKU.TEST.ISAM
JW1      SHRDAENT SYSTEM   M/S SYSDSN   JW.COMN.SOURCE
JPO      DAES348  SYSTEM   M/S SYSDSN   COM.SYSF.IV450.XA.LISTING
*** END OF DATA ***









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

You can manipulate the output by specifying a function in the*Function* field and appropriate parameters in the*Token*fields. Available functions are described below:

List (default)

This causes outstanding ENQs from thread to be displayed using the major and minor names specified in the *Token* fields to limit the extent of the display. Specify the major name as the Token 1 parameter, and the minor name as the Token 2 parameter.

Deq (z/OS only)

This function enables you to DEQ an outstanding ENQ. The ENQ to be freed must be identified uniquely using the major and minor names in the*Token* fields, otherwise the request will fail. Specify the major name as the Token 1 parameter, and the minor name as the Token 2 parameter.

Note:
Invalid use of this function could effect the integrity of your system, as other users assume that they have an ENQ outstanding for a resource which the DEQ function then frees. This function should therefore only be used when you are sure that an application has abended, or has left the system without freeing the ENQ.

User

This requests a list of all outstanding ENQs for a specific user. The user ID can be specified as the Token 1 parameter.

Term

This requests a list of all outstanding ENQs issued from a specific terminal name.The terminal name or TID number must be specified as the Token 1 parameter.

Thrd

This requests to list of all outstanding ENQs issued from a specific thread. The thread number must be specified as the Token 1 parameter.

Meaning of the output fields by column heading:

Heading	Meaning
User-ID	User ID running in the thread at the time the ENQ was issued.
LU-name	Terminal name upon which the user was running when the ENQ was issued.
Task-ID	Task that the ENQ was issued for. When an ENQ occurs for a task other than a Com-plete task, the Task-ID is SYSTEM. This can occur when a user request results in a system ENQ for the whole region, for example in the case of dynamic allocation.
S/T	Scope and type of the ENQ. Possible values for the scope:
	P Step M System R Reserve X Systems Possible values for type: S Shared E Exclusive
	For more information, see the section *System ENQs* below.
Major	Major name.
Minor	Minor name.

Press PF3 to return to the NQ function menu.

Press PF4 to switch to the sytem ENQ overview using the specified major and minor name parameters.

System ENQs

Selecting the SE subfunction from the NQ menu, or pressing PF4 from the Thread ENQs display with the minor and major names specified, results in a display of system ENQs similar to the following:

```
14:30:44        Tid     4           COM-5.1.          User JPO          05/06/97
                             --- System Enqueues ---                        UQSC
 more ....:
 QName ...:                      Refresh Time:  0          Min Waiters ..:  0
 RName ...:                                                Mode(Char/Hex): C
 Jobname .:
 Trap ....:                                                            Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ------------------------------------------- -------- ---- ---- ----
 SYSDSN
 SYSVSAM
 COMPLETE
 SPFUSER
 SYSIKJUA
 ADABASLX
 SPFEDIT
 ADABAS
 DFSOC001
 SAGCTIMO
 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

If there are more entries than will fit on one screen, the cursor is positioned in the field *more*. If all entries are shown on this screen, the cursor is positioned in the field *QName* field.

To scroll the list (cursor in field *more*), press ENTER; if there no more entries than on this screen (cursor in Qname field), ENTER takes you to the next information screen, see the section *Selection Modes* below;

To switch to the thread ENQs display, press PF4;

To return to the NQ menu, press PF3.

The input fields are described in the section *Selection Modes* below.

Meaning of the information fields according to column heading:

Heading	Meaning
QNAME	High-level qualifier.
RNAME	Low-level qualifier. This is the name of a resource enqueued under a given high-level qualifier (queue name).
JOBNAME	User of the listed resource.
STAT	Status of individual Rname (resource) entry:
	OWNS requestor is owner. WAIT requestor is waiter.
ENQ. TYPE	Scope and type of ownership by the Qname entry. Possible values for scope:
	P Step M System R Reserve X Systems Possible values for type: S Shared E Exclusive
	Example: M/E = system, exclusive. See also table below.
WAIT CNT	Wait count, that is, the number of resources that are waiting for allocation to the resource currently in use.

The abbreviations listed below the bottom line of the Systems Enqueues display indicate the possible values in the ENQ. TYPE field and describe the type of control of the resource that was requested by the active task:

Abbreviation	Explanation
P = Step	Indicates that the resource can be used only within an address space.
M = System	Indicates that the resource can be used by more than one address space.
X = Systems	Indicates that the resource can be shared between systems.
R = Reserve	Indicates that the task has exclusive control over the resource and also has a hardware reserve on the DASD volume associated with the resource.
E = Exclusive	Indicates that the task has exclusive control over the resource.
S = Shared	Indicates that if a resource is not modified while under control of a task, enqueueing requires shared status.

Selection Modes

You can select information concerning a certain queue using the input fields in the top half of the System Enqueues display.

Selecting all Entries

If you leave all fields blank, all queues currently found in the operating system are displayed:

```
14:30:44        Tid     4           COM-5.1.          User JPO          05/06/97
                             --- System Enqueues ---                        UQSC
 more ....:
 QName ...:                      Refresh Time:  0          Min Waiters ..:  0
 RName ...:                                                Mode(Char/Hex): C
 Jobname .:
 Trap ....:                                                            Enq  Wait
 QName    RName                                             Jobname  Stat Type Cnt
 -------- ------------------------------------------- -------- ---- ---- ----
 SYSDSN
 SYSVSAM
 COMPLETE
 SPFUSER
 SYSIKJUA
 ADABASLX
 SPFEDIT
 ADABAS
 DFSOC001
 SAGCTIMO
 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

You can place the cursor on any Qname in the list and press ENTER. This has the same effect as entering the name in the *Qname* input field: all information pertaining to this Qname is displayed (see below).

Selection by Qname

If you only know the Qname, you can enter it in the *Qname* field. When you press ENTER, all information about the specified Qname is displayed, for example:

```
14:32:54        Tid     4           COM-5.1.          User JPO          05/06/97
                            --- System Enqueues ---                         UQSC
 more ....:
 QName ...: COMPLETE            Refresh Time:  0          Min Waiters ..:  0
 RName ...:                                               Mode(Char/Hex): C
 Jobname .:
 Trap ....:                                                        Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ------------------------------------------- -------- ---- ---- ----
 COMPLETE BATCH                                          COMPLETE OWNS M/E   00
                                                         COMPLETE OWNS M/E   00
                                                         TESTNAT  OWNS M/E   00
                                                         COMPSGE  OWNS X/E   00
                                                         COM444   OWNS M/E   00
                                                         COM450   OWNS X/E   00
                                                         SUPPORT  OWNS M/E   00

 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

You can place the cursor on any Rname or Jobname in the list and press ENTER. This has the same effect as entering the name in the corresponding input field: all information pertaining to the Qname with this Rname or Jobname is displayed (see below).

Selection by Rname

If you only know the Rname, you can enter it in the *Rname* field. When you press ENTER, all information about the Qname with this specified Rname is displayed, for example:

```
14:41:28        Tid     4           COM-5.1.          User JPO          05/06/97
                            --- System Enqueues ---                         UQSC
 more ....:
 QName ...:                     Refresh Time:  0          Min Waiters ..:  0
 RName ...: TIMON                                         Mode(Char/Hex): C
 Jobname .:
 Trap ....:                                                        Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ------------------------------------------- -------- ---- ---- ----
 SAGCTIMO TIMON                                          COM450   OWNS P/E   00
                                                         COMPSGE  OWNS P/E   00
                                                         COM444   OWNS P/E   00









 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

You can place the cursor on any Jobname in the list and press ENTER. This has the same effect as entering the name in the *Jobname* input field: all information pertaining to this Jobname is displayed (see below).

Selection by Jobname

If you only know the Jobname, you can enter it in the *Jobname*field. When you press ENTER, all information about the Qname with this specified Jobname is displayed, for example:

```
14:43:11        Tid     4           COM-5.1.          User JPO          05/06/97
                             --- System Enqueues ---                        UQSC
 more ....:
 QName ...:                      Refresh Time:  0          Min Waiters ..:  0
 RName ...:                                                Mode(Char/Hex): C
 Jobname .: COM450
 Trap ....:                                                            Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ---------------------------------------------- -------- ---- ---- ----
 SYSDSN   COM.SYSF.V45.MAPS                              COM450   OWNS M/S   00
          OPS.SYSF.V5.ADALOAD                            COM450   OWNS M/S   00
 SYSVSAM  COM.SYSF.V45.VSAM.CAPTUR1.DATAUCAT.USR836...  COM450   OWNS X/E   00
 SYSDSN   COM.SYSF.IV450.XA.LOAD                         COM450   OWNS M/S   00
          COM.SYSF.IV450.ZAP.LOAD                        COM450   OWNS M/S   00
          TPF.SYSF.IV450.LOAD                            COM450   OWNS M/S   00
          PUB.SYSF.USER.LOAD                             COM450   OWNS M/S   00
          COM.SYSF.V45.USER.SOURCE                       COM450   OWNS M/S   00
 SAGCTIMO TIMON                                          COM450   OWNS P/E   00
 SYSDSN   COM.SYSF.V45.ROLL1                             COM450   OWNS M/S   00

 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

Selection by Rname and Jobname

If you only know both the Rname and the Jobname, you can enter them in the corresponding fields. When you press ENTER, all information about this Rname and Jobname is displayed, for example:

```
14:44:15        Tid     4           COM-5.1.          User JPO          05/06/97
                             --- System Enqueues ---                        UQSC
 more ....:
 QName ...:                      Refresh Time:  0          Min Waiters ..:  0
 RName ...: TIMON                                          Mode(Char/Hex): C
 Jobname .: COM450
 Trap ....:                                                          Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ------------------------------------------- -------- ---- ---- ----
 SAGCTIMO TIMON                                         COM450   OWNS P/E   00









 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

If there are more entries than fit on the screen, the cursor is positioned in the more field. Press ENTER to scroll the list. You can manipulate the output by entering other, valid values in the input fields.

To return to the NQ menu, press PF3.

Display Modes

Rname Mode

The Rname Mode parameter displayed at the top of the System Enqueue display screen can be used to set the mode of the Rname entries for a given Qname to either character or hexadecimal format. The values permitted are:

C	Rnames are displayed in character mode.
H	Rnames are displayed in hexadecimal mode.

Note:
*If you specify Mode=H, data input in the field Rname to select specific minor names also must be in hexadecimal mode (for example, if Mode=H and you wish to list enqueues for Rnames starting with SYS, input data in field*Rname must be E2E8E2).

```
14:45:13        Tid     4           COM-5.1.          User JPO          05/06/97
                             --- System Enqueues ---                        UQSC
 more ....:
 QName ...: COMPLETE             Refresh Time:  0          Min Waiters ..:  0
 RName ...:                                                Mode(Char/Hex): H
 Jobname .:
 Trap ....:                                                            Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ------------------------------------------- -------- ---- ---- ----
 COMPLETE C2C1E3C3C8404040                              COMPLETE OWNS M/E   00
          00D75DC8                                      COMPLETE OWNS M/E   00
          00D73B98                                      TESTNAT  OWNS M/E   00
          00D79110                                      COM444   OWNS M/E   00
          F8                                            COM450   OWNS X/E   00
          00D74CB0                                      SUPPORT  OWNS M/E   00


 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

Setting the Rname entry to hex mode is useful in instances in which the Rname is an address (which is shown in the Rname column only as "...." when Rname mode is set to character).

Refresh Time

The Refresh Time parameter displayed at the top of the System Enqueues display screen can be used to set the refresh rate to a one- or two-digit value specifying the time in seconds that is to pass before the System Enqueues screen is automatically redisplayed with the latest data.

The Refresh Time parameter enables you to bypass the need to keep pressing ENTER in order to see the most current status; entering a Refresh Time parameter causes the screen to be refreshed automatically regularly.

Note:
To reset the Refresh Time, interrupt the system enqueue monitor by pressing the RESET key and then press ENTER, or by pressing the ATTN key on SNA devices.

Displaying Enqueue/Dequeue Conflicts

The *Min Waiters* field can be used to set the number of resources waiting for allocation to the resource currently in use.

To search for specific enqueue conflicts, first enter precise Qname/Rname/ Jobname information in the appropriate fields on the System Enqueues display screen, or select them by placing the cursor on the desired Qname and pressing ENTER).

Set the Refresh Time and the Min Waiters parameters to the desired values and mark the Trap option with an X. (Note that the Trap option automatically sets the Refresh Time value to zero if an enqueue conflict with the given Min Waiters is encountered.)

```
14:50:46        Tid     4           COM-5.1.          User JPO          05/06/97
                            --- System Enqueues ---                        UQSC
 more ....:
 QName ...: SYSDSN               Refresh Time:  1          Min Waiters ..:  1
 RName ...: COM.SYSF.IV450.XA.LOAD                         Mode(Char/Hex): C
 Jobname .:
 Trap ....: X                                                     Enq  Wait
 QName    RName                                          Jobname  Stat Type Cnt
 -------- ---------------------------------------------- -------- ---- ---- ----
 SYSDSN   COM.SYSF.IV450.XA.LOAD                         COM450   OWNS M/S   01
                                                         JPOBR14  WAIT M/E   01





 -- P = Step, M = System, X = Systems, R = Reserve, E = Exclusive, S = Shared --
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End   ThEnq
```

The system ENQ monitor then searches for enqueue conflicts with the given criteria, and the desired information is displayed.

Function PW- User Password Maintenance

The User Password Maintenance function is an alternative to changing your password on the ULOG screen during logon. The PW function allows you to update your password without having to log off from your current session.

The change your password, select function PW from the utilities menu. This displays the user password maintenance screen:

```
16:26:13        TID     6           COM-5.1.          User SAGAWW       05/21/97
                      --- Com-plete Password Maintenance ---                UPWD





            Password:
        New password:
```

Enter your current password and the new password in the appropriate input fields and press ENTER. The following message asks you to confirm the password change:

UPW0003 - CONFIRM PASSWORD CHANGE

Enter the new password again and press ENTER to complete the change.

Function SD - SD File Display Utility (Administrator)

Note:
This function is not applicable if your installation uses dynamic SD files. Each dynamic SD file is a standard VSAM relative record dataset which can be handled using standard tools like IDCAMS.

This utility provides some administrator functions that allow you to maintain SD data sets.

Sensitive subfunctions of UUTIL SD can be restricted to individuals or groups of individuals by security exit UUSEX1.

If you select the SD function from the Administrator menu, the SD library statistics menu is displayed, the items of information given are self-explanatory:

```
10:37:49        TID    11          COM-5.1.          User MBE          03/12/97
                              -- SD library --                             USD0


Max number of SD files:......................     251
Existing SD files:...........................      71 =   28.28 %
Free entries:................................     180

Number of blocks for SD and paging files:...    3904
Currently used:..............................     459 =   11.75 %

-------------------------------------------------------------------------------

Number of blocks for online dumps:..........    2093
Currently used:..............................    1157 =   55.27 %

-------------------------------------------------------------------------------

Blocksize:..................................    4080


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

Press PF5 to display the SD files directory:

```
09:41:39        TID     4          COM-5.1.           User MBE          11/18/97
                            -- SD files directory --                        USD1

 D Name       Tid  Use-cnt    Lrecl    Hirec   Maxrec    Level    Time last open
 ---------------------------------------------------------------------------
 . CMEDIT     SHR        0     4048     1000     1000             97/10/02 13:41
 . COK      32769        0     2860        1        5        1    97/11/11 15:19
 . COK      32774        0     1800        3        4        6    97/09/30 09:51
 . COK  1   32769        0     1800       14       15        1    97/11/04 12:48
 . RDCHO340   SHR        0      128        1        1             97/11/18 08:54
 . RDCRT340   SHR        0      256        5        5             97/11/18 08:55









 Reposition to:                                                            END
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Data        Up    Down                    Delete
```

From this menu you can perform any of the following:

- Scroll the SD files directory forward PF8 or backwards PF7.
- List the SD files directory starting from the name entered in the input field after the prompt *Reposition to*.
- Delete SD files. To delete SD files, select the appropriate lines by entering "D" in the first column and pressing PF12. If you try to delete an SD file while it is in use by other programs, it is deleted only after the last program closes it. However, you are recommended not to delete SD files while they are in use.
- Display records of an SD file. Select the appropriate line by placing the cursor on it and pressing PF5.

The following figure illustrates the display of a record of an SD file:

```
10:39:13        TID    11          COM-5.1.          User MBE          03/12/97
                             -- SD file display --                          USD2
Record:          1              Name: JPO                        Hirec:      8
Displacement: 0000              Tid:  32772                      Lrecl:   1800
-------------------------------------------------------------------------------
0000  00020000 00018000 00000050 00000000         * ..........&.... *
0010  00001222 20540093 040FD1D7 D64BC3D6         * .......l..JPO.CO *
0020  D4D54BC3 D5E3D340 40404040 40404040         * MN.CNTL          *
0030  40404040 40404040 40404040 40404040         *                  *
0040  40404040 4040D7D9 E3C4D4D7 E7C1C3D6         *       PRTDMPXACO *
0050  D4F8F1F1 D1D7D640 40400003 00010002         * M811JPO   ...... *
0060  00000000 6161D1D7 D6C4E4D4 D740D1D6         * ....//JPODUMP JO *
0070  C24040D1 D7D66BC3 D3C1E2E2 7EC76BD4         * B  JPO,CLASS=G,M *
0080  E2C7C3D3 C1E2E27E E7404040 40404040         * SGCLASS=X        *
0090  40404040 40404040 40404040 40404040         *                  *
00A0  40404040 40404040 40404040 F0F0F0F0         *             0000 *
00B0  F0F1F0F0 00040002 00030000 0000615C         * 0100........../* *
00C0  D1D6C2D7 C1D9D440 D3C9D5C5 E27EF9F9         * JOBPARM LINES=99 *
00D0  F9F94040 40404040 40404040 40404040         * 99               *
00E0  40404040 40404040 40404040 40404040         *                  *
00F0  40404040 40404040 40404040 40404040         *                  *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                 End         Zap         Up    Down  Prev  Next
```

From this display, you can perform any of the following:

- Scroll the record forward PF8 or backwards PF7.
- Display part of a record starting with any displacement, entered in the field prompted *Displacement* in the header of the display.
- Go to the previous record PF9 or to the next record PF10.
- Display any existing record of the SD file by entering the record number in the field prompted *Record* in the header of the display.
- Overwrite SD file name and/or TID number to display any other SD file.
- Change contents of a record (PF5). After you have pressed PF5, both the hexadecimal and the character part of the display become unprotected and can be modified. You will be requested to confirm the change before the updated record is rewritten.

Function SI- User Site Information

This facility enables you or a group of users to create up to 64 pages of site-specific information.

If you select the SI function from the utilities menu, the User Site Info System menu is displayed, for example:

```
COMINF0001 Please select an item (enter A(dd), M(odify), or S(how)
 13:55:04        TID     6            COM-5.1.         User SAGAWW       05/21/97
 Page 1                 ---  User Site Info System  ---                     UNFO


 Sel Info-item........... C/* Userid..     Sel Info-item........... C/* Userid..
     BROADCAST









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

Meaning of the fields according to column headers:

Column	Meaning
Sel	Input field for information item maintenance. Possible input values: S Displays information item. A In an unused line, add an information item. M Modify the selected item.
Info-item	Name or short description (up to 20 characters) of the information item. Type a value in this field when using the A or M selection code. C/* Use this field to restrict write access to the item. Possible values: <Blank> No restriction. C Restricts write access to control users. * Only you can modify the item. When you press ENTER, your user ID appears in the Userid field.
Userid	Displays the user ID of the user who added/modified the item using the asterisk option (*) in the C/* field.

Display Info Item

When you use the S option to display an information item, the item is displayed in read mode.

- To return to the information item after displaying an item, press either the CLEAR key or PF3;
- To exit to COM-PASS directly from the item display screen, press PF12 .

Add Info Item

When you wish to add an information item, enter the A option in the*Sel* field of a blank line and press ENTER. The *Info-item*and*C/**columns of this line are opened for input. Fill these fields as required and press ENTER. An edit screen appears in which you can type your text.

- Press PF5 to save the new item. Then press PF3 or the CLEAR key to return to the information item menu;
- Press PF12 to return directly to COM-PASS.

Modify Info Item

When you select the M option and press ENTER to modify an information item, The *Info-item* and *C/** columns of this line are openend for input. You can change any these fields if you are authorized (see above). Press ENTER, and the item is displayed in edit mode. You can modify the text as required.

- Press PF5 to save the modified item. Then press PF3 or the CLEAR key to return to the information item menu;
- Press PF12 to return directly to COM-PASS.

  The editor recognizes the following two special editing commands:

- a > (greater than sign) in the first column of a line inserts a blank line before that line;
- a ? (question mark) in the first column of a line deletes that line.

BROADCAST Info

If you name an information item BROADCAST, the logon broadcasting mechanism is automatically activated, which causes the information in this item to be displayed along with the ’Logon Successful’ message whenever any user logs on.

You can use this facility to display system-wide information of interest to all users. An example of a broadcast screen is shown in the following figure:

```
13:56:30        TID     6           COM-4.6.          User SAGAWW       05/21/94
                        ---  User site info system  ---                    UNF1
 >                                                                            <
 >                        W e l c o m e   t o   t h e                         <
 >                                                                            <
 >                     w o n d e r f u l   w o r l d   o f                    <
 >                                                                            <
 >  CCCCC   OOOOO   MM   MM           PPPPP   LL      EEEEEEE TTTTTTTT EEEEEEE <
 > CCCCCCC OOOOOOO MM MMM MM          PPPPPP  LL      EEEEEEE TTTTTTTT EEEEEEE <
 > CC      OO   OO MM  M  MM          PP   PP LL      EE         TT    EE      <
 > CC      OO   OO MM     MM  ******  PPPPPP  LL      EEEEE      TT    EEEEE   <
 > CC      OO   OO MM     MM  ******  PPPPP   LL      EEEEE      TT    EEEEE   <
 > CC      OO   OO MM     MM          PP      LL      EE         TT    EE      <
 > CCCCCCC OOOOOOO MM     MM          PP      LLLLLLL EEEEEEE    TT    EEEEEEE <
 >  CCCCC   OOOOO  MM     MM          PP      LLLLLLL EEEEEEE    TT    EEEEEEE <
 >                                                                            <
 >                                                                            <
 >                                                                            <
 >                           >>>  Version 4.6.1  <<<                          <
 >                                                                            <

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

When defining a BROADCAST message, you can use some special characters which you must type in column 75 (the last text column):

- @ (’at’ sign), which causes the text line to rotate left to right;
- # (hash sign), which causes the text line to blink;
- $ (dollar sign), which rotates the text line from right to left;
- % (percent sign), which causes inverse blinking of the text line.

Function TO - Trace Options Maintenance (Administrator)

Trace options are normally set using appropriate sysparms. The TO function of UUTIL enables the system programmer to modify some of these options online. However, any changes made with the TO function are only active for the duration of the current Com-plete run. When Com-plete is restarted, the values as set in the sysparms are in force again.

If you select the TO option from the utilities menu, the trace maintenance screen is displayed, for example:

```
16:45:58     TID     5          COM-5.1.            User ID MBE        12.11.97
                           -- Trace Options --                                UTR1

 Trace Classes:                                Restrict Trace to TIB:           0

 GENERIC    N   SVCVSE    N                    Additional Options
 QTIB       Y
 OP         N                                  Trace at abend                   N
 FIXBPOOL   N                                  Capture Trace data               N
 VTAM       Y
 ROLL       N
 DISPATCH   N                                  Open SYSTRACE for APPC Trace     O
 ACCESS     N                                  APPC Trace Options:
 SDFILE     N                                    Dump TIB                       Y
 LIBR       N                                    Dump RPL / NIB / RPL6          N
 RESOURCE   N                                    Dump Buffers/Data Areas        Y
 ENQDEQ     N
 OSIM       N
 ENTREXIT   N                                  DYNALLOC Message Level:         12


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

The various trace classes can be set by typing *Y*(on)*N*(off) against the listed trace classes and pressing ENTER. The traces can be restricted to a specific TIB if its TIB number is specified in the corresponding field.

A DD statement for SYSTRACE must be present in the startup JCL if the extended APPC trace is to be opened. Note that the otion field will contain "O" if the trace file is open, ’Y’ if open was requested (but not completed) and "N" if the file is closed. For more details refer to System Programming documentation section *VTAM Interface*.

A message is written to the operator console for each modification made to this screen indicating the changes, userID and TID number of the user that made the changes.

Function TT - TIBTAB Maintenance (Administrator)

This function allows you to maintain the Com-plete TIB definition table used during Com-plete startup to initialize the TIBTAB. During initialization, depending on sysparm TIBTAB, either the TIBTAB load module is loaded or an empty TIBTAB is created. Afterwards, in both cases, the TIBTAB is completed dynamically using this TIB definition table. For details about TIBTAB initialization, please refer to the section on TIBTAB in the Com-plete System Programming documentation.

In the TIB definition table, you can maintain TIBs supported via one of the access methods VTAM or ACCESS. Terminals connected to Com-plete using other access methods must be defined in the TIBTAB load module. The TIB definition table resides in Com-plete's system dataset. By default, the table is empty.

As a side benefit, this function can also be used to perform operations against the in-storage TIBTAB currently being used by Com-plete.

If you select the TT option from the Utilities menu, the TIBTAB Maintenance Functions Menu is displayed:

```
12:37:59        TID    13          COM-5.1.          User MBE          19.04.97
                            -- TIBTAB Maintenance --                        UDT0

 Definitions for STARTUP  Fc Operand    Active TIBTAB            Fc Operand
 ------------------------ -- --------   ------------------------ -- -----------
 List TIB definitions.... LD (prefix)   List Terminals.......... LT (any)
 Show TIB definition..... SD name       Show single TIB......... ST name/number
 Add TIB definition...... AD name       Add TIB................. AT name
 Update TIB definition... UD name       Update TIB.............. UT name/number
 Delete TIB definition... DD name       Delete TIB.............. DT name/number
                                        Copy TIB to definition.. CT name/number



 Select function ........
 TIB name / number ......
 System .................               Options for COPY:
                                        Copy with TID .......... N (Y/N)
                                        Replace ................ N (Y/N)


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

You can select any of the TT subfunctions by entering the appropriate function code and an operand. Generally speaking, the TIB name must be entered to identify a TIB definition; a TIB in the active TIBTAB may be identified by either TIB name or TIB number (TID).

The *System* input field prompted is relevant only for sites sharing Com-plete system datasets among more than one system. Here, the appropriate Patch Character can be entered to maintain startup TIB definitions related to a local system. By default, if this field is left empty, the Global system is assumed. For functions dealing with the active TIBTAB, this field is ignored.

For subfunction CT (Copy) and for the Copy feature of subfunction LT, you can specify whether you want the TIDs to be copied or cleared and if existing TIB definitions shall be replaced or not. All other subfunctions ignore these options.

Note that operations against TIB definitions have no influence on the TIBTAB currently used by Com-plete, they become effective only when Com-plete is initialized next time. Additonally, operations (except Copy) performed on the active TIBTAB do not influence TIB definitions for startup.

List

Use the subfunctions LD / LT to list contents of the TIB definition table or of the active TIBTAB respectively, for example (LD):

```
12:41:57        TID    13          COM-5.1.          User MBE          19.04.97
 System: Global         -- TIB Definitions for Startup --                   UDT1

 FC Name       TID  Type   Device   UserID  Prty Lin Col Status  Alloc Note
 ---------------------------------------------------------------------------
 .  DAEEC306    14  VTAM   3270  L            1   24  80         start


 FC:  S how  U pdate  D elete               Reposition to:                   END
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End                           Forwd
```

You can specify a name or prefix you want the listing to be started from, either in the Operand field on the Functions menu or in the field prompted *Reposition to* on the List screen. The listing will be in alphabetical order by TIB name.

With the LT function, you can also specify a TIB number (TID) in these fields to request listing of the TIBTAB in ascending order by TID. By default, if you select the LD subfunction without an operand, TIB definitions are listed in alphabetical order by TIB name. If you select the LT subfunction without an operand, the active TIBTAB is listed in ascending order by TID.

Meaning of the output fields by column heading:

Heading	Meaning
FC	Input field for subfunction call.
Name	The name of the TIB.
TID	The Terminal Identification number. For subfunction LD, zero indicates that any free TID can be allocated to this TIB.
Type	The access method, e.g., VTAM or ACCESS.
Device	The device type of the terminal.
UserID	Filled out by subfunction LT only. The User ID, if any, currently logged on from the terminal connected to this TIB.
Prty	The dispatching priority (0-3), where 0 is the lowest priority.
Lin	The maximum number of lines to be used in a display or output.
Col	The length of the line (number of columns) to be used.
Status	Possible values for subfunction LT are: active The terminal is currently allocated to Com-plete. deleted The terminal as been deleted, e.g., using operator commands DELETE or IGNORE, or it has been de- fined to be brought up deleted. error A permanent I/O error has occurred at the terminal.
	For subfunction LD, this field is empty, except for TIB definitions with the parameter DELETED=YES specified.
Alloc	For subfunction LT: The way this TIB has been allocated. Possible values are:
	fixed — TIB is defined in the TIBTAB load module. start — TIB has been allocated during Com-plete startup due to an entry in the TIB definition table. autom — TIB has been allocated automatically, e.g., during user logon processing. admin — TIB has been allocated explicitely using the AT sub- function of this utility. - — For subfunction LD: Always "start".
Note	After execution of a subfunction for the appropriate line: Result of the subfunction call.

The Show, Update, Delete, and Copy subfunctions are also available from the menu displayed by the appropriate List subfunction. When called from the List Menu, Delete and Copy subfunctions are executed immediately one after the other, without the TIB Parameter Menu being displayed each time for confirmation.

The TIB Parameter Menu

All of the subfunctions Show, Add, Update, Delete, and Copy use the same TIB Parameter Menu to display and / or enable input or modification of TIB attributes, for example:

```
16:13:22        TID    12          COM-5.1.          User MBE          19.04.97
System: Global         -- TIB Definitions for Startup --                   UDT2

TIB/LU name: DAEEC306         TIB number (TID):   14
-------------------------------------------------------------------------------
Access method / CUU: VTAM     (VTAM/ACCESS/cuu)
Device type:........ 3270  L                                Hardcopy device:. No
Access node number:.     0

Alternate TID:......     0                                  Share (VTAM):.... No
Permanent HC TID:...     0                                  Acquire (VTAM):.. No

Lines X line length:    24 X  80                            Stall:........... No
Alternate screen:...    24 X  80                            Deleted:......... No
VTAM RUsize:........     0
                                                            Compress output:. Yes
                     1 2 3 4 5 6 7 8
Receive msg classes: X X   X                                In "ALL" group:.. Yes
User options:.......

Printer type:.......             Forms:
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  End
```

Most of the TIB parameters shown on this screen are equivalent to the parameters of the TIB macro used for generation of the TIBTAB load module. For detailed description of TIB parameters, please refer to the section on TIBTAB in the Com-plete System Programming documentation.

For printer devices, the following additional parameters kept in an extension of the TIB are available:

Heading	Meaning
Printer type	Eight-character printer type or class specification which can be analyzed, e.g., by logical output drivers. There are no restrictions on the contents of this field.
Forms	For printer devices supporting multiple printout forms: Specifies up to eight forms supported by a printer at one time. If one of these form names is specified for a printout in the Printout Spool Control Block (PSCB), and if a logical output driver exists, the printout is scheduled to the printer undependently on the form currently mounted for the printer. The logical output driver must analyze the form specified for the printout and take the appropriate actions.

If any of these parameters is specified for a printer, field TIBAPPRF contains the 31 bit address of a 40 byte block of following structure (residing above the 16MB line where applicable):

Location		Length	Format	Contents
Dec	Hex			
0	0	8	Character	Printer type
8	8	4	Character	Form 1
12	C	4	Character	Form 2
16	10	4	Character	Form 3
20	14	4	Character	Form 4
24	18	4	Character	Form 5
28	1c	4	Character	Form 6
32	20	4	Character	Form 7
36	24	4	Character	Form 8

The default value for all fields is a character string of spaces. If none of the parameters is specified for a printer, field TIBAPPRF contains a value of hexadecimal zeroes.

Show

Using the SD / ST subfunctions, you can display the parameter settings specified for a TIB definition or the values currently in effect for a TIB in the active TIBTAB.

Add

Using the AD / AT subfunctions, you can add a new TIB definition or explicitly allocate another TIB from the active TIBTAB. The TIB name to be added must be specified in the Operand field of the Functions menu. The TIB Parameters Menu is displayed, with input fields for all parameters.

All parameter values are defaulted to the values of the TIB or TIB definition last used in a Show, Add, Update, Delete, or Copy subfunction during the current UUTIL TT session. If these values do not match your requirements, you can enter the access method and device type desired and press PF8 to set up the default values for this combination.

Note that the only access methods supported are VTAM and ACCESS.

You can specify a fixed TID to be used. If this TID is not free, allocation will fail with an appropriate warning message (immediately after you press PF5 for subfunction AT, during next Com-plete startup for subfunction AD). If you leave the TID zero, the TIB will be created with any free TID.

Specify the desired parameter values and press PF5 to store the new TIB definition or to allocate a TIB with the indicated attributes.

Update a TIB Definition

When you call the UD subfunction, the TIB Parameters Menu appears, displaying the TIB definition. You can now modify any parameter values and update the TIB definition using the PF5 key.

Update a TIB in the Active TIBTAB

Using the UT subfunction, you can modify some of the parameters of a TIB in the active TIBTAB.

Note:
Unpredictable results may occur if you change a TIB in the active TIBTAB while it is in use, e.g., for terminal I/O. Therefore, the UT subfunction should be used very carefully.

When you call the UT subfunction, the TIB Parameters Menu appears, displaying the parameter values currently in effect for the TIB.

You can modify most of the parameters. TID, access method, and device type cannot be changed. When you press PF5, first the TIB is checked for any changes occurred since you entered the UT menu. If contents of any important field of the TIB have changed in the meantime, your modification will be denied. Otherwise, the TIB will be updated.

Delete a TIB Definition

When you call the DD subfunction, the TIB Parameters Menu appears, displaying the TIB definition. You can now delete the TIB definition by pressing the PF5 key. From the menu displayed by the LD subfunction, you can delete multiple TIB definitions at one time without being requested to confirm each of them.

Free a "Deleted" TIB from the Active TIBTAB

Using the DT subfunction, you can free a TIBTAB entry which has the status "deleted" (see description of subfunction LT above in this section). If you want to free a TIB which has status "active" or "error", it must be "deleted" first using operator command IGNORE.

Copy Contents of a TIB from the Active TIBTAB to a TIB Definition

Subfunction CT allows you to save contents of a TIB as a TIB definition for next Com-plete initialization. You can specify the TIB by name or by TID, and you can set additional options at the TIBTAB Maintenance Functions Menu as described above in this section. From the menu displayed by the LT subfunction, you can copy multiple TIBs at one time without being requested to confirm each of them.

Function TU - Terminal / User ID Group Maintenance (Administrator)

The name of a group of terminals, lines, and / or User IDs can be used in operator commands and as a target for message switching.

A group may contain any combination of terminals, lines, and / or User IDs. Each terminal or line may be identified by either its number (TID/LID) or by its name.

When a group name is specified in an operator command or as a receiver of a message, it is converted to a list of terminal identification numbers (TIDs).

Note:
If a group contains a terminal name and more than one terminal with this name exist, only the lowest of their TIDs is taken into account when this TID list is being created.

The TU function of UUTIL allows you to maintain these groups.

If you select the TU option from the utilities menu, a screen is displayed listing the names of all groups currently defined, for example:

```
11:47:40        TID    13          COM-5.1.          User MBE          19.04.97
 System: Global             -- TIB / UserID Groups --                       UTU0

 Enter group name: DEMO     System:   or select group by cursor
 ------------------------------------------------------------------------------
 DEMO









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

You can enter the name of a group you want to modify or add, or select a group name from the list by placing the cursor on it and pressing ENTER to display the group maintenance menu, for example:

```
11:49:45        TID    13          COM-5.1.          User MBE          19.04.97
 System: Global        -- TIB / UserID Group Maintenance --                 UTU1

 Group:  DEMO         Add: T =          ( T=TIBname / T=TIBnumber / U=userID )
 ------------------------------------------------------------------------------
 T=_______1
 T=PRINT014
 U=USERID1
 U=USERID2
 U=USERID3









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Save        Up    Down                    Remove
```

From this menu, you can perform any of the following:

- Scroll the listing of the group forward PF8 and backwards PF7
- Add a TIB or an User ID to the group. To achieve this, specify the item you want to add in the input fields after the Add prompt and press ENTER
- Remove a TIB or an User ID from the group. Place the cursor at the member you want to remove and press PF12. If you remove the last member from a group, the group will be deleted immediately

Additions and removals (except deletion of the whole group) become effective after you press PF5.

Function UD - Maintain User Defaults

This function allows you to specify various parameters to be used by COM-PASS and other Com-plete utilities. If you select the UD option from the Utilities menu, the screen displayed shows which parameters can be specified:

```
16:42:42        TID    12          Com-plete         User MBE          13.11.00
System: Global          -- User Default Parameters --                      UDEF

----------------------------   Global  Defaults   ----------------------------

DSN                                                  Volume        Device
Libid/Member     /
                                                   PF01    PF02    PF03    PF04
                                                   PF05    PF06    PF07    PF08
Language....   0   Case.......                     PF09    PF10    PF11    PF12
                                                   PF13    PF14    PF15    PF16
Recallchar...                                      PF17    PF18    PF19    PF20
Suspendchar..      Suspendkey. PA1                 PF21    PF22    PF23    PF24
Jumpchar.....      Jumpkey.... PF24
Printer DUMMY      HC Key..... PA2                 PF 13-24 = 01-12 (Y/N): N
Server.
----------------------------   Utility Defaults   ----------------------------

UDS Function                                UPDS    Function
UQ  Function      Keywords  JB= SKU       SI= CC  DS= 1    LL= 80   PP= 1

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

Some of the parameters (such as Suspendchar or Suspendkey) override the defaults set by the Com-plete system administrator. The Utility Defaults have no global predefined value and you can set them here. For the meaning of the fields, see the description of the same screen in the next section *Function UM - User Maintenance*.

Function UL - Maintain User Lib IDs

This function allows you to specify two-letter short IDs for libraries or dummy libraries for PC file transfer support. If you select the UL function from the utilities menu, the user short IDs are displayed, for example:

```
16:59:36        TID     6           COM-5.1.          User SAGAWW       05/21/97
 System: Global                                                            ULID
 Page 1 of 2                --- Library Id Table ---
                                                          Lib
 Id  DSName..................................... VOLSER Typ
 5L  COM.SYSF.V46.USER.LOAD
 5U  COM.SYSF.V46.UA.SOURCE
 CY  COM.SYSF.IV441.SYSTEM
 WS  COM.SYSF.SAGAWW.SOURCE
 CO  COM.SYSF.IV460.XA.LISTING
 5Z  COM.COMN.C460.ZAPS
 5D  COM.SYSF.IV460.DOC
 L5  COM.SYSF.IV460.XA.LOAD
 45  COM.BETA.IV460.SYSTEM
 PC  THE.PC.LIBRARY                                        M
 CW  COM.SYSF.DEV.WORK




     Lib-Typ  =  V (Panvalet),  L (Librarian),  M (Personal Computer)

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

Meaning of the items according to column header:

Header	Meaning		
ID	Short ID of the library. You can use this ID in all Com-plete functions.		
DSNAME	Full name of the data set referred to by the short ID.		
VOLSER			
	z/OS	The volume serial number is required only if the library is not cataloged.	
	VSE	The volume serial number is required.	
Lib Typ	Type of library. Possible options:		
	M	Specifies a dummy library for transfer functions between PDS members and PC files. For more information, see the section on UEDIT.	
	z/OS	’’	partitioned dataset
		V	PANVALET library
		L	LIBRARIAN library
	VSE	A	VSAM library
		B	Non-VSAM

Press PF5 to save all entered data. PF3 or the CLEAR key return you to the Utilities menu.

Function UM - User Maintenance (Administrator)

The UM function of the UUTIL utility is designed for use at a control terminal to:

- Maintain user data;
- Maintain the COM-PASS environment;
- Display Com-plete user ID definitions;
- Display active users of Com-plete.

  Com-plete definitions of users who are authorized to access Com-plete are stored on the so-called system data container and are referenced by the ULOG utility program each time a user attempts to log on to the Com-plete system. Each user definition contains the following information:

- User ID;
- Account number or group number for the user ID;
- Password for the user ID; number of invalid password attempts since the last successful logon;
- Authorization code for the user ID;
- Control status to be given the user ID;
- Sending and receiving class codes to be assigned to this user ID for sending and receiving messages and for printout spooling;
- Date and time of the most recent ULOG ON session;
- COM-PASS defaults and PF key settings.

When a terminal user is identified to the Com-plete system with the ULOG utility program, the information from the user ID data set is used to create a logon information block. After logging on successfully, this information is used to create the user ID accounting block (see also *UED Edit Control Block* in the Com-plete System Programming documentation). Statistical information for the user ID is accumulated and maintained by using the user ID accounting block, which in turn is written to the SMF data set. Each user ID accounting block is purged from the Com-plete system when either the associated user logs off or Com-plete terminates.

User Maintenance Functions

After invoking this function from the UUTIL main menu, the user maintenance menu is displayed. This menu enables the definition, control, and maintenance functions to be performed:

```
COMULM0021-* Please enter operation or hit 'CLEAR'
 17:45:12        TID    11          COM-5.1.          USER SAGAWW       30.11.97
                                                                            ULM1
                       ---  User ID Maintenance Menu  ---
 Operation                Fc Operand         Operation                Fc Operand
 ------------------------ -- -------         ------------------------ -- -------
 Define a User Id........ AD 1,2,3,4         Display a User Id....... DS 1,2
 Update a User Id........ UP 1,2             Display selected Users.. DI 6
 Update selected User Ids UA 6               Display Active Users.... DA 6
 Delete a User Id........ DE 1,2
 Lock   User ID(s)....... LO 1,2,6           Maintain COM-PASS menu.. CM 5
 Unlock User ID(s)....... UN 1,2,6           Set Maintenance Password MA


 Fc:     (1) User Id..:           (2) System:       (6) Selection:
         (3) copy from:           (4) System:
         (5) Language.:   1

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

Most operations provided speak for themselves. The numbers listed in the*Operand* field following each operation relate to the relevant operands for the function, which can be specified in the bottom half of the screen. in the bottom half of the screen.

Note:
If you are creating user profiles for the first time, you should consider defining the COM-PASS environment first. See the description of function CM (Maintain COM-PASS menu) below.

Creating a New User Profile

To add a new user, select the AD operation and supply the new user ID and optionally system ID. You can also specify an existing user ID in the *copy from*field. If no *copy from* user ID is specified, $MODLSTD is implicitly used. You can modify the $MODLSTD user definition to set defaults for all subsequent *ADD*requests.

The following screen illustrates an example user definition screen:

```
COMULM0052-* Please enter additional arguments...
 16:54:39        TID     5          COM-5.1.          User MBE          12.11.97
                         ---  User ID Maintenance  ---                      ULM3

 Add      User NEWBOY    System Global
 User Name....
                                                          Last update.: SKU
 This is a Model UserId  NO                                         on: 12:41:15
 Based on Model UserId.                                             at: *  ¥ Ø
 Control Status........  YES                                       Sys: 7
 Account Number........
 Authorization Code....       0
 User Password.........  PASSWORD
 Exempt from Autologoff  YES
 COM-PASS User.........  YES
 Suspend 9 Programs....  YES
 Non-Menu Programs.....  YES
 Start-Up Program......  USTACK
 Dispatch Priority.....  1
 Send Message Codes....  1   2   3   4 X 5   6   7 X
 Receive Message Codes.  1   2   3   4 X 5   6 X 7 X

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Updat             fwd   MenuL             Quit
```

Note:
From this screen, PF9 takes you straight to the third user definition screen, namely the menu program selection screen (see below).

Having completed your modifications, press PF5 to save the profile.

Meaning of the options:

Option	Meaning
This is a Model User ID	YES means that this user ID can ONLY be used to serve as a model definition for a logon via one of the Com-plete facilities that use model IDs (applymod 57, NSC link,...).This user ID cannot be specified on the ULOG ON screen.
Based on Model User ID	Enter an existing model user ID (optional). If the user logs on, the model user’s definitions are taken. The advantage of using this option is that to change the definitions of a user group based on the same model, only the model User Profile need be changed.
Control Status	Specifies CONTROL (YES) or NONCONTROL (NO) status. Control status is required to enable usage of the "control user" utilities. Administrator status is attained through knowledge of the maintenance password.
Account number	Specifies a 1 to 12 digit account code or group number to be assigned to the user ID.
Authorization code	Specifies the authorization code to be be assigned to the specified user ID. Represents an integer from -32768 to +32767.

Option	Meaning
User Password	Specifies the password to be assigned to this user ID. Note that if the password option is selected at Com-plete initialization time, this password must be entered along with the user ID in order to gain access to the Com-plete system.
Exempt from	A value of "YES" indicates that this user is not subject to autologoff the inactivity value supplied through the AUTOLOGOFF sysparm value. The default value is "NO".
COM-PASS user	This option acts as a "COM-PASS switch". You may decide not to let a user become a COM-PASS user until all definitions are complete. The Profile can be updated to make the user (or all users) a COM-PASS user in one operation.
Suspend	n is a value from one to nine, indicating the number of programs this user can suspend. The default value for n is nine (9).
	At logon time, n is compared to the STACKMAXIMUM sysparm value. The number of programs the user can suspend is n or the STACKMAXIMUM value, whichever is the smaller.
	This feature is used to convert Com-plete from a non-COM-PASS production system to a COM-PASS production system.
Non-menu programs	A value of "NO" indicates that this user can only access those programs that appear in his COM-PASS menu. "YES" indicates that he can also access programs that do not appear in his COM-PASS menu.
Startup program	This option acts as a "startup switch". If the user is to be automatically supplied with a startup transaction after logon, enter the name of the transaction here. If the user is to receive the COM-PASS menu, specify USTACK.
Send Message Codes	Specifies the authorization class code(s) to be assigned to this user ID for sending messages and printout spooling. Authorization class code 4 (used for Com-plete messages) is always assigned even if not specified. The authorization class codes are fully described in the section UM - Message Switching/Printout Spooling Utility.
Receive Message Codes	Specifies the authorization class code(s) to be assigned to this user ID for receiving messages and printout spooling. The authorization class codes are full described in the section UM - Message Switching/Printout Spooling Utility. If RMC=(0), this user ID is not allowed to receive messages or printouts. If SMC=(0), this user ID is not allowed to send messages or printouts.

Press PF8 to proceed to the second screen:

```
16:51:22        TID    12          Com-plete          User MBE         13.11.00
System: Global          -- User Default Parameters --                       UDEF

----------------------------   Global  Defaults   -----------------------------

DSN                                                  Volume        Device
Libid/Member     /
                                                   PF01    PF02    PF03    PF04
                                                   PF05    PF06    PF07    PF08
Language....   0   Case.......                     PF09    PF10    PF11    PF12
                                                   PF13    PF14    PF15    PF16
Recallchar...                                      PF17    PF18    PF19    PF20
Suspendchar..      Suspendkey. PA1                 PF21    PF22    PF23    PF24
Jumpchar.....      Jumpkey.... PF24
Printer DUMMY      HC Key..... PA2                 PF 13-24 = 01-12 (Y/N): N
Server.
----------------------------   Utility Defaults   -----------------------------

UDS Function                                UPDS    Function
UQ  Function      Keywords  JB= SKU       SI= CC  DS= 1    LL= 80   PP= 1

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  End         Updat       bwd   fwd                     Quit
```

In this screen, you can:

- Specify the short-ID of the library, as well as the member to be used as default when the UPDS utility is invoked;
- Specify the default volume serial number and device type when the UDS utility is invoked;
- Set the default language and case for screens;
- Assign PF keys to levels on the COM-PASS menu by entering either a suspend level number (1 through 9) or a transaction Profile identity (A through I) next to the PF key;
- Set COM-PASS characteristics;
- Specify the default printer;
- Specify an LPD print server. When this field is filled, all printouts from this user are routed to this server, using the printer name entered for the print queue name. Enter an asterisk (*) in the first position of this field to route printouts to the user’s workstation. Note that this feature works only when the user connects to Com-plete’s individuasl telnet tn3270 port, and an LPD server is active on the user’s workstation.
- Specify default functions when UPDS or UDS in invoked, as well as a default function and default keyword values when UQ is invoked.

The PF key and default settings can initially be defined on this screen and modified later by the user. PF keys can be modified on the main menu, and default settings can be modified with the UD function.

Press PF8 to display the menu program selection screen for the user. This screen lists the programs defined for the COM-PASS menu (see function CM, Maintain COM-PASS menu, below). Using the Page and ID fields, you can specify on which COM-PASS menu page the program is to appear and with which letter it can be called (the programs are also listed alphabetically according to the ID on the user’s

COM-PASS menu):

```
15:06:52                      User ID  Maintenance                     Page    1
 Sys Global Lang   1          Menu Program Selection                        ULMA

 PageID .... Menu Text .....  Program      PageID .... Menu Text .....  Program
  1  E  THE Editor            UEDIT                                     TESTCOK
  1  B  PDS functions         UPDS
  1  I  Map definition        UMAP
        Switch to lower case  LOW                                       TESTCOK1
        Switch to upper case  UP
        Online spooling       USPOOL
  1  F  Online dumps          UDUMP
  1  A  Storage display       USTOR
  1  D  JES functions et al   UQ
  1  C  Online control        UCTRL
        Utilities Menu        UUTIL
  1  G




      Used IDs -> Page1: ABCDEFG.I   Page2: .........   Page3: .........
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   Quit        Updat       bwd   fwd               Reset
```

When all the User Profile information has been entered, saved the profile by pressing PF5 until you see the message User *userid* successfully added.

Updating a User Profile

Existing User Profiles can be updated. Select function UA with a user ID to update a single user definition, or specify Y in the selection field to select a group of users according to the displayed criteria. The user definitions are then presented one by one.

Deleting a User Profile

Redundant user profiles can be deleted using option DE from the user ID maintenance menu. After the profile to be deleted is selected, the profile is displayed and you are asked to verify that the profile is to be deleted.

Locking/Unlocking Users

A user or group of users can be locked and subsequently unlocked from using Com-plete, for example, while maintenance is being carried out or while the user is away from the installation for a long period of time.

Select option LO/UN from the user ID maintenance menu to do this. You can specify a specific user ID, or specify Y in the selection field to lock/unlock all users belonging to a particular account.

Displaying Profiles and Users

The DS option enables individual User Profiles to be displayed and updated. The DI option enables all defined users (or selected groups) to be displayed. The DA option enables all active users (or selected groups) to be displayed. Specify Y in the selection field to specify selection

criteria for the DI and DA functions.

Maintenance Password

You can change the maintenance password using option MA from the user ID maintenance menu.

Maintain COM-PASS Menu

The CM function allows you to set up the COM-PASS environment, which you can subsequently tailor for each User Profile or group of User Profiles (functions AD, UP, UA).

You can define the program names and program descriptions to appear on the COM-PASS menu, a one-letter ID to be associated with each program, the menu page, as well as the conditions under which the program is to appear. Additionally, you can link or unlink specific menu lines to user IDs.

The COM-PASS menu definition function operates in two modes:

- PROGRAM mode, used to define the program names to appear on the user's COM-PASS menu
- TEXT mode, meaning there is only ONE set of program names, the program descriptions can be added in various languages, temporarily activated using the LANG command from COM-PASS, or perrmanently set for the user by the administrator using the UM UP function. The user can modify the language using UUTIL function UD.

When you select the CM option, the COM-PASS menu screen is displayed:

```
17:30:34              User ID Maintenance - COM-PASS Menu              Page   1
 System..: Global                                                          ULM0
 Language:   1                              Propagate into
 S Nbr  Menu Text...........  Programs      P/E  Ovw  Free   Message............
     1  THE Editor            UEDIT         0     N    N
     2  PDS functions         UPDS          0     N    N
     3  Map definition        UMAP          0     N    N
     4  Switch to lower case  LOW           0     N    N
     5  Switch to upper case  UP            0     N    N
     6  Online spooling       USPOOL        0     N    N
     7  Online dumps          UDUMP         0     N    N
     8  Storage display       USTOR         0     N    N
     9  JES functions et al   UQ            0     N    N
    10  Online control        UCTRL         0     N    N
    11  Utilities Menu        UUTIL         0     N    N
    12                                      0     N    N
    13                                      0     N    N
    14                                      0     N    N
    15                                      0     N    N
    16                                      0     N    N

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End         Updat       bwd   fwd         Text        Quit
```

Initially the screen is clear; up to 240 programs (8 screen pages) can be added. Enter the menu text and program names in the appropriate columns as in the above example, using PF10 to toggle between text and program name input.

The other input fields have the following meaning:

Column	Meaning
S	Enter L (for LINK) or U (for UNLINK) for the entry to be linked or unlinked and specify one or more of the following parameters under the heading Propagate into:
P/E	Specifies the Page/Entry number (1A to 3I) of the COM-PASS menu slot where the menu program is to appear.
Ovw	Set to Y to force the selected menu program into the slot specified in P/E, otherwise it will be linked only if the selected slot is empty.
Free	Specifies that the first empty slot on the user's COM-PASS menu will be used for the selected menu program.

For both the L and U selection, the following screen will be presented:

```
COMULM0052-* Please enter additional arguments...
 08:44:46        TID    10          COM-5.1.          User SAGAWW       09.12.97
                                                                             ULM7
                          --- User ID Maintenance ---
                              Selection  criteria


 Account number ...:
 Authorization code:     0
 Control User (Y/N):
 Startup program ..:
 TID number .......:     0
 UserId ...........:
 System Id ........:









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec              EndIn                                                Aband
```

Specify criteria to select the range of users the selected menu entry will be linked to/unlinked from and press ENTER.The COM-PASS menu is redisplayed. Now press PF5 to save the definition. Return to the user maintenance menu by pressing the CLEAR key or PF3.

Function UP - PF Key Defaults (Administrator)

This function allows you to assign commands to PF keys in the various Com-plete utilities. These settings are installation defaults and can be modified by users for the utilities for which they are authorized with the UD function. Pressing a PF key in the utility has the same effect as if the assigned command was entered in the command line.

If you select the UP option from the utility menu, the list of utilities for which this function is supported is displayed, for example:

```
17:11:55        TID     4          COM-5.1.          User MBE          21.04.97
 System: Lcl *            ---  PF-Key Maintenance  ---                      UPF0

 Nbr Utility     Nbr Utility     Nbr Utility     Nbr Utility     Nbr Utility
 --- --------    --- --------    --- --------    --- --------    --- --------
   1 COM-PASS     13              25              37              49
   2 UQ           14              26              38              50
   3 UPDS         15              27              39              51
   4 USTOR        16              28              40              52
   5 UDUMP        17              29              41              53
   6 UDEBUG       18              30              42              54
   7              19              31              43              55
   8              20              32              44              56
   9              21              33              45              57
  10              22              34              46              58
  11              23              35              47              59
  12              24              36              48              60

 Please select Utility number..:
           and System Id.......:

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

Select the utility for which you wish to define PF Keys by typing the number in the *Utility Number* input field. If you have multiple Com-pletes installed, and you do not enter data into the *System ID* field, your Global PF Key definitions for the specified utility are defined. If you require special PF Key definitions for one Com-plete system only, enter its ID in this field (in the above example, the current Com-plete system has an ID of 8).

When you select a utility and press ENTER, the PF definition screen is displayed, for example (no system ID specified):

```
COMUPF0015-* New PF key table will be added...
 17:13:43        TID     4          COM-5.1.          User MBE          21.04.97
 System: Global                                                           UPF1
                              Dynamic PF-Key Table
                               UQ       (Private)                       Dis-
 PFnr Command                                                           play   >
 ---- ---------------------------------------------------------------  ----   -
 PF01
 PF02
 PF03
 PF04
 PF05
 PF06
 PF07
 PF08
 PF09
 PF10
 PF11
 PF12
 ---- ---------------------------------------------------------------    -    -
      Override undefined keys with system default keys: N

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
             Mode  End         Updat       bwd   fwd         Purge       Quit
```

You can enter or modify data in the input fields. Meaning of the fields according to column header:

Column	Meaning
Command	The command to be executed when the PF key is pressed. Valid commands are: Com-plete control functions. Com-plete commands, optionally with parameters, to execute a utility program, for example: **`*UQ Q,JB=name,RR=105=UEDIT PC(COMSM1)`** Program-specific commands. For example, for UQ: **`A,RR=3`** If the required command string exceeds 64 characters, press PF2. This displays the screen in which you can enter up to 79 characters for each command string.
Display	Enter any character here to cause the assigned command to be displayed in the command line when the PF key is pressed. The displayed command can be modified before you press ENTER to execute it. If you use underscore characters in the assigned command string, the cursor is placed on the first underscore when it is displayed.
Mo	An asterisk (*) is automatically displayed in this column if you have entered more than 64 characters in the command string using PF2.

Press PF5 to save the definitions. Press PF3 to leave this function.

Function ZA - Zap and Applymod Display (Administrator)

This menu-driven function serves as a display utility for the Com-plete applymods that are in force for the current session.

If you select the function code ZA from the UUTIL main menu, the Applymod Display appears on your terminal screen:

```
13:10:58     TID    14            Dev--621       User COK           02.10.02
                                                                     UZA1
                        ----    Applymod Display      ---


     Fc Function                                Fc Function
     -- --------------------                    -- --------------------
     AL Applymod List ......                    AU Used Applymods .....




     Select Function ..:
     From Applymod ....:   0
     To   Applymod ....:   0


     You are running Com-plete  621   Patchlevel    05
                     APS        271   Patchlevel    06

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

Meaning of the functions:

Function	Meaning
AL	Displays a list of all available applymods
AU	Displays the applymods that are currently effective for the Com-plete session within the range specified in the From Applymod and To Applymod fields.

ZO - Zap Overview

To display an overview about the status of zaps, select the ZO function on the Zap Display menu. This displays the following screen:

```
10:43:43        TID    16          COM-5.1.          User MBE          22.04.97
 System: Lcl *                --- Zap overview ---                          UZA2

   COM-V460  I  ....+....1 I ....+....2 I ....+....3 I ....+....4 I ....+....5 I
 ------------+-------------+------------+------------+------------+------------+
  001 - 050  I  .......... I .......... I .......... I .......... I .......... I
  051 - 100  I  .......... I .......... I .......... I .......... I .......... I
  101 - 150  I  .......... I .......... I .......... I .......... I .......... I
  151 - 200  I  .......... I .......... I .......... I .......... I .......... I
  201 - 250  I  .......... I .......... I .......... I .......... I .......... I
  251 - 300  I  .......... I .......... I .......... I .......... I .......... I
  301 - 350  I  .......... I .......... I .......... I .......... I .......... I
  351 - 400  I  .......... I .......... I .......... I .......... I .......... I
  401 - 450  I  .......... I .......... I .......... I .......... I .......... I
  451 - 500  I  .......... I .......... I .......... I .......... I .......... I
  501 - 550  I  .......... I .......... I .......... I .......... I .......... I
  551 - 600  I  .......... I .......... I .......... I .......... I .......... I
  601 - 650  I  .......... I .......... I .......... I .......... I .......... I
  651 - 700  I  .......... I .......... I .......... I .......... I .......... I
  701 - 750  I  .......... I .......... I .......... I .......... I .......... I

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End                     bwd   fwd                     Quit
```

The digit in every position that represents a specific zap number denotes the SM level of Com-plete for which the zap was applied. Zaps applied by your installation to solve a specific problem between SM levels will be shown as a plus sign (+).

ZS - Display Status of a Single Zap

To retrieve information about the status of a single zap, select the ZS function and enter the number in the *Zap number* field. The status information on the selected zap is displayed in the message line of the screen, for example:

```
comzap0004-* Zap CP46100 is  - NOT -  applied
10:45:18        TID    16          COM-5.1.          User MBE          22.04.97
                                                                          UZA1
                        --- Zap and Applymod Display ---


        Function . . . . . . Fc                  Function . . . . . . Fc
        -------------------- --                  -------------------- --
        Zap Overview ....... ZO                  Applymod List ...... AL
        Single Zap status .. ZS                  Used Applymods ..... AU


                             Select Function .... ZS

                             <--  and operands  -->

         Zap number ........ 100                 From Applymod .....   0
                                                   to Applymod .....   0




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

AL - Display Applymod List

To display a list of all available applymods, select the AL function on the Zap and Applymod Display screen:

```
11:23:29        TID    13          COM-5.1.          User MBE          22.04.97
                            --- Applymod Display ---                       UZA3
 S Nbr St Applymod Usage                                                   Mod
 - --- -- ------------------------------------------------------------ ---
     1 On No page eject at start of a hardcopy                            Y
     2 On Page eject after a hardcopy                                     Y
     3    Call ULSRPSFS for each internal Com-plete operation
     4    UEDIT does not delete SD file at termination                    Y
     5    Display date and time on the hello message                      Y
     6    Noncontrol users may issue operator commands                    Y
     7    Noncontrol terminals may issue TID functions                    Y
     8    ... Currently no function assigned to this Applymod
     9    ... Currently no function assigned to this Applymod
    10    ... Currently no function assigned to this Applymod
    11    ULOG: Display the Account field                                 Y
    12    UQ: Pass '//*UQ ...' card to UUQEX1, except '//*UQ ALLOW'       Y
    13    Mapping system rewrites map constant fields                     Y
    14    Priority messages can be sent from batch programs               Y
    15    No MSG/PO if it gets an I/O error                               Y
    16    UEDIT Submit default is NOSAVE                                  Y

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                   End                     bwd   fwd                     Quit
```

Applymods currently set are indicated by "On" in the column headed "St". Dynamic applymods are indicated in this display "Y" in the column headed "M".

Modifying Applymods (Control Users only)

Applymods that are defined as dynamic can be activated or deactivated from this list by entering "F" (Flip) for the applymod in the column headed "S".

Note:
Applymod modifications are in effect system-wide during the current Com-plete session only. If you want the applymod to be effective for a longer period, use the APPLYMOD sysparm and restart Com-plete.

Displaying Information on Applymods

You can display information about an applymod by selecting it with the "I" line command in the column headed "S".

For additional details on applymods, see *Getchr Information Table* in the Com-plete System Programming documentation or enter "H" for HELP in the "S" column.

AU - Display Used Applymods

To display the applymods that are currently in force for the Com-plete session, select the AU function from the Zap and Applymod Display. The following screen is displayed:

```
11:33:54        TID    13          COM-5.1.          User MBE          22.04.97
                            --- Applymod Display ---                        UZA3
 S Nbr St Applymod Usage                                                    Mod
 - --- -- ------------------------------------------------------------ ---
     1 On No page eject at start of a hardcopy                             Y
     2 On Page eject after a hardcopy                                      Y
    29 On Screen format taken from VTAM bind area                          Y
    52 On Logon possible through the DATA parameter of VTAM Logon          Y
    57 On Allow any userid to logon, password verified by external secsys  Y
    61 On European date in map header                                      Y
    74 On Don’t pagefix Nucleus or Tibtab
    76 On Don’t pagefix Threads
    83 On Don’t keep Password for RJE                                      Y
     0
     0
     0
     0
     0
     0
     0

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

The handling of used applymods is the same as for the applymod overview discribed above.