

Mactra

**The Access Control Facility
Utilities Manual**

acf2[®]

The Access Control Facility

UTILITIES MANUAL

for

acf2/MVS Release 4.1 Installations

Base Manual Dated: January 15, 1985
Includes Revisions Dated: January 31, 1986

Doc. Nr. ABP0004-02



© Copyright SKK, Inc., U.S.A., 1978, 1981, 1982, 1983, 1984, 1985.
All rights reserved.

Reproduction of this manual without written
permission of SKK, Inc. is strictly prohibited.

Printed in U.S.A.

ACF2 is a Registered Trademark of SKK, Inc.

acf2/MVS is a proprietary product developed and maintained by:

SKK, Inc.
10400 West Higgins Road
Rosemont, Illinois 60018-3790

Business Office: (312) 635-1040
Product Support: (312) 635-3000
TELEX: 206-186 (SKK ROSM)

A 24 hour answering service on (312) 825-5150 is available
for emergency assistance outside of normal business hours.

ACF2 UTILITIES MANUAL

<u>Chapter</u>	<u>page</u>
INTRODUCTION	1
GENERAL INFORMATION ON ACF2 UTILITIES	2
Execution of Report Generators Through the Use of JCL	
Supplied with ACF2	2
General Functions of the REPORTS JCL	2
Input and Output Files for Report Generators	3
Required Access for Executing Report Generators	5
Parameters	5
Executing ACF2 Report Generators through the use of ISPF	
Screens	8
Using the ACF2/ISPF Screens for Report Selection	8
HELP Screens for Executing a Report Generator	10
Execution of Other ACF2 Utilities through the use of JCL	11
Files and Parameters for Other ACF2 Utilities	11
Executing Other ACF2 Utilities through the use of ISPF	
Screens	11
Using the ACF2/ISPF Screens for Utility Selection	11
HELP Screens for Executing a Utility Program	13
Execution of ACF2 Utilities Via the TSO CALL Command	13
Input Records for ACF2 Report Generators	14
Report Generator Sort Sequence	15
ACF2 Reports When ACF2 Is Inactive	16
Executing ACFRPTSL When ACF2 Is Inactive	16
Executing ACFRPTXR When ACF2 Is Inactive	16
ACFRPTCR - TSO COMMAND STATISTICS LOG	17
Files	17
Parameters	17
Sort Sequence	18
ACFRPTCR - Sample Output (acf2/MVS Releases 4.0 and Above)	19
ACFRPTCR - Sample Output (acf2/MVS Version 3.1.5 and Earlier)	21
Field Descriptions (acf2/MVS Release 4.0 and Above)	22
Field Descriptions (acf2/MVS Versions 3.1.5 and Earlier)	23
ACFRPTDS - DATASET/PROGRAM EVENT LOG	25
Files	25
Parameters	25
Sort Sequence	29
ACFRPTDS - Sample Output (Terminal Format)	30
ACFRPTDS - Sample Output (Printer Format)	31
ACFRPTDS - Sample Output (Summary Format)	32

NEXTKEY Reporting	33
ACFRPTDS - Sample Output (Excess NEXTKEYs)	34
Field Descriptions	35
 ACFRPTEL - INFORMATION STORAGE UPDATE LOG	 44
Files	44
Parameters	44
Sort Sequence	46
ACFRPTEL - Sample Output (SUMMARY Parameter)	47
ACFRPTEL - Sample Output (DETAIL Parameter)	48
Field Descriptions	49
 ACFRPTIX - DATASET INDEX REPORT	 52
Files	52
Parameters	52
ACFRPTIX - Sample Output	53
Field Description	54
 ACFRPTJL - RESTRICTED LOGONID JOB LOG	 56
Files	56
Parameters	56
Sort Sequence	57
ACFRPTJL - Sample Output	58
Field Descriptions	59
 ACFRPTLL - LOGONID MODIFICATION LOG	 61
Files	61
Parameters	61
ACFRPTLL - Sample Output (SUMMARY Parameter)	62
ACFRPTLL - Sample Output (DETAIL Parameter)	63
ACFRPTLL - Sample Output (DETAIL Parameter)	64
FIELD DESCRIPTIONS	64
Additional Fields for the DETAIL Parameter	66
 ACFRPTNV - THE ENVIRONMENT REPORT	 67
Files	67
Parameters	67
Sort Sequence	68
ACFRPTNV--Sample Output	69
Field Descriptions	70
 ACFRPTPP - THE PRE-PROCESSOR	 71
Files	72
Parameters	73
ACFRPTPP - Sample Output	77
Field Descriptions	79

ACFRPTPW - INVALID PASSWORD/AUTHORITY LOG	80
Files	80
Parameters	80
Sort Sequence	81
ACFRPTPW - Sample Output	82
Field Descriptions	83
ACFRPTRL - RULE-ID MODIFICATION LOG	85
Files	85
Parameters	85
ACFRPTRL - Sample Output	86
Field Descriptions	86.1
ACFRPTRV - GENERALIZED RESOURCE EVENT LOG	87
Files	87
Parameters	87
Sort Sequence	88
ACFRPTRV - Sample Output (Terminal Format)	89
ACFRPTRV - Sample Output (Printer Format)	90
Field Descriptions	91
ACFRPTRX - THE LOGONID ACCESS REPORT	95
Files	96
Parameters	98
Format and Fields of the Logonid Access Report	99
Header Section	99
User Information Section	100
Logonid Access Report Section	100
ACFRPTRX Reason Codes and Message Lines	102
ACFRPTRX - Sample Output (Dataset Logonid Access)	104
ACFRPTRX - Sample Output (Generalized Resource Logonid Access)	105
Other Processing Options	106
ACFRPTSL - SELECTED LOGONID LIST	107
Files	107
Parameters	108
ACFRPTSL - Sample Output (Short Format)	112
ACFRPTSL - Sample Output (Full Format)	113
Field Descriptions	114
ACFRPTXR - THE CROSS-REFERENCE REPORT	115
ACFRPTXR Exit Considerations	115
ACFRPTXR Processing	115
ACFRPTXR Access Reason Codes	116
Input/Output Files	117
Parameters	120
Field Descriptions	122

ACFRPTXR - Sample JCL (Dataset Access Cross-Reference) . . .	124
ACFRPTXR - Other Processing Options (Dataset Access Cross-Reference)	125
ACFRPTXR - Sample Output (Dataset Access Cross Reference) . .	125
ACFRPTXR - Sample Rule Record Summary (Dataset Access Cross Reference)	126
ACFRPTXR - Sample JCL (Generalized Resource Cross-Reference)	126
ACFRPTXR - Other Processing Options (Generalized Resource Cross-Reference)	127
ACFRPTXR - Sample Output (Generalized Resource Cross-Reference)	127
ACFRPTXR - Rule Record Summary (Generalized Resource Cross-Reference)	128
 INSTALLATION EXITS FOR REPORT GENERATORS	 129
 DATABASE RECOVERY	 131
Pre-Planning for Recovery	131
ACF2 Recovery Features	131
The Automatic Backup Facility	132
@DDSN ACFFDR Macros And GSO BACKUP Record	132
SMF Journalling of Database Modifications	133
Contingency Procedure for Using the Alternate Clusters . . .	133
Restoring the Primary Clusters	134
ACFRECVR - The Recovery Utility	134
Files	135
Parameters	135
ACFRECVR - Sample Output	138
Field Descriptions	139
Sample Recovery Jobstream	143
The ACFRECVR Procedure	145
The ACFBKUP Procedure	147
 UTILITIES FOR ACF2 ADMINISTRATION	 149
ACFBATCH - BATCH COMMAND PROCESSOR	150
Files	150
Parameters	151
Examples	151
ACFCOMP - TSO Ruleset Compile Command	152
Syntax	152
Methods of Compilation	152
Parameters	153
ACFBCOMP - Batch Rule Compiler	155
Files	155
ACFBCOMP (Batch Compiler) Example	155
ACFBDCMP - Batch Rule Decompiler	157
Files	157
Examples	157
ACFNRULE - New Rule Utility	159
Files	159
Parameters	159

Examples	160
ACFBSYNC - The Batch Sync Utility	161
Files	161
Parameters	161
Example	162
 OTHER ACF2 UTILITIES	 163
 ACFSUB - TSO Production Job Submitter	 164
Files	164
Parameters	164
Syntax Examples	165
JOBCOPY Utility - Batch Production Job Submission	166
Files	166
Parameters	166
Examples	166
Prerequisites for the Use of JOBCOPY	167
ACFERASE - The Data Disposal Utility	168
Files	168
Parameters	169
Examples	170
ACFDEL - TSO Data Disposal Command	171
Syntax	171
Parameters	171
Example	172
 INDEX	 173

INTRODUCTION

This manual contains the following chapters:

- * General Information describes the common parameters and files for executing the ACF2 report generators and utilities.
- * Report Generators describes the files and parameters specific to each report generator. It also gives samples and field descriptions of the reports.
- * Database Recovery describes ACF2 facilities for database recovery.
- * Batch Programs describes utilities for batch processing of ACF2 commands, addition and deletion of single rules in ACF2 rule sets, compilation and decompilation of rule sets, dataset erasure, and other functions.
- * Other Utilities describes the TSO production job submitter utility.

NOTE:

All ACF2 TSO commands and functions can be executed in the MVS batch environment through the use of the background Terminal Monitor Program (TMP). This manual provides only a brief description of the batch TMP. For more information, see the IBM "OS/VS2 MVS TSO Command Package: User's Guide and Reference (SC28-0748)."

GENERAL INFORMATION ON ACF2 UTILITIES

This section provides information common to ACF2 report generators and batch programs. It discusses:

1. Report generator execution through the use of JCL supplied with acf2/MVS
2. Report generator execution through the use of ISPF screens
3. Batch program execution through the use of JCL (includes database recovery and other utilities)
4. Batch program execution through the use of ISPF screens
5. ACF2 utilities execution through the use of the TSO CALL command

EXECUTION OF REPORT GENERATORS THROUGH THE USE OF JCL SUPPLIED WITH ACF2

A prototype JCL procedure is provided on the distribution tape with the acf2/MVS product. This JCL is stored in the REPORTS member of the ACF2.ACFJOBS dataset at your installation. You will need to modify this JCL or create your own JCL for regular use. This section, along with subsequent sections on each individual report generator, provide information for modifying or creating the REPORTS JCL and executing the report generators.

General Functions of the REPORTS JCL

The sample REPORTS JCL, or a user-written substitute for that jobstream, is intended to perform the following functions:

1. Preprocessing (separation) of each different type of SMF record into its own intermediate file. A utility called ACFRPTPP performs this operation. Afterwards, each report generator can access the appropriate intermediate file created by ACFRPTPP.
2. Sorting of each intermediate file of SMF records. Your installation's SORT utility program performs this operation. Each intermediate file may be sorted according to its intended use.
3. Execution of the report generators. All ACF2 report generators are re-entrant and generally run in a 128K region. However, some report generators may require a slightly larger region size, depending upon the number of SMF records processed. Such

requirements are highlighted in the individual descriptions of the report generators.

Record Preprocessing and Sorting Is Optional. None of the report generators are dependent on the use of ACFRPTPP (the SMF preprocessor utility) or on sorting of the input records. All report generators may be run as standalone programs using unprocessed SMF data.

However, if sorting of the SMF data is required, it is recommended that ACFRPTPP be used to move into an intermediate file those records of the subtype to be sorted. In addition, use of the preprocessor can improve the efficiency of report generation. Once the ACFRPTPP utility is run, the report generators will need to read only the records they will process rather than all the records in the SMF file.

Report Generators Not Requiring Preprocessing. The following report generators, discussed later in this manual, are not dependent upon SMF data and are not affected by the use of the ACFRPTPP utility:

1. ACFRPTIX
2. ACFRPTRX
3. ACFRPTSL (unless the ACFRPTSL report parameter SMF is specified)
4. ACFRPTXR

| This manual provides prototype jobstreams for execution of these report generators.

Input and Output Files for Report Generators

This section gives the DD name and description of the input/output files common to most ACF2 report generators.

RECxxxxx

| These are the input files containing SMF records that your installation collects and maintains. The ACF2 report generators can use these files for input. These files must have DD names beginning with the characters "REC" For example:

```
//RECMAN1 DD DSN=SYS1.MAN1,DISP=SHR  
//RECMAN2 DD DSN=SYS1.MAN2,DISP=SHR
```

In the supplied JCL, these files provide input for the ACFRPTPP utility program, which in turn creates intermediate files for input to each of the report generators. The RECxxxxx files can be either VSAM or non-VSAM. However, they cannot be concatenated.

SYSIN

This file provides an additional method of specifying parameters to ACF2 report generators. Parameters can also be specified by using the PARM field of a JCL EXEC statement. The SYSIN file allows an installation to specify a set of parameters that exceeds 100 characters. Use of the JCL PARM parameter is discussed in this chapter in the section on parameters.

The SYSIN file may be defined in one of the following formats:

- * F or FB. The last 8 characters of each record are assumed to be a sequence number and are ignored.

- * VB. The first 8 characters are assumed to be the sequence field and are ignored.

For example, an installation can create a SYSIN file, like the following one, by using the TSO EDIT command:

```
TITLE(DATASET LOGGING RECORD)
JOBMASK(SSD-)
SDATE(84170)
EDATE(84189)
```

Continuation of Parameter Values. All records in the SYSIN file are assumed to be an extension of the JCL EXEC statement PARM field. Any parameter value specified in a record may be continued in the next record in the file. A dash (-) as the last nonblank character of a record indicates a continuation in the next record. The contents of the next record are concatenated to the preceding record at the position of the dash (the dash itself is omitted).

SYSPRINT

This file specifies where the report output will be sent. Output can be directed to a printer or to a listing dataset. The record format is VB. Specification of the BLKSIZE parameter is optional; the default is 3665.

Report generator output is generally 80 characters wide for most reports. This width allows for convenient report browsing on an 80-character display screen. However, some report have a wider format for use with printer-directed output. To find out the maximum record length for each format, refer to the explanation of each report generator.

Required Access for Executing Report Generators

Any user executing report generators must have read access (via ACF2 access rules) to all input datasets. Such datasets may include live or backup SMF datasets and/or live or backup ACF2 databases.

To execute report generators that process data related to Logonids or UID string, the user must also have the appropriate privilege levels (i.e., SECURITY, ACCOUNT, AUDIT, LEADER, or CONSULT) along with the appropriate scope. Otherwise, the generated reports may be incomplete.

The installation should carefully consider the subset of its user community to which access to live or backup SMF datasets will be granted. ACF2 SMF records contain database record images that are used in the database recovery process and may contain sensitive data about the installation.

One method which may be employed to control SMF data access is to use ACF2 program pathing facilities. When creating access rules for SMF datasets, the installation can identify the report generators in the access rule LIB and PGM parameters. Program pathing can effectively limit data access by allowing it only through execution of the individual ACF2 programs.

Parameters

Parameters can be specified for ACF2 report generators in either or both of the following ways:

1. Through the PARM parameter of the EXEC statement in the JCL. For example:

```
//DSLOGS EXEC PGM=ACFRPTDS,REGION=128K,  
//      PARM=('TITLE(DATASET LOGGING RECORDS)',  
//      'MASK(SYS1.-)', 'SDATE(84170)', 'EDATE(84.174)')
```

2. Through the SYSIN file. The installation supplies a SYSIN DD statement and control record file as previously discussed in the section "Input and Output Files for Report Generators."

```
//DSLOGS EXEC PGM=ACFRPTDS,REGION=128K  
//SYSIN DD DSN=ADMIN.WORK.PARMS(DS),DISP=SHR
```

Also, note that if a particular parameter is specified more than once, the last specified value for the parameter is used. For example, if a user specifies:

```
PARM=('SDATE(83001)', 'EDATE(83005)', 'SDATE(83002)')
```

In this example, a value of 83002 is used for the SDATE parameter.

This section lists those parameters that operate in an identical fashion for most report generators. Other parameters are described in the sections related to the particular report generators.

LINECNT(nnnnnnnn) - DEFAULT=60

The LINECNT parameter specifies the number of output lines to be printed on a page. To prevent splitting of information, ACF2 report generators that issue multiple line reports check whether a complete report item will fit on a page. The maximum number of output lines per page is limited only by the physical constraints of the output media being used, or to 999,999,999 lines.

TITLE(string) - DEFAULT = First 35 characters of the PARM parameter

The TITLE parameter specifies a character string to be added to other title information at the top of the report. This character string can be up to 35 characters in length. If this parameter is unspecified, then the report generator will use the first 35 characters in the PARM field of the EXEC statement. If this character string is longer than 35 characters, the leftmost 35 characters are used.

SDATE(yyddd) - DEFAULT=00000

The SDATE parameter specifies the beginning Julian date for which report information will be selected. Any input SMF records generated prior to the SDATE value will be ignored.

EDATE(yyddd) - DEFAULT=99365

The EDATE parameter specifies the ending Julian date for which report information will be selected. When combined with SDATE parameter, this parameter creates a window for report content. The defaults for SDATE and EDATE cause all available records to be processed.

STIME (hhmm) - DEFAULT=0000

The STIME parameter specifies the beginning of the time interval for which SMF records will be selected. This time is based on a 24-hour clock. Any SMF records generated before this specified time of day will be ignored.

ETIME (hhmm) - DEFAULT=2359

The ETIME parameter specifies the end of time interval for which SMF records will be selected. This time is based on a 24-hour clock. Any SMF records generated after this specified time of day will be ignored.

SELECT(nnn nnn...nnn)/NOSELECT - DEFAULT=Number defined in the @SMF macro of the ACFFDR

The SELECT parameter is used to define the SMF record numbers used by ACF2. Generally, this parameter is not necessary because the report generators can extract the proper record numbers from the ACFFDR of the running system.

However, if ACF2 is not available in the running system, or if the installation has changed the SMF record numbers, the proper record numbers must be supplied.

All SMF records generated by acf2/MVS Release 4.0 and above are processed under a single record number. When processing such SMF records, specify this single record number in the SELECT parameter. (This record number is defined in the ACF2 parameter of the @SMF macro in the ACFFDR.)

When you need to process SMF data generated under prior releases of acf2/MVS, specify the SMF record number(s) required by the particular report. Under prior releases, the record number is different for each ACF2 record type. Each record number is specified by the appropriate record-type parameter in the @SMF macro of the ACFFDR that was active when the record was generated. For a more complete discussion of ACF2-generated SMF record types, refer to the chapter on the ACFRPTPP utility.

You can specify the single record number for acf2/MVS Release 4.0 and above along with those record numbers used by prior releases.

The NOSELECT parameter specifies no selection of records based on SMF record numbers or ACF2 subtypes. All records satisfying the other specified parameters (SDATE, STIME, SYSID, etc.) are processed. You should specify this parameter only when your input file(s) contain records that have been preprocessed by the ACFRPTPP utility and selected especially for the report generator you want to execute. Otherwise, use of this parameter may lead to unpredictable results, such as abends or meaningless report output.

SYSID(sysid) - DEFAULT=all systems

The SYSID parameter specifies the ACF2 system identifier that was active on the system when the SMF records in question were generated. You can specify a single system identifier or a system identifier mask. You cannot specify multiple masks or a series of identifiers.

HEX/NOHEX - DEFAULT=NOHEX

The HEX option is provided in the ACF2 report generators to cause all selected SMF records to be printed in hexadecimal dump format. This option is provided primarily for diagnostic purposes.

EXECUTING ACF2 REPORT GENERATORS THROUGH THE USE OF ISPF SCREENS

All ACF2 report generators can be executed via IBM's Interactive System Productivity Facility (ISPF). Through this facility, the installation can execute a report generator online and have the results displayed at the terminal screen.

Each user should contact system maintenance personnel for information on the availability and use of SPF. Before proceeding, the user should also be familiar with the execution requirements of the report programs as documented in this manual.

Using the ACF2/ISPF Screens for Report Selection

To select a report generator for execution, a user performs the following steps:

1. From TSO READY mode, enter the SPF command to bring up the ISPF primary selection menu. (Check with your installation for the exact command syntax.)
2. From the ISPF primary selection menu, enter the appropriate selection code for displaying the ACF2 SPF Option Selection Menu.

----- ACF2 SPF OPTION SELECTION MENU -----

SELECT OPTION====>

- 1 RULES - PROCESS ACF2 ACCESS AND GENERALIZED RESOURCE RULES
 - 2 LOGONIDS - ACF2 LOGONID CREATION/MAINTENANCE FACILITY
 - 3 SYSTEM - ACF2 SHOW COMMANDS
 - 4 REPORTS - ACF2 REPORT PROGRAM PROCESSOR
 - 5 UTILITIES - PROCESS ACF2 UTILITIES
 - 6 GSO - GLOBAL SYSTEM OPTIONS SERVICES
-

3. Enter the digit "4" to display the ACF2 Report Program Processor Menu:

----- ACF2 REPORT PROGRAM PROCESSOR MENU -----

SELECT OPTION====>

- 0 ACFRPTPP - ACF2 SMF RECORD PRE-PROCESSOR
- 1 ACFRPTCR - TSO COMMAND STATISTICS LOG
- 2 ACFRPTDS - DATASET/PROGRAM EVENT LOG
- 3 ACFRPTL - INFORMATION STORAGE UPDATE LOG
- 4 ACFRPTIX - DATASET INDEX REPORT
- 5 ACFRPTJL - RESTRICTED LOGONID JOB LOG
- 6 ACFRPTLL - LOGONID MODIFICATION LOG
- 7 ACFRPTNV - ACF2 ENVIRONMENT
- 8 ACFRPTPW - INVALID PASSWORD/AUTHORITY LOG
- 9 ACFRPTRL - RULE-ID MODIFICATION LOG
- A ACFRPTRV - GENERALIZED RESOURCE EVENT LOG
- B ACFRPTRX - THE LOGONID ACCESS REPORT
- C ACFRPTSL - SELECTED LOGONID LIST
- D ACFRPTXR - THE CROSS-REFERENCE REPORT

4. Enter the appropriate number to select the desired report. A panel will be displayed, allowing entry of parameters for the report generator. For example:

```
----- ACFRPTDS - DATASET/PROGRAM EVENT LOG -----  
  
COMMAND ===>  
  
OPTIONAL PARAMETERS FOR ACFRPTDS:  
TITLE      ===>                               35 CHARACTERS MAX  
  
SPECIFIC PARAMETERS FOR ACFRPTDS:  
DSNAME MASK ===> (ALL) NDSNAME MASK ===> (NONE )  
LOGONID MASK ===> (ALL) NLOGONID MASK ===> (NONE )  
  
REPORT TYPE: PICK ONE OR MORE BY PLACING AN "X" NEXT TO OPTION  
ALL ==> LOGGING ==> VIO ==> TRACE ==> PGMNAME ==>  
TAPE ==> INSTALL ==> UNKNOWN ==>  
  
OUTPUT FORMAT: DEFAULT = 'T' FOR TERMINAL FORMAT  
FORMAT      ===> T ENTER: 'P' FOR PRINTER, 'S' FOR SUMMARY  
OUTPUT LIST NAME:  
LIST ID     ===>  
SPECIFY INPUT DATASET(S) FOR ACFRPTDS  
//RECMAN1   ===>  
//RECMAN2   ===>  
//RECMAN3   ===>  
//RECMAN4   ===>  
LINECNT ===> SDATE ===> EDATE ===>
```

5. Refer to the general and specific report generator descriptions in this manual for assistance in entering the parameters.

An efficient method of running a series of reports is to first run the ACFRPTPP report utility, and then run other report generators from the data collected and separated by ACFRPTPP.

HELP Screens for Executing a Report Generator

For further instructions on entering the parameters and specifying files, the user may press the PF1 key to view a tutorial.

EXECUTION OF OTHER ACF2 UTILITIES THROUGH THE USE OF JCL

This manual illustrates sample JCL jobstreams for executing each of the other ACF2 utility programs described. The user can submit these jobsteps as individual jobs or combine them to form larger, more comprehensive jobs.

Files and Parameters for Other ACF2 Utilities

Refer to the individual description of the particular ACF2 utility for an explanation of all input and output files.

EXECUTING OTHER ACF2 UTILITIES THROUGH THE USE OF ISPF SCREENS

Three ACF2 utilities can be executed via IBM's Interactive System Productivity Facility (ISPF). These utilities are ACFDEL, ACFBSYNC, and ACFCLEAN. Through ISPF, the user can execute these batch programs online and have the results displayed at the terminal screen or, if necessary, to the printer.

Each user should contact the appropriate system maintenance personnel at the site for information on the availability and use of ISPF. Before proceeding, the user should also be familiar with the execution requirements of the batch program to be run as documented in this manual.

Using the ACF2/ISPF Screens for Utility Selection

To select one of the other ACF2 utilities for execution, a user performs the following steps:

1. From TSO READY mode, enter the SPF command to bring up the ISPF primary selection menu. (Check with your installation for the exact command syntax.)

2. From the ISPF primary selection menu, enter the appropriate selection code for displaying the ACF2 SPF Option Selection Menu.

----- ACF2 SPF OPTION SELECTION MENU -----

SELECT OPTION=====>

- 1 RULES - PROCESS ACF2 ACCESS AND GENERALIZED RESOURCE RULES
- 2 LOGONIDS - ACF2 LOGONID CREATION/MAINTENANCE FACILITY
- 3 SYSTEM - ACF2 SHOW COMMANDS
- 4 REPORTS - ACF2 REPORT PROGRAM PROCESSOR
- 5 UTILITIES - PROCESS ACF2 UTILITIES
- 6 GSO - GLOBAL SYSTEM OPTIONS SERVICES

-
3. Enter the digit 5 to display the ACF2 Utility Processor Menu:

----- ACF2 UTILITY PROCESSOR MENU -----

SELECT OPTION=====>

- 1 ACFDEL - THE DATA DISPOSAL UTILITY
 - 2 ACFSYNCH - THE LOGONID BROADCAST PROCESSOR
 - 3 ACFCLEAN - ACF2 REPORT PROGRAM PROCESSOR DATASET CLEANUP
-

4. Enter the appropriate number to select the desired utility. A panel will be displayed, allowing the user to enter parameters for the batch program:

----- ACFDEL - THE DATA DISPOSAL UTILITY -----

COMMAND ===>

OPTIONAL PARAMETERS FOR ACFDEL: PLACE AN "X" NEXT TO OPTION ...

NOUNCATALOG	===>	DEFAULTS = UNCATALOG
NOSCRATCH	===>	SCRATCH
NOERASE	===>	ERASE

ISPF LIBRARY:

PROJECT	===>	PROJECT
LIBRARY	===>	LIBRARY
TYPE	===>	TYPE

OTHER PARTITIONED OR SEQUENTIAL DATASET:

DATASET NAME	===>	
VOLUME SERIAL	===>	(If not cataloged)

5. Refer to the sections of this manual describing the ACFERASE for the ACFDEL functions and ACFBSYNC utilities for descriptions of parameters for ACF2 Utility Process Menu selections #1 and #2, respectively. Selection #3 (ACFCLEAN) is a simple CLIST, which does not require specification of files or parameters.

HELP Screens for Executing a Utility Program

For further instructions on using the ACFDEL, ACFBSYNC, and ACFCLEAN utilities, press the PF1 key to view a tutorial.

EXECUTION OF ACF2 UTILITIES VIA THE TSO CALL COMMAND

An installation can also execute the ACF2 utilities by first creating a CLIST dataset containing the CALL and other necessary commands for executing the utility. In doing so, the installation should be aware that:

1. The parameters for the utility can be specified as CALL parameter operands.

2. In addition, parameters can be specified in a SYSIN file. A dash in single quotes '-' must be specified as the last nonblank character in the CALL program parameter.
3. If a dash has been specified as the last nonblank character of the CALL program parameter and no SYSIN file has been allocated, then the user will be prompted for input via the terminal. Each input prompt consists of the last two characters of the report generator name followed by a question mark.

INPUT RECORDS FOR ACF2 REPORT GENERATORS

The input records for the various ACF2 report generators are classified by record numbers. For records written under acf2/MVS Version 3.1.5 and earlier, a unique record number identifies the ACF2 record type. For records written, under acf2/MVS Release 4.0 and above, one combined record number includes all ACF2 record types; a subtype indicator identifies the ACF2 record type.

To run reports that include SMF records written by systems running under acf2/MVS Version 3.1.5 or earlier, the installation default SMF record numbers can be specified in the ACF2 Field Definition Record (ACFFDR). If the ACFFDR module is not available through LINKLIST, STEPLIB, or JOBLIB, then the user should specify the SMF record number(s) by means of the report generator's SELECT parameter. For example, the user might specify the parameter SELECT(225) when running the ACFRPTCR report generator. The default numbers may be different at your installation.

The table below shows the name of each report generator (except ACFRPTNV) and report title. Listed for each report generator is the corresponding operand name used for specifying the report generator's default record number(s) (acf2/MVS Version 3.1.5 or earlier) in the @SMF macro of the ACF2-supplied ACFFDR. This ACFFDR must be one supplied with acf2/MVS Version 3.1.5 or earlier. Note that the installation may have changed these defaults. The ACF SHOW SYSTEM subcommand will list the installation defaults.

<u>REPORT NAME</u>	<u>REPORT TITLE</u>	<u>FDR @SMF OPERAND</u>	<u>RECORD NUMBER DEFAULT</u>
ACFRPTCR	TSO Command Statistics	COMMAND	225
ACFRPTDS	Data Set Access Journal	DSN	221
ACFRPTTEL	Information Storage Update	INFO	226
ACFRPTIX	Access Index Report	LID/RULE	222/223
ACFRPTJL	Restricted Logonid Job Log	JTRACE	224
ACFRPTLL	Logonid Modification Log	LID	222
ACFRPTPW	Invalid Password Authority	PSWD	220
ACFRPTRL	Rule-ID Modification Log	RULE	223
ACFRPTRV	Generalized Resource Log	RSRC	227
ACFRPTRX	Logonid Access Report	N/A	N/A
ACFRPTSL	Logonid Superlist Report	LID	222
ACFRPTXR	Cross Reference Report	N/A	N/A

Reports that are run using SMF data written under acf2/MVS Version 3.1.5 or earlier may be executed by using these defaults.

However, if a user specifies the above record numbers through the SELECT parameter of a report generator, and the user wants to also select records with the new combined SMF record number (acf2/MVS Release 4.0 and later), then the user must specify the combined SMF record number (default of 230) in the SELECT parameter. (For example, SELECT(222,230) would be specified for the ACFRPTLL report.)

For the ACFRPTIX report generator, the user should specify the appropriate ACFRPTIX report parameter with the SMF record numbers as defined by the installation. For example:

```
SELLID(222,230)
SELRULE(223,230)
```

REPORT GENERATOR SORT SEQUENCE

The prototype JCL in ACF2.ACFJOBS also sorts the records within several ACF2 SMF record subtypes to provide more meaningful reports. Again, the JCL may require local modification to customize the reports as desired. The sort sequences that are provided with the prototype JCL are explained, as applicable, in each of the following report generator explanations.

Some reports, such as the data base update journals, do not require sorting steps. The input for these utilities is usually generated in chronological sequence. However, a sort step may be used to ensure that records are in date and time stamp sequence, particularly if multiple SMF files are being processed.

ACF2 REPORTS WHEN ACF2 IS INACTIVE

All ACF2 report generators except ACFRPTRX are executable without ACF2 active on the system. If the ACFFDR module (used when ACF2 is active) is available (via the LOAD macro), then the default SMF record number(s) will be retrieved from the ACFFDR and need not be specified by the user.

If the ACFFDR is not available, the SMF record number(s) must be specified by the user. An authorized user can run the report generators ACFRPTCR, ACFRPTDS, ACFRPTTEL, ACFRPTJL, ACFRPTLL, ACFRPTNV, ACFRPTPW, ACFRPTRL, and ACFRPTRV by specifying the SELECT report parameter with the appropriate SMF record numbers.

Executing ACFRPTSL When ACF2 Is Inactive

If ACF2 is not active, ACFRPTSL can use the ACF2 sequential LID backup file or an SMF file as input. If an SMF file is used, specify the appropriate LID SMF record number in the SELECT parameter. See also the section on ACFRPTSL in this manual.

Executing ACFRPTXR When ACF2 Is Inactive

ACFRPTXR can be executed when ACF2 is not active. Use the ACFRECVR utility and create a set of VSAM clusters from the ACF2 sequential backup files. After creating the required VSAM clusters, point ACFRPTXR to these files and execute the program. See also the section entitled "ACFRPTXR - The Cross-Reference Report" in this manual.

ACFRPTCR - TSO COMMAND STATISTICS LOG

The ACFRPTCR utility formats the audit information collected by the ACF2/TSO interface routines. This information is collected only if the CMDREC field is specified in the GSO OPTS record. (See the chapter on GSO records in the acf2/MVS Administrator's Guide.) Alternatively, this information can be collected for a individual user by specifying the TSO-TRC attribute in the User's Logonid record.

There are two formats for this report. The format used is dependent upon whether the input SMF records were generated by a system running acf2/MVS Release 4.0 or above, or a system running a prior release.

Prior to acf2/MVS Release 4.0, the command trace records contained resource usage information similar to that provided by IBM's PCF program product. This report will continue to display the information found in these historical records.

For acf2/MVS Release 4.0, the command limiting facility was restructured and generalized. The TSO interface for command limiting was redesigned, and the resource usage information is no longer available in ACF2 command trace SMF records. However, the PCF program product may be used concurrently with ACF2 to provide this information in the standard IBM command resource usage SMF records.

The report formats are explained after their sample output in this section.

FILES

The ACFRPTCR utility uses the SYSIN file and the standard report generator input and output files (RECxxxxx and SYSPRINT) as discussed in the General Information section of this manual.

PARAMETERS

ACFRPTCR accepts the following parameters discussed in the general information section of this manual:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SELECT, SYSID, and HEX.

The following parameters are specific to ACFRPTCR:

JOBMASK(job-mask,job-mask,...) - DEFAULT=***** (all)

The JOBMASK parameter specifies that records appearing on this report be limited to those pertaining to the job(s) indicated by the jobname(s) or jobname mask(s). Multiple jobname masks must be separated by commas or spaces.

MASK(logonid-mask) - DEFAULT=***** (all)

The MASK parameter specifies that records appearing on the report be limited to those pertaining to the Logonid(s) indicated by the Logonid(s) or Logonid mask(s). Note that, for TSO, the Logonid and jobname are usually the same.

UID(uid-mask) - DEFAULT=-

The UID parameter specifies that the records appearing on the report should be limited to those pertaining to the user(s) indicated by the UID-mask. Any value specified for this parameter will be expanded with asterisks to the full 24-character length of the UID string.

BUFFER/NOBUFFER - DEFAULT=NOBUFFER

When the BUFFER option is requested, the ACFRPTCR utility will print the usual output and also the command exactly as entered by the user. The default of NOBUFFER results in only a single line for each command. (See sample output below.)

UPPER/NOUPPER - DEFAULT=NOUPPER

The UPPER option is used when the ACFRPTCR output is being directed to a device that does not support lower case letters or a device that does not have a character set fold option (lower case letters are not automatically translated to uppercase). This option only applies to the command buffer displayed when the BUFFER parameter is specified. All other report output is always displayed in uppercase.

SORT SEQUENCE

The recommended sort sequence for the ACFRPTCR report is by jobname (major), date, and time. The installation can perform this sorting through the its own routine or by modifying and using the prototype JCL provided with the acf2/MVS product.

ACFRPTCR - SAMPLE OUTPUT (ACF2/MVS RELEASES 4.0 AND ABOVE)

The following two sample outputs show the ACFRPTCR report format produced from combined SMF records, written by systems running acf2/MVS Release 4.0 and above. These two reports show the difference between the BUFFER and NOBUFFER report parameters. This first report was produced by using the default report parameter NOBUFFER:

ACF2 UTILITY LIBRARY - ACFRPTCR - TSO COMMAND STATISTICS - PAGE 1
DATE 12/21/84 (84.356) TIME 11.45 ADMJDL NOBUFFER

84.355 12/20 12.54 JOB=ADMJDL UID=ABC73ADMJDL SYSTEM ID=CPU1

SEQUENCE NUMBER	COMMAND NAME	TIME OF DAY	CALLING PROGRAM
00	TSOEXEC	12.52	ISRPTC
01	TSOEXEC	12.52	ISRPTC
02	ACFRPT	12.52	ISRPTC
03 *	FREE	12.52	ISRPTC
04 *	ATTR	12.52	ISRPTC
05 *	ALLOC	12.52	ISRPTC

NOTE: 'C' INDICATES CLIST OR INVALID COMMAND, * INDICATES COMMAND
WITHIN CLIST

On the ACFRPTCR report, the command trace entries are grouped into sections. Those entries in one section are those entries that were written to SMF as one record.

The above example shows one section. The section heading line (third printed line) shows the Julian and Gregorian calendar dates on which the record was written. This line also shows the time of day at which the record was written, jobname associated with the commands being traced, UID of the submitter, and system ID.

This second example report was produced by specifying the parameter
BUFFER:

ACF2 UTILITY LIBRARY - ACFRPTCR - TSO COMMAND STATISTICS - PAGE 1
DATE 12/21/84 (84.356) TIME 11.46 ADMJDL BUFFER

84.355 12/20 12.54 JOB=ADMJDL UID=ABC73ADMJDL SYSTEM ID=CPU1

SEQUENCE NUMBER	COMMAND NAME	TIME OF DAY	CALLING PROGRAM
00	TSOEXEC TSOEXEC acucb *	12.52	ISRPTC
01	TSOEXEC TSOEXEC call 'sys1.linklib(iefbr14)'	12.52	ISRPTC
02	ACFRPT ACFRPT cr	12.52	ISRPTC
03 *	FREE FREE FILE(RECMAN1 RECMAN2 RECMAN3) A TTR(BLK)	12.52	ISRPTC
04 *	ATTR ATTR BLK RECFM(F B) LRECL(80) BLKSIZE(80)	12.52	ISRPTC
05 *	ALLOC ALLOC FILE(RECMAN1) DA('SYS1.MAN1') SHR	12.52	ISRPTC

NOTE: 'C' INDICATES CLIST OR INVALID COMMAND, * INDICATES COMMAND
WITHIN CLIST

ACFRPTCR - SAMPLE OUTPUT (ACF2/MVS VERSION 3.1.5 AND EARLIER)

The following two sample outputs show the ACFRPTCR report format produced from records written by systems running acf2/MVS Version 3.1.5 and earlier. These two reports show the difference between the BUFFER and NOBUFFER options of the ACFRPTCR utility. This first report was produced by using the default report parameter NOBUFFER:

ACF2 UTILITY LIBRARY - ACFRPTCR - TSO COMMAND STATISTICS - PAGE 1
DATE 07/31 (78.212) TIME 16.37 LID001

78.212 07/31 09.15 LID001 TMP NEST LEVEL= 00 LOCLID001

SEQUENCE NUMBER	COMMAND NAME	CPU-TIME (SEC)	REAL-TIME (SEC)	SERVICE UNITS	TGETS	TPUTS	COMP-CODE /ABEND
01	C STARTUP	.11	2.72	357	00	00	
02	* FILESTAT	.09	2.38	162	00	00	
03	* EXEC	.14	.96	251	00	00	
04	* FREE	.06	.11	92	00	00	
05	* ALLOCATE	.22	3.12	357	00	00	
06	* PFK	.05	5.17	84	00	01	
07	* END	.03	.34	65	00	00	
08	RULES	.11	5.08	219	00	01	
09	Q	.28	33.30	506	07	08	
10	ACF	.19	119.50	347	03	12	
11	RULE	.09	1.83	204	00	00	
12	Q	.44	309.79	833	07	10	
13	ACF	.72	166.57	1,140	09	35	
14	RECORDS	.10	6.74	198	00	00	

NOTE: 'C' INDICATES CLIST OR INVALID COMMAND, * INDICATES COMMAND
WITHIN CLIST

The section heading line indicates the date and time the SMF record was issued. The TMP NEST LEVEL is incremented by one each time the TSO TMP is called as a subroutine. The last field in the title line is the user's UID string.

This second example report was produced by specifying the parameter BUFFER:

ACF2 UTILITY LIBRARY - ACFRPTCR - TSO COMMAND STATISTICS - PAGE 1
DATE 07/31 (78.212) TIME 16.45 LID001 BUFFER MASK(LID001**)

78.212 07/31 10.10 LID001 TMP NEST LEVEL= 00 LOCLID001

SEQUENCE NUMBER	COMMAND NAME	CPU-TIME (SEC)	REAL-TIME (SEC)	SERVICE UNITS	TGETS	TPUTS	COMP-CODE /ABEND
01	LISTC listc	.28	15.80	466	00	14	
02	ALLOC alloc fi(sysprint) da(*)	.04	1.66	81	00	00	
03	ALLOC alloc fi(records) da('sys1.manx' 'sys1.many') shr	.13	1.09	217	00	00	
04	TERM term lines(48)	.07	1.22	80	00	00	
05	LOGOFF logoff	.02	.15	116	00	00	

NOTE: 'C' INDICATES CLIST OR INVALID COMMAND, * INDICATES COMMAND WITHIN CLIST

The section heading line above the column titles indicates the date and time when the SMF record was issued. The TMP NEST LEVEL is incremented by one each time the TSO TMP is called as a subroutine. The last field in the title line is the user's UID string.

The various fields of the report are described below.

FIELD DESCRIPTIONS (ACF2/MVS RELEASE 4.0 AND ABOVE)

The following fields appear on the ACFRPTCR report when it is produced from SMF records generated on systems running acf2/MVS Releases 4.0 and above:

SEQUENCE NUMBER

This field is the sequence number of the command with respect to the session during which it was executed. Because of MVS SMF processing, the output SMF records do not necessarily reflect the order in which the commands were issued. The command sequence field provides a means of determining this order. This sequence number is incremented by one for each command issued.

COMMAND NAME

This field indicates the name of the command or CLIST that was issued.

Under acf2/MVS Version 3.1.5 and earlier, the acf2 system may be generated to recognize the shortest non-ambiguous abbreviation of commands. This feature is allowed through ACF2 command limiting. Such abbreviated command names are expanded. Any command name on this report is listed in entirety. Under acf2/MVS Releases 4.0 and later, this feature is not supported.

An asterisk * preceding a command name indicates that the command was issued from a CLIST. A "C" preceding a name command name indicates that the command was an implicit CLIST command.

Implicit CLISTs are recognized by a special notation (a percent sign % prefix under TSO) or by their absence in the user's command limiting list. Implicit CLISTs which are entered without the special notation and are not disallowed by the command limiting list (perhaps because the user does not have a command limiting list) are assumed by ACF2 to be real commands, even if they eventually result in the execution of a CLIST (e.g., because the IBM PCF program allowed the command as a CLIST or because the command was not found in the command library search).

TIME OF DAY

Indicates, in minutes and seconds, the time of day at which the command was issued. This time is based on a 24-hour clock.

CALLING PROGRAM

Indicates the name of the module that called the command.

FIELD DESCRIPTIONS (ACF2/MVS VERSIONS 3.1.5 AND EARLIER)

The following fields appear on the ACFRPTCR report when it is produced from SMF records generated on systems running acf2/MVS Versions 3.1.5 and earlier:

SEQUENCE NUMBER

This field is the sequence number of the command with respect to the session during which it was executed. Because of MVS SMF processing, the output SMF records do not necessarily reflect the order in which the commands were issued. The command sequence field provides a means of determining this order. This sequence number is incremented by one for each command issued.

COMMAND NAME

This field indicates the name of the command or CLIST that was issued.

Under acf2/MVS Version 3.1.5 and earlier, the ACF2 system may be generated to recognize the shortest non-ambiguous abbreviation of commands. This feature is allowed through ACF2 command limiting. Such abbreviated command names are expanded. Any command name on this report is listed in entirety. Under acf2/MVS Releases 4.0 and later, this feature is not supported.

An asterisk * preceding a command name indicates that the command was issued from a CLIST. A "C" preceding a name command name indicates that the command was an implicit CLIST command.

Implicit CLISTs are recognized by a special notation (a percent sign % prefix under TSO) or by their absence in the user's command limiting list. Implicit CLISTs which are entered without the special notation and are not disallowed by the command limiting list (perhaps because the user does not have a command limiting list) are assumed by ACF2 to be real commands, even if they eventually result in the execution of a CLIST (e.g., because the IBM PCF program allowed the command as a CLIST or because the command was not found in the command library search).

CPU-TIME

This field shows the CPU time used by a command, including both the TCB and SRB times.

REAL-TIME

This field shows the elapsed "wall clock" time during execution of a command. This real time is a function of several factors that include system utilization, I/O, and time spent waiting for a user's response to prompts at the terminal.

SERVICE UNITS

This field indicates the total amount of resources utilized by a command. This amount is calculated by the System Resource Manager (SRM). The larger this number, the more resources utilized by the command.

TGETS

This field shows the total number of requests for terminal input that a command has issued. In general, commands that accept subcommands will issue many more TGETs during execution than commands that do not accept subcommands.

TPUTS

This field indicates the number of lines that a command has outputted to the terminal.

COMP-CODE/ABEND

This field indicates the success or failure of a command's execution. If the command abends during processing, the abend code will be listed along with a "S" indicating system abends or "U" indicating user abends.

ACFRPTDS - DATASET/PROGRAM EVENT LOG

ACF2 journals an event in the SMF records when (a) a request for dataset or program access is invalid or (b) that event was specifically required to be journalled (for example, by an access rule, an installation exit, TRACE set to ON in a Logonid record, NEXTKEY error, etc.) These journal records can be placed in one of four categories: dataset loggings, dataset access violations, dataset access trace requests, and program use loggings and violations.

It should be noted that many dataset and program accesses occur that are not logged. This situation arises for accesses to the user's owned data (as defined to ACF2), or when the access rules specify that an access is to be allowed without being journalled. An ACF2 security officer may request that all accesses for a particular user be logged by setting the TRACE attribute in the user's Logonid record. When this occurs, a trace journal record is written describing the access. Also written are any violation/logging records that are normally written for the access.

The information in the SMF records describes the user and job information, the access environment (e.g., program pathing functions), and the type of access requested. The ACFRPTDS utility formats the information for all of these different accesses. The information can be requested in hex format, edited for printer output format (133 characters per line), or for terminal output format (80 characters per line). In certain circumstances, however, (such as when the dataset name or the access parameter list is invalid) a particular record could also appear on the report in hex format regardless of whether the HEX option is selected.

FILES

ACFRPTDS uses the standard SYSPRINT, SYSIN, and RECxxxx files as described in the General Information section of this manual.

PARAMETERS

ACFRPTDS accepts the following parameters described in the General Information section of this manual:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SELECT, SYSID, and HEX.

The following parameters are specific to ACFRPTDS:

MASK(dsn-mask) - DEFAULT=- (all)

The MASK parameter allows the user to request information for a particular dataset or group of datasets. This function is useful when investigating the accesses to a particular user's datasets. For example, to format a report of only the SYS1 dataset loggings, specify MASK(SYS1.-).

NMASK(dsn-mask) - DEFAULT=(none)

The NMASK parameter specifies a dataset-name mask. This mask allows a user to exclude, from the report, information pertaining to a certain dataset or group of datasets. For example, a parameter of NMASK(SYS1.-) excludes from the report any information pertaining to the SYS1 datasets. Any dataset names specified by this parameter will override any dataset set names specified by the MASK parameter.

LIDMASK(logonid-mask) - DEFAULT=*** (all)**

This parameter allows the user to investigate access requests for a particular Logonid or group of Logonids. The default specification requests information for all Logonids for which an access has been journalled.

NLIDMASK(logonid-mask) - DEFAULT=(none)

The NLIDMASK parameter specifies a Logonid mask. This mask allows a user to exclude, from the report, information pertaining to a certain Logonid or group of Logonids. For example, a parameter of NLIDMASK(PAY-) excludes from the report any information pertaining to Logonids beginning with the letters PAY. Any Logonids specified by this parameter will override those specified by the LIDMASK parameter.

UID(uid-mask) - DEFAULT=-

The UID parameter specifies that the records appearing on the report be limited to those pertaining to the user(s) indicated by the uid-mask.

JOBMASK(job-mask,job-mask,...) - DEFAULT=*** (all)**

The JOBMASK parameter specifies that records appearing on the report be limited to those pertaining to the job(s) indicated by the jobname mask(s). Commas or spaces must separate multiple jobname masks.

SIZE(nnnnn) - DEFAULT=2500

The SIZE parameter defines the number of elements to be allowed for in the prefix/Logonid cross-reference table. A cross-reference entry is built for each prefix to Logonid combination. Each element in this table is 20 bytes in length so the table will take 20 times SIZE bytes of memory. The default table size will occupy 50,000 bytes (49K) of memory.

SHORT - DEFAULT=(printing of all detailed information)

The SHORT option requests that only the cross-reference table be printed for this run of ACFRPTDS. The cross-reference table provides a listing of dataset prefixes, and the Logonids that accessed datasets with that prefix, showing the access counts.

LOGGING / VIO / TRACE / PGMNAME / TAPE / INSTALL / UNKNOWN / ALL -
DEFAULT=ALL

These keywords request which of various types of records be formatted for a run of ACFRPTDS. Any combination of these keywords may be specified; if none of these keywords is specified then the default of ALL will take effect. These parameters operate in an inclusive OR fashion. For example, a specification of VIO and PGMNAME would result in a report detailing every access to a protected program and every dataset access that resulted in a violation of ACF2 access controls.

The following table describes the keywords and the type of record to which the keyword refers:

LOGGING Requests all records produced for accesses that were allowed but for which the access rule requested a journal record. LOGGING records are also issued in the situation when the access is allowed through the user's SECURITY, NON-CNCL, or READALL privilege. These privileges can override the recommendation of an access rule.

VIO Requests records produced because of an attempted violation of access controls.

TRACE Specifies that records produced for a user because the TRACE attribute was specified in his/her Logonid record be printed. In addition, NKEYLOOP and KEYEXCES trace records are also processed, if present. Trace records are written regardless of whether the access is denied or logged.

PGMNAME Specifies that the report be limited to showing all logging or violation records written for attempts to access datasets through protected or logged programs. It shows all trace records written for access attempts made through any program.

|
| The installation can specify protected and logged
| programs respectively by means of the GSO PPGM and
| LOGPGM records on the Infostorage database. See
| the chapter on GSO records in the acf2/MVS
| Administrator's Guide.

- TAPE** Specifies that the report be limited to records written for tape access requests validated on the volume level (as opposed tape access requests validated at the dataset name level). Validation on the volume level occurs when the volser has been specified on the secured volume list (SECVOLS record in the Inforstorage database) or the DSNGEN installation exit has been taken.
- INSTALL** Installation records are issued whenever any of the ACF2 dataset validation installation exits (VIOEXIT, DSNGEN, and VLDEXIT as specified in the ACF2 Field Definition Record module) requests that the access be journalled to SMF.
- UNKNOWN** Unknown type records are issued whenever the ACF2 dataset access validation SVC detects an error condition, such as an invalid parameter list. These records indicate an access attempt for which no proper determination could be made. In this case the access is aborted and the UNKNOWN type record (INVPARMS) is produced. The report output will contain whatever information could be determined, but may contain invalid data and be printed in hexadecimal notation.
- ALL** Requests that information for all journalled accesses be formatted. However, if the MASK parameter is specified, the report will not contain program records.

PRINTER/SUMMARY/TERMINAL - DEFAULT=TERMINAL

ACFRPTDS supports three output formats. Only one of these keywords may be specified per report. If none of these keywords is specified, then the default of TERMINAL is used. The following is a brief description of each keyword.

PRINTER This report format provides a three-line detail section for each record (133 characters per line.) Note that if TRACE records are processed, this detail section contains from 3 to 5 lines.

SUMMARY This report format provides a one-line detail section for each record (133 characters per line). Each detail section contains minimal information about the dataset being accessed and the user involved.

TERMINAL This default format provides a five-line detail section for each record. This format is suitable for use on a limited display screen. An optional 4 line report can be produced by using the NOEXTEND parameter. This format usually fits on an 80 character screen width with an occasional wraparound due to long dataset names. Note that if TRACE records are processed, the number of lines per logging record will range from 6 to 12.

NOEXTEND/EXTEND - DEFAULT=EXTEND

This parameter further defines the default terminal report format. The EXTEND parameter provides the maximum information from each record. The NOEXTEND parameter provides consistency with the previous 4-line format of this report generator.

SORT SEQUENCE

The recommended sort sequence for the ACFRPTDS report first splits the report into four separate groups - logging records, violation records, trace records, and program journal records. Records in each of these sections are then sorted as follows:

Data Set Logging Report - Dataset Name (major), Logonid, Date, and Time.

Data Set Violation Report - Dataset Name (major), Logonid, Date, and Time.

Data Set Trace Report - Logonid (major), Date, and Time.

Program Violations and Loggings - Program Name (major), Logonid, Date, and Time.

The installation can perform this sorting through the its own routine or through the prototype JCL or ISPF screens provided with the acf2/MVS product.

ACFRPTDS - SAMPLE OUTPUT (TERMINAL FORMAT)

The following examples show the ACFRPTDS output when the TERMINAL option the default, is specified.

ACF2 UTILITY LIBRARY - ACFRPTDS - DATASET ACCESS JOURNAL - PAGE 1
DATE 05/11/80 (80.132) TIME 16.02

LABA 80.132 05/11 09.02 VOLUME LOGGING
ONLINEA VOL=EXP019 DDN=EXPDATA0 DSN=EXP019.DATA.TAPE
SAVEDATA VOL=PUBL01 PGM=LABSAVE LIB=LAB1.LOAD
JOB 302 TP-OPN **BLP** ACCESS NAM=LABA SUPPORT
CPUA SRC=TERM02 LIB-PGM UID=LBALABA

HAL1 80.132 05/11 09.01 DATASET LOGGING
HAL1 VOL=PUBL01 DDN=SYS00039 DSN=SYS1.MACLIB
TSOUSER VOL= PGM=R3QENDSV LIB=SYS1.LINKLIB
TSU 2905 DA-OPN OUTPUT ACCESS NAM=LOCK, HERBERT A
CPUA SRC=RDR1 UID=SPSHAL1

JFH 80.132 05/11 09.01 DATASET LOGGING SEC-OFF
JFHRCVR VOL=PUBL09 DDN=DDPUBLO9 DSN=TEST.DATA
IEHMOVE VOL= PGM=IEHMOVE LIB=SYS1.LINKLIB
JOB 301 DA-OPN OUTPUT NORULE NAM=HOOVER, JOHN F
CPUB SRC=TERM01 UID=SPSJFH

Below are the names of the fields shown in the report listing above. These fields are discussed in the Field Descriptions section for this report.

logonid jdate gdate time record inst stape RKEY=rule
jobname VOL=dsnvol DDN=ddname DSN=dataset
stepname VOL=libvol PGM=pgmname LIB=library
jobid major minor rmrc NAM=name
cpuid SRC=source path UID=uid
nextkey:

*Note: The data printed on the fifth line of each record does not appear when displaying the Terminal Format output with the NOEXTEND option selected.

ACFRPTDS - SAMPLE OUTPUT (PRINTER FORMAT)

ACFRPTDS - Sample Output Printer Format

ACF2 UTILITY LIBRARY - ACFRPTDS - DATASET ACCESS JOURNAL - PAGE 1
DATE 05/11/80 (80.132) TIME 16.04 PRINTER

DATASET	ACCESS TYPE	RM-RC	INST	STAPE	UID	NAME	JOB #	CPUID	LIBRARY	DDNAME	LVOL	TI	ME
	VOL	RULE	LOG TYPE	PATH	LID	SNAME			PROGRAM	DATE			
EXP019.DATA.TAPE					LBALABA				LAB1.LOAD				
	TP-OPN	**BLP**	ACCESS		LABA	LABA SUPPORT			LABSAVE	EXPDATA0	PUBL01		
	EXP019	EXP019	VOL LOG	LIB-PGM	ONLINEA	SAVEDATA	JOB 302	CPUA	INTRDR	80.132	05/11	09	.02
SYS1.MACLIB					SPSHAL1				SYS1.LINKLIB				
	DA-OPN	OUTPUT	ACCESS		HAL1	LOCK, HERBERT A			R3QENDSV	SYS00039			
	PUBL01	SYS1	DSET LOG		HAL1	TSOUSER	TSU 2905	CPUA	TERM01	80.132	05/11	09	.01
TEST.DATA					SPSJFH				SYS1.LINKLIB				
	DA-OPN	OUTPUT	NORULE	SEC-OFF	JFH	HOOVER, JOHN F			IEHMOVE	DDPUBL09			
	PUBL09	TEST	DSET LOG		JFHRCVR	IEHMOVE	JOB 301	CPUB	RDR1	80.132	05/11	09	.01
dataset	major	minor	rmrc	inst	uid	name			library	ddname	libvol		
	dsnvol	rule	record	path	Logonid	stepname	jobid	cpuid	pgmname	jdate	gdate	ti	me

This report output shows the ACFRPTDS utility run with the PRINTER option (note the title field taken from the JCL parm string). The sample output is shown using the same input SMF records (the same access situations) as were shown in the terminal output example on the previous page.

ACFRPTDS - SAMPLE OUTPUT (SUMMARY FORMAT)

ACFRPTDS - Sample Output (Summary Format)

ACF2 UTILITY LIBRARY - ACFRPTDS - DATASET ACCESS JOURNAL
DATE 05/11/80 (80.132) TIME 16.04 SUMMARY

- PAGE 1

DATE	TIME	JNAME	PROGRAM	LID	NAME	LOG-TYPE DATASET NAME
80.132	05/11	09.01	HAL1	R3QENDSV	HAL1	LOCK, HERBERT A
80.132	05/11	09.01	JFHRCVR	IEHMOVE	JFH	HOOVER, JOHN F
80.132	05/11	09.02	ONLINA	LABSAVE	LABA	LABA SUPPORT
						DSET LOG SYS1.MACLIB
						DSET LOG TEST.DATA
						VOL LOG EXP019.DATA.TAPE

NEXTKEY REPORTING

The sample ACFRPTDS report on the next page lists an event where the maximum number of NEXTKEY parameters was exceeded during a dataset validation.

In the sample, user NPDTEST1 requested ALLOCate access for dataset "T314NK01.TEST.DATASET1". However, the rule entry for the dataset directed ACF2 to another rule key via the NEXTKEY rule option. ACF2 allows a maximum of 25 NEXTKEYs when validating access to a dataset. Eventually, the rule key used to validate the access request in the sample was T314NK26, which was the 26th rule set searched by ACF2 during validation processing. Therefore, a KEYEXCES condition occurred.

The first logging entry on the sample is from a TRACE record and indicates that a KEYEXCES condition occurred when ACF2 was directed to the 26th rule set. Note that the NEXTKEY field of the logging entry lists all rule sets that were searched during ACF2 validation.

These trace records are a valuable aid in determining where and why a KEYEXCES condition occurred. In addition, if a NEXTKEY loop occurs, the easiest method of determining where the loop occurred is the TRACE record. When a NEXTKEY loop occurs, the rsrc field of the report indicates an NKEYLOOP condition. The NEXTKEY field on the report lists all rule sets that were referenced during ACF2 validation.

The second logging entry is a dataset violation record and indicates that ACF2 aborted the access request due to a KEYEXCES condition. The RKEY field indicates the rule key being processed when the access was aborted.

ACFRPTDS - SAMPLE OUTPUT (EXCESS NEXTKEYS)

ACF2 UTILITY LIBRARY - ACFRPTDS DATASET ACCESS JOURNAL
DATE 06/02/83 (83.153) TIME 09.49 NEXTKEY SAMPLE

NPDTST1 83.153 06/02 09.39 DATASET TRACE REQ RKEY=T314NK26
NPDTST1 VOL=WORK02 DDN= DSN=T314NKO1.TEST.DATASET1
TS73659 VOL= PGM=IDFSD060 LIB=SYS1.LINKLIB
JOB 8041 ALLOC KEYEXCES NAM=JOHN SMITH
SKK1 SRC=LV437 UID=PER99NPDTST10123
NEXTKEY: T314NKO1 T314NKO2 T314NKO3 T314NKO4 T314NKO5 T314NKO6
T314NKO7 T314NKO8 T314NKO9 T314NKO10 T314NKO11 T314NKO12 T314NKO13
T314NKO14 T314NKO15 T314NKO16 T314NKO17 T314NKO18 T314NKO19 T314NKO20
T314NKO21 T314NKO22 T314NKO23 T314NKO24 T314NKO25

NPDTST1 83.153 06/02 09.39 DATASET VIOLATION RKEY=T314NK26
NPDTST1 VOL=WORK02 DDN= DSN=T314NKO1.TEST.DATASET1
TS73659 VOL= PGM=IEFSD060 LIB=SYS1.LINKLIB
JOB 8041 ALLOC KEYEXCES NAM=JOHN SMITH
SKK1 SRC=LV437 UID=PER99NPDTST10123

The following list shows the various names of the fields in the report outputs shown above. These fields and their meaning are discussed in the "Fields" section.

logonid jdate gdate time record inst stape RKEY=rule
jobname VOL=dsnvol DDN=ddname DSN=dataset
stepname VOL=libvol PGM=pgmname LIB=library
jobid major minor rmrc NAM=name
cpuid SRC=source path UID=uid
nextkey:

*Note: The data printed on the fifth line of each record does not appear when displaying the Terminal Format output with the NOEXTEND option selected.

FIELD DESCRIPTIONS

logonid

This is the Logonid of the user who attempted the dataset access.

jdate

The Julian date on which the access or attempted access occurred.

gdate

The corresponding Gregorian date. The format of this date field will be either MM/DD or DD/MM depending on ACF2 generation options.

time

The time of day when the access or attempted access occurred.

record

The type of security record being formatted. The list below details the various keywords that can be present in this field. The keywords are abbreviated in the printer output format as shown within the parenthesis.

PROGRAM (PROG) - Indicates this record was issued for program access validation rather than dataset validation.

VOLUME (VOL) - Indicates the access was validated at a volume level. The dataset name in this case may be @volser.VOLUME as defined by ACF2 volume protection.

DATASET (DSET) - The access is to a dataset.

INVPARMS - The access request validation parameter list was invalid. As much information as could be determined is listed in the report. The record will also be printed in hexadecimal notation.

VIOLATION (VIO) - This journal record was issued because the access violated ACF2 access controls.

LOGGING (LOG) - The access as defined by this journal record was allowed but logged either because the access rule requested logging or the access was allowed because the user was a security officer or non-cancellable.

TRACE (TRC) - This access was journalled because the user involved was marked with the TRACE attribute in his Logonid record. A trace record may be accompanied by a logging or violation record depending on the access rules. In addition, a trace record is automatically written whenever a NKEYLOOP or KEYEXCES condition occurs (see rmrc description above).

inst

This field indicates that the logging was created because of an installation exit or installation specification as detailed in the following list.

- PRE-VALD The installation pre-validation exit requested that the access be journalled.
- DSNGEN The dataset name generator exit requested that the access be journalled.
- DSNPOST The dataset post-validation exit requested that the access be journalled.
- VIO-EXIT The installation violation exit caused this access to be journalled.
- SEC-OFF This access was allowed because the user was a security officer.
- NON-CANC This access was allowed because the Logonid was marked with the NON-CNCL attribute.
- READ-ALL This access for input was allowed because the Logonid had the READALL attribute (read-only non-cancellable).

stape

This field reflects any special information concerning this access. The keywords in the table below define the values this field may take. For further information on any of these keywords, refer to the acf2/MVS Administrator's Guide for explanations of the related GSO record in the acf2/MVS Administrator's Guide.

- TAPE The dataset being referenced is a tape dataset.
- MANT-PGM The program named in the PGM field is defined as a maintenance program via the MAINT record in the Infostorage database.
- BLP-PGM The program named in the PGM field is allowed to use bypass label processing access for tapes, as defined by the BLPPGM record in the Infostorage database.
- PGM-LOG The program named in the PGM field is specified in a GSO LOGPGM record in the Infostorage database.
- BLP-LOG This access (using BLP) was authorized because the TAPE-BLP or TAPE-LBL attribute was present in the Logonid record associated with the access; or the program used was listed in the GSO BLPPGM record. A logging has been generated because the installation has specified the BLPLOG field in the OPTS GSO record. See the chapter on GSO records in the acf2/MVS Administrator's Guide.
- BLP-VIO Bypass label processing was not authorized for processing this tape volume.

RKEY=ruleid

The name (**\$KEY**) of the access rule set used to process this request. This information is optional in the terminal format report and is displayed only if the rule set used is not the same as the dataset high level index, (e.g., if a NEXTKEY rule was used for validation).

jobname

The name of the job. For time sharing (TSO) sessions the jobname will generally be the same as the Logonid.

VOL=dsnvol

The volser of the volume on which the accessed dataset resides.

DDN=ddname

The DDNAME of the DDCARD through which access was being attempted, if applicable. Note: If DDN=SYSUDUMP, SYSMDUMP, SYSABEND, ABENDAID, or ABNLTERM, and if DSN is a JES assigned dsn (such as JES2.JOB00381.S00104), then this record is on the report because the user was in a program pathing or an execute-only environment, did not have DUMPAUTH in his Logonid record, and the system attempted a dump (for a non-ACF2 ABEND of the job). In this case, ACF2 is not using access rules to determine that this is a violation, but the Logonids DUMPAUTH authority. Writing a rule for **\$KEY(JES2)** would not affect this processing. Other records without JES-type dsns but with DDN=SYSUDUMP, SYSMDUMP, SYSABEND, ABENDAID or ABNLTERM are also likely to be on the report for the same reason.

DSN=dsname

The name of the dataset for which access was being attempted. This is the actual name used for dataset validation and rule interpretation and reflects any editing by installation exits or by ACF2. Note: If this data set name is invalid, it will result in the entire record being printed in hexadecimal notation.

stepname

The name of the job step active at the time of the access or attempted access.

VOL=libvol

The volser of the volume on which the program library was found.

PGM=pgmname

The name of the program attempting the access. This name will be the ACF2 translated name of a multi-module program if a structural model is defined. This means that the name in this report will be the active load module name, not just the name of the initiator-loaded load module.

LIB=library

The library from which the program was loaded. Certain library names have a specific meaning, such as:

SYS1.LINKLIB - This name is used for any library in the system linklist or the link pack area.

UNKNOWN.LIBRARY - This name is substituted when ACF2 cannot determine the library from which the program is executing. Following the name is a code detailing the reason why the library could not be identified. The various reasons are described in the table below.

- 0 - Access validation in a program path environment, after previously encountering a 913.
- 1 - The JOBLIB and/or STEPLIB DD cards define a concatenated group of libraries, and more than one of these libraries are not specified in the LINKLST GSO record. For further information, see the chapter on GSO records in the acf2/MVS Administrator's Guide.
- 2 - The online interface (e.g., the TSO TMP) detected an invalid state, or some other reason has previously caused an UNKNOWN.LIBRARY in the same job.
- 3 - The TCB and RB structure did not match the ACF2 structure model. (For example, a module was invoked (by LINK or ATTACH) that was not defined in the structure model, or the module did not have the attributes (RENT or SYSLIB) that were specified for that module in the structure model.)
- 4 - The online path structure was found to be invalid. (For example, the TCB structure found did not reflect the normal TMP environment.)
- 5 - The load modules which have been linked to or attached by the ACF2-determined active load module (using the structure model) are not from an APF-authorized library (including linklist libraries and the link pack area).
- 6 - The TSO TMP was run in batch with a JOBLIB or STEPLIB specification.
- 7 - The online CALL command was executed in an invalid environment.

8 - The program executing came from a library that was not on the RESVOLS list.

The above codes mention structures involving program paths and control block interrelationships for the operating system. If problems exist they should be referred to ACF2 maintenance personnel at the installation. (ACF2 program structure models are described in the acf2/MVS System Programmer's Guide. See TSO Generalized Path Control.)

jobid

The JES2 or JES3 assigned job number.

major

This field describes the system/ACF2 component through which access is being attempted. This field may take on any of the values in the table below.

DA-OPN DADSM open issued the request.

DA-EOV DADSM E-O-V issued the request.

VS-OPN VSAM open issued the request.

TP-OPN Processing issued the request. This request occurs before volume mount and label verification.

TP-EOV Tape EOVS issued the request. This request occurs before the new volume has been mounted and its label verified.

CATLG AMS/CMS issued the request.

ALLOC DADSM request for new dataset allocation.

RENAME DADSM rename operation requested (to/from names are indeterminate).

REN-FR DADSM rename operation requested (original dataset name).

REN-TO DADSM rename operation requested (new dataset name).

DELETE DADSM dataset scratch requested.

INSTLL This violation is in response to a user security request (FDR, FDRDSF, ASM2, etc.).

EXTRNL This is an external request for a multi-user address space subsystem (WYLBUR, ROSCOE, JES2-userproc, CRJE, ACEP, etc).

PRGNAM This request was for program execution authorization by the initiator.

- TP-XTD Tape open processing during volume verification. This request occurs after volume mount and label verification processing.
- TP-XPB Tape open processing after all final volume verification and label processing has occurred, and the system has updated all O-C-E workarea control blocks.
- TP-XOV Tape E-O-V processing after new volume verification and label processing are updated. All internal O-C-E workarea control blocks are updated.
- CVOL This request was issued by catalog management CVOL processing. To determine the type of CVOL request being made, see the "minor" field.

minor

The type of access being performed. The major and minor fields combine to detail the exact nature of the dataset access environment. The possible values the minor field may contain are listed below.

- INPUT The dataset being processed is to be read only.
- RDBACK The dataset is being processed for input and being read backwards. Readback requires an access rule specifying write access rather than just read access.
- IN/OUT The request specified that the dataset was being opened for input and output processing. The JCL for the program may be modified to specify only input processing for this case by specifying LABEL=(,,IN) on the appropriate DDCARD. This access type is standard for FORTRAN files and will result in a security violation if only read access is allowed and the JCL LABEL parameter is not specified to limit processing to input-only.
- UPDATE This access will be to read records from the dataset and update them in place.
- OUT/IN This access will both write and read the dataset. The LABEL parameter in the JCL may be used to cause this access to be a write access by specifying LABEL=(,,OUT).
- OUTPUT The dataset being accessed will be written only.
- EXECUTE For PRGNAM access, the program is to be executed.
- DEFINE CMS functions, catalog entry is being created.
- ALTER CMS functions, catalog entry is being modified.

- DELETE CMS functions, catalog entry is being deleted. This does not require that the dataset also be deleted.
- **BLP**** The access is to a tape dataset and the JCL specified bypass label processing access via the LABEL=(,BLP) DD parameter.
- CATLG CVOL Catalog request (generates an entry in the index of the catalog).
- UNCAT CVOL Uncatalog request (removes an entry from the index of the catalog).
- RECAT CVOL Recatalog request (replaces an entry in the index of the catalog).
- BLDX CVOL Build Index request (creates a new index in the catalog).
- BLDG CVOL Build GDG Index request (builds an index for generation data groups).
- BLDA CVOL Build Alias request (assigns an alias to an index).
- LNKX CVOL Link request (connects two volumes together).
- DLTX CVOL Delete Index request (removes an index from the catalog).
- DLTA CVOL Delete Alias request (deletes an alias previously assigned to an index).
- DRPX CVOL Disconnect request (disconnects two volumes).
- UNKNOWN CVOL request of an unknown nature (none of the requests listed above).

rmrc

The return code from the ACF2 access rule record manager and interpreter.

\$MODEQT The access to the dataset was allowed because a \$MODE(QUIET) control card was specified in the access rule. See note A.

ACCESS An access rule was found that matched the environment when this access was attempted. The rule specified that access was to be allowed.

BLPLOG The access was logged because a BLP tape access was attempted and the GSO OPTS record indicated BLPLOG.

BLPVIO The user requested BLP access to a tape dataset and did not have that authority. Access was denied.

DASDUNSC The ACF2 GSO options specified by the installation indicated that this DASD dataset was not to be protected by ACF2.

DUMPAUTH A dump was requested and the user had opened a dataset in which program pathing allowed the user access to the dataset. In this case, the user must have the DUMPAUTH Logonid attribute to allow the dump to be taken.

EXITALLW An installation exit allowed access to the dataset.

EXITVIO An installation exit denied access to the requested dataset.

INITFAIL Scope record initialization failed.

INV-CMD The control block structure for a command under TSO did not match the structure defined for that command in the ACF99@RB command structure definition module.

INV-JSL The program pathing code was unable to correctly determine the library for the jobstep program. An invalid return code was detected from one of the information gathering routines of BLDL. The library is defaulted to 'SYS1.LINKLIB'.

INV-TMP The control block structure for the TSO TMP was found to be invalid. One of the programs that make up the TMP or front-ends the TMP was not from an APF authorized library.

I/O-ERR Access denied because TAPE OPEN intercept could not read the tape label.

MAINT The access to this dataset is allowed because the requester has the MAINT Logonid authority and the environment (program and library) matched a GSO MAINT record entry.

NKEYLOOP Access was denied because the NEXTKEY parameter on one of the rules pointed directly or indirectly to itself.

NOACCESS An access rule indicated to prevent access.

NOACUCB Access to the dataset was allowed because no ACUCB was available with which to validate the request.

- NO-CDE No CDE was available to determine the active program name for program pathing. Normally happens during a dump to a program pathed dataset for S806 abends.
- NON-APF One of the programs that was used by a TSO command did not come from an APF authorized library.
- NORECORD The access ruleset did not exist.
- NORULE An access rule in a ruleset could not be located that matched the environment.
- NOTEST An invalid path for dataset access exists. A rule allowed access to a DSN via program pathing but the program is being executed under TSO TEST.
- NOTPPGM A program name only check was made and the program was not on the GSO PPGM list.
- OWNED Access was allowed because the high level index of the dataset being accessed matched the prefix of the user making the request.
- PATHERR Unknown program pathing error.
- PPGMVIO The user request to execute a program listed on the GSO PPGM record was denied. The user did not have the required authority to do so.
- SCOPFAIL External scope processing failed.
- SCOPESEC The access was allowed because the requester was a scoped SECURITY officer and the dataset high level index was within the officer's scope.
- SPECIAL The access was allowed because of "special" authority associated with this request. This would normally apply to some implicit operation done on behalf of a user who did not directly request the action.
- SYNTAX An invalid parameter list was passed to the ACCESS RULE INTERPRETER.

- TAPEUNSC The ACF2 GSO options specified by the installation indicated that this TAPE dataset was not to be protected by ACF2.
- \$MODELG The access to the dataset was allowed and logged because a \$MODE(LOG) control card was specified in the access rule. See NOTE A.
- \$MODEWR The access to the dataset was allowed because a \$MODE(WARN) control card was specified in the access rule. See NOTE A.
- \$MODEAB The access to the dataset was denied because a \$MODE(ABORT) control card was specified in the access rule. See NOTE A.
- NOMODEQT The access to the dataset was allowed because the installation was in RULE MODE (as indicated on the GSO OPTS record). And, the no-mode option was QUIET mode. See NOTE B.
- NOMODELG The access to the dataset was allowed and logged because the installation was in RULE MODE (as indicated on the GSO OPTS record). And, the no-mode option for that mode was specified LOG. See NOTE B.
- NOMODEWR The access to the dataset was allowed and logged because the installation was in RULE MODE and the no-mode parameter was WARN as indicated on the GSO OPTS record. See NOTE B.
- NOMODEAB The access to the dataset was denied because the installation was in RULE MODE and the no-mode parameter was ABORT as indicated on the GSO OPTS record. See NOTE B.
- NORECQT The access to the dataset was allowed because the installation was in RULE MODE and the no-record parameter was specified as QUIET (indicated on the GSO OPTS record). See NOTE C.
- NORECLG The access to the dataset was allowed and logged because the installation was in RULE MODE and the no-record parameter was LOG (specified in the GSO OPTS record).
- NORECWR The access to the dataset was allowed because the installation was in RULE MODE and the no-record parameter was WARN (as specified in the GSO OPTS record). See NOTE C.
- NORECAB The access to the dataset was denied because the installation was in RULE MODE and the no-record parameter was ABORT (as shown in the GSO OPTS record).

RULELOG Access was allowed and logged because a rule indicated allow and log access.

QUIET Access was allowed because ACF2 was in QUIET mode.

LOG Access was allowed because ACF2 was in LOG mode.

WARN Access was allowed because ACF2 was in WARN mode.

KEYEXCES The access was denied because more than 25 levels of NEXTKEY were found.

NOTE A: A \$MODE control card would only apply when the installation is in RULE MODE and based on the rule, access would otherwise have been denied. The rule indicated prevent or no rule in the rule set applied.

NOTE B: The no-mode condition applies when:
1) The installation is in RULE MODE.
2) Based on the rule, access would otherwise have been denied (rule indicated prevent or no rule in the rule set applied).
3) The \$MODE control card was not specified in the access ruleset.

NOTE C: The no-record condition applies when:
1) The installation is in RULE MODE.
2) Based on the rule, access would otherwise have been denied (rule indicated prevent or no rule in the rule set applied).
3) A \$MODE control card was not specified in the ruleset.

ACFRPTEL - INFORMATION STORAGE UPDATE LOG

The ACFRPTEL report generator processes the journal SMF records issued for ACF2 recovery purposes and provides a listing of each change to the Information Storage Facility data base. These changes include updates to entry records and generalized resource rule sets.

FILES

ACFRPTEL uses SYSPRINT, SYSIN, and RECxxxxx as described in the General Information section of this manual.

PARAMETERS

ACFRPTEL accepts the following common parameters:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SELECT, SYSID, and HEX.

The following parameters are specific to ACFRPTEL:

TYPE(type-mask) - DEFAULT=- (all)

The type parameter for ACFRPTEL allows the user to request update information for a specific type of resource rules or Infostorage database records (i.e., entry, scope, shift/zone, or control records). The following list describes the default names for various types in ACF2. Additional type codes can be defined locally:

- CFC - CICS file control rule sets
- CKC - CICS transaction control rule sets
- CPB - CICS DL/I request rule sets
- CPC - CICS program control rule sets
- CTD - CICS transient data rule sets
- CTS - CICS temporary storage rule sets
- DAT - IDMS area control rule sets
- GSO - Global System Option records
- IAG - IMS application group name rule sets
- ITR - IMS transaction control rule sets
- OID - Operator Identification records
- PGM - IDMS program control rule sets
- PGN - IDMS non-protected program control rule sets
- SCP - Scope lists

SFT - Time shift records
SGP - Input source group records
SRC - Input source records
SSC - IDMS subschema rule sets
TAC - TSO account rule sets
TPR - TSO procedure rule sets
TSK - IDMS task control rule sets
ZON - Time zone records

Note: This parameter is selecting only by the three character type code, and does not differentiate between storage classes. Therefore, to prevent possibly confusing reports, it is recommended that you do not assign type codes which duplicate other type codes, even if they are in different storage classes.

ID(id-mask) - DEFAULT=- (all)

The ID parameter specifies a mask for the actual name of the generalized resource rule set or entry list. Combined with the TYPE parameter the ID parameter allows the user to search for specific Information Storage Facility updates.

MASK(logonid-mask) - DEFAULT=***** (all)

The MASK parameter specifies selection of records by ACFRPTTEL based upon the Logonid of the person making the change thereby providing a summary of activity by a single person or group of people.

JOBMASK(job-mask,job-mask,...) - DEFAULT=***** (all)

The JOBMASK parameter specifies that records appearing on the report be limited to those pertaining to the job(s) indicated by the jobname mask(s). Commas or blank spaces must separate multiple jobname(s) or jobname mask(s).

DETAIL/SUMMARY - DEFAULT=SUMMARY

The DETAIL parameter produces additional lines of information on the report for any INSERT or CHANGE subcommand entries made to update the Infostorage database. This information includes: the name of each field changed in the Infostorage database record; the old value of each field; and the new value of each field. The SUMMARY parameter produces only one line of information for the INSERT or CHANGE subcommand entry. SUMMARY is the default.

| SORT SEQUENCE

| The recommended sort sequence for the ACFRPTEL report is by storage
| class (major), type code, record name, date, and then time. The
| installation can perform this sorting through the its own routine or by
| modifying and using the prototype JCL provided with the acf2/MVS
| product.

(ACFRPTEL continued on the next page.)

ACFRPTEL - SAMPLE OUTPUT (SUMMARY PARAMETER)ACFRPTEL - Sample Output

The following are two examples of the report issued by ACFRPTEL. The first example shows the report when the default parameter of SUMMARY is in effect:

ACF2 UTILITY LIBRARY - ACFRPTEL - INFORMATION STORAGE UPDATE LOG - PAGE
DATE 06/15/84 (83.167) TIME 16.38

DATE	TIME	JNAME	LID	MODULE	FUNCTION	CPU	C-TYP-NAME		
84.158	06/06	17.02	ACTISO	ACTISO	ACF60CMD	INSERT	CPUA	T-SFT-TEST	
84.158	06/06	17.15	ACTISO	ACTISO	ACF60CMD	DELETE	CPUA	T-SFT-TEST	
84.158	06/06	18:50	ADMISO	ADMISO	ACFOAENT	REPLACE	CPUA	C-GSO-ABC1	LINKLST
84.158	06/06	18:51	ADMISO	ADMISO	ACFOAENT	REPLACE	CPUA	C-GSO-ABC1	LINKLST
84.158	06/06	18:51	ADMISO	ADMISO	ACFOAENT	REPLACE	CPUA	C-GSO-ABC1	LOGPGM
84.158	06/06	18:54	ADMISO	ADMISO	ACFOAENT	REPLACE	CPUA	C-GSO-ABC1	OPTS
84.158	06/06	18:55	ADMISO	ADMISO	ACFOAENT	REPLACE	CPUA	C-GSO-ABC1	OPTS
84.158	06/06	19/12	ADMISO	ADMISO	ACFOAENT	REPLACE	CPUA	C-GSO-ABC1	OPTS

In this example report, the first entry shows that, on June 6th, 1984 at 5:02 p.m., the user ACTISO inserted a shift record named TEST. The insertion was done under the jobname ACTISO by using the ACF60CMD module on CPUA. Other entries can be read in a similar fashion.

ACFRPTEL - SAMPLE OUTPUT (DETAIL PARAMETER)

This second example shows the report when the DETAIL parameter is specified:

```
ACF2 UTILITY LIBRARY - ACFRPTEL - INFORMATION STORAGE UPDATE LOG - PAGE
DATE 06/15/84 (83.167) TIME 16.38
  DATE    TIME  JNAME    LID      MODULE  FUNCTION CPU  C-TYP-NAME
  FIELD    OLD VALUE                                NEW VALUE
84.158 06/06 17.02 ACTISO  ACTISO  ACF60CMD INSERT   CUA  T-SFT-TEST
84.158 06/06 17.15 ACTISO  ACTISO  ACF60CMD DELETE   CUA  T-SFT-TEST
      *** NO FIELDS CHANGED ***
84.158 06/06 18:50 ADMISO  ADMISO  ACFOAENT REPLACE  CUA  C-GSO-ABC1  LINKLST
      LIBRARY      SYS1.LINKLIB
                                SYS1.LINKLIB,
                                ACF2SRC.WORK.LOAD,
                                ACTRSO.R314.LOAD
84.158 06/06 18:51 ADMISO  ADMISO  ACFOAENT REPLACE  CUA  C-GSO-ABC1  LINKLST
      LIBRARY      SYS1.LINKLIB
                                ACF2SRC.WORK.LOAD,
                                ACTRSO.R314.LOAD
84.158 06/06 18:51 ADMISO  ADMISO  ACFOAENT REPLACE  CUA  C-GSO-ABC1  LOGPGM
      PGMS          AMASPZAP,IMASPZAP
                                AMASPZAP,IMASPZAP,
                                INCORZAP
84.158 06/06 18:54 ADMISO  ADMISO  ACFOAENT REPLACE  CUA  C-GSO-ABC1  OPTS
      $NOSORT      NO$NOSORT
      BLPLOG       NOBLPLOG
      CENTRAL     NOCENTRAL
      CHANGE      CHANGE
      CONSOLE     NOROLL
      LABNUM      0
                                5
84.158 06/06 18:55 ADMISO  ADMISO  ACFOAENT REPLACE  CUA  C-GSO-ABC1  OPTS
      MODE        ABORT
                                RULE,ABORT,ABORT
84.158 06/06 19/12 ADMISO  ADMISO  ACFOAENT REPLACE  CUA  C-GSO-ABC1  OPTS
      DSNPOST     ---NULLS---
      VLDEXIT     ---NULLS---
                                POSTVLD
                                ABCVALD
```

This example report lists the same entries as the previous example, except it also shows the changed information, before and after the change. Other entries can be read in a similar fashion. See the "Field Description" for a further explanation of how to interpret the additional fields reported under the DETAIL parameter.

FIELD DESCRIPTIONS

The following text describes the fields of the ACFRPTEL reports:

DATE

The Julian and Gregorian date on which the update was made. The format of this date is either MM/DD or DD/MM based on ACF2 generation options.

TIME

The time at which the update was made.

JNAME

The name of the job under which the updates were made. If this is a TSO session then, in general, the jobname and the Logonid will be the same.

LID

The Logonid of the user who made the update.

MODULE

The name of the program that issued the update request. This name is provided to ACF2 by the program making the update request.

FUNCTION

The type of update requested. This field can take any of the following values:

INSERT Request was to insert a new record or generalized resource rule set into the Infostorage database. This request function indicates no previous record existed.

REPLACE The update request specified insertion but the record or generalized resource rule set already existed. ACF2 converts INSERT requests to REPLACE unless instructed not to do so.

DELETE The update request specified deletion of a record or generalized resource rule set from the Infostorage database.

C-TYPE-NAME

The storage class, type code, and name of the record being changed. The storage class and type code may be one of the following letters:

- C Control(Global System Option) record--with type code of GSO.
- E Entry record--with type code of SRC, SGP, or OID.
- R Generalized resource rule set--with type code of CFC, CKC, CPB, CPC, CTD, CTS, DAT, IAG, ITR, PGM, PGN, SSC, TAC, TPR, or TSK.
- S Scopelist record--with type code of SCP.
- T Shift or Zone record--with type code of SFT or ZON.

The name of the record may be one of the following:

- * Record-id for entry records
- * Key (designated in the \$KEY control card) for generalized resource rule sets
- * Scopelist name
- * Name of the shift or zone
- * Control record name

USING

The id of the model record specified in the USING parameter.

The following fields appear as an additional line of information on control record updates when the DETAIL parameter is specified:

FIELD

The name of the field in the record that was updated.

OLD VALUE

The value of the field before the update was made.

NEW VALUE

The value of the field after the update.

When the DETAIL parameter is in effect, the values before and after the change to the Infostorage database may be reported as follows:

- * If the value of a field is too long, it will be continued onto more than one line of the report.
- * If a field contains no value, then the message --- NULLS --- will appear for the value of that field on the report.

- | * If the value of a field cannot be reconstructed (such as a
| password), then the message --- NON PRINTABLE --- will be reported
| for the value of that field.

- | * If the user printing the report does not have authorization to list
| the value of a field, then the message AUTH-REQ'D will be reported
| for the value of that field.

- | * If no fields have been changed, then the message *** NO FIELDS
| CHANGED *** will be reported instead the old and new values of the
| field.

ACFRPTIX - DATASET INDEX REPORT

The ACFRPTIX utility is designed to aid the security officer in determining when the access environment for a particular dataset prefix has changed. To perform this function, ACFRPTIX searches the ACF2 data base update journal records.

When ACFRPTIX encounters a Logonid update record, it checks the owned dataset prefix field for a match against the prefix being searched. The owned dataset prefix in the Logonid record indicates the datasets a user can access without any SMF journalling. The listing provided shows changes to a Logonid with a PREFIX matching the search argument, this does not indicate that the owned dataset prefix was changed simultaneously. ACFRPTIX ignores UPDATE changes from job validation to reduce output volume and because this type of record could not possibly change the access environment.

ACFRPTIX also checks access rule update records for matches against the search prefix. A match indicates an update to the access rule set for the prefix being searched. Optionally, ACFRPTIX can decompile and list the new access rule set. Execution of ACFRPTIX requires a region size of 256K.

FILES

ACFRPTIX uses SYSPRINT, SYSIN, and RECxxxxx as described in the General Information section of this manual.

DETAIL

ACFRPTIX uses the DETAIL file to output the decompiled rule set. This file must be specified if the DETAIL parameter is in effect (even if by default). The DCB options for this file default to LRECL=137, BLKSIZE=3665, and RECFM=VBA.

PARAMETERS

ACFRPTIX accepts the following standard parameters:

SDATE, EDATE, TITLE, LINECNT, STIME, ETIME, SYSID, and HEX.

In addition, the following parameters are accepted:

SELLID(nn,nn,...) - DEFAULT=ACFFDR value

The SELLID parameter defines the Logonid SMF journal record number. The default value, like the SELECT parameter default, is available only if the ACFFDR is available on the executing CPU. This default value is specified in the @SMF macro of the ACFFDR.

SELRULE(nn,nn,...) - DEFAULT=ACFFDR value

The SELRULE parameter defines the SMF record number(s) for the ACF2 accessrule data base update journal. Again, the default is not available unless the ACFFDR is available on the executing CPU. This default value is specified in the @SMF macro of the ACFFDR.

PREFIX(mask) - DEFAULT=***** (all)

The PREFIX parameter specifies the dataset high level index that is to be used as a search argument. ACFRPTIX checks the Logonid owned dataset prefix field and the key of each access rule set processed.

JOBMASK(job-mask,job-mask,...) - DEFAULT=-

The JOBMASK parameter specifies that records appearing on the report be limited to those pertaining to the job(s) indicated by the jobname mask(s). Commas must separate multiple masks.

DETAIL/NODETAIL - DEFAULT=DETAIL

The DETAIL parameter requests that access rule set updates be decompiled and the result displayed using the DETAIL output file. Specifying NODETAIL will result in only a summary listing being produced. When NODETAIL is specified, the DETAIL output file is not needed.

ACFRPTIX - SAMPLE OUTPUT

ACF2 UTILITY LIBRARY - ACFRPTIX - ACCESS INDEX REPORT - PAGE 1
DATE 05/05/83 (83.125) TIME 13.35 PREFIX((*****))

DATE	TIME	TYPE	KEY	CHANGER	JOBNAME	CHANGE	CPU	DET
83.125	09/13	07.50	LID	ACFUSER	ACFUSER	CHANGE	168A	
83.125	09/13	07.50	LID	ACF2	ACF2	INSERT	168A	
83.125	09/13	08.40	RULE	ACFUSER	ACFUSER	REPLACE	168A	1
83.125	09/13	09.40	RULE	ACF2	ACF2	INSERT	168A	3

The example shows the main report from ACFRPTIX. The detail report which would also exist for this run contains a similar heading and the decompilation of the rule sets.

FIELD DESCRIPTION

The fields in this report are described below:

DATE

The date on which the Logonid record or access rule set was updated. The format is MM/DD or DD/MM depending on ACF2 generating options.

TIME

The time of day for the update.

TYPE

The type of record for which this summary report is being issued as defined in the following table.

LID A Logonid record has been found whose owned dataset prefix matches the search prefix.

RULE An access rule set whose key matches the search argument has been found.

KEY

The Logonid being changed or the key of the access rule set updated.

CHANGER

The Logonid of the user who initiated the change. If the system operator issues an "F ACF2,RESET(logonid)" command, this field is set to OPER.

JOBNAME

The name of the job running at the time of the change. For TSO this name will usually be the same as the Logonid.

CHANGE

The type of update performed to the record.

For Logonids:

INSERT A new record has been inserted.

CHANGE An existing record has been changed.

DELETE/UPDATE changes are not listed by ACFRPTIX.

For access rules:

INSERT A new record has been inserted.

REPLACE An old record has been replaced.

DELETE The access rule set has been deleted.

CPU

The SMF CPU identification from which the update request was issued.

DET

For access rule sets that are decompiled, this field contains the page number in the detail report for the start of the decompilation listing for that rule.

ACFRPTJL - RESTRICTED LOGONID JOB LOG

The ACFRPTJL utility lists all system accesses by Logonids marked with the RESTRICT attribute. The RESTRICT attribute indicates that the Logonid does not have an associated password. Since these Logonids are intended primarily for use by production jobs, their use must be carefully controlled. Additionally, since the ACF2 default Logonid is defined with the RESTRICT attribute this log provides a summary of its usage.

The report produced by ACFRPTJL indicates the path of submission for jobs using a Restricted Logonid. This is important since Logonids marked with the RESTRICT attribute are generally required to be submitted by APF authorized programs (except for the default Logonid).

FILES

SYSPRINT, SYSIN, and RECxxxxx are used by ACFRPTJL as described in the General Information section of this manual.

PARAMETERS

ACFRPTJL accepts the following parameters as described in the General Information section of this manual:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SYSID, HEX, and SELECT.

The following parameters are processed specifically by ACFRPTJL:

MASK(logonid-mask) - DEFAULT=***** (all)

The MASK parameter allows the user to request that the report include information for a specific Logonid or group of Logonids. Specifying 'MASK(dfltlid)' will provide a listing of each use of the default Logonid.

JOBMASK(job-mask,job-mask,...) - DEFAULT=-

The JOBMASK parameter specifies that records appearing on the report be limited to those pertaining to the job(s) indicated by the jobname mask(s). Commas or blank spaces must separate multiple masks.

| SORT SEQUENCE

| The recommended sort sequence for the ACFRPTJL report is by Logonid
| (major), date, and then time. The installation can perform this sorting
| through the its own routine or by modifying and using the prototype JCL
| provided with the acf2/MVS product.

| (ACFRPTJL continued on next page.)

ACFRPTJL - SAMPLE OUTPUT

ACFRPTJL - Sample Output

The following listing shows the output produced by the ACFRPTJL report generator. Following the example is a discussion of the various fields in the report.

ACF2 UTILITY LIBRARY - ACFRPTJL - RESTRICTED LOGONID JOB LOG - PAGE 1
DATE 05/07/83 (83.127) TIME 15.34

DATE	TIME	LOGONID	JOBNAME	SUBMIT'R	SOURCE	PROGRAM	CPU	LAB
83.127	05/07	08.42	ACFDFT	ASM001		READER1		CPUA
83.127	05/07	08.44	ACFDFT	ASM002		READER2		CPUA
83.127	05/07	10.20	SQSHID	SQUISH	S-JOBCOPY	STC	*JOBCOPY	CPUA
83.127	05/07	12.24	ACCTUSER	SMFDUMP	S-JOBCOPY	STC	*JOBCOPY	CPUA
83.127	05/07	12.25	PRODID12	PAY134		READER3		CPUA CPUB

FIELD DESCRIPTIONS

DATE

The Julian and Gregorian date on which the job was validated. The format of this date is either MM/DD or DD/MM based on ACF2 generation options.

TIME

The time at which the job was validated. The validation date and time will generally differ from the reader date and time by only a fraction of a second.

LOGONID

The Logonid under which the job was validated.

JOBNAME

The name of the job that used the Restricted Logonid. Logonids marked with the RESTRICT attribute can only be used for background jobs.

SUBMIT'R

The Logonid that submitted the job under a restricted Logonid. The characters "S-" preceding the Logonid indicate that the job was submitted from a started task. The Logonid following the "S-" is either the started task procedure name or the default started task procedure name. For further information on the default Logonid for started tasks, refer to the acf2/MVS Administrator's Guide. See the explanation of the DFLTSTC field of the OPTS record in the chapter on the CONTOL Infostorage class.

SOURCE

The logical input source through which the job was submitted.

PROGRAM

The name of the program that submitted the job using a Restricted Logonid. An asterisk (*) preceding the name indicates that the program was APF authorized. The program name in this field is the name of the load module that did the actual submission of the job and may not be the same as the program specified in the JCL EXEC statement.

CPU

The SMF CPU identifier of the CPU where job validation occurred. This is not necessarily the CPU where the job was originally submitted nor the one where the job was executed. For JES2 sites, this job validation is done at JCL Conversion time. For JES3 sites, this is done at Input Services time. NOTE: If LAB (Lookaside Buffer support) is being used, the LAB field (see below) indicates the CPUid where the restricted Logonid was validated.

LAB

The LAB (Lookaside Buffer support) field indicates that validation was performed on the specified CPUid. LAB support allows ACF2 information, such as the Logonid record, to be transmitted from one CPU to another. Whenever a restricted Logonid is transmitted, the LAB field indicates the CPUid where the restricted Logonid was validated. If LAB is blank, then the restricted Logonid was validated at the CPUid indicated in the CPU field.

ACFRPTLL - LOGONID MODIFICATION LOG

The ACFRPTLL report generator uses the SMF records issued for ACF2 recovery purposes to provide an update activity report for the ACF2 Logonid data base.

ACF2 Logonid records are updated for two distinct reasons. The first of these is maintenance of the Logonid data base. The second group of changes occurs during JESx and LOGON validation where the Logonid access counts, time and source of the last access, and possibly the password are changed. An installation generally will not be interested in the large volume of validation updates, so the ACFRPTLL utility provides parameters which suppress this information.

FILES

ACFRPTLL accepts the standard I/O files SYSPRINT, SYSIN, and RECxxxxx.

PARAMETERS

The following general parameters are accepted by ACFRPTLL:

| LINECNT, TITLE, SDATE, EDATE, SDATE, EDATE, SYSID, HEX, and SELECT.

The following parameters are specific to ACFRPTLL:

UPDATE/NOUPDATE - DEFAULT=NOUPDATE

The UPDATE parameter, when specified, requests a summary of Logonid modifications including any LAB, JESx, and LOGON validation updates. The default of NOUPDATE causes only updates other than validation updates to be listed. The default of NOUPDATE was chosen because of the volume of validation updates (one for every job and TSO session).

MASK(logonid-mask) - DEFAULT=***** (all)

The MASK parameter is used to select update reporting for a specific Logonid record. Note that this mask is compared to the Logonid of the record being changed, not the Logonid of the changer.

JOBMASK(job-mask,job-mask,...) - DEFAULT=- (all)

| The JOBMASK parameter specifies that records appearing on the
| report be limited to those pertaining to the job(s) indicated by
| the jobname mask(s). Commas or blank spaces must separate multiple
| masks.

DETAIL/SUMMARY - DEFAULT=SUMMARY

The DETAIL parameter produces additional report lines that highlight changes made to any Logonid records (via the ACF INSERT or CHANGE subcommand). Each of these additional lines show: the name of the field whose value has changed; the old value of the field; and the new value of the field. The SUMMARY parameter produces the report without these additional report lines.

ACFRPTLL - SAMPLE OUTPUT (SUMMARY PARAMETER)

Three examples of the ACFRPTLL report are shown below. The first example shows the report when the default parameter of SUMMARY is in effect:

```
ACF2 UTILITY LIBRARY - ACFRPTLL - LOGONID MODIFICATION LOG - PAGE 1
DATE 08/13/84 (84.226) TIME 14.03
  DATE      TIME  LOGONID  JOBNAME  CHANGER  CHANGE  CPU  LAB  USING
84.219 08/06 11.22 PAYJKL   PAYISO   PAYISO   CHANGE  CPUTA  PAYMOD
84.219 08/06 14.22 PAYTEST  PAYARD   PAYARD   INSERT  CPUTA
84.219 08/06 14.24 PAYTEST  PAYARD   PAYARD   DELETE  CPUTA
84.219 08/06 15.31 ACTRRC   ACTISO   ACTISO   CHANGE  CPUTA
84.219 08/06 15.31 ACTRRC   ACTISO   ACTISO   CHANGE  CPUTA
84.219 08/06 15.33 ACTRRC   ACTISO   ACTISO   CHANGE  CPUTA
84.219 08/06 16.42 ACTPNC1  ACTPNC   ACTISO   CHANGE  CPUTA
84.220 08/07 09.01 ACTRRC   ACTISO   ACTISO   CHANGE  CPUTA
84.220 08/07 09.20 PAYTNS   PAYARD   PAYARD   CHANGE  CPUTA
84.220 08/07 15.18 PAYTNS   PAYISO   PAYISO   CHANGE  CPUTA
84.220 08/07 15.19 PAYTNS   PAYISO   PAYISO   CHANGE  CPUTA
84.221 08/08 10.30 PAYTNS   PAYISO   PAYISO   CHANGE  CPUTA
```

In the above example report, the first entry was made at at 11:22 a.m. on August 16th. A record for the Logonid PAYJKL was inserted by PAYISO, the payroll security officer. The record was processed through cpu A using PAYMOD as a model Logonid record. Other entries can be read in a similar fashion.

ACFRPTLL - SAMPLE OUTPUT (DETAIL PARAMETER)

The sample below shows the report when the DETAIL parameter is specified:

ACF2 UTILITY LIBRARY - ACFRPTLL - LOGONID MODIFICATION LOG - PAGE 1
DATE 08/13/84 (84.226) TIME 11.15

DATE	TIME	LOGONID	JOBNAME	CHANGER	CHANGE	CPU	LAB	USING
FIELD	OLD VALUE			NEW VALUE				
84.219	08/06	11.22	PAYJKL	PAYISO	PAYISO	CHANGE	CPUA	PAYMOD
TRACE	NOTRACE			TRACE				
84.219	08/06	14.22	PAYTEST	PAYARD	PAYARD	INSERT	CPUA	
JOB	NOJOB			JOB				
MAXDAYS	00000			00010				
MINDAYS	00000			00005				
MONITOR	NOMONITOR			MONITOR				
NAME	---NULLS---			ED JARVIS				
PHONE	---NULLS---			EXT. 233				
RULEVLD	NORULEVLD			RULEVLD				
TRACE	NOTRACE			TRACE				
TSO	NOTSO			TSO				
84.219	08/06	14.24	PAYTEST	PAYARD	PAYARD	DELETE	CPUA	
84.219	08/06	15.31	ACTRRC	ACTISO	ACTISO	CHANGE	CPUA	
OPERATOR	NOOPERATOR			OPERATOR				

This report contains a portion of the entries shown previously. However, it also shows the Logonid record fields that were changed in each entry. For example, in the first entry, the TRACE attribute was specified in the Logonid record for PAYJKL. Other entries can be read in a similar fashion. See the field descriptions in the next section, for a further explanation of how to interpret the additional fields reported under the DETAIL parameter.

ACFRPTLL - SAMPLE OUTPUT (DETAIL PARAMETER)

When both the DETAIL and UPDATE parameters are specified, the report appears as shown in the previous example; however, LAB, JESx, and LOGON validation updates are also included on the report. For example, entries for system validation updates are indicated by the word UPDATE in the CHANGE column of the report.

ACF2 UTILITY LIBRARY - ACFRPTLL - LOGONID MODIFICATION LOG - PAGE 10
DATE 12/18/84 (84.353) TIME 14.49 LOGONID MODIFICATION REPORT

DATE	TIME	LOGONID	JOBNAME	CHANGER	CHANGE	CPU	LAB	USING
FIELD		OLD VALUE			NEW VALUE			
84.352	12/17	12.12	PETETEST	PETE315	PETE315	INSERT	SKK1	USRPEG
	NAME	---NULLS---			EXPIRE TEST			
84.352	12/17	12.14	PETETEST	PETE315	PETE315	CHANGE	SKK1	
	PASSWORD	---NON PRINTABLE---			---NON PRINTABLE---			
84.352	12/17	12.15	PETE315	PETE315	PETE315	CHANGE	SKK1	
	EXPIRE	---NULLS---			12/17/84			
84.352	12/17	12.18	SSDRS01	MSTJCL00		UPDATE	SKK1	
		*** NO FIELDS CHANGED ***						
84.352	12/17	12.20	TSS123	TSSFAD	TSSFAD	INSERT	SKK1	TSSFAD
	RESTRICT	NORESTRIC			RESTRICT			
84.352	12/17	12.20	TSS123	TSSFAD	TSSFAD	CHANGE	SKK1	
	MAXDAYS	00000			00060			
	MINDAYS	00000			00060			

FIELD DESCRIPTIONS

The following text describes the fields of the ACFRPTLL reports:

DATE

The Julian and Gregorian date on which the update was made. The format of this date is either MM/DD or DD/MM based on ACF2 generation options.

TIME

The time at which the update was made.

LOGONID

The Logonid of the updated record.

JOBNAME

The name of the job under which the updates were made. If this is a TSO session then, in general, the jobname and the changer will be the same.

CHANGER

The Logonid of the user who issued the change request. This field will be JESx for batch job validation or MSTRJCL for LOGON validation records.

CHANGE

Indicates the type of update performed as per the following list.

- INSERT A new Logonid record has been inserted into the Logonid database.
- CHANGE An old Logonid has been changed.
- DELETE A Logonid has been deleted from the Logonid database.
- UPDATE Indicates that the Logonid was updated during LOGON or job validation processing. This type of record will appear only if the UPDATE option was specified for ACFRPTLL processing.

NOTE: The following change indicators are displayed only when the UPDATE parameter is specified.

- LAB-ADD Indicates that a LAB (Lookaside Buffer) entry was inserted from the CPU indicated in the LAB field to the CPU indicated in the CPU field.
- LAB-DELL Indicates that the system operator issued a command to delete one or more LAB entries based on the Logonid name. The LAB entries were deleted from the CPUid referenced in the CPU field.
- LAB-DELN Indicates that the system operator issued a command to delete one or more LAB entries based on the path or node used to create the LAB entry. The LAB entries were deleted from the CPUid referenced in the CPU field.
- LAB-DELS Indicates that the system operator issued a command to delete one or more LAB entries based on the input source-id of the entries. The LAB entries were deleted from the CPUid referenced in the CPU field.
- LAB-DELU Indicates that a LAB entry was deleted after all normal processing was completed, e.g., the job using the LAB entry ended. The LAB entry was deleted from the CPUid referenced in the CPU field.

CPU

The SMF CPU name from which the change was executed.

LAB

The SMF CPUid where the Logonid record originated. Valid only when LAB (Lookaside Buffer) support is being used to transmit ACF2 Logonid record information across CPUs.

Additional Fields for the DETAIL Parameter

An additional line of information will appear on the ACFRPTLL report when the user specifies the DETAIL parameter. This additional line contains the following fields:

FIELD

Name of the field of the Logonid record that was updated (via an ACF INSERT or CHANGE subcommand).

OLD VALUE

Value of the field before the update was made.

NEW VALUE

Value of the field after the update.

In certain instances, the values reported in the OLD VALUE and NEW VALUE fields will appear as follows when the DETAIL parameter is in effect:

- * If the value of a field is too long, it will be continued onto more than one line of the report.
- * If a field contains no value, then the message --- NULLS --- will appear as the value of the field.
- * If the value of a field cannot be reconstructed (as in the case of a password), then the message --- NON PRINTABLE --- will be reported for the value of that field.
- * If the user printing the report does not have authorization to list the value of a particular field, then the message AUTH-REQ'D will be reported for the value of the field.
- * If no fields have been changed, then the message *** NO FIELDS CHANGED *** will be reported for the old and new values of the field.

ACFRPTNV - THE ENVIRONMENT REPORT

The ACFRPTNV report generator produces loggings of each start (S ACF2), STOP or PURGE (P ACF2), and MODIFY (F ACF2) operator command issued. This report generator also produces logging of system IPLs and possible losses of SMF data. These events represent changes in the ACF2 security environment.

This report promotes auditing of the ACF2 security environment by allowing an installation to track the flow of security-related activities within the system. For instance, by noting the time between IPL and ACF2 startup, the installation can detect the passage of any significant period of time during which ACF2 did not have full control of system security. (ACF2 provides some control over job submission and data access even when stopped.) If ACF2 is stopped and then restarted, the installation should be able to document the reason for such an occurrence. Also, the installation must investigate any losses of SMF data (SMF record type 7), since the lost data may include loggings of access violations.

In addition, the logging of ACF2 modify commands allows an installation to review when ACF2 cross-reference tables, resident rules and directories in memory, and GSO records in the Infostorage database may have been altered. Such journalling also informs the installation of when ACF2 database backups have taken place.

FILES

ACFRPTNV uses the SYSPRINT, SYSIN, and RECxxxxx files, as described in the General Information chapter of this manual. Any number of SMF input files can be used--DISK, TAPE, or VSAM. However, concatenation of input files is not allowed.

PARAMETERS

ACFRPTNV accepts the following common parameters, which are described in the general information chapter of this manual:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SYSID, SELECT, and HEX.

CPUID(cpuid-mask) - DEFAULT=*** (all)

Indicates that the report should be limited to records written by the specified CPU(s). Only one individual CPUID or CPUID mask can be specified.

TRACE - DEFAULT=no tracing

Specifies the listing of all SMF records being written, including trace records written by the ACF2 main task. The default is for no report of trace records. ACFRPTNV does not provide a NOTRACE parameter.

DBLSPC/NODBLSPC - DEFAULT=NODBLSPC

Specifies double spacing of entries on the Environment report. The default is no double spacing.

HEADER - DEFAULT=no header lines for messages/replies

Specifies the printing of a header line before each report line for an ACF2 main task message or reply (WTO/WTOR console communications). The header line supplies the date, time, console, CPU, and system identifiers for a message or reply. The default is for no listing of header lines--only the listing of each ACF2 command and the listing of the text of any subsequent messages or replies. However, note that ACFRPTNV does not provide a NOHEADER parameter.

SORT SEQUENCE

The recommended sort sequence for the ACFRPTNV report is by CPU identification (major), and then timestamp. The installation can perform this sorting through the its own routine or by modifying and using the prototype JCL provided with the acf2/MVS product.

ACFRPTNV--SAMPLE OUTPUTACFRPTNV - Sample Output

Below is an example of the report issued by ACFRPTNV. The heading *** S M F D A T A L O S T *** appears with extra lines of information only if an SMF DATA LOST record (Type 07) is encountered in the SMF data stream. This example does not show any data lost:

ACF2 UTILITY LIBRARY - ACFRPTNV - ENVIRONMENT REPORT 1
DATE 06/15/84 (84.167) TIME 14.32

ENVIRONMENT	DATE	TIME	CPU	CID	ACF2 CMD RESULT
MVS SYSTEM IPL	84.159	06/07	06:15	CPUA	
ACF2 START	84.159	06/07	06:15	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='SHOW'	84.159	06/07	08:38	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='SHOWSYS'	84.159	06/07	09:11	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='REFRESH(LINKLST)'	84.159	06/07	10:54	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='REFRESH(LINKLST)'	84.159	06/07	10:57	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='RESET(PAYJSD)'	84.159	06/07	12:30	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='RESET(PAYJSD)'	84.159	06/07	12:30	CPUA 00	COMMAND SUCCESSFUL
ACF2 MODIFY PARAM='RESET(PAYJSD)'	84.159	06/07	12:30	CPUA 00	INVALID PARMS
ACF2 MODIFY PARAM='REFRESH(LINKLST)'	84.159	06/07	12:30	CPUA 00	COMMAND SUCCESSFUL
ACF2 STOP	84.159	06/07	21:14	CPUA 00	COMMAND SUCCESSFUL
MVS SYSTEM SHUTDOWN	84.159	06/07	06:15	CPUA	

FIELD DESCRIPTIONS

The following text describes the fields of the ACFRPTNV report:

ENVIRONMENT

The type of SMF or ACF2 record being written along with any commands or parameters/options specified. System SMF records can be either for system IPLs (SMF type 0) or for data lost (SMF type 7). ACF2 records can be for type ACF2 starts (S ACF2), ACF2 modifications (F ACF2), or ACF2 stops (P ACF2).

IPL DATE

Date on which the SMF or ACF2 record was written. The date appears in both Julian date format and in the format mm/dd.

TIME

Time at which the SMF record for the IPL was written. This time is in the format hh:mm.

CPU

Four-character identification of the CPU which journalled the SMF record.

CID

Identification of the console from which the logged event originated.

ACF2 CMD RESULT

The result of the ACF2 operator command issued. The printed message can be COMMAND SUCCESSFUL, COMMAND FAILED, or INVALID PARMS.

ACFRPTPP - THE PRE-PROCESSOR

The ACFRPTPP record pre-processor utility can reduce overhead when multiple report generators are executed using the same SMF data. Additionally, this report generator provides other functions.

ACFRPTPP accepts files of SMF records, and separates these records into intermediate files. Other report generators can use these intermediate files and thus avoid reading through all of the SMF records. In addition, each intermediate file can be sorted into a unique sequence appropriate for the intended report program.

In acf2/MVS Versions 3.1.5 and prior environments, ACFRPTPP identifies each ACF2 record type by its own unique record number (using the IBM-defined record type field of the SMF record). A one-to-one correspondence exists between each ACF2 record type and the installation-defined record number.

In environments of acf2/MVS Releases 4.0 and above, all ACF2 records are written under one single combined record number (in the IBM-defined record type field). Each ACF2 record contains a subtype indicator identifying the specific ACF2 record type.

For example, using the defaults provided with acf2/MVS Version 3.1.5, records for the ACFRPTRL report are identified by record number 223. Using the defaults provided with acf2/MVS Release 4.0 and above, these records are identified by record number 230 and subtype R.

However, ACFRPTPP can process records created under any release of acf2/MVS. Those records created under acf2/MVS Versions 3.1.5 and earlier are automatically converted by ACFRPTPP to the new record format just described for acf2/MVS Release 4.0 and above. The ability to process historical data has been maintained.

Conversion of Existing Installation-developed Programs. Your installation may have existing programs that utilize processed data from ACFRPTPP under acf2/MVS Version 3.1.5 or earlier. The ACFSMF macro may allow for easy conversion of these programs so that they can use the record formats for acf2/MVS Release 4.0 and above. See the acf2/MVS System Programmer's Guide for a description of the ACFSMF macro.

FILES

RECxxxxx

These ddnames identify the files containing the input SMF records, as described earlier in the general information chapter of this manual. ACFRPTPP accepts one SMF input file per ddname. Do NOT concatenate SMF input files.

SYSPRINT

ACFRPTPP uses the SYSPRINT file for message and summary report output. Sample summary reports can be found later in this section.

SMFxxxxx

ACFRPTPP creates intermediate files identified by ddnames that begin with the characters SMF. These files can be used by the installation to collect any combination of SMF records desired. The installation can then use these files as input to sort procedures, ACF2 report generators, and installation-developed programs.

The following, standard files are ACF2-defined:

<u>ddname</u>	<u>Description of File</u>
SMFAR	Rules database modification journal records
SMFCR	TSO command trace records
SMFDR1	Dataset access logging records
SMFDR2	Dataset access violation records
SMFDR3	Dataset access trace records
SMFDR4	Program access violation/logging records
SMFER	Infostorage database modification journal records
SMFJR	Logonid database modification journal records
SMFNR	Environment records
SMFPR	System entry violation records
SMFTR	RESTRICTed Logonid journal records
SMFVR	Generalized Resource Facility event journal records

The installation can define additional files by including one or more ddnames of the following formats:

SMF#nnn

This ddname format allows the user to specify the number of the particular SMF record type to be extracted. The ddname includes a number ranging from 0 to 255 that identifies the selected record type. For example, a file with the ddname SMF#0 collects type 0 records (System IPLs).

SMF\$xxxx

This ddname format allows the user to collect from 1 to 4 ACF2 SMF record subtypes in a single dataset. Each suffix letter in the ddname can specify a valid ACF2 SMF record subtype. Below are some example ddnames of this format along with a description of the corresponding ACF2 SMF record subtype:

SMF\$A - GSO initialization

SMF\$G - GSO record processing

SMF\$AG - Both GSO initialization and GSO record processing

SMFxxxxx

This ddname format allows a user to completely specify the contents of the file of extracted records. Any combination of 1 to 5 characters (excluding \$ or # as the first character) can be used to generate a unique ddname. Any ddnames for standard, ACF2-defined files (as listed in the table above) are reserved.

When using this ddname format during execution of ACFRPTPP, the user must also specify the SMFxxxxx(nnn/x,nnn/x,...,nnn/x,="description") report parameter, which is described below along with other report parameters. This parameter defines the records to be collected in the user-defined file.

The DCB attributes for both ACF2- and user-defined intermediate files are LRECL=32756,RECFM=VBS,BLKSIZE=3665. The BLKSIZE may be overridden.

Omission of any of the ddnames for any of these intermediate files causes processing of that file to be skipped. However, ACFRPTPP produces a summary of all input SMF records encountered, regardless of record number or applicability to an output file.

PARAMETERS

ACFRPTPP accepts the following parameters, as discussed in the general information chapter of this manual:

.LINECNT, SDATE, EDATE, STIME, ETIME, SYSID, TITLE, and HEX.

In addition, the following parameters are accepted by ACFRPTPP:

MASK(logonid-mask) - DEFAULT=***** (all)

This parameter specifies records being selected be limited to the Logonid, or Logonids, indicated by the Logonid mask(s). Use of this parameter allows the full set of ACF2 reports to be produced for an individual Logonid or a set of Logonids.

JOBMASK(job-mask,job-mask,...) - DEFAULT=***** (all)

The **JOBMASK** parameter specifies that records processed by ACFRPTPP be limited to those pertaining to the job(s) indicated by the jobname mask(s). Multiple jobname masks must be separated by commas or spaces. The default is to process records pertaining to all jobs.

SMFxx(nnn,nnn,...)

Each of these optional parameters define the record number(s) for SMF records used by the ACF2 report generators. Multiple record numbers in any single parameter must be separated by commas or spaces.

The following table shows each parameter name, default SMF record number (as specified in the @SMF macro of the ACF2-supplied default Field Definition Record (ACFFDR)), and record description. This table gives the default SMF record number(s) for both acf2/MVS Versions 3.1.5 and earlier and acf2/MVS Release 4.0 and above.

PARAMETER NAME	RECORD NUMBER	RECORD NUMBER	DESCRIPTION OF RECORD
	3.1.5 & Earlier	4.0 & Later	
SMFAR	223	230-R	Rules database modification journal record
SMFCR	225	230-T	TSO command trace record
SMFDR	221	230-D	Dataset access event journal record
SMFER	226	230-E	Infostorage modification journal record
SMFJR	222	230-L	Logonid database modification journal record
SMFPR	220	230-P	System entry violation journal record
SMFTR	224	230-J	Restricted Logonid trace record
SMFVR	227	230-V	Generalized resource event journal record
SMFNR		0,7 230-A 230-G	ACF2 environment record

Note: Records associated with each parameter name have their own record number under acf2/MVS Versions 3.1.5 and earlier. Under the defaults supplied with acf2/MVS Releases 4.0 and above, these records are all associated with record number 230 and an ACF2 subtype (denoted by an alphabetic letter).

The record numbers 220 through 227 shown are the old defaults under acf2/MVS Versions 3.1.5 and earlier.

For the parameters listed above, the name can be specified with characters SMF omitted (for example, AR(223,230)).

The default SMF record number for each ACF2 record type are specified in the @SMF macro of the ACF2FDR. (See the section on the ACF2 Field Definition Record in the acf2/MVS System Programmer's Guide.)

Alternatively, an installation can use the following parameter:

SMF\$x(nnn,nnn,...,nnn)

This parameter defines the record number(s) for ACF2 SMF record types written by all versions of acf2/MVS. For instance, SMF\$R(223,230) defines the record numbers for Rules database modification records produced by any release of acf2/MVS. This parameter is equivalent to specifying the parameter SMFAR(223,230).

Multiple record numbers in this parameter must be separated by commas or spaces. The characters SMF can be omitted from the name of this parameter (e.g., \$R(223,230)).

For specification of the combined record number (ACF2-defined default of 230), use of the SELECT parameter is preferred over this parameter. The SELECT parameter, described below, allows the combined record number to be defined for several ACF2 record types simultaneously.

SMFxxxxx(nnn/x,nnn/x,...,nnn/x,"description")

This parameter defines the contents of an intermediate output file with a ddname format of SMFxxxxx (described in the previous section). "xxxxx" corresponds with the last 1-5 characters of the ddname. For example, this parameter would be called SMFTEST if the corresponding ddname of the file is SMFTEST. In parentheses are the numbers or letter codes (separated by commas or spaces) that define which SMF record types will be collected in the intermediate file.

For example, SMFTEST(0,R,"TEST FILE") specifies a file of System IPL and Rules Database modification records. The description, TEST FILE, will appear on the ACFRPTPP summary report. This description can be up to 16 characters long.

Any pair of special characters can be substituted for the double quotes surrounding the file description. The placement of the second special character marks the end of the description. For instance, SMFTEST(0,R,"TEST" FILE) will result in the description TEST on the ACFRPTPP summary report. The remaining characters, FILE, will be ignored and considered as a comment. If omitted, the second special character will be treated as if it were placed just before the closing parenthesis. Please be careful in using single quotes as the special characters, since they may be confused with other delimiting single quotes within the JCL PARM field. Single

quotes should always be coded in pairs (i.e., a closing quote should always be coded).

This parameter name can be specified with the characters SMF omitted (e.g., TEST(0,R,=SHORT FORM)).

SELECT(nnn,nnn,...,nnn)

This parameter specifies the record number(s) associated with the combined record type (written by acf2/MVS Releases 4.0 and above). Multiple record numbers must be separated by commas.

By default, the combined record number is 230, as specified in the ACF2 field of the supplied @SMF macro in the ACF2 Field Definition Record (ACFFDR).

This parameter must specify the ACF2 combined record number that was in effect when the SMF records were written.

The defaults for the SMFxxxxx and SELECT parameters are in effect only when ACF2 is running or when the ACFFDR is available (loadable) on the CPU that is executing the ACFRPTPP utility. Note that if any SMF record number is specified through these parameters, then the numbers of all SMF records to be processed must be explicitly specified; no default record numbers will be in effect.

NOTE: Since the ACFRPTPP utility acts as a pre-processor for the other ACF2 report generators, the SDATE, EDATE, STIME, and ETIME parameters specified for ACFRPTPP may override the SDATE and EDATE parameters specified for the individual report generator. For example, an SDATE of 85010 for ACFRPTPP would mean that an SDATE of 85001 for a report generator would have no effect.

ACFRPTPP - SAMPLE OUTPUT

ACFRPTPP Sample Output

The sample output report below shows the report produced by the ACFRPTPP report generator. This report summarizes the record types for the various ACF2 SMF records and the number of records processed by ACFRPTPP.

ACF2 UTILITY LIBRARY - ACFRPTPP - SMF RECORD PRE-PROCESSOR - PAGE 1
DATE 08/01/84 (84.214) TIME 10.30

-- RECORD SELECTION SUMMARY - BY DDNAME ---

DDNAME	DESCRIPTION	COUNT	SELECTION
SMFAR	ACCESS	19	223-R,230-*
SMFCR	COMMAND	00	225-T,230-*
SMFDR	DATASET	2,540	221-D,230-*
SMFDR1	LOGGING	2,463	221-D,230-*
SMFDR2	VIOLATION	65	221-D,230-*
SMFDR3	TRACE	12	221-D,230-*
SMFDR4	PGMNAME	19	221-D,230-*
SMFER	RES JOURN	12	226-E,230-*
SMFJR	LOGONID	1,150	222-L,230-*
SMFNR	ENVIRON	04	A,G,00,07,230-*
SMFPR	PASSWORD	122	220-P,230-*
SMFTR	JOB TRACE	18	224-J,230-*
SMFVR	RES VIO	63	227-V,230-*

* - INDICATES ACF2 COMBINED SMF NUMBER

--- TOTAL RECORDS PROCESSED ---

READ=35,290 SELECTED=3,928 WRITTEN=6,487

ACF2 UTILITY LIBRARY - ACFRPTPP - SMF RECORD PRE-PROCESSOR - PAGE 2
DATE 08/01/84 (84.214) TIME 10.30

-- SMF RECORDS INPUT SUMMARY - BY DDNAME ---

DDNAME	----- STARTING ----->				<----- ENDING ----->				COUNT
	<----- PHYSICAL ----->		<----- LOGICAL ----->		<----- PHYSICAL ----->		<----- LOGICAL ----->		
	DATE	TIME	DATE	TIME	DATE	TIME	DATE	TIME	
REC0001	07/26/84	01.26	07/26/84	01.26	08/01/84	10.29	08/01/84	10.29	48,108

ACFRPTPP - SAMPLE OUTPUT (CONTINUED)

ACF2 UTILITY LIBRARY - ACFRPTPP - SMF RECORD PRE-PROCESSOR - PAGE 3
DATE 08/01/84 (84.214) TIME 10.30

-- SMF RECORDS INPUT SUMMARY - BY TYPE --

	--0--	--1--	--2--	--3--	--4--	--5--	--6--	--7--	--8--	--9--
0-	971	0	0	0	2223	1293	905	0	13	37
10-	6	28	0	0	0	0	0	0	0	0
20-	1935	195	14	99	0	0	2382	0	0	0
30-	6531	2	513	0	540	539	0	0	0	0
40-	15292	0	0	12	0	9	0	20	15	0
50-	0	0	0	0	0	11	0	24	9	0
60-	0	0	0	0	0	0	0	0	0	0
70-	93	93	1488	93	93	373	4186	91	0	0
80-	0	0	0	0	0	0	0	0	0	0
90-	35	0	0	0	0	0	0	0	0	0
100-	0	0	0	0	0	0	0	0	0	0
110-	0	0	0	0	0	0	0	0	0	0
120-	0	0	0	0	0	0	0	0	0	0
130-	0	0	0	0	0	0	0	0	0	0
140-	0	0	0	0	0	0	0	0	0	0
150-	0	0	0	0	0	0	0	0	0	0
160-	0	0	0	0	0	0	0	0	0	0
170-	0	0	0	0	0	0	0	0	0	0
180-	0	0	0	0	0	0	0	0	0	0
190-	0	0	0	0	0	0	0	0	0	0
200-	0	0	0	0	0	0	0	0	0	0
210-	0	0	0	0	0	0	0	0	0	0
220-	0	12	12	0	0	0	0	0	0	0
230-	7921	0	0	0	0	0	0	0	0	0
240-	0	0	0	0	0	0	0	0	0	0
250-	0	0	0	0	0	0	0	0	0	0
	--0--	--1--	--2--	--3--	--4--	--5--	--6--	--7--	--8--	--9--

FIELD DESCRIPTIONS

The fields of the ACFRPTPP report are described in terms of these sections:

- RECORD SELECTION SUMMARY - BY DDNAME -- This section lists, by ddname, each file which was provided for ACFRPTPP output (i.e., those ddnames which begin with SMF). For each file, the report provides a description, the number of records written into the file, and the corresponding SMF record number and/or ACF2 subtype of records requested for the file.

- TOTAL RECORDS PROCESSED -- This section shows the total number of records that have been: (1) read from all of the SMF input files (DD name formats of RECxxxxx); (2) selected from the input records for the purpose of output; and (3) written to all of the output intermediate files.

- SMF RECORDS INPUT SUMMARY - BY DDNAME -- This section shows the DD name of each SMF input file processed by ACFRPTPP. For each file, this section shows the physical starting and ending date, and time (i.e., both the date and time that the first record and the last record were written). This section also shows the logical starting and ending date, and time (i.e., both the date and time from the earliest and the latest records in the file).

- SMF RECORDS INPUT SUMMARY - BY TYPE -- This section shows the number of records read for each SMF record type. The types are identified by SMF record number (IBM record type field of the record). To interpret which record number corresponds with each total shown on the table, add the number to the left of the row where the total appears to the number at the top of the column where the total appears.

ACFRPTPW - INVALID PASSWORD/AUTHORITY LOG

ACF2 journals each unsuccessful attempt to gain access to the system along with the reason for the unsuccessful attempt. Also, if a user has the LOGSHIFT privilege on in his Logonid record and gains entry to the system outside of the shift/time controls, the access is logged to SMF and then reported on the ACFRPTPW report.

ACF2 security officers should monitor this report for excessive invalid password violations, invalid submission paths for RESTRICTed Logonids, and similar potential problems.

FILES

ACFRPTPW uses the standard SYSPRINT, SYSIN, and RECxxxxx I/O files.

PARAMETERS

The following parameters are accepted by ACFRPTPW as discussed in the General Information section of this manual:

| LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SYSID, SELECT, and HEX.

In addition the following parameters are accepted by ACFRPTPW:

MASK(logonid-mask) - DEFAULT=***** (all)

The MASK parameter for ACFRPTPW specifies an eight character Logonid mask to be compared against the Logonid that was incorrectly used for system access.

| JOBMASK (job-mask,job-mask,...) - DEFAULT=-

| The JOBMASK parameter specifies that records appearing on the
| report be limited to those pertaining to the job(s) indicated by
| the jobname mask(s). Commas must separate multiple masks. The
| default is records pertaining to all jobs.

SORT SEQUENCE

The recommended sort sequence for the ACFRPTPW report is by Logonid (major), date, and then time. The installation can perform this sorting through the its own routine or by modifying and using the prototype JCL provided with the acf2/MVS product.

(ACFRPTPW continued on the next page.)

ACFRPTPW - SAMPLE OUTPUTACFRPTPW Sample OutputACF2 UTILITY LIBRARY - ACFRPTPW - INVALID PASSWORD/AUTHORITY LOG - PAGE 1
DATE 03/07/82 (82.066) TIME 18.05

DATE	TIME	LID	JNAME	SUBMIT'R	SOURCE	PROGRAM	RC	L	CPU	LAB
82.066	03/07	07.24	AJBO	AJBO	P-LOGON	UCB-2C1		12	CPUA	
82.066	03/07	08.21	ACF1	JAVOLNK	JAVO	TR01T001	SUBMIT	4	CPUA	
82.066	03/07	09.25	SKA1	HAL1LIST	HAL1	TR02T005	SUBMIT	8	CPUA	CPUB
82.066	03/07	10.21	HAL2	HAL2JUNK		READER1		12	CPUA	
82.066	03/07	08.21	YSO	SMFDUMP	P-JOBCOPY	STC	*JOBCOPY	9	CPUA	
82.066	03/07	14.59	SSDDF	SSDDF	P-LOGON	UCB-2C2		61	CPUA	
82.066	03/07	16.50	SSDISO	SSDISO	P-LOGON	UCB-2C3		63	CPUA	
82.066	03/07	14.10	SSDJOB	SSDJOB	P-LOGON	UCB-2C4		135 *	CPUA	
82.066	03/07	15.13	SSDTJH	TJHJOB	P-LOGON	UCB-205		254 *	CPUA	

FIELD DESCRIPTIONS

The following discussion describes the fields in the ACFRPTW report:

DATE

The Julian and Gregorian date on which the job was validated. The format of this date is either mm/dd or dd/mm based on ACF2 generation options.

TIME

The time at which the job was validated. The validation date and time will generally differ from the reader date and time by only a fraction of a second.

LID

The Logonid under which the job was validated.

JNAME

The name of the job associated with the invalid system entry attempt.

SUBMIT'R

The Logonid that submitted the job using an invalid Logonid. The characters 'P-' preceding the SUBMIT'R field indicates that the job was submitted from a started task and that the name is a started task procedure name rather than a Logonid.

SOURCE

The logical input source through which the job was submitted.

PROGRAM

The name of the program that submitted the job using a Restricted Logonid. An asterisk ('*') preceding the name indicates that the program was APF authorized. The program name in this field is the name of the load module that did the actual submission of the job and may not be the same as the program specified in the JCL EXEC statement.

RC

This field indicates the Reason Code for which the access was denied or logged. The number corresponds to an ACF2 message number of the form ACF01xxx where the xxx is replaced by the reason code from the report (e.g., look up message ACF01004 for an RC of 4). Some common reason codes and the messages associated with them are listed in the following table.

4 LOGONID 111 NOT FOUND

8 UNAUTHORIZED INPUT SOURCE FOR LOGONID 111

9 LOGONID NOT VALID FOR SUBMISSION BY PROGRAM ppp

12 PASSWORD NOT MATCHED

13 LOGONID 111 SUSPENDED BECAUSE OF PASSWORD VIOLATIONS

17 PASSWORD FOR LOGONID 111 HAS EXPIRED

19 PASSWORD LESS THAN MINIMUM LENGTH

61 OUTSIDE OF SHIFT CONTROLS

63 SHIFT RECORD NOT FOUND

135 LOGSHIFT ALLOWED SYSTEM ACCESS

254 This return code is the result of MON-LOG being in the Logonid record. There is no message associated with this return code.

255 This return code is issued by the installation's New Password Exit (NEWPXIT). The message text and meaning are dependent upon the exit code.

For more information about these and other reason codes, see the acf2/MVS Messages Manual.

L

Indicates whether this entry represents an allowed access that generated a logging record, or whether this entry represents an attempt to access the system that has been denied. A "*" indicates allowed and logged access. A blank indicates that access was denied.

CPU

The SMF CPU identifier of the CPU where job validation occurred.

LAB

Indicates the SMF CPU identifier where the Lookaside Buffer (LAB) entry was created. Note that this field is displayed only when a LAB entry was used during system access validation.

ACFRPTRL - RULE-ID MODIFICATION LOG

ACF2 journals each update to the access rule data base for recovery purposes. This report formats some of the information in the SMF record to show recent update activity to the data base. The DETAIL parameter allows reporting of a before/after image if an access rule has been changed.

FILES

ACFRPTRL uses SYSPRINT, SYSIN, and RECxxxxx as described in the General Information section of this manual.

PARAMETERS

ACFRPTRL accepts the following general parameters:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SYSID, SELECT, and HEX.

In addition the following parameters are accepted by ACFRPTRL:

MASK(rulemask) - DEFAULT=***** (all)

The MASK parameter specifies an eight character mask used to select update reports for a specific dataset index or group of indices. The default causes all access rule update information to be output.

JOBMASK(job-mask,job-mask,...) - DEFAULT=-

The JOBMASK parameter specifies that records appearing on the report be limited to those pertaining to the job(s) indicated by the jobname mask(s). Commas must separate multiple masks.

SUMMARY - DEFAULT-SUMMARY

One line of information will appear on the report for each INSERT, CHANGE, or DELETE command that modified an access rule.

DETAIL - DEFAULT-SUMMARY

The DETAIL parameter will produce additional reporting for a modified access rule. Rule images will be printed in the same format as output from a DECOMP command. Only rule entries which have a difference between the BEFORE and AFTER images are printed; rule images which match are not.

BEFORE - DEFAULT= NONE

Additional reporting for a modified access rule will be produced in the BEFORE image only. A deleted rule set will be printed in its entirety.

AFTER - DEFAULT=NONE

Additional reporting for a modified access rule will be produced in the AFTER image only. A new rule set will be printed in its entirety.

ACFRPTRL - SAMPLE OUTPUT

ACF2 UTILITY LIBRARY - ACFRPTRL - RULE-ID MODIFICATION LOG - PAGE 1
DATE 07/31/78 (78.212) TIME 08.34

DATE	TIME	RULE-ID	JOBNAME	CHANGER	CHANGE	CPUID
		OLD VALUE			NEW VALUE	
78.212	07/31	08.03	SYS1	PTSTRT	PTSTRT	CHANGED CPU1
		LINKLIB	UID(APPL) R(A) E(A)		LINKLIB	UID(APPL) E(A)
		NEW.LPALIB	UID(SPG) R(A)-W(L) A(L)		NEW.LINKLIB	UID(SPG) R(A) W(L) A(L)
78.212	07/31	10.25	PROD01	PLMWER	PLMWER	INSERT CPU1
78.212	07/31	03.41	TEST05	PLMWER	PLMWER	DELETE CPU1

FIELD DESCRIPTIONS

The following paragraphs describe the fields in the output report:

DATE

The Julian and Gregorian date on which the update was made. The format of this date is either MM/DD or DD/MM based on ACF2 generation options.

TIME

The time at which the update was made.

RULE-ID

The key of the updated access rule set.

JOBNAME

The name of the job under which the updates were made. If this is a TSO session then, in general, the jobname and the changer will be the same.

CHANGER

The Logonid of the user who issued the update request.

CHANGE

Indicates the type of update performed as per the following list:

INSERT A new access rule set record has been inserted into the access rule data base.

REPLACE An old access rule set has been changed.

DELETE An access rule set has been deleted from the access rule data base.

CPU

The SMF CPU name from which the change was executed.

| **OLD VALUE**

| Contains the image of the rule before it is changed.

| **NEW VALUE**

| Contains the image of the rule after a change has occurred.

| This page intentionally left blank.

ACFRPTRV - GENERALIZED RESOURCE EVENT LOG

The generalized resource facility produces journal information based upon the results of resource validation requests. The ACFRPTRV utility processes the SMF records issued by the Generalized Resource Facility producing a report describing the nature of resource accesses, the user requesting the access, and the final disposition of the access.

There are three separate types of resource events that are logged by ACF2. These types are categorized as loggings, violations, and trace requests. It should be noted that many resource events are not logged if the resource rule specifies that the request be allowed. Trace requests occur when a security officer specifies that a specific user's access be journalled. When a user is being TRACEd, ACF2 performs normal resource validation processing and writes any logging or violation records; ACF2 then writes a TRACE record.

FILES

ACFRPTRV uses the standard SYSPRINT, SYSIN, and RECxxxxx files as discussed in the General Information section of this manual.

PARAMETERS

ACFRPTRV supports the following parameters discussed in the General Information section:

LINECNT, TITLE, SDATE, EDATE, STIME, ETIME, SELECT, SYSID, UID, and HEX.

In addition ACFRPTRV supports the following parameters:

LOG / VIO / TRACE / ALL - DEFAULT=ALL

This group of parameters specifies the type of records to be processed by ACFRPTRV. Any combination of these parameters may be specified. If none of these parameters are specified, the default of ALL is taken. These parameters act in an inclusive OR fashion, so that, for example, a specification of VIO and TRACE will cause all violation and trace records to be listed. The individual parameters are discussed in the following table.

TRACE Requests that all records produced as a result of the TRACE attribute being present in the Logonid record. If the access was LOGged or was a violation, TRACEd requests may have more than one SMF record written.

VIO Requests that all accesses which violated generalized resource rules be formatted.

LOG Requests that accesses which were allowed but the resource rule specified LOG be processed. LOGgings also occur when a security officer, non-cancellable, or read-only/non-cancellable Logonid issues a request that would normally have been prevented.

ALL Requests that all of the above types of journal information be listed.

| MASK(logonid-mask) - DEFAULT=(all)

| Requests that records for Logonids matching the specified mask be
| selected.

PRINTER - DEFAULT=terminal format

ACFRPTRV supports two different types of output format. The standard default format is designed to fit a limited width display terminal. The second format report, specified with the PRINTER option, is designed for output to a 133 column line printer.

TYPE(type-mask) - DEFAULT=*** (all)

Requests that only records for resources under the matching type code(s) be selected.

ID(id-mask) - DEFAULT=- (all)

Requests that only records with resource name(s) matching the id-mask be selected.

SORT SEQUENCE

The recommended sort sequence for the ACFRPTRV report is by Infostorage record key (major), Logonid, date, and then time. The Infostorage record key consists of the storage class, type code, and record name or key of the generalized resource rule set. The installation can perform this sorting through its own routine or by modifying and using the prototype JCL provided with the acf2/MVS product.

ACFRPTRV - SAMPLE OUTPUT (TERMINAL FORMAT)

ACFRPTRV - Sample Output (Terminal Format)

The following report example shows the terminal format output. The fields within the report are described later in this section.

ACF2 UTILITY LIBRARY - ACFRPTV - GENERALIZED RESOURCE LOG - PAGE 1
DATE 05/07/79 (79.127) TIME 15.35

REQUESTED RESOURCE				SOURCE			CPU MODULE		DISP	LOOKUP KEY				SERV
UID	DATE	TIME	JNAME	LID		NAME			DSP-MOD	KEY-MOD	INT	PST	FIN	
R-TAC-SYS00														
SYSAXYZ4				R01T0001	CPUA	LOGON	RULE		LOG R-TAC-SYS000					
79.127	05/07	08.25	XYZ4	XYZ4		XYZ INC			-	-	-	-	4	
									0	4	4	0	4	
R-TPR-IKJACCNT														
SYSAACF0				T2268B	CPUA	LOGON	NO-RULE		LOG R-TPR-IKJACCNT					
79.127	05/07	08.30	ACF0	ACF0		SECURITY OFFICER			-	-	-	-	4	
									0	0	8	0	4	
R-TPR-SYSAPL														
UEOLJHLO				B607T05	CPUA	LOGON	RULE		LOG R-TPR-SYSAPL					
79.127	05/07	08.55	JHLO	JHLO		LINCOLN, JOHN H			-	-	-	-	4	
									0	4	4	0	4	
R-CFC-INDEX														
UEOLJHL2			CICS001	CICS003	CPUA	CICS001	RULE		R-CFC-INDEX				UPDT	
79.127	05/07	09.10		JHL2		JOHNSON, MARIE L			-	-	-	-	16	
									0	0	16	0	16	

reqsrc				source			cpu module		disp	rec rulekey				serv
uid	gdate	time	jname	lid		name			dspmod	keymod	int	pst	fin	
jdate	gdate	time	jname	lid		name			dspmod	keymod	int	pst	fin	
									pre	rmc	int	pst	fin	

Descriptions of these report fields are listed following the next example (PRINTER format).

ACFRPTRV - SAMPLE OUTPUT (PRINTER FORMAT)

ACFRPTRV - Sample Output (Printer Format)

ACF2 UTILITY LIBRARY - ACFRPTV - GENERALIZED RESOURCE LOG - PAGE 1
DATE 05/07/79 (79.127) TIME 15.35 PRINTER

DATE	TIME	SOURCE	JNAME	LID	NAME	DISP	SERV	LOOKUP-KEY
PRE PST RMC INT FIN UID					CPU MODULE KEY-MOD DSP-MOD			REQUESTED RESOURCE
79.127	05/07	08.25	R01T0001	XYZ4	XYZ INC	RULE	LOG	R-TAC-SYS000
0 0 4 4 4			SYSAXYZ4	XYZ4	CPUA LOGON -			R-TAC-SYS000
79.127	05/07	08.30	T2268B	ACF0	SECURITY OFFICER	NO-RULE	LOG	R-TAC-1KJACCNT
0 0 8 0 4			SYSAACF0	ACF0	CPUA LOGON -	SEC-OFF		R-TAC-1KJACCNT
79.127	05/07	08.55	B607T05	JHL0	LINCOLN, JOHN H	RULE	LOG	R-TPR-SYSAPL
0 0 4 4 4			UEOLJHL0	JHL0	CPUA LOGON -			R-TPR-SYSAPL
79.127	05/07	09.10	CICS003	CICS001	JOHNSON, MARIE L	RULE	*VIO UPDT	R-CFC-INDEX
0 0 0 16 16			UEDLSHL2	JHL2	CPUA CICS001 -			R-CFC-INDEX
jdate	gdate	time	source	jname	name	disp	rec	rulekey
pre	pst	rmc	int	fin	uid	cpu	module	keymod
						dspmod	serv	reqsrc

This report output shows the ACFRPTV utility run with the PRINTER option (note the title field taken from the JCL parm string). The sample output is shown using the same input SMF records (the same access situations) as shown in the previous ACFRPTV report sample.

FIELD DESCRIPTIONS

The following describes the fields in the ACFRPTRV reports:

jdate

The Julian date on which the resource access was issued.

gdate

The Gregorian date on which the access was attempted. The format of this date is MM/DD or DD/MM/YY depending on ACF2 generation options.

time

The time of day when the access attempt occurred.

source

The logical input source from which the resource request was issued.

jname

The name of the job under which the access was issued.

lid

The Logonid of the user issuing the request.

name

The name of the user making the request.

disp

The name of the element that determined the disposition of this request as per the following table:

NO-REC A record matching the 'rulekey' could not be found in the ACF2 data base.

NO-RULE A rule matching the environment of the request could not be found in the rule set.

RULE The disposition was determined by a generalized resource rule in the rule set.

rec

A three character code indicating whether the record is a logging, violation, or trace record. Violation records are highlighted with an asterisk ('*') before this field.

rulekey

The name of the generalized resource rule set that was used to validate the request. This name shows any modifications to the resource name from the pre-validation exit or the ACF2 directory.

pre

The return code from the installation pre-validation exit.
Possible return codes are:

- 0 - Continue normal processing.
- 4 - Logonid (logonid) not found.
- 8 - Allow and log request.
- 12 - Allow request and reverify password.
- 16 - Allow, log request, and reverify password.
- 20 - Prevent request.

pst

The return code from the installation post-validation exit.
Possible return codes are:

- 0 - Continue normal processing.
- 4 - Allow request.
- 8 - Allow and log request.
- 12 - Allow request and reverify password.
- 16 - Allow, log request, and reverify password.
- 20 - Prevent request.

rmc

The return code from the ACF2 Generalized Resource Record Manager.
Possible return codes are:

- 0 - Record was already resident.
- 4 - I/O needed to obtain record.
- 8 - Record not found.

int

The return code from the generalized resource rule interpreter.
Possible return codes are:

- 0 - Allow request.
- 4 - Allow and log request.
- 8 - Allow request and reverify password.
- 12 - Allow, log request, and reverify password.

- 16 - Prevent access.
- 20 - No rule applies.
- 24 - Rule record not proper format.

fin

The final return code from the ACF2 generalized resource validation function. Possible return codes are:

- 0 - Allow.
- 4 - Allow and log.
- 8 - Allow request and reverify password.
- 12 - Allow request, log, and reverify password.
- 16 - Prevent request.

uid

The requestor's user identification string.

CPU

The SMF name of the CPU which validated this resource request.

module

The name of the requesting module as identified in the generalized resource request parameter list. This may be a user supplied name. Some possible ACF2 standard values are:

ACF\$Cxxx For CICS entries, the module name is the name of the ACF2/CICS parameter module for that CICS system.

IMS-AGN IMS Application Group Name validation.

IMS-TRAN IMS Transaction entered from a terminal.

IMS-TLNK IMS Transaction entered from an MSC Link.

IMS-PTP IMS Program-to-Program switch.

IMS-DL/I IMS DL/I call to a modifiable PCB.

IMS-/SET IMS /SET command entered.

IMS-/LCK IMS /LOCK command entered.

LOGON TSO Logon Processing validation.

keymod

ACF2 modifies the resource name to perform its database lookup operations. This field indicates what resource validation component modified the key.

PRE-VALD The installation pre-validation exit altered the request key.

DIRECTRY An ACF2 directory matched the key and modified it.

PREV/DIR Both the installation pre-validation exit and the ACF2 directory modified the request key.

SERV

The type of service requested. This field is only applicable for CICS file and IDMS area access resource loggings. The possible values are:

READ The access request was for read only.

ADD The access request was to add new records to an existing file.

DEL The request was to delete (erase) existing records.

UPDT The request was to modify existing records.

dspmod

The disposition can be modified by various exits and conditions. This field in the report details these conditions.

PRE-VALD An installation pre-validation exit altered the final request disposition.

PST-VALD An installation post-validation exit altered the final request disposition.

NON-CNCL The requestor Logonid was marked as non-cancellable so the request was allowed.

SEC-OFF The requestor was a security officer so the request was allowed.

ABORT The request was unconditionally aborted.

reqsrc

The name of the resource for which access is being requested.

ACFRPTRX - THE LOGONID ACCESS REPORT

The ACFRPTRX report generator produces a Logonid Access Report showing all dataset access rules or generalized resource rules that apply to a specific Logonid (LID) mask and/or a User Identification String (UID) mask.

For each LID/UID processed, ACFRPTRX searches the online Access Rules data base or the online Information Storage data base and compares the input UID string against the UID string of each rule entry. If the UIDs match, ACFRPTRX prints the rule key (\$KEY) and the entire rule entry. ACFRPTRX also determines if the LID/UID has authority to change (%CHANGE or %RCHANGE authority) any access rules or generalized resource rules and prints this information as part of the report.

Additionally, all accesses allowed because of special ACF2 privileges (such as the NON-CNCL, READALL, PREFIX, Security attributes, etc.) are highlighted by a message line and an access reason code. Also, if an LID/UID has no access authority or has been suspended, ACFRPTRX indicates these conditions with a descriptive message line. See the section entitled "ACFRPTRX Reason Codes and Message Lines" for more information.

The user running the report must have either the SECURITY, ACCOUNT, or AUDIT attribute to use the online ACF2 databases. Additionally, only those Logonid records and rule records which that user has normal ACF2 access to will be processed for that report.

NOTE: ACFRPTRX simulates normal ACF2 rule interpretation and checking but does not attempt to simulate the actions of installation dataset or generalized resource validation exits.

FILES

ACFRPTRX accepts input from several sources. The standard SYSIN file can be used to process one LID/UID mask pattern. The SYSUT1 and SYSUT2 files define work space for ACFRPTRX processing. For processing multiple LID/UID mask patterns, the SYSIDLST file can be used. All output from ACFRPTRX is written to the SYSPRINT file.

ACFRPTRX can be used to examine the access controls in place during a given time frame. Historical data can be specified as input through the LOGONIDS, RULES, and INFOSTG files. To use this facility, the appropriate ACF2 databases must be built from SMF backup files or from backup copies of the ACF2 databases. The ACFRECVR utility can be used to build these data bases (see the section entitled "ACFRECVR - The Recovery Utility" in this manual). Note that scope lists are not taken into consideration when ACFRPTRX processes input data from the LOGONIDS, RULES, or INFOSTG files.

SYSPRINT

The SYSPRINT file is used for message and report output. If ACFRPTRX is run in the time sharing environment and this file is not allocated, the report output will be directed to the TSO terminal. It is not recommended that ACFRPTRX be executed online under time-sharing (TSO), however, due to the amount of ACF2 database checking required.

SYSIN

The SYSIN file may be used to specify input parameter information for ACFRPTRX. ACFRPTRX accepts all parameter input from either the SYSIN file, the JCL parameter field, or both. The parameters specified in the SYSIN file supersede those specified in the JCL parameter field.

MVS Note: If ACFRPTRX is run in the time sharing environment (TSO) and the SYSIN file is not allocated to the TSO terminal, ACFRPTRX prompts for input parameter information at the terminal via the string "RX?".

SYSUT1

This is one of two scratch files required by ACFRPTRX. In batch, it can be specified with a DD statement such as:

```
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
```

Space allocation for this file is based on the number of ACF2 Logonid records specified by the UID and LID report parameters. If these parameters are not specified, space allocation, by default, is based on the total number of Logonid records in the database.

Also by default, the following DCB parameters are in effect:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,BUFNO=5)
```

SYSUT2

This scratch file stores the decompiled access or generalized resource rule sets for ACFRPTRX. In batch, this file can be specified with a DD statement such as:

```
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
```

Space allocation for this file is based on the number of lines that comprise the decompiled access or generalized resource rules.

By default, the following DCB parameters are in effect:

```
DCB=(RECFM=VBA,LRECL=85,BLKSIZE=3665,BUFNO=5)
```

SYSIDLST

The SYSIDLST file is an optional file and may be used to specify a list of LID masks or UID masks. Operation of ACFRPTRX for a single LID or UID may be specified in the JCL parameter field. The format of the SYSIDLST input statement is:

```
LID(lid-mask) UID(uid-mask)
```

LID(lid-mask) - DEFAULT=-

Specifies a Logonid mask. ACFRPTRX generates a Logonid Access Report for each Logonid which matches the mask pattern.

UID(uid-mask) - DEFAULT=-

Specifies a User Identification String (UID) mask. ACFRPTRX generates a Logonid Access Report for each UID which matches the mask pattern.

Either LID or UID must be specified through the LID and/or UID parameters or through the SYSIDLST file. Otherwise, no listing is produced.

MVS Note: If ACFRPTRX is run in the time sharing environment and the SYSIDLST file is not allocated or is allocated to the TSO terminal, then ACFRPTRX will prompt for input at the terminal with the string 'ID?'. ACFRPTRX should not be run online for large volume processing.

LOGONIDS

ACFRPTRX may be directed to use the current online ACF2 databases to make its determinations or, alternately, other prebuilt VSAM clusters can be provided to ACFRPTRX for this purpose. The LOGONIDS file should point to an alternate Logonid database. Valid only when NOACF2 is specified.

RULES

The RULES file is used in conjunction with the LOGONIDS file to have ACFRPTRX operate on historical data. The RULES file is valid

only when the DSET and NOACF2 parameters are specified. The RULES file should point to an alternate dataset access rules data base.

INFOSTG

The INFOSTG file is used in conjunction with the LOGONIDS file and directs ACFRPTRX to operate on historical data. INFOSTG is valid only when the RSRC and NOACF2 parameters are specified. The INFOSTG file should point to an alternate information storage data base.

PARAMETERS

ACFRPTRX accepts parameters from the JCL parameter field and the SYSIN file in combination. ACFRPTRX accepts the following parameters discussed in the General Information section of this manual: LINECNT, TITLE, and SYSID.

In addition, the following parameters are specific to ACFRPTRX:

ACF2/NOACF2 - DEFAULT=ACF2

If ACF2 is specified, ACFRPTRX uses the online ACF2 clusters. The ACF2 system must be active on this CPU for this type of processing.

If NOACF2 is specified, the report will be based on historical data provided by the RULES, LOGONIDS, or INFOSTG input files. Note that when historical data is specified as input, ACFRPTRX will not take scope lists into consideration when access authorization checking is performed.

NOTE: The user running the report must have either the SECURITY, ACCOUNT, or AUDIT attribute to use the online data sets. Additionally, only those Logonid records and rule records which that user has normal ACF2 access to will be processed for that report. Rule and Logonid information for rule records or Logonid records outside the scope of the requestor will never appear on the reports in "ACF2" mode.

DSET/RSRC - DEFAULT=DSET

If DSET is specified, ACFRPTRX processes dataset access rules (see also the RMASK parameter below). If RSRC is specified, ACFRPTRX processes generalized resource rules (see also the TYPE and RMASK parameters below).

LID(logonid-mask) - DEFAULT=-

Specifies a Logonid mask. ACFRPTRX generates a Logonid Access Report for each Logonid which matches the mask pattern. The LID parameter is required unless the UID parameter (described below) or the SYSIDLST file is used. If SYSIDLST is used to supply input parameters, any LID specification in the JCL parameter field is ignored. If neither the LID parameter nor the UID parameter is present in the JCL parameter field, ACFRPTRX expects to receive input from the SYSIDLST file.

UID(uid-mask) - DEFAULT=-

Specifies a User Identification String (UID) mask. ACFRPTRX generates a Logonid Access Report for each UID which matches the mask pattern. The UID parameter is optional when the LID parameter is specified. If both LID and UID are specified, they are processed using AND logic, meaning that only Logonids that match both the LID and UID mask patterns are processed. If SYSIDLST is used to supply input parameters, any UID specification in the JCL parameter field is ignored.

TYPE(type) - DEFAULT=none

(Required if RSRC is specified.) Specifies a three character generalized resource type. Only the specified generalized resource type will be included in the Logonid Access Report. TYPE is a required parameter valid only when the RSRC parameter is also specified.

RMASK(access-rule-mask/resource-rule-mask) - DEFAULT=none

This optional parameter specifies a mask for rule set keys, and may be used to produce a Logonid Access Report for a selected group of rule keys or a single rule key. When processing dataset access rules (DSET parameter), RMASK is a dataset rule key mask. For generalized resource processing (RSRC parameter), RMASK is a generalized resource name mask. When RMASK is specified, only those rules that match the specified mask are included in the report.

FORMAT AND FIELDS OF THE LOGONID ACCESS REPORT

The ACFRPTRX report has three sections: header, user information, and the Logonid Access Report.

Header Section

The header contains a standard ACF2 report title line, the execution date and time, the input parameters specified in the JCL parameter field, the number of Logonid records selected for processing, and the number of rules selected for processing. The report title is repeated on each page of the report.

ACF2 UTILITY LIBRARY - ACFRPTRX - LOGONID ACCESS REPORT - PAGE 1
DATE 02/11/82 (82.042) TIME 09.52

INPUT PARAMETERS: DSET LID(DOCDLW)

LID FILE PROCESSING COMPLETE, RECORDS SELECTED = 00001
RULE FILE PROCESSING COMPLETE, RECORDS SELECTED = 00227

User Information Section

For each LID/UID processed, ACFRPTRX prints a user information section. A sample user information section is illustrated below:

```
LID: DOCDLW   UID: SHS99DOCDLW
NAME: DEE WILLIAMS
```

Each field can be described as follows:

LID

Logonid being processed.

UID

User Identification String (UID) of the Logonid being processed.

NAME

The value stored in the NAME field of the LID being processed. Generally, this is the user's name.

Logonid Access Report Section

A Logonid access report is generated for each LID/UID processed. This section details all access rules that apply to the LID/UID being processed. Additionally, message lines may be printed at various points in the report. See the section entitled "ACFRPTRX Reason Codes and Message Lines" for more information.

A typical entry on the Logonid Access report is illustrated below followed by a description of each report field:

```
$KEY(DOC)
STORED: 04/20/83-14:28 BY: DOCMGR
$MODE(ABORT)
$PREFIX(DOCDATA)
$USERDATA(PROFILE)
%CHANGE *****DOCMGR
%RCHANGE *****DOCS
```

\$KEY

The key of the dataset access or generalized resource rule.

STORED

Date and time the rule was last stored. The format of this field is MM/DD/YY, DD/MM/YY, or YY/MM/DD, depending on ACF2 generator options.

BY

Logonid of the user that last stored the rule.

\$MODE

This line appears only when a \$MODE control card is stored with the access rule set. The \$MODE value is also displayed.

\$NOSORT

This line appears only when a \$NOSORT control card is stored with the access rule set.

\$PREFIX

This line appears only when a \$PREFIX control card is stored with the access rule set. The \$PREFIX value is also displayed.

\$USERDATA

The line appears only when a \$USERDATA control card is stored with the access rule set. The \$USERDATA value is also displayed.

%CHANGE

This line appears only when the LID/UID being processed is allowed to change the rule control cards and rule entries for the rule key (\$KEY). See also "ACFRPTRX Reason Codes and Message Lines".

%RCHANGE

This line appears only when the LID/UID being processed is allowed to change the rule entries for the rule key (\$KEY). See also "ACFRPTRX Reason Codes and Message Lines".

rule entry

Each rule entry in the rule set that applies to the LID/UID being processed. Possible fields that could appear in an access rule entry are:

dsn VOL(volser-mask) UID(userid-mask) -
LIB(library-name-mask) PGM(program-name-mask) -
UNTIL(date)/FOR(days) SHIFT(shift) -
SOURCE(source-name) DDN(ddn-mask) -
READ(A/L/P) WRITE(A/L/P) ALLOC(A/L/P) -
EXEC(A/L/P) USER(local-data) NEXTKEY(nextkey)

Possible fields that could appear in a generalized resource rule entry are:

UID(uid) SHIFT(shift) SOURCE(source) -
SERVICE(READ,ADD,UPDATE,DELETE) DATA(data) -
UNTIL(date)/FOR(days) VERIFY ALLOW/LOG/PREVENT

ACFRPTRX REASON CODES AND MESSAGE LINES

ACFRPTRX prints message lines when access to a dataset or generalized resource is allowed because of "special" ACF2 access authorization. Each message line appears with an access code indicating the reason why access is allowed. Message lines are also generated if the LID/UID being processed has been suspended, cancelled, or cannot access any datasets or generalized resources. The access reason codes are explained below followed by a description of each possible message line:

- NC - Non Cancellable - The Logonid is non-cancellable by ACF2 (this user has the NON-CNCL attribute in his Logonid record).
- O - Owner - The user's "owned dataset prefix" (Logonid record PREFIX field) matches the high level index for the dataset in question. This code is valid only for dataset processing.
- RA - Read-only/Non-cancellable - This Logonid has the READALL attribute and is not cancellable by ACF2 as long as the dataset is being opened for input (read only). This code is valid only for dataset access processing.
- SC - Scoped Security Officer - The Logonid is that of a scoped security officer whose DSNSCOPE matches the high level index of the dataset (this user has SECURITY and DSNSCOPE or SCPLIST in his Logonid record).
- SE - Security Officer (unscoped) - The Logonid is that of an unrestricted security officer (this user has the SECURITY attribute and no DSNSCOPE or SCPLIST limits in his Logonid record).

The possible message lines and their meanings are:

**** USER HAS ACCESS TO ALL DATASETS AS: SE , NC
APPEARS: After the "NAME:" line in DSET mode.
MEANS: User is an unrestricted Security Officer, has the NON-CNCL attribute, or both.

**** USER HAS READ ACCESS TO ALL DATASETS AS: RA
APPEARS: After the "NAME:" line in DSET mode.
MEANS: User has the READALL attribute.

**** USER HAS ACCESS TO NO DATASETS
APPEARS: After the "NAME:" line in DSET mode.
MEANS: The user did not match any rules or %CHANGE entries, and does not have any special ACF2 access authority. This user can not access any datasets.

**** USER HAS ACCESS TO ALL DATASETS FOR THIS KEY AS: O , SC
APPEARS: After display of \$KEY for the rule set.
MEANS: User's PREFIX matched the rule key and/or the user's scope matched the rule key.

**** USER HAS ACCESS TO ALL RESOURCES AS: SE , NC
APPEARS: After the "NAME:" line in RSRC mode.
MEANS: User is either an unrestricted Security Officer or has the NON-CNCL attribute.

**** USER HAS ACCESS TO NO RESOURCES
APPEARS: After the "NAME:" line in RSRC mode.
MEANS: The user did not match any rules or %CHANGE entries, and does not have any special ACF2 access authority. This user can not access any resources.

**** USER HAS ACCESS TO ALL RESOURCES FOR THIS KEY AS: SC
APPEARS: After display of \$KEY for the rule set.
MEANS: User's scope matched the rule key.

**** USER CAN CHANGE RULE
APPEARS: After display of a %CHANGE entry.
MEANS: User's UID string matched one or more masks in the %CHANGE entry which gives the user authority to change both the control cards and rule entries in the rule set.

**** USER HAS %CHANGE, BUT ALSO HAS NO-STORE
APPEARS: After display of a %CHANGE card.
MEANS: User's UID string matched the change mask, but the user also has the NO-STORE attribute.

**** LID: lid UID: uid * CANCELLED
APPEARS: After LID/UID line for the user being processed.
MEANS: This LID/UID has been cancelled.

**** LID: lid UID: uid * SUSPENDED
APPEARS: After LID/UID line for the user being processed.
MEANS: This LID/UID has been suspended.

**** NO LID/UID FOUND TO MATCH SYSIDLST PARMS
APPEARS: After display of SYSIDLST parameters.
MEANS: LID/UID combination not found on the SYSUT1 file.

**** USER CAN CHANGE ANY OF THE RULE ENTRIES
APPEARS: After display of %RCHANGE control card.
MEANS: User's UID matched one or more masks in the %RCHANGE entry which gives the user authority to change all rule entries in rule set. However, the user cannot change any of the control cards in the rule set.

ACFRPTRX - SAMPLE OUTPUT (DATASET LOGONID ACCESS)

This JCL produces the sample Logonid Dataset Access Report illustrated below. The report shows all dataset access rules that apply to the Logonid "DOCDLW".

```
/** THIS JOB PRODUCES A DATASET LOGONID ACCESS REPORT FOR
/** LOGONID DOCDLW
/**
/**REPORT EXEC PGM=ACFRPTRX,REGION=640K,
//      PARM='DSET,LID(DOCDLW)'
/**
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(2,0)),DCB=BUFNO=30
//SYSUT2   DD  UNIT=SYSDA,SPACE=(CYL,(2,0)),DCB=BUFNO=30
```

On the sample report, three rule keys (\$KEY) and one message line are displayed. According to the sample report, the Logonid "DOCDLW" has access to all datasets stored under the rule key of "DOCDLW" as the owner - 0. Additionally, two rule entries stored under the key of "DOCGDS" allow "DOCDLW" to access the dataset "DOCGDS.LIB.TEXT" and one rule entry stored under the rule key of "DOCJLS" allows "DOCDLW" to access the dataset "DOCJLS.JIMSLIB.TEXT".

ACF2 UTILITY LIBRARY - ACFRPTRX - LOGONID ACCESS REPORT - PAGE 1
DATE 02/11/82 (82.042) TIME 09.53 DSET,LID(DOCDLW)

INPUT PARAMETERS: DSET LID(DOCDLW)

LID FILE PROCESSING COMPLETE, RECORDS SELECTED = 00001
RULE FILE PROCESSING COMPLETE, RECORDS SELECTED = 00009

LID: DOCDLW UID: SHS99DOCDLW
NAME: DEE WILLIAMS

\$KEY(DOCDLW)
**** USER HAS ACCESS TO ALL DATASETS FOR THIS KEY AS: 0
STORED: 10/01/81-09:14 BY: DOCDLW
AVL0024.TEXT UID(SH***DOC) READ(A) EXEC(A)
DEESLIB.TEXT UID(SH***DOC) READ(A) WRITE(A) EXEC(A)
DEESLIB.TEXT UID(SH) READ(A) EXEC(A)

\$KEY(DOCGDS)
STORED: 08/06/81-09:09 BY: DOCGDS
LIB.TEXT UID(SH***DOC) READ(A) WRITE(A) EXEC(A)
LIB.TEXT UID(SH) READ(A) EXEC(A)

\$KEY(DOCJLS)
STORED: 09/01/81-15:35 BY: DOCJLS
JIMSLIB.TEXT UID(SH***DOC) READ(A) WRITE(A) EXEC(A)

ACFRPTRX - SAMPLE OUTPUT (GENERALIZED RESOURCE LOGONID ACCESS)

This JCL produces the sample generalized resource Logonid Access report shown below for the Logonid "DOCDLW".

```
//SAMPLE JOB 1,'ACFRPTRX REPORT',MSGCLASS=A
/**
/** THIS JOB PRODUCES A GENERALIZED RESOURCE LOGONID ACCESS
/** REPORT FOR LOGONID DOCDLW
/**
//REPORT EXEC PGM=ACFRPTRX,REGION=640K,
// PARM='RSRC,LID(DOCDLW),TYPE(CKC)'
/**
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(2,0)),DCB=BUFNO=30
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(2,0)),DCB=BUFNO=30
```

According to the sample report, the Logonid "DOCDLW" has access authority for two TYPE(CKC) resources.

ACF2 UTILITY LIBRARY - ACFRPTRX - LOGONID ACCESS REPORT - PAGE 1
DATE 02/11/82 (82.042) TIME 13.34

INPUT PARAMETERS: RSRC LID(DOCDLW) TYPE(CKC)

LID FILE PROCESSING COMPLETE, RECORDS SELECTED = 00001
RULE FILE PROCESSING COMPLETE, RECORDS SELECTED = 00008

LID: DOCDLW UID: SHS99DOCDLW
NAME: DEE WILLIAMS

\$KEY(ABC*) TYPE(CKC)
STORED: 01/12/82-22:42 BY: CICSISO
UID(SH) ALLOW

\$KEY(C1C2C3**) TYPE(CKC)
STORED: 10/13/81-15:28 BY: CICSISO
UID(SH) ALLOW

OTHER PROCESSING OPTIONS

There are numerous ways to specify input parameters.

For example, the following parameter statement causes ACFRPTRX to produce a Logonid Access Report for all rule entries stored under the key "SYS1" that apply to any Logonid beginning with "DOC":

```
//      PARM='DSET,LID(DOC-),RMASK(SYS1-)'  
...(other JCL statements)
```

To process a list of LIDs/UIDs, use the SYSIDLST input file. The following example combines SYSIDLST and the JCL parameter field to process three different LID/UID patterns.

```
//      PARM='DSET'  
...(other JCL statements)  
//SYSIDLST DD *,DCB=BLKSIZE=80  
LID(DOCTMS) UID(***1)  
LID(TSSTRD) UID(***2)  
LID(NPD*** ) UID(***3)
```


ACFRPTSL - SELECTED LOGONID LIST

The ACFRPTSL utility provides a listing of all Logonid records (users defined to ACF2) which match the set of selection criteria specified in the report (JCL) parameters. It provides the ability to copy all Logonid records or to select and print (in long or short formats) specific Logonid records. The IF parameter allows for flexible record selection criteria and the SFLDS parameter allows for flexible field printing and editing.

FILES

ACFRPTSL uses the standard SYSPRINT, SYSIN and RECxxxxx files as described in the General Information section of this manual. Note: RECxxxxx is only used if INPUT(SMF) is specified. The following files are also used:

BACKUP

This is the ddname for type BKUP input. This is a VB format dataset produced by the ACF2 automatic backup facility.

SAVEREC

This is the ddname for the copy function output file. This copy function will only occur if this file has been defined. The DCB options are LRECL=1236 and RECFM=VB. The BLKSIZE parameter may be specified in the JCL; the default is BLKSIZE=3665. The format of the output record is dependent upon the input type. If the input type is SMF, the format is ACF2 Logonid modification SMF records. If the input type is ACF2 or BKUP, the format is the same as ACF2 Logonid database records.

SYSLIB

The SYSLIB file defines a partitioned dataset that contains the ACFDDLID definition module. This module defines the fields in the ACF2 Logonid record for ACFRPTSL IF statement processing. Note: The ACFDDLID module is produced during ACFFDR generation from the @CFDE macro entries.

PARAMETERS

Parameters will be taken from both the JCL parm string and the SYSIN file. If both formats are used, they will be combined for processing. ACFRPTSL accepts the following parameters described in the General Information section of this manual: LINECNT, TITLE, SDATE, EDATE, SELECT and HEX. The following parameters are specific to ACFRPTSL:

MASK(LIDMASK) - DEFAULT=*** (all)**

The mask parameter is used to select a particular Logonid or group of Logonids for processing. The default specifies that all Logonids be processed. The mask parameter makes its selections before the "IF" parameter, so that if a Logonid is not selected here it never reaches the "IF" parameter processing.

UPDATE/NOUPDATE - DEFAULT=NOUPDATE

The update parameter, when specified, requests a summary of Logonids modifications including any JESx and logon validation updates. The default of NOUPDATE causes only updates other than validation updates to be listed. This default was chosen because of the volume of validation updates (one for every job and TSO session). This parameter is used only if INPUT(SMF) was specified.

INPUT(type) - DEFAULT=(none)

This parameter specifies the type of input to be processed. The subparameter must be one of the following keywords designating the file attributes. Omission of this parameter will result in a message being printed and immediate termination of ACFRPTSL.

SMF Input is accepted as ACF2 Logonid modification SMF records.

BKUP Input is accepted as a VB format file as produced by the ACF2 automatic backup facility.

ACF2 Input is accepted as unformatted records from the ACF2 Logonid database. Note: To use the ACF2 Logonid data base for input requires that the person executing the report generator have a SECURITY, ACCOUNT or AUDIT attribute. In addition, only those Logonid records that the person running the report has normal ACF2 authority to access will be included in the report output. Thus, if a user requested all Logonids with the IMS attribute, but had a UIDSCOPE which allowed him only to access Logonids records for users in a specific department, then the report output would only list those Logonids for users with the IMS attribute and who were in that department.

REPORT(type) - DEFAULT=(none)

This parameter specifies the format of the report issued by ACFRPTSL. The subparameter must be one of the following keywords. Note: omission of this parameter will result in a message being printed and immediate termination of ACFRPTSL.

SHORT This format is a one-line summary consisting of the Logonid, name, date, time, and changer (if INPUT SMF specified), and it easily fits an 80 character screen width. This format can be extended (multiple lines if needed) to include other fields from the Logonid record by the use of the SFLDS parameter.

FULL This display format is a replica of the output from the ACF command list subcommand, and fits on a 133 character print line.

NONE This subparameter specifies that no report be generated. This can be useful if ACFRPTSL is going to be used as a copy utility for Logonid records (see SAVEREC under FILES).

SFLDS(field-list) - DEFAULT=(none)

This parameter is valid only if the REPORT(SHORT) parameter is specified. The field-list contains the external field names (as defined in the ACFEFD @CFDE entries) which correspond to the names of fields of the selected Logonid records that are to be formatted along with the basic short format fields. The fields are formatted in the order specified and headings are produced for each field. If too many fields are specified for one line, then multiple lines will be generated. If the field-list has to be continued on more than one line, a hyphen must be placed at the end of the line; also, the second and subsequent lines must start in column 1.

IF(field-name-operators) - DEFAULT=(none)

This parameter is formatted similar to a high level programming language IF statement and allows for the definition of flexible (though complex) record selection criteria. The "variables" available for processing are the various Logonid record field names and/or constants (defined below). The available "operators" are also defined below. The full IF expression is evaluated as an algebraic expression yielding a "true" (i.e., select record) or "false" (i.e., bypass record) value. If the result of the IF expression is a quantity, then "non-zero" is considered "true" (selected) and a zero value is considered "false" (not selected). Parentheses can be used to group expressions to override the normal precedence order.

IF Expression "Constants": When a constant (specific value) needs to be specified in the IF statement, the following formats may be used:

<u>Format</u>	<u>Contents</u>	<u>Type</u>
'aaaa' or C'aaaa'	Alphanumerics	Character fields
nnnn	Numerics	Binary number fields
X'xx'	Hex numbers	Hex fields
B'n'	1 or 0	Bit (flag) fields ("1"="on", "0"="off")
P'nn'	Numerics	Packed decimal fields
D'mm/dd/yy' or D'dd/mm/yy' or D'yy/mm/dd'	Numerics with dividing slashes (which format used is based on local system option). See note below.	Date field

Note: Date field could be a TOD clock field or stored as packed decimal in the Logonid record, but is displayed on outputs as a date, with or without a time. Time-of-day fields will be treated as date fields only (no comparison on the time portion of the field) by the IF processor.

IF Expression Operators (in order of precedence) are displayed on the following page.

<u>Precedence</u>	<u>Character</u>	<u>Symbol</u>	<u>Meaning</u>
1.	NOT	■	NOT
2.	OR		OR
3.	AND	&	AND
4.a.	EQ	=	EQUAL
b.	NE	■=	NOT EQUAL
c.	LE	<=	LESS THAN OR EQUAL TO
d.	GE	>=	GREATER THAN OR EQUAL TO
e.	LT	<	LESS THAN
f.	GT	>	GREATER THAN
5.		-	designates negative value
6.a.		*	TIMES
b.		/	DIVIDED BY
7.a.		+	PLUS
b.		-	MINUS
8.			Concatenated to (i.e., used between field names, in sequence, to show concatenation of fields). Note this can be used to build a UID string, since "UID" is not an actual field in the Logonid record and thus cannot be referenced directly.

Either the symbols given or the character abbreviation (where available) is acceptable to the program.

Only one IF statement will be accepted by the program. However multiple criteria may be specified in one statement. If the criteria needs to be continued on more than one line, a hyphen must be placed at the end of the line; also, the second and subsequent lines must start in column 1. For example, to select all Logonid records for users whose system access count (ACC-CNT) is greater than 1000 and who are either Security Officers or Auditors, the IF statement would be:

```
IF((ACC-CNT > 1000) AND-
  (SECURITY OR AUDIT))
```

ACFRPTSL - SAMPLE OUTPUT (SHORT FORMAT)

The following example illustrates the short format of the report as it would appear at the terminal:

ACF2 UTILITY LIBRARY - LOGONID SUPERLIST REPORT - PAGE 1
DATE 08/14/83 (83.227) TIME 12.54 -

LOGONID	NAME	DATE	TIME	CHANGER	ACC-CNT	STC
ACFSTCID	DEFAULT STC ID	08/14/83	12:18		15,032	YES
#PROD	RESTRICTED PROD ID	08/14/83	12:48		27,767	NO
JOB	STC JOB	08/14/83	12:49		18,205	YES

ACFRPTSL - SAMPLE OUTPUT (FULL FORMAT)

The following example illustrates the full format of the report:

ACF2 UTILITY LIBRARY - ACFRPTSL - LOGONID SUPERLIST REPORT - PAGE 1
DATE 08/14/80 (80.227) TIME 12.07 -

AAAAO ACSAAAO ANDERSON,ARTHUR A
SITE(A) DEPARTMT(CS)
ACCESS ACC-CNT(0) ACC-DATE(00/00/00) ACC-TIME(00:00)
MISCELLANEOUS PREFIX(AAAO) TSOCMDS(ACFCLUSR)
TSO CMD-LONG INTERCOM JCL LGN-ACCT LGN-PROC LGN-TIME
PMT-ACCT PROMPT TSOACCT(1234) TSOPROC(AALOGONA)
TSORGN(512) TSOSIZE(1,024) TSOUNIT(SYSTSO) VLD-ACCT
VLD-PROC WTP
STATISTICS PSWD-DAT(00/00/00) PSWD-TOD(06/06/78-22:42)
PSWD-VIO(0) SEC-VIC(0) UPD-TOD(04/23/80-12:55)

AAA1 BTAAAA1 ALBERTS,ANDREW A
SITE(B) DEPARTMT(TA)
ACCESS ACC-CNT(143) ACC-DATE(07/02/80) ACC-TIME(17:30)
MISCELLANEOUS PREFIX(AAA1) TSOCMDS(ACFCLUSR)
TSO CHAR(BS) CMD-LONG DFT-PFX(AAA1) INTERCOM JCL LGN-ACCT
LGN-PROC LGN-TIME PMT-ACCT PROMPT TSOACCT(1234)
TSOPROC(AALOGONB) TSORGN(512) TSOSIZE(1,234)
TSOTIME(10) TSOUNIT(SYSTSO) VLD-ACCT VLD-PROC WTP
STATISTICS PSWD-DAT(04/22/80) PSWD-TOD(03/27/80-13:25) PSWD-VIO(2)
SEC-VIO(0) UPD-TOD(07/02/80-17:30)

AAA3 CCEAAA3 AARDVARK,ALICE A
SITE(C) DEPARTMT(CE)
ACCESS ACC-CNT(0) ACC-DATE(00/00/00) ACC-TIME(00:00)
MISCELLANEOUS PREFIX(AAA3) TSOCMDS(ACFCLUSR)
TSO CMD-LONG INTERCOM JCL LGN-ACCT LGN-PROC LGN-TIME
PMT-ACCT PROMPT TSOACCT(999) TSOPROC(AALOGONA)
TSORGN(512) TSOSIZE(1,234) TSOUNIT(SYSTSO)
VLD-ACCT VLD-PROC WTP
STATISTICS PSWD-DAT(06/18/80) PSWD-TOD(06/06/78-22:43)
PSWD-VIO(1) SEC-VIO(0) UPD-TOD(06/18/80-10:48)

FIELD DESCRIPTIONS

The fields in the short format of the ACFRPTSL report are:

LOGONID

The Logonid of the user.

NAME

The name of the user.

DATE

The Julian and Gregorian date on which the last update was made. The format of this date is either MM/DD/YY, DD/MM/YY, or YY/MM/DD, based on ACF2 generation options.

TIME

The time at which the update was made.

CHANGER

The Logonid of the user who initiated the change request. This field will be JES2 or JES3 for batch job validation or MSTRJCL for logon validation records. Note: This field will only have an entry if INPUT(SMF) was specified.

FIELD1, FIELD2...

These are the fields that were specified in the SFLDS parameter. For definitions of these fields, refer to the @CFDE entries of the Field Definition Record and/or the acf2/MVS Implementatio Planning Guide and the General Information Manual. (Note: In the example above only two fields were specified, ACC-CNT and STC).

ACFRPTXR - THE CROSS-REFERENCE REPORT

ACFRPTXR provides a facility for determining which users will have access to a specified dataset or generalized resource based on standard ACF2 security controls. For each dataset or generalized resource requested, ACFRPTXR creates a list of applicable rules, displays all the Logonids of users who have access to that dataset or generalized resource, and also displays the reason that Logonid has access to the dataset or generalized resource.

ACFRPTXR Exit Considerations

ACFRPTXR does not support any installation dataset or generalized resource validation exits, except in the case of alternate rule selection. This situation usually arises when an installation dataset pre-validation exit changes the rule key used to validate access to a dataset. ACFRPTXR has parameters to simulate this situation where desired. Other results displayed are subject to alteration due to local exit coding.

ACFRPTXR Processing

For each dataset or generalized resource processed by ACFRPTXR, a report will be generated detailing the input parameters specified and information about the applicable rule set. ACFRPTXR checks each rule entry on an individual basis and compares the User Identification String (UID) in each rule entry to the UID of each Logonid record. If the UIDs match, the rule is considered applicable to the Logonid being processed and the entire rule is printed along with the Logonid.

Whenever an applicable rule is found, ACFRPTXR prints the rule. If the LID parameter is specified then ACFRPTXR will also list all Logonids that have access to the dataset or generalized resource being processed, either because of the UID string mask in the rule or because of any special ACF2 authorizations. The reason for each Logonid gaining access to the dataset or generalized resource is listed with that Logonid as described below.

Note that those users with special authorization to access the dataset or generalized resource are listed with each applicable rule. If the UID mask in the rule includes all users (a UID mask of UID(-) or none specified), ACFRPTXR will print only a single message indicating that all users have the indicated access authority. Likewise, if a totally masked dataset name (-) is used, the report will show that read access is allowed for all users specified.

If no rules in the rule set apply or the rule set that would be associated with this dataset or generalized resource is not found, then ACFRPTXR prints a list of Logonids that have access to the dataset or generalized resource through other special ACF2 authorizations (e.g., because of NON-CNCL, SECURITY, READALL, or PREFIX authorities) as described on the next page.

ACFRPTXR Access Reason Codes

For each Logonid listed, ACFRPTXR details the reasons why that Logonid has access to the dataset or generalized resource in question. This information is listed as a series of codes in parentheses following the Logonid. These codes can be listed in any combination and will denote all modes of access that would apply.

NOTE: If "U" is the only mode of access, then it is not listed (no data will appear in parentheses following that Logonid). If codes do appear, they are the only ones which apply (e.g., if (NC,SE) appears, the person has access because of NON-CNCL and SECURITY attributes but does not have access via the rule itself).

The possible codes are:

- NC - Non Cancellable - The Logonid is non-cancellable by ACF2 (this user has the NON-CNCL attribute in his Logonid record).
- O - Owner - The user's "owned dataset prefix" (Logonid record PREFIX field) matches the high level index for the dataset in question. This code is valid only for dataset processing.
- RA - Read-only/Non-cancellable - This Logonid is not cancellable by ACF2 as long as the dataset is being opened for input (read only). This code is valid only for dataset access processing.
- SC - Scoped Security Officer - The Logonid is that of a scoped security officer whose DSNSCOPE matches the high level index of the dataset (this user has SECURITY and DSNSCOPE or SCPLIST in his Logonid record).
- SE - Security Officer (unscoped) - The Logonid is that of an unrestricted security officer (this user has the SECURITY attribute and no DSNSCOPE or SCPLIST limits in his Logonid record).
- U - UID match - The UID string in the access rule set matches the UID of this user. If only this code applies, no codes are specified in parentheses.

INPUT/OUTPUT FILES

ACFRPTXR accepts input from several sources. The standard SYSIN file can be used to process one dataset name or generalized resource. The SYSUT1 file defines work space for dataset access processing. The SYSUT2 file defines work space for generalized resource processing. For processing multiple datasets, the SYSDSLST file can be used. For processing multiple generalized resource types, the SYSRSLST can be used. All output from ACFRPTXR is written to the SYSPRINT file.

ACFRPTXR can be used to examine the access controls in place during a given time frame. Historical data can be specified as input through the LOGONIDS, RULES, and INFOSTG files. To use this facility, the appropriate ACF2 databases must be built from SMF backup files or from backup copies of the ACF2 databases. The ACFRECVR utility can be used to build these databases (see the section entitled "ACFRECVR - The Recovery Utility" in this manual). Note that scope lists are not taken into consideration when ACFRPTXR processes input data from the LOGONIDS, RULES, or INFOSTG files.

SYSPRINT

The SYSPRINT file is used for message and report output. If ACFRPTXR is run in the time sharing environment and this file is not allocated, the report output will be directed to the TSO terminal.

SYSIN

The SYSIN file may be used to specify input parameter information for ACFRPTXR. ACFRPTXR accepts all parameter input from either the SYSIN file, the JCL parameter field, or both. The parameters specified in the SYSIN file supersede those specified in the JCL parameter field.

MVS Note: If ACFRPTXR is run in the time sharing environment (TSO) and the SYSIN file is not allocated to the TSO terminal, ACFRPTXR prompts for input parameter information at the terminal via the string "XR?".

SYSUT1

This is one of two scratch files required by ACFRPTXR. In batch, it can be specified with a DD statement such as:

```
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
```

Space allocation for this file is based on the number of ACF2 Logonid records specified by the UID and LID report parameters. If these parameters are not specified, then the space allocation, by default, is based on the total number of access or generalized resource rules on the respective database.

Below are some space allocation estimates:

<u>Device</u>	<u>Records in Storage per Buffer</u>	<u>Records in Storage per Cylinder</u>
IBM 3330	220	4180
IBM 3350	323	9690
IBM 3380	554	8310

SYSUT2

This scratch file stores the decompiled access and generalized resource rule sets for ACFRPTXR. In batch, it can be specified with a DD statement such as:

```
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
```

Space allocation for this file is based on the number of access or generalized resource rules as follows:

<u>Device</u>	<u>Records in Storage per Buffer</u>	<u>Records in Storage per Cylinder</u>
IBM 3330	296	4440
IBM 3350	433	12990
IBM 3380	743	11145

Note that the RECFM and BLKSIZE DCB parameters do not have to be specified, since RECFM is undefined and BLKSIZE is dynamically optimized for the particular device being used.

SYSDSLST

The SYSDSLST file is an optional file and may be used to specify a list of dataset names to be processed by ACFRPTXR. This file is used only when the DSET input parameter is selected. Operation of ACFRPTXR for a single dataset may be specified in the JCL parameter field.

If the DSN parameter is not specified on the JCL parameter field, ACFRPTXR expects to process the SYSDSLST input file. With this facility a list of multiple dataset names may be processed. This type of processing is useful when all the datasets in the system are requested. Large volume requests should not be done online. The format of the SYSDSLST input statement is:

```
dsn [vol] RKEY(key)
```

Where the dsn and vol parameters are positional (i.e., must appear in this sequence and before RKEY if present).

dsn - The dsn parameter is mandatory and specifies a fully qualified dataset name to be processed by ACFRPTXR (no quotes).

vol - This parameter is optional and is the name of the volume on which the dataset resides. If omitted then all volser specifications in the access rule set are ignored (all match).

RKEY(key) - This parameter is optional and may be used to specify an alternate access rule key to be used instead of the dataset high level index.

Note: If ACFRPTXR is run in the time sharing environment and the SYSDSLST file is not allocated or is allocated to the TSO terminal, then ACFRPTXR will prompt for input at the terminal with the string 'DS?'. This file may not be specified if the DSN, RKEY, or VOL parameters are specified in the JCL parameter field. For batch jobs, if neither the JCL parameter field, SYSIN file, nor SYSDSLST are present, the job terminates with an error message.

SYSRSLST

The SYSRSLST file is an optional file which may be used to specify a list of generalized resource names. The SYSRSLST file is valid only when used in conjunction with the RSRC input parameter. If only one resource name is to be processed, then specify the resource name in the TYPE and NAME parameter fields. The format of SYSRSLST input is as follows:

TYPE(type) NAME(name)

Where the type and name parameters are positional (i.e., must appear in this sequence, if present).

TYPE(type) - The three character generalized resource type to be processed. For example, "TYPE(IAG)" represents generalized resource rule sets for IMS transactions.

NAME(name) - The key under which the rule is stored. For example, "TYPE(IAG) NAME(ACFR)" specifies that an access report be produced for the IMS transaction named "ACFR". The NAME keyword may be specified as "NAME(-)" resulting in a cross-reference report entry for each resource name stored under a particular generalized resource type.

Note: If ACFRPTXR is run in the time sharing environment and the SYSRSLST file is not allocated or is allocated to the TSO terminal, then ACFRPTXR will prompt for input at the terminal with the string 'RS?'. This file may not be specified if the TYPE and NAME parameters are specified in the JCL parameter field. For batch jobs, if neither the JCL parameter field, SYSIN file, nor SYSRSLST are present, the job terminates with an error message.

LOGONIDS

ACFRPTXR may be directed to use the current online ACF2 databases to make its determinations or, alternately, other prebuilt VSAM clusters can be provided to ACFRPTXR for this purpose. The LOGONIDS file should point to an alternate Logonid database. Valid only when NOACF2 is specified.

RULES

The RULES file is used in conjunction with the LOGONIDS file to have ACFRPTXR operate on historical data. The RULES file is valid only when the DSET and NOACF2 parameters are specified. The RULES file should point to an alternate dataset access rules database.

INFOSTG

The INFOSTG file is used in conjunction with the LOGONIDS file and directs ACFRPTXR to operate on historical data. INFOSTG is valid only when the RSRC and NOACF2 parameters are specified. The INFOSTG file should point to an alternate information storage data base.

PARAMETERS

ACFRPTXR accepts parameters from the JCL parameter field and the SYSIN file in combination. ACFRPTXR accepts the following parameters discussed in the General Information section of this manual: LINECNT, TITLE.

In addition, the following parameters are specific to ACFRPTXR:

ACF2/NOACF2 - DEFAULT=none (specification of this parameter is required.)

If ACF2 is specified, ACFRPTXR uses the online ACF2 clusters. The ACF2 system must be active on this CPU for this type of processing.

If NOACF2 is specified, the report will be based on historical data provided by the RULES, LOGONIDS, or INFOSTG input files. Note that when historical data is specified as input, ACFRPTXR will not take scope lists into consideration when access authorization checking is performed.

NOTE: The user running the report must have either the SECURITY, ACCOUNT, or AUDIT attribute to use the online data sets. Additionally, only those Logonid records and rule records which that user has normal ACF2 access to will be processed for that report. Rule and Logonid information for rule records or Logonid records outside the scope of the requestor will never appear on the reports in "ACF2" mode.

LID/NOLID - DEFAULT=LID

LID indicates that ACFRPTXR is to create a cross-reference report and list all of the Logonids that have access to the specified dataset or generalized resource.

NOLID suppresses listing of Logonids. Only the applicable dataset or generalized resource rule sets are listed.

RRSUM/NORRSUM - DEFAULT=RRSUM

If RRSUM is specified, the additional Rule Record Summary portion of ACFRPTXR is produced at the end of the report. This includes an entry for each rule record (e.g., high level index, \$KEY value, or resource TYPE/NAME combination) used in producing the report. This portion of the report is also where the detailed Logonid lists for each %CHANGE and/or %RCHANGE record encountered are displayed (assuming the LID option is also specified). Thus, where the message "%CHANGE DATA EXISTS" or "%RCHANGE DATA EXISTS" appears in the main part of the report after the RULE KEY line, the related specific UID/LID entries are printed in the Rule Record Summary part of the report.

DSET/RSRC - DEFAULT=DSET

If DSET is specified, ACFRPTXR processes dataset access rules. The DSN, RKEY, and VOL input parameters may be provided through either the JCL parameter field or the SYSDSLST input file. See also the DSN, RKEY, and VOL parameters below.

If RSRC is specified, ACFRPTXR processes generalized resource rules. The TYPE and NAME input parameters may be provided through either the JCL parameter field or the SYSRSLST input file. See also the TYPE and NAME parameters below.

DSN(dsn) - DEFAULT=none

This parameter may be specified to cause ACFRPTXR to act on a single dataset name without the need for the SYSDSLST file. This parameter is valid only when the DSET parameter is also specified and cannot be used in conjunction with the SYSDSLST input file. The name specified must be fully qualified (regardless of time sharing option) but must not be specified in quotes. The dataset name high level index will be used as the key to identify the applicable access rule set to be used unless the RKEY parameter (see below) is also specified. In the case where a full rule set for a particular \$KEY is desired, the DSN field must be defined as "DSN(-)" and the applicable \$KEY value defined in the RKEY parameter (see below).

RKEY(key) DEFAULT=none

This parameter is valid only when the DSET parameter is also specified. RKEY has two uses:

1. RKEY(key) can be used in conjunction with the DSN parameter to specify the key of the rule set to be used to validate the dataset access. This is similar to the concept of using the ACF2 dataset pre-validation exit to perform the same function at run time. It is only necessary to specify RKEY when some rule record other than the one under the dsn high level index is to be used for rule checking.
2. RKEY(key) can be used in conjunction with a DSN parameter of "-" (hyphen) when you wish to list all the rule entries for a particular key.

VOL(volser) - DEFAULT=none

An optional parameter for use with DSN and specifies the volume serial number of the volume on which the dataset resides. If no volume serial is specified, then all volume information in the access rule set is ignored (volume masks specified in the rules will all be considered as matches). VOL is valid only when the DSET parameter is also specified.

TYPE(type) - DEFAULT=none

Specifies the three character generalized resource type to be processed. TYPE is valid only when the RSRC and NAME parameters are also specified. If the TYPE and NAME parameters are not specified in the JCL parameter field, the SYSRSLST file will be used for input. TYPE and NAME can not be specified in both the JCL parameter field and the SYSRSLST file.

NAME(name) - DEFAULT=none

Specifies the name of the resource to be processed. NAME is valid only when used in conjunction with the RSRC and TYPE parameters. NAME may be specified as "-" to process every NAME present within the specified TYPE. If the TYPE and NAME parameters are not specified in the JCL parameter field, the SYSRSLST file will be used for input. TYPE and NAME can not be specified on both the JCL parameter field and the SYSRSLST file.

FIELD DESCRIPTIONS

The following fields appear in the report output:

DATASET

The fully qualified dataset name for which a report was requested.

RULE KEY

The key under which the rules were searched. This would normally be the same as the \$KEY value of the generalized resource type or dataset access rule. For dataset access rules; however, the rule key may not be equivalent to the \$KEY value, because the RKEY parameter option can specify that an alternate \$KEY be used to test the access criteria for a dataset. In this case, the rule key shown on the report is the value specified in the RKEY parameter.

STORED

The date and time this rule record was last changed. The format of this field is either MM/DD/YY, DD/MM/YY, or YY/MM/DD, based on ACF2 generation options.

CONTROLS

Indicates all of the control cards that are contained in the rule set. Possible entries include %CHANGE, %RCHANGE, \$MODE, \$PREFIX, \$NOSORT, \$OWNER, or \$USERDATA. %CHANGE and %RCHANGE users are listed in the Rule Record Summary portion of the report. Note that for the \$MODE, \$PREFIX, and \$OWNER control cards, the value of each is also listed in the CONTROLS line.

BY

The Logonid of the user who last changed (stored) this rule record.

rule entry

The individual matching rule entry as it appears in the rule record. Possible fields which could appear in an access rule entry are:

dsn VOL(volser-mask) UID(userid-mask) -
LIB(library-name-mask) PGM(program-name-mask) -
UNTIL(date)/FOR(days) SHIFT(shift) -
SOURCE(source-name) DDN(ddn-mask) READ(A/L/P) -
WRITE(A/L/P) ALLOC(A/L/P) EXEC(A/L/P) -
USER(local-data) NEXTKEY(nextkey)

Possible fields that could appear in a generalized resource rule entry are:

UID(uid) SHIFT(shift) SOURCE(source) -
SERVICE(READ,ADD,UPDATE,DELETE) DATA(data) -
UNTIL(date)/FOR(days) VERIFY ALLOW/LOG/PREVENT

lid1 through lidn

The individual Logonids which have access to the specified dataset or generalized resource, either because they match the UID in the rule (U or nothing after the Logonid) or because of other special ACF2 privileges (NC, O, RA, SC, or SE after the Logonid). The listing of individual Logonids matching each rule entry is optional (see LID/NOLID option).

(rc)

The reason codes specifying why ACF2 would allow this Logonid to have access to this dataset or generalized resource. If a code does not appear on the report, it means the user has access only as specified by that rule. Possible codes are:

O Owner ("owned data set" PREFIX matches).

NC Non-cancellable (NON-CNCL attribute).

- RA Read-only/non-cancellable Logonid (READALL attribute). This reason code applies only to dataset access rules.
- SC Scoped Security Officer (SECURITY attribute and matching DSNSCOPE or SCPLIST value).
- SE Unrestricted Security Officer (SECURITY attribute and no DSNSCOPE or SCPLIST value).
- U UID match (user's UID string matches rule UID field). Not listed if only condition met.

ACFRPTXR - SAMPLE JCL (DATASET ACCESS CROSS-REFERENCE)

The JCL shown below produces the sample dataset access cross-reference report illustrated on the next page. Note that the "Rule Record Summary" report is included following the dataset access cross-reference report.

```
//SAMPLE JOB 1,'ACFRPTXR REPORT',MSGCLASS=A
//*
//* THIS JOB PRODUCES THE ACFRPTXR CROSS-REFERENCE REPORT FOR
//* DATASET SYS1.LINKLIB
//*
//REPORT EXEC PGM=ACFRPTXR,REGION=640K,
// PARM='DSET,ACF2,RRSUM,DSN(SYS1.LINKLIB)'
//*
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(2,0)),DCB=BUFNO=30
```

On the sample report, five rule entries were found that applied to the specified dataset, SYS1.LINKLIB. Each rule entry and its applicable authorities, conditions, and list of matching Logonids is displayed. Information entries will also appear for each new high level index (after the RULE KEY line) whenever the rule set contains a %CHANGE, %RCHANGE, \$NOSORT, \$OWNER, \$MODE, \$PREFIX, or \$USERDATA entry.

Additional information about the %CHANGE and %RCHANGE users appears in the optional Rule Record Summary portion of the report. Also, if the input request included a volume name and a volume name rule also exists (@volser.VOLUME), this information will be indicated after the RULE KEY line.

ACFRPTXR - OTHER PROCESSING OPTIONS (DATASET ACCESS CROSS-REFERENCE)

There are numerous ways to specify input parameters. For example, the following parameter statement will produce an access cross-reference report for every "SYS1" dataset:

```
//      PARM='DSET,RRSUM,DSN(-),RKEY(SYS1),ACF2'  
...(other JCL statements)
```

To process a list of datasets, use the SYSDSLST input file. The following example combines SYSDSLST and the JCL parameter field to process three datasets:

```
//      PARM='DSET,ACF2'  
...(other JCL statements)  
//SYSDSLST DD *,DCB=BLKSIZE=80  
USER01.FILE1  
USER01.FILE2  
USER01.FILE3
```

ACFRPTXR - SAMPLE OUTPUT (DATASET ACCESS CROSS REFERENCE)

ACF2 UTILITY LIBRARY - ACFRPTXR - CROSS REFERENCE REPORT - PAGE 1
DATE 01/18/82 (82.018) TIME 14.06 SAMPLE OF XR REPORT

DATASET: SYS1.LINKLIB
STORED: 01/15/82-16:41 BY: NPDLLV
CONTROLS: %CHANGE DATA EXISTS %RCHANGE DATA EXISTS \$MODE(ABORT)

LOGONIDS THAT HAVE ACCESS WITHOUT RULES
ACFSTCID(NC) ACFUSER(SE) ADMJM(SE) BACKUP(NC) CICSCVT(NC,SE)
NET(NC) NPDLLV(SE) SECOFF(SE) TONE3(NC,SE) TSO(NC)
TSSISO(NC) TSSJD(NC)

LINKLIB UID(SHM**SSD) LIB(LINKLIB) PGM(HMASMP) READ(A) WRITE(L) EXEC(A)
SSDBSS(U,NC) SSDDCF SSDDCF1 SSDFAS(U,NC) SSDKGS SSDLP

LINKLIB UID(SHM**TSS) LIB(LINKLIB) PGM(HMASMP) READ(A) WRITE(L) EXEC(A)
TSSLP TSSRJT

LINKLIB UID(SHM**TSS) READ(A) WRITE(L) EXEC(A)
TSSLP TSSRJT

LINKLIB UID(SHS**TSS) LIB(LINKLIB) PGM(HMASMP) READ(A) WRITE(L) EXEC(A)
TSSDF TSSDF# TSSFAS TSSJD(U,NC) TSSJD1 TSSJEC
TSSNJG TSSPK TSSRMC

ACFRPTXR - SAMPLE RULE RECORD SUMMARY (DATASET ACCESS CROSS REFERENCE)

ACF2 UTILITY LIBRARY - ACFRPTXR - RULE RECORD SUMMARY - PAGE 1
DATE 01/18/82 (82.018) TIME 14.06 SAMPLE OF XR REPORT

DATASET KEY: SYS1
STORED: 01/25/82-16:41 %CHANGE DATA BEING PROCESSED

LOGONIDS THAT UPDATE THIS RULE WITHOUT ANY %CHANGE OR %RCHANGE
ACFUSER(SE) SECOFF(SE) SSDISO(SE) SSDJHS(U,SE)
CICSBAT(NC,SE) SSDGTF(NC,SE)

%CHANGE *****SSDLH
SSDLH(U,NC,SE) SSDLH1(U,NC)

LOGONIDS THAT CAN UPDATE RULE ENTRIES IN THIS RULE

%RCHANGE *****SSD
SSDCK SSDDMP SSDFAS

ACFRPTXR - SAMPLE JCL (GENERALIZED RESOURCE CROSS-REFERENCE)

The JCL shown below produces the sample generalized resource report illustrated on the next page. Note that the "Rule Record Summary" report is included following the generalized resource cross-reference report.

```
//SAMPLE JOB 1,'ACFRPTXR REPORT',MSGCLASS=A
/**
/** THIS JOB PRODUCES THE ACFRPTXR CROSS-REFERENCE REPORT FOR
/** GENERALIZED RESOURCES
/**
//REPORT EXEC PGM=ACFRPTXR,REGION=640K,
// PARM='RSRC,ACF2,RRSUM,TYPE(CKC),NAME(ACFR)'
/**
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(2,0)),DCB=BUFNO=30
```

On the sample report, two rule entries were found that applied to the specified generalized resource, the CICS transaction named ACFR. Each rule entry and its applicable authorities, conditions, and list of matching Logonids is displayed.

Additional detailed information about the %CHANGE users appears in the optional Rule Record Summary portion of the report.

ACFRPTXR - OTHER PROCESSING OPTIONS (GENERALIZED RESOURCE CROSS-REFERENCE)

There are numerous ways to specify input parameters. For example, the following parameter statement will produce an access cross-reference report for every generalized resource name which is stored under resource type "CKC".

```
//      PARM='RSRC,RRSUM,TYPE(CKC),NAME(-)'  
...(other JCL statements)
```

cp 11 To process a list of generalized resource types, use the SYSRSLST file. The following example combines SYSRSLST and the JCL parameter field to process three types of generalized resources:

```
//      PARM='RSRC,ACF2'  
...(other JCL statements)  
//SYSRSLST DD *,DCB=BLKSIZE=80  
TYPE(CKC) NAME(ACFR)  
TYPE(IAG) NAME(UPDATE)  
TYPE(TAC) NAME(PROCS)
```

ACFRPTXR - SAMPLE OUTPUT (GENERALIZED RESOURCE CROSS-REFERENCE)

ACF2 UTILITY LIBRARY - ACFRPTXR - CROSS REFERENCE REPORT - PAGE 1
DATE 01/18/82 (82.018) TIME 14.06 XR REPORT

RESOURCE TYPE: CKC RESOURCE NAME: ACFR

RULE KEY: RCKCACFR
STORED: 09/21/81-09:07 BY: NPDLLV
CONTROLS: NO %CHANGE DATA

LOGONIDS THAT HAVE ACCESS WITHOUT RULES

ACFSTCID(NC)	ACFUSER(SE)	ADMJM(SE)	BACKUP(NC)	CICSCVT(NC,SE)
NET(NC)	NPDLLV(SE)	SECOFF(SE)	TONE3(NC,SE)	TSO(NC)
TSSISO(NC)	TSSJD(NC)			

UID(SHM) ALLOW

ADMJM(U,SE)	NPDLLV(U,SE)	SSDBSS(U,NC)	SSDDCF	SSDDCF1
TSSLP	TSSRJT	USRVCA		

UID(SH**CTSS) ALLOW
NO LOGONIDS MATCH

ACFRPTXR - RULE RECORD SUMMARY (GENERALIZED RESOURCE CROSS-REFERENCE)

ACF2 UTILITY LIBRARY - ACFRPTXR - RULE RECORD SUMMARY - PAGE 1
DATE 01/18/82 (82.018) TIME 14.06 XR REPORT

RESOURCE KEY: RCKCACFR
STORED: 09/21/81-09:07 RULE USED, NO %CHANGE DATA

LOGONIDS THAT CAN UPDATE THIS RULE
ACFUSER(SE) ADMJM(SE) NPDLLV(SE) SECOFF(SE) SSDISO(SE)

INSTALLATION EXITS FOR REPORT GENERATORS

ACF2 provides installation exits for its report generators for specialized selection functions. The exit is taken just prior to printing the record. One use of these exits is to check authorization of the user before printing the record. Another possibility, in installations with several security officers, is to select the records pertinent to the security officer running the job.

The various exits must be link-edited into the appropriate report generator load modules. All ACF2 report generators are reentrant; if the installation exit is not reentrant then the combination must be linked without the RENT attribute.

The exit linkage follows standard OS conventions. Register 15 has the entry point address; register 14 has a return address; register 13 points to a savearea; and register 1 points to a parameter list. The exit must save and restore the registers upon return.

The parameter list passed to the exit is two words, with the high order bit in the second word always on to indicate the end of the parameter list. The only exception is ACFRPTSL, which has a three word parameter list. Correspondingly, the high order bit in the third word is always on to indicate the end of its parameter list. Entries in a parameter list are:

- +0 - Address of SMF record being processed (for all report generators except ACFRPTSL) or the address of the Logonid record (for ACFRPTSL).
- +4 - Address of user's ACUCB (ACF2 User Control Block) or zero if not available. The exit may use the ACUCB to obtain the user's Logonid or UID string.
- +8 - Address of the original input record (applies to ACFRPTSL only). The original record could be an SMF record, a backup file record, or a record from the Logonid data set.

Upon return, the exit must set a return code in R15 indicating the action to be taken. A return code of zero (0) indicates that the record should be printed. A non-zero return code will cause the record to be skipped.

The following table lists the ACF2 report generators that have exits, the entry points for the report generators, and the names of the exit csects.

<u>Report Generator</u>	<u>Entry Point</u>	<u>Exit Csect Name</u>
ACFRPTCR	ACF46RCR	ACFRCRXT
ACFRPTDS	ACF44RDS	ACFRDSXT
ACFRPTTEL	ACF47REL	ACFRELXT
ACFRPTJL	ACF43RJL	ACFRJLXT
ACFRPTLL	ACF42RLL	ACFRLLXT
ACFRPTNV	ACF51RNV	ACFRNVXT
ACFRPTPW	ACF41RPW	ACFRPWXT
ACFRPTRL	ACF45RRL	ACFRRLXT
ACFRPTRV	ACF48RRV	ACFRRVXT
ACFRPTSL	ACF49RSL	ACFRSLXT

DATABASE RECOVERY

ACF2 provides the facilities necessary for recovery of its three primary VSAM clusters in the event they become inaccessible or unusable. Each installation should review, modify, and test these JCL procedures as necessary. Note: the backup and recovery procedures outlined below describe one method of ensuring the recoverability of ACF2's VSAM data bases. However, it is the responsibility of each installation to devise and test recovery methods which are suitable to their particular environment.

PRE-PLANNING FOR RECOVERY

During the installation process, the DEFINE job should be modified to reflect the dataset names of the primary and alternate clusters. These names should be synchronized with the names specified in the @DDSN macro in the ACFFDR and also with the ACFRECVR and ACFBKUP procedures located in SYS1.PROCLIB. Additionally, space for both the primary and alternate ACF2 VSAM clusters should be initialized.

An alternate COMMNDxx member should be inserted into SYS1.PARMLIB that does not perform an automatic start of ACF2.

ACF2 RECOVERY FEATURES

The ACF2 VSAM clusters can be corrupted for several reasons: corruption of the volume on which the clusters reside; VSAM error caused by a new PTF; a hardware error while the clusters are being updated; or various other causes. However, even in such cases, a system with ACF2 can be IPLed. If ACF2 becomes non-functional because the VSAM clusters are not available, simply IPL the system specifying an alternate COMMNDxx member from SYS1.PARMLIB that does not automatically start ACF2.

ACF2 provides several facilities to aid in the backup and recovery of the primary VSAM clusters. These include an automatic backup facility, the @DDSN macro of the ACFFDR, the GSO BACKUP record, the creation of an SMF record for each modification made to an ACF2 database, the ACFBKUP procedure, and the ACFRECVR procedure.

The Automatic Backup Facility

At the installation's option, the ACF2 system task will make a copy of each primary VSAM cluster onto sequential datasets once per 24 hour period. This facility is activated by specifying the appropriate information in the GSO BACKUP record of the Infostorage database and the dataset names for both the primary and alternate clusters in the @DDSN macro of the ACF2FDR. (The BACKUP record is described in the chapter on GSO records in the acf2/MVS Administrator's Guide. The @DDSN macro is explained in the chapter on the ACF2 Field Definition Record in the acf2/MVS System Programmer's Guide.)

Optionally, the operator can initiate backup processing by issuing an "F ACF2,BACKUP" command through any system console at any time.

The ACF2 backup process includes copying each of the three primary VSAM clusters into a "scratch" dataset. This insures that a complete copy of each cluster can be made before overwriting the permanent backup datasets. The permanent backup datasets should not be created on the same volume on which the primary VSAM clusters reside. Procedures should be established for the retention of backup copies for a reasonable period of time. It should be noted that these backup copies can also be used as input to certain report generators. For example, previous versions of the backup files may be needed to produce such historical reports as the ACFRPTXR Cross-reference Report.

@DDSN ACF2FDR Macros And GSO BACKUP Record

During the ACF2 installation process, it is necessary to define the primary and alternate VSAM clusters and also to define a sequential dataset for each cluster which will be used during backup processing. The DEFINE job, supplied on the ACF2 distribution tape, will allocate the space required for these datasets. (Refer to the installation procedure section of the acf2/MVS System Programmer's Guide.)

The @DDSN macro of the ACF2FDR indicates to the ACFMAIN task the dataset names assigned for both the primary and alternate clusters, and the backup sequential dataset for each set of clusters.

The GSO BACKUP record specifies the CPU on which the automatic backup processing will be performed, and the time-of-day when ACF2 should initiate its automatic backup processing. The BACKUP record has an optional field, STRING='str'. The string may be any valid operator command. After successful creation of the backup sequential datasets, ACF2 will issue this string to the system. For example, this facility could be used to initiate further local processing of the backup dataset. See also "The ACFBKUP Procedure" section later in this chapter for a sample backup procedure provided with ACF2.

SMF Journaling of Database Modifications

ACF2 journals each modification of its databases to SMF. By combining this SMF data with the backup copies, an up-to-date copy of each primary VSAM cluster can be recreated. The ACFRECVR procedure will combine SMF data with the backup datasets and may be initiated from the system console. See also "ACFRECVR - The Recovery Utility" in this manual.

CONTINGENCY PROCEDURE FOR USING THE ALTERNATE CLUSTERS

In the event the primary VSAM clusters become inaccessible or unusable, the contingency procedure outlined below illustrates a step-by-step method for using the alternate clusters. The example has been included to show the inter-relationship between the various ACF2 recovery facilities and should not be misconstrued to be a "fail-safe" procedure which will work for every installation. This example assumes that the alternate clusters have been maintained with the primary clusters' backup data, as described in the section on the ACFBKUP procedure later in this chapter. Also note that the sample commands depicted in this procedure will probably be different at each installation.

- * IPL the system specifying the alternate COMMNDxx member that does not automatically start ACF2.
- * Initiate the ACFRECVR procedure from the console or through batch. This procedure executes the ACFRECVR utility which reads the SMF data base modification journal records created by ACF2 and applies any changes necessary to bring the alternate clusters up-to-date with the primary clusters. If ACFRECVR is being initiated from the console, it may be necessary to execute the procedure two or three times, depending on how many sources of SMF input must be processed. The following operator command will initiate ACFRECVR from the console specifying SYS1.MANX as input:

```
S ACFRECVR,ACFX=SYS1,ALT=ALT,DSNAME='SYS1.MANX',DISP=SHR
```

- * Now start ACF2 from the system console. Specify the alternate clusters and NOBACKUP parameter. The NOBACKUP parameter turns off the automatic backup facility. (The STRING field of the GSO BACKUP record may cause a job to be submitted and abend.) The following operator command starts ACF2 with the alternate clusters and turns off the automatic backup facility:

```
S ACF2,PARM='NOBACKUP,DDSNS(ALT)'
```

- * ACF2 should be up and running on the alternate VSAM clusters.
- * If the SMF records are processed in a non-ascending order, which could be the case in a multi-CPU environment, a record that had been deleted may exist in the alternate clusters after ACFRECVR

processing completes. To insure against the inadvertant existence of such records, it is recommended that the ACFRPTLL, ACFRPTRL, and ACFRPTL report generators be run and the output reviewed.

RESTORING THE PRIMARY CLUSTERS

At the earliest possible time, the installation should correct the problem with the primary VSAM clusters. Once the primary clusters are accessible:

- * Force a backup of the alternate clusters by issuing an F ACF2, BACKUP command from the system console. Be aware that the STRING field of the the GSO BACKUP record may cause a job to be submitted and to abend. The next step will ensure that the primary clusters are restored without the occurrence of such an abend.
- * Force ACF2 to REPRO the alternate sequential backup datasets into the primary VSAM clusters. The ACFBKUP procedure may be initiated from the system console to accomplish this function:

```
S ACFBKUP,PRIM='A',ALT=' '
```

- * Shutdown ACF2 by entering the command "P ACF2" from the system console. Note that once ACF2 is stopped, the operator must reply to all access request verifications. If this procedure is being followed, the operator should reply "W" to cause the access requests to wait until ACF2 is restarted (next step).
- * Restart ACF2 from the system console.
- * ACF2 should be up and running again on the primary VSAM clusters.

ACFRECVR - THE RECOVERY UTILITY

The ACFRECVR utility processes the database update SMF records produced by ACF2 and forward merges them into the VSAM clusters to produce an up-to-date set of databases. Each input SMF record contains a time-of-day stamp that is checked to insure that the most recent record exists in the cluster. Note that space for the output VSAM clusters must be initialized before running ACFRECVR.

Additionally, the procedure ACFRECVR is placed into the target system's SYS1.PROCLIB during the installation process. This procedure executes the ACFRECVR utility and may be initiated from the system console. Refer to the section on the ACFRECVR procedure, towards the end of this chapter.

Files

SMFxxxxx

These files contain the input SMF records to be processed by the ACFRECVR utility. All DD names with the first three characters of "SMF" will be selected for input processing. Any combination of VSAM and non-VSAM input is acceptable. All SMF records produced since the last ACF2 database backup must be provided to ACFRECVR for forward merge purposes for complete recovery. ACFRECVR checks time stamps in the records to insure that the latest update remains in the file.

NOTE: Do not concatenate SMF input files. ACFRECVR will terminate with an abnormal return code if SMF files are concatenated.

LOGONIDS

The DD statement with a DD name of LOGONIDS defines the new Logonid VSAM cluster. This dataset must be specified with a disposition of old (DISP=OLD). Logonid records from the ACF2 permanent backup dataset must also be REPRO'd into this file before ACFRECVR processes the dataset.

INFOSTG

The DD statement with a DD name of INFOSTG defines the new Information Storage VSAM cluster to be built. Again, specify DISP=OLD. Also, REPRO the backup information records into this file.

RULES

The DD statement with a DD name of RULES defines the access rule VSAM cluster. Again, specify DISP=OLD. Also, REPRO the backup access rules into this file.

SYSPRINT

This is the utility output file. See the section "Report Generators - General Information" for details concerning specifications for this file. The ACFRECVR utility provides a report of each record processed and the disposition of that record.

Parameters

The ACFRECVR utility accepts parameters through the JCL "PARM=" field, or through SYSIN, or both, in the same manner as the ACF2 report generators. The following parameters for the ACFRECVR utility are described in the "Report Generators - General Information" section of this manual:

LINECNT, SDATE, EDATE, STIME, ETIME, SYSID, and HEX.

The following parameters are specific to ACFRECVR:

RULE/RULE(nnn)/NORULE - DEFAULT=RULE(nnn), where nnn is defined in the ACFFDR

The RULE parameter defines the SMF record number for Access Rule database update journal records. If NORULE is specified, then the Access Rule cluster is not updated during the recovery process. If the RULE parameter is specified without an SMF record number, then the value specified in the ACFFDR will be used during the recovery process to define the SMF record number for Access Rule database update journal records.

LID/LID(nnn)/NOLID - DEFAULT=LID(nnn), where nnn is defined in the ACFFDR

The LID parameter defines the SMF record number for Logonid database update journal records. If NOLID is specified, then the Logonid cluster is not updated during the recovery process. If the LID parameter is specified without an SMF record number, then the value specified in the ACFFDR will be used during the recovery process to define the SMF record number for Logonid database update journal records.

RSRC/RSRC(nnn)/NORSRC - DEFAULT=RSRC(nnn), where nnn is defined in the ACFFDR

The RSRC parameter defines the SMF record number for Infostorage database update journal records. If NORSRC is specified, then the Infostorage cluster is not updated during the recovery process. If the RSRC parameter is specified without a record number, then the value specified in the ACFFDR will be used during the recovery process to define the SMF record number for Infostorage database update journal records.

Note: If the RULE, LID, and RSRC parameters are not specified, then the defaults for these parameters will be used. However, if any of these parameters is specified, then the unspecified parameters will default to NORULE, NOLID, or NORSRC, respectively. For example, if RULE is specified, then NOLID and NORSRC will be assumed.

JOBMASK(job-mask,job-mask,...) - DEFAULT=***** (all)

The JOBMASK parameter specifies that records processed by ACFRECVR be limited to those pertaining to the job(s) indicated by the jobname(s) or jobname mask(s). Multiple jobnames or jobname masks must be separated by commas or blank spaces. The default is processing of records pertaining to all jobs.

KEYMASK(record-key-mask) - DEFAULT=all record keys

The KEYMASK parameter allows selected records to be restored to a database cluster without disturbing other records. The selected records are specified by record key, as follows:

* If the LID parameter is specified, then the KEYMASK specifies a 1- to 8-character Logonid mask.

- * If the RULE parameter is specified, then the KEYMASK specifies a 1- to 8-character mask for access rule sets.
- * If the RSRC parameter is specified, then the KEYMASK specifies a 1- to 44-character mask for an Infostorage record key. The first character is the storage class, and the next three characters are the type code.

LAB/NOLAB - Default=NOLAB

Indicates whether LAB (Lookaside Buffer Support) entries should appear on the ACFRECVR report. Note that these records are not used to perform any ACF2 database updating. The default of NOLAB causes ACFRECVR to bypass these records during ACFRECVR processing. Use the LAB option to obtain a complete report of all LAB entry processing.

ACFRECVR - Sample Output

ACFRECVR - Sample Output

The following is an example of the output produced by the recovery utility ACFRECVR. The report details each record processed and the status of the record.

ACF2 UTILITY LIBRARY - ACFRECVR - VSAM FILE RECOVERY UTILITY PAGE 1
DATE 09/11/79 (79.254) 10.45 LAST LID 79.254-09.00 LAST RULE 79.253-14:23 LAST INFSTG 79.250-11.30

DATE	TIME	CLUSTER	KEY	JOBNAME	CHANGER	CHANGE	STATUS
79.254	09/11	07.00	LOGONID HAL1	MSTRJCL		UPDATE	BYP-TOD STAMP*
79.253	09/10	12.25	RULE-ID SYS1	ACF2	ACF2	REPLACE	REC-REPLACED
79.250	09/07	11.30	INFSTG ESRCT1T001	ACF2	ACF2	INSERT	REC-INSERTED

Field Descriptions

Below is a description of the fields of the ACFRECVR output report.

The second title line (at the top of each report page) shows the data and time on which the recovery utility was run. This title line also shows the latest time stamp from the VSAM clusters. The time stamp indicates the backup level of each cluster before the recovery utility began processing.

DATE

The Julian and Gregorian dates from the SMF record being processed. The format of the Gregorian date is either MM/DD or DD/MM, depending on ACF2 generation options.

TIME

The time-of-day at which the record was issued. This time is in hours and minutes (24-hour clock). The date and time listed in the report are taken from the SMF record time stamp. Internally, ACFRECVR compares time-of-day clock format time stamps which are considerably more accurate.

CLUSTER

This field indicates the ACF2 cluster to which the record applies. This field can take any of the following values.

LOGONID - Logonid cluster.

RULE-ID - Access Rule cluster.

INFOSTG - Infostorage cluster.

KEY

This field displays the key of the ACF2 record that was processed. The following table shows what the key represents:

<u>Type of Record</u>	<u>Record Key Consists Of:</u>
Logonid record	Logonid
Access rule set	Access rule set key
Infostorage record	Storage class, type code, record name

JOBNAME

The name of the job from which the update was made. MSTRJCL and JESx come from updates made during job validation processing.

CHANGER

The Logonid of the user who caused the change to occur if applicable.

CHANGE

The type of record update represented by this SMF record.

For Logonids:

INSERT A new Logonid was inserted into the data base.

CHANGE An old Logonid was changed. This type of change is from a user request.

DELETE The indicated Logonid was deleted from the ACF2 database.

UPDATE The Logonid was changed during job validation processing.

For access rules and information storage records:

INSERT A new record was inserted into the ACF2 database.

REPLACE An old record was replaced.

DELETE The indicated record was deleted from the ACF2 database.

STATUS

This field indicates the action taken for the record by the ACFRECVR utility. The table below details the possible actions.

REC-ERASED The input SMF record was for a deleted record. The corresponding record in the ACF2 database was erased.

REC-REPLACED The record input has a time stamp greater than the record in the ACF2 database. This indicates that the input record represents a more recent update so the input record is stored in the ACF2 database.

REC-INSERTED The input record key does not match any currently in the ACF2 database; the input record is inserted into the database.

BYP-NO RECORD* The input SMF record indicated that the data base record should be deleted but no record with a corresponding key was found in the ACF2 database.

BYP-INV KEY * An internal consistency check of the SMF record has failed, indicating the SMF record is invalid. Notify ACF2 maintenance personnel of this error.

BYP-NO ROOM * The new record could not be stored in the data base being recovered. This error will seem to occur randomly because of VSAM CA/CI split processing. After the recovery utility completes executing, reorganize the cluster in question (reallocate if necessary), and rerun the recovery utility without change.

BYP-INV TYPE * The recovery utility did not recognize the update function code in the SMF record being processed. Notify ACF2 maintenance personnel of this error and ensure that the recovery utility is at the proper release level.

BYP-TOD STAMP* The record was bypassed because the time-of-day stamp checks indicated that the record in the ACF2 database was more recent than the record represented by the SMF record.

BYP-VSAM ERR * A VSAM error has occurred during ACFRECVR processing. A message will be issued indicating the cause of the error. Correct the error condition and rerun the ACFRECVR utility without change.

BYP-TYPE SKIP* The database in which this record is to be stored is not being processed by the recovery utility because of a NORULE, NOLID, or NORSRC parameter specification.

NOTE: The following status indicators are displayed only when the LAB parameter is specified. Also, these LAB records are not used to perform any ACF2 database updating. They appear for reporting purposes only.

BYP-LAB INSERT* Indicates that a LAB (Lookaside Buffer) entry was inserted from the CPU indicated in the LAB field to the CPU indicated in the CPU field.

BYP-LAB DEL L* Indicates that the system operator issued a command to delete one or more LAB entries based on the Logonid name. The LAB entries were deleted from the CPUid referenced in the CPU field.

BYP-LAB DEL N* Indicates that the system operator issued a command to delete one or more LAB entries based on the path or node used to create the LAB entry. The LAB entries were deleted from the CPUid referenced in the CPU field.

|
|
|
BYP-LAB DEL S* Indicates that the system operator issued a command to delete one or more LAB entries based on the input source-id of the LAB entries. The LAB entries were deleted from the CPUid referenced in the CPU field.

BYP-LAB DEL U* Indicates that a LAB entry was deleted after all normal processing was completed, e.g., the job using the LAB entry ended. The LAB entry was deleted from the CPUid referenced in the CPU field.

Note: The asterisk ('*') is used to highlight bypassed status messages. If the status message indicates an error during processing, first correct the error and then rerun the recovery utility without changing the JCL. ACFRECVR time stamp checks will insure that the most recent record is stored in the data base.

Sample Recovery Jobstream

The sample RECOVER JCL which follows uses the alternate VSAM clusters established by the DEFINE job during the ACF2 installation process. The sample assumes that output VSAM clusters have been initialized and are ready to be used. Each installation should tailor the RECOVER JCL as necessary and test it prior to production use of the ACF2 system. As part of the DEFINE job, the alternate clusters and backup datasets are allocated with the primary clusters and backup datasets. This is done to help the installation begin thinking about disaster recovery procedures. Toward this end, two new started procedures are installed in SYS1.PROCLIB; ACFBKUP and ACFRECVR. A description and JCL listing of the ACFRECVR procedure follows the sample RECOVER jobstream. The ACFBKUP procedure is then described following the ACFRECVR procedure.

```
//RECOVER JOB 1,'RECOVERY UTILITY',MSGCLASS=A,TYPRUN=HOLD
//*
//*
//* THIS JOB SHOULD BE TAILORED TO EACH INSTALLATION AND
//* KEPT FOR EMERGENCY USE ONLY.
//*
//* RECOVER THE ACF2 DATABASES INTO ALTERNATE CLUSTERS.
//*
//* THE ALTERNATE CLUSTERS USED BY THIS PROCEDURE SHOULD HAVE
//* ALREADY BEEN DEFINED BY THE DEFINE JOB.
//*
//RECOVER PROC SOUT=A, (MVS) SYSOUT OUTPUT CLASS
// ACFX=SYS1, (MVS) ACF2 1ST LEVEL INDEX
// ALT=A, (MVS) ALTERNATE CLUSTER NAME
// PRIM=, (MVS) PRIMARY CLUSTER NAME
// DMYX=, (MVS) 'DUMMY,' IF NO MANX
// DSX='SYS1.MANX', (MVS) MANX DATASET NAME
// DMY=, (MVS) 'DUMMY,' IF NO MANY
// DSY='SYS1.MANY', (MVS) MANY DATASET NAME
// DMYT='DUMMY,', (MVS) 'DUMMY,' IF NO HIST TAP
// DST='SYS1.BACKUP.MAN(0)' (MVS) SMF HISTORY TAPE DSN
//*
//REPRO EXEC PGM=IDCAMS
//*
//*
//* REPRO THE ACF2 BACKUP DATASETS INTO THE APPROPRIATE
//* ACF2 ALTERNATE VSAM DATABASE CLUSTERS.
//*
//SYSPRINT DD SYSOUT=&SOUT
//*
//*
//* INPUT ACF2 BACKUP DATASETS.
//*
//BKLIDS DD DISP=OLD,DSN=&ACFX..ACF2.&PRIM.BKLIDS
//BKRULES DD DISP=OLD,DSN=&ACFX..ACF2.&PRIM.BKRULES
//BKINFO DD DISP=OLD,DSN=&ACFX..ACF2.&PRIM.BKINFO
//*
//*
//* OUTPUT ACF2 ALTERNATE DATABASE CLUSTERS.
```

```

/**
//ALIDS DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.LOGON
//ARULES DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.RULES
//AINFO DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.INFOSTG
/**
//MERGE EXEC PGM=ACFRECVR,COND=(4,LT)
/**
/**
/** THIS STEP MERGES THE SMF JOURNAL RECORDS WITH THE
/** VSAM CLUSTERS AND PRODUCES AN UP-TO-DATE DATABASE. THE
/** ORDER OF INPUT OF THE SMF RECORDS IS UNIMPORTANT AS
/** THE TIME AND DATE STAMP IN THE SMF RECORD WILL
/** DETERMINE WHETHER THE RECORD IS USED OR NOT. THIS STEP
/** MAY BE RERUN AS MANY TIMES AS NEEDED.
/**
//SYSPRINT DD SYSOUT=&SOUT
/**
/**
/** OUTPUT ALTERNATE ACF2 DATABASE CLUSTERS.
/**
//LOGONIDS DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.LOGON
//RULES DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.RULES
//INFOSTG DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.INFOSTG
/**
/**
/** INPUT SMF FILES.
/**
//SMFREX DD &DMYX.DISP=SHR,DSN=&DSX
//SMFREY DD &DMYY.DISP=SHR,DSN=&DSY
//SMFRECT DD &DMYT.DISP=OLD,DSN=&DST
/**
// PEND
/**
//ACF2 EXEC RECOVER
/**
/**
/** CONTROL STREAM.
/**
//REPRO.SYSIN DD *
/**
/**
/** COPY THE ACF2 BACKUP DATASETS INTO THE ALTERNATE ACF2
/** DATABASE CLUSTERS.
/**
| REPRO INFILE(BKLIDS) OUTFILE(ALIDS) REPLACE REUSE
| REPRO INFILE(BKRULES) OUTFILE(ARULES) REPLACE REUSE
| REPRO INFILE(BKINFO) OUTFILE(AINFO) REPLACE REUSE
/**

```

THE ACFRECVR PROCEDURE

The ACFRECVR procedure is placed into the target system's SYS1.PROCLIB during the ACF2 installation process. This procedure may be initiated from the system console. Before using ACFRECVR, synchronize the primary and alternate VSAM cluster dsnames with those specified within the @DDSN macro of the ACF2. (Refer to the acf2/MVS System Programmer's Guide.)

The ACFRECVR procedure may be used in the event the primary VSAM clusters are damaged and it becomes necessary to bring up ACF2 on the alternate clusters. This procedure will execute the ACFRECVR utility using one source of input. Input may be SYS1.MANX, or SYS1.MANY, or an SMF generation tape. For complete recovery of the clusters, the ACFRECVR procedure should be run once for each source of SMF data before ACF2 is restarted. Also, see the section "Contingency Procedure For Using the Alternate Clusters," earlier in this chapter.

Submission of batch jobs and use of online systems should not be allowed until ACFRECVR completes processing and ACF2 has been restarted. Certainly, updating of the ACF2 database VSAM clusters should not be allowed while ACFRECVR is processing.

Note that the ACFRECVR procedure accepts only one source of input per execution. Therefore, it is sometimes necessary to initiate ACFRECVR two or three times in order to bring the alternate clusters completely up-to-date with the primary clusters. Alternatively, the ACFRECVR batch utility program may be used to recover from up to three SMF input sources into the alternate clusters.

```
//ACFRECVR PROC SOUT=A,           (MVS) SYSOUT OUTPUT CLASS
//      ACFX=SYS1,              (MVS) ACF2 1ST LEVEL INDEX
//      ALT=A                    (MVS) ALTERNATE CLUSTER NAME
//*
//*
//*      THIS PROCEDURE IS PROVIDED AS A RECOVERY AID. ITS FUNCTION
//*      IS TO EXECUTE THE ACFRECVR PROGRAM TO RECONSTRUCT THE ACF2
//*      VSAM CLUSTERS. THE DEFAULT FOR THIS PROCEDURE IS TO RECOVER
//*      INTO THE ALTERNATE VSAM CLUSTERS BUT THE PRIMARY CLUSTERS
//*      MAY BE USED PROVIDED ACF2 IS NOT ACTIVE AT THE TIME. INPUT
//*      TO THE RECOVERY UTILITY IS THE SMF DATA JOURNALED BY ACF2.
//*      EITHER THE ONLINE SMF FILES MAY BE USED OR A JOURNAL TAPE
//*      CONTAINING DUMPED SMF DATA. THE ORDER IN WHICH THE SMF DATA
//*      IS NOT IMPORTANT. THIS PROCEDURE WILL ONLY PROCESS ONE SMF
//*      FILE AT A TIME AS IT IS CURRENT DEFINED. TO EXECUTE THIS
//*      PROCEDURE SPECIFYING THE ONLINE SMF DATASET SYS1.MANX
//*      ISSUE THE FOLLOWING COMMAND AT THE OPERATOR'S CONSOLE:
//*
//*      S ACFRECVR,SOUT=A,ACFX=SYS1,ALT=A,DSNAME=SYS1.MANX,DISP=SHR,
//*          UNIT=SYSDA,VOL=SER=SYSRES
//*
//*      NOTE THAT IN THIS EXAMPLE THE UNIT AND VOL=SER PARAMETERS
//*      NEED NOT BE CODED AS THE SYS1.MANX DATASET CAN BE ACCESSED
//*      VIA THE CATALOG. THEY ARE PROVIDED TO SHOW ONLY THAT THESE
//*      PARAMETERS ARE AVAILABLE.
```

```
//*  
//* THIS JOB SHOULD BE TAILORED TO EACH INSTALLATION AND  
//* KEPT FOR EMERGENCY USE ONLY.  
//*  
//IEFPROC EXEC PGM=ACFRECVR  
//*  
//*  
//* THIS STEP MERGES THE SMF JOURNAL RECORDS WITH THE  
//* VSAM CLUSTERS AND PRODUCES AN UP-TO-DATE DATABASE. THE  
//* ORDER OF INPUT OF THE SMF RECORDS IS UNIMPORTANT AS  
//* THE TIME AND DATE STAMP IN THE SMF RECORD WILL  
//* DETERMINE WHETHER THE RECORD IS USED OR NOT. THIS STEP  
//* MAY BE RERUN AS MANY TIMES AS NEEDED.  
//*  
//SYSPRINT DD SYSOUT=&SOUT  
//SYSUDUMP DD SYSOUT=&SOUT  
//*  
//*  
//* OUTPUT ALTERNATE ACF2 DATABASE CLUSTERS.  
//*  
//LOGONIDS DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.LOGON  
//RULES DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.RULES  
//INFSTG DD DISP=OLD,DSN=&ACFX..ACF2.&ALT.INFSTG  
//*  
//*  
//* INPUT SMF FILES.  
//*  
//SMFRECX DD DDNAME=IEFRDER  
//IEFRDER DD DISP=SHR,DSNAME=SMFIN  
//*
```


THE ACFBKUP PROCEDURE

The ACFBKUP procedure is placed into the target system's SYS1.PROCLIB during the ACF2 installation process. ACFBKUP will REPRO the primary sequential backup datasets into the alternate VSAM clusters. The most convenient method of initiating ACFBKUP is through the use of the STRING operand within the GSO BACKUP record of the Infostorage database. (See the chapter on GSO records in the acf2/MVS Administrator's Guide.) When ACFBKUP is used in this manner, the alternate clusters will always be within 24 hours of the primary clusters.

Before using ACFBKUP, synchronize the primary and alternate VSAM cluster dsnames with those specified in the @DDSN macro of the ACF2 FDR. Also, initialize the alternate VSAM clusters before executing ACFBKUP. See also the acf2/MVS System Programmer's Guide, and the earlier sections of this chapter.

```
//ACFBKUP  PROC SOUT=Z,           (MVS) SYSOUT OUTPUT CLASS
//          ACFX=SYS1,           (MVS) ACF2 1ST LEVEL INDEX
//          ALT=A,               (MVS) ALT CLUSTER DSNAME MODIFIER
//          PRIM=,               (MVS) PRIMARY CLUSTER DSNAME MODIFIER
//          INDS='SYS1.PROCLIB', (MVS) SYSIN DATASET NAME
//          INU=SYSDA,           (MVS) SYSIN DATASET UNIT
//          INVOL=SYSRES,        (MVS) SYSIN DATASET VOLUME
//          INC=' '              (MVS) ', ' = SYSIN NOT CATALOGED
//*
//*
//*      THIS PROCEDURE IS PROVIDED AS A RECOVERY AID. ITS FUNCTION
//*      IS TO USE THE CONTENTS OF THE PRIMARY BACKUP DATASETS TO
//*      UPDATE THE CONTENTS OF THE ALTERNATE VSAM CLUSTERS. IN THIS
//*      WAY, THE ALTERNATE CLUSTERS WILL ALWAYS BE WITHIN 24 HOURS
//*      OF THE PRIMARY CLUSTERS. TO MAKE BEST USE OF THIS PROCEDURE
//*      THE @BACKUP MACRO OF THE ACF2 FDR GENERATION SHOULD SPECIFY:
//*
//*          STRING='S ACFBKUP'
//*          STRING='S ACFBKUP.PNN'
//*
//*      SO THAT THIS PROCEDURE IS ALWAYS EXECUTED AS SOON AS THE
//*      ACF2 BACKUP PROCESS HAS COMPLETED.
//*
//*      USE IDCAMS REPRO TO BUILD THE ALTERNATE CLUSTERS FROM
//*      THE BACKUP DATASETS.
//*
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=&SOUT
//*
//*
//*      INPUT ACF2 BACKUP DATASETS
//*
//BKLIDS   DD  DISP=SHR,DSN=&ACFX..ACF2.&PRIM.BKLIDS
//BKRULES  DD  DISP=SHR,DSN=&ACFX..ACF2.&PRIM.BKRULES
//BKINFO   DD  DISP=SHR,DSN=&ACFX..ACF2.&PRIM.BKINFO
//*
//*
```

```
/**      OUTPUT ALTERNATE CLUSTERS.  
/**  
//ALIDS DD  . DISP=OLD,DSN=&ACFX..ACF2.&ALT.LOGON  
//ARULES DD  DISP=OLD,DSN=&ACFX..ACF2.&ALT.RULES  
//AINFO  DD  DISP=OLD,DSN=&ACFX..ACF2.&ALT.INFOTG  
/**  
//SYSIN  DD  DSN=&INDS(ACFBKUPI),  
//        DISP=SHR&INC.UNIT=&INU,VOL=SER=&INVOL
```

UTILITIES FOR ACF2 ADMINISTRATION

This chapter discusses the following utilities designed for more convenient administration of ACF2:

<u>Utility Name</u>	<u>Type of Utility</u>	<u>Purpose</u>
ACFBATCH	Batch program	Batch execution of a sequence of ACF subcommands.
ACFCOMP	TSO Command	Compilation of ACF2 access or generalized resource rule set.
ACFBCOMP	Batch program	Batch compilation of rule sets.
ACFBDCMP	Batch program	Batch decompilation of rule sets into MVS datasets.
ACFNRULE	TSO Command or Batch program	Addition, replacement, or deletion of selected rules.
ACFBSYNC	Batch program	Synchronization of the TSO BROADCAST dataset with the ACF2 Logonid database.

ACFBATCH - BATCH COMMAND PROCESSOR

The ACFBATCH utility is provided as a facility for executing the standard ACF2 commands in batch. These subcommands include those used to maintain the ACF2 Logonid data base (INSERT, CHANGE, DELETE, LIST), those used to maintain the ACF2 rules data bases (COMPILE, DECOMPILE, STORE, DELETE, TEST), and various other subcommands (SET, SHOW, HELP, etc.) The equivalent facility is already available at MVS sites by executing the TMP (Terminal Monitoring Program) in background, but MVS sites can also use ACFBATCH.

Files

SYSHELP

This input file is necessary whenever the HELP subcommand is being used (this file provides the HELP data).

SYSPRINT

This output file is used for the output listings produced by the various commands. Used only when "PGM=ACFBATCH" is specified in the JCL.

SYSTSPRT

This file is used for the output listings produced by the various subcommands. Used only when "PGM=IKFEFT01" (TMP in background) is specified in the JCL.

SYSIN

This input file is used when specifying multiple input lines (subcommands and/or data) to the ACFBATCH utility. Used only when "PGM=ACFBATCH" is specified in the JCL.

SYSTSIN

This input file is used when specifying multiple input lines (subcommands and/or data) when the TMP in background is used to execute the ACF command. Used only when "PGM=IKFEFT01" (TMP in background) is specified in the JCL.

anyddns

These files (none, one, or more) can have any DD name and would be used to name the dataset(s) whenever rules are being compiled out of existing dataset(s) or decompiled into dataset(s). These do not refer to the ACF2 databases themselves.

Parameters

The only parameter which can be specified to ACFBATCH via the JOB card PARM= field is the ACF2 command to be executed. The most common usage of ACFBATCH would be to execute the ACF command (and its subcommands), so the default value (when none is specified) is "PARM=ACF".

No PARM= field or PARM=ACF

This is the default and the ACF command will be invoked. The input entries in the SYSIN DD * stream would then be various ACF subcommands and data.

PARM=ACFCOMP (* /pdsname (member))

This invokes the ACFCOMP command ALL. If PARM=ACFCOMP *, then the entries in the SYSIN DD * stream would be input to the compiler. The first line would begin "\$KEY(key)", as shown below. After compilation, the ruleset is automatically stored. This parameter allows compilation of either an access or a generalized resource rule set.

Examples

Using ACFBATCH to perform ACF subcommands:

```
//ACFJOB1 EXEC PGM=ACFBATCH
//SYSPRINT DD SYSOUT=A
//SYSHELP DD DSN=SYS1.HELP,DISP=SHR
//SYSIN DD *
CHANGE UID(ABC****) JOB NOTAPE-BLP
SET TERSE
LIST IF(JOB)
SET VERBOSE
LIST IF(SEcurity)
SHOW STATE
HELP FIELDS
/*
```

Using ACFBATCH to execute ACFCOMP command to compile a rule. Data is being provided as input in the job stream:

```
//BACFCOMP EXEC PGM=ACFBATCH,PARM='ACFCOMP *'
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
$KEY(ABC123)
TEST.DATA UID(ABC****) R(A) W(A)
SPECIAL.DATA UID(ABC****) R(A)
SPECIAL.DATA UID(ABC12*) R(A) W(A)
/*
```

See the acf2/MVS Administrator's Guide for a full description of the ACF2 commands and subcommands available.

ACFCOMP - TSO RULESET COMPILE COMMAND

The TSO command ACFCOMP provides an alternative means for compiling access rule sets. This command is issued from TSO ready mode.

Syntax

The TSO ACFCOMP command follows one of these syntax formats:

ACFCOMP [STORE/NOSTORE] [FORCE/NOFORCE]

ACFCOMP * [STORE/NOSTORE] [FORCE/NOFORCE] [MAXRULE(nnn)]

ACFCOMP dsname [LIST/NOLIST] [STORE/NOSTORE] [FORCE/NOFORCE]
[MAXRULE(nnn)] [ALL]

Methods of Compilation

ACF2 provides two ways of compiling access rule sets:

1. Directly at the terminal
2. From a partitioned dataset (PDS)

Compiling Directly at the Terminal. You can enter an access rule set directly from the terminal by first entering the ACFCOMP command without an asterisk or dataset name:

```
READY  
acfcomp
```

ACF2 will respond with:

```
ACF70010 ACF COMPILER ENTERED
```

You can enter your rule set text just as you would under the ACF COMPILE subcommand. The access rule set is automatically stored after you compile it.

Compiling from a Partitioned Dataset (PDS). You can also create access rule sets by first entering the control cards and rule entries into a PDS member. Each control card or rule entry must be on a separate line. The last line does not have to be a blank line. For example:

```
$KEY(PAYROLL)  
WORK.MASTER UID(****PAYNLT) R(A) W(A) E(A)  
WORK.BACKUP UID(****PAYISO) R(A) W(L) E(A)
```

After entering the control cards and rule entries into the PDS member, you can issue the ACFCOMP command with the name of the PDS and member. For example:

```
acf
set rule
acfcamp work.text(rule)
```

Parameters

The ACFCOMP command takes the following parameters:

An asterisk indicates that the text that follows will be input to the compiler. The system will prompt you to enter the access rule text directly from the terminal.

(no parameters)

Use of the ACFCOMP command without parameters is equivalent to specifying an asterisk. However, the valid parameters that you can specify differ slightly. See the syntax above.

dsname

A partitioned dataset (PDS) and member name specifies the PDS and member that contains the access rule text to be compiled. The PDS name follows TSO conventions. (Your high-level index is assumed unless you specify the entire PDS name and enclose it in single quotes. For example, 'PAYNLT.WORK.TEXT(RULE)').

If you do not specify a member name, ACF2 will prompt you for one. To compile input from all PDS members, specify the ALL parameter. An access rule set cannot be compiled from a sequential dataset.

LIST

The LIST parameter causes the input to the compiler to be displayed on your screen or printed on your listing during compilation of a rule set. NOLIST causes no such display or printed list. LIST is the default. This parameter does not apply for compilations directly from the terminal.

STORE/NOSTORE

The STORE parameter causes the rule set to be automatically stored at compilation time. NOSTORE causes no automatic storing of the rule set; you must issue the STORE subcommand to store the rule set. STORE is the default.

FORCE/NOFORCE

The FORCE parameter allows the access rule set to be stored regardless of whether it currently exists. NOFORCE allows the access rule set to be stored only if it does not already exist. FORCE is the default.

MAXRULE(nnn)

The MAXRULE parameter specifies a number from 0 through 999 that limits the size of the rule set that you can input to the compiler. This number is a scaling factor and has no direct correlation with the number of rules in a rule set. (Different rules require different amounts of space.) The default is 250. If a rule set compiles correctly but encounters space problems when you try to store it, try specifying MAXRULE with a value less than 250. Alternatively, if a rule set contains a lot of duplication (i.e., input source names, etc.), you can specify a higher value for MAXRULE to allow the compiler to accept a larger rule set. However, using the NEXTKEY parameter in your access rules is the preferable way to handle large rule sets.

ALL

The ALL parameter causes compilation and storing of access rule sets from all members of a specified partitioned dataset (PDS). For example:

```
READY  
acfcomp work.text all
```

If any members of a PDS do not contain an access rule set text, then do not specify this parameter.

ACFBCOMP - BATCH RULE COMPILER

The ACFBCOMP utility may be used to compile ACF2 rule sets in the batch environment.

Files

RULES

A single access or generalized resource rule set to be compiled. This dataset can be in either of two formats: card image or variable. Card image datasets must have DCB attributes of RECFM=FB and LRECL=80; columns 73-80 may contain sequence numbers. Variable format datasets must have DCB attributes of RECFM=VB and LRECL=255; for variable format datasets columns 1-8 are ignored by the compiler and may contain sequence numbers. These DCB attributes and sequencing conventions are standard for TSO DATA and CLIST dataset types.

SYSPRINT

This file is used for the output listing of the rule set being compiled along with any messages produced.

The ACFBCOMP facility accepts the standard (ACF COMPILE subcommand) parameters in the JCL parm field and passes them to the compiler (e.g., MAXRULE, LIST, NOLIST, STORE, NOSTORE).

ACFBCOMP (Batch Compiler) Example

The following example shows how the ACFBCOMP utility is executed in batch.

```
//COMPILE EXEC PGM=ACFBCOMP,REGION=128K
//SYSPRINT DD SYSOUT=A
//RULES DD *
$KEY( . . . ) [TYPE(type)]
      .
      .
/*
```

If the TYPE() parameter is present, the ACFBCOMP utility assumes that the rule set is for a generalized resource. Otherwise if the TYPE parameter is absent, it is assumed to be an access rule set.

Only one rule set may be compiled with each execution of the ACFBCOMP utility. To compile multiple rule sets from a partitioned dataset (PDS) the ACFCOMP command or the COMPILE subcommand of the ACF command must be used, as shown in the following examples.

Note that the SYSTSIN input file and the SYSTSPRT output file replace the SYSIN and SYSPRINT files, respectively, in the following examples. See "ACFBATCH - Batch Command Processor" for a description of these files.

Using the ACFCOMP command with TTP in background:

```
//COMPALL EXEC PGM=IKJEFT01,REGION=256K,DYNAMNBR=5
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
ACFCOMP 'rules.pds' ALL
```

Using the COMPILE subcommand of the ACF command with TMP in background:

```
//COMPILE EXEC PGM=IKJEFT01,REGION=256K,DYNAMNBR=5
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
ACF
COMPILE *
$KEY(ABC123)
TEST.DATA UID(ABC*****) R(A) W(A)
SPECIAL.DATA UID(ABC*****) R(A)
SPECIAL.DATA UID(ABC12*) R(A) W(A)
(NULL LINE must be placed prior to STORE command)
STORE
END
/*
```

ACFBDCMP - BATCH RULE DECOMPILER

The ACFBDCMP utility provides for decompilation of ACF2 access and generalized resource rule sets in the batch environment.

Any output message from the ACF2 decompiler will be issued to the JOB LOG. On an MVS system, messages will appear at the user's terminal if ACFBDCMP is executed under TSO and WTP has been specified for that user.

ACFBDCMP uses the JCL "PARM=" field to specify the name of the rule set to be decompiled. For access rule sets, specify simply the name. For generalized resource rule sets specify the resource type and name separated by a slash (type/name). Because slash is a special character, the PARM field must be enclosed in quotes. A decompile of these two types of rule sets is illustrated in the following examples.

Files

RULES

The RULES file is used by ACFBDCMP to output the results of the decompilation. The RULES output file must have DCB attributes of either LRECL=80, RECFM=FB (in which case the output will be card image format) or LRECL=255, RECFM=VB (in which case the output will be of TSO CLIST format). The output datasets will be sequence numbered and continuation cards will be created as needed for the card image dataset format. If no DCB attributes are specified the default will be variable format output and a BLKSIZE=3665 will be used.

Examples

For an access rule set:

```
//DECOMP EXEC PGM=ACFBDCMP,REGION=128K,PARM=SYS1  
//RULES DD DSN=SYS1.RULES(SYS1),DISP=OLD
```

For a generalized resource rule set:

```
//DECOMP EXEC PGM=ACFBDCMP,REGION=128K,PARM='TPR/IKJACCNT'  
//RULES DD DSN=SYS1.PROCS(IKJACCNT),DISP=SHR
```

The ACFBDCMP utility will decompile only one rule set per execution. To decompile multiple rule sets the DECOMP subcommand of the ACF command must be used. The DECOMP subcommand will create a partitioned dataset (PDS) containing one member for each rule set decompiled.

Batch example (using the TMP in background):

Note that the SYSTSIN input file and the SYSTSPRT output file replace the SYSIN and SYSPRINT files, respectively, in the following examples. See "ACFBATCH - Batch Command Processor" for a description of these files. To decompile all ACF2 access rule sets:

```
//DECALL EXEC PGM=IKJEFT01,REGION=256K,DYNAMNBR=5
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
ACF
DECOMP LIKE(-) INTO('SYS1.RULES')
```

For generalized resource rule sets, the "DD *" input would be:

```
ACF
SET RESOURCE(TPR)
DECOMP LIKE(-) INTO('SYS1.PROCS')
/*
```

ACFNRULE - NEW RULE UTILITY

The ACFNRULE utility provides a simple means of adding rules to access and generalized resource rule sets. ACFNRULE also provides a delete function to delete unwanted rules in the set. ACFNRULE may be executed both as a batch program or as a TSO command; as a TSO command ACFNRULE provides an additional verification facility to the delete function.

After ACFNRULE is executed, the modified rule set is automatically stored.

Files

SYSPRINT

The SYSPRINT file is used for the compiler output listing. As part of its processing, ACFNRULE must decompile the rule set being processed. Any messages issued by the decompiler will appear either in the JOB LOG (for a batch job) or at the terminal (if run as a TSO command and WTP has been specified). If run as a TSO command, the SYSPRINT file need not be specified as output will be sent to the terminal.

Parameters

ACFNRULE parameters are entered on the command line in the TSO environment and from the JCL parm field in batch.

TYPE(type)

If the ACFNRULE routine is to process a generalized resource rule set, the TYPE parameter is used to indicate the type of resource being processed. If no TYPE parameter is specified, ACFNRULE assumes that an access rule set is to be processed.

KEY(name)

The name of the rule set to be processed. For generalized resource rule sets this is the name of the resource and for access rule sets this is the key of the rule set to be processed.

DELETE(string)

The DELETE parameter specifies that any line containing the specified string be deleted from the rule set being processed. Lines being deleted will be listed on the SYSPRINT file or at the terminal if ACFNRULE is executed under TSO.

ADD(rule)

The ADD parameter specifies any valid ACF2 access or generalized resource rule to be added to the rule set being processed. ACFNRULE adds the rule to the end of the rule set; the ACF2 compiler will reorder the rule set as needed during the compilation pass.

LIST/NOLIST - DEFAULT=LIST

Specifies whether the rule set should be listed upon compilation. This listing is sent either to SYSPRINT (for a batch job) or to the terminal (if run as a TSO command).

VERIFY/NOVERIFY - DEFAULT=VERIFY - ignored in batch

The VERIFY option requests that the user be prompted for rule deletion. After listing the rules that would be deleted, ACFNRULE asks for a YES or NO to actually delete the rules. If the user specifies NO, ACFNRULE terminates. The VERIFY function is not available in the batch environment.

Examples

ACFNRULE in batch:

```
//ADDRULE EXEC PGM=ACFNRULE,REGION=128K,  
// PARM='KEY(SYS1),DEL(PARMLIB),ADD(PARMLIB UID(ABC) READ(A))'  
//SYSPRINT DD SYSOUT=A
```

Equivalent in TSO:

```
ACFNRULE KEY(SYS1) DEL(PARMLIB) ADD(PARMLIB UID(ABC) READ(A))
```

ACFNRULE may become more useful if various CLISTS are written to aid in common functions. The following CLIST adds a rule to a set allowing consultants to read a dataset for three days.

```
PROC 1 DSN FOR(3)  
ACFNRULE KEY(&SYSUID) ADD(&DSN UID(**C) READ(A) FOR(&FOR)) NOVERIFY
```

ACFBSYNC - THE BATCH SYNC UTILITY

The ACFBSYNC utility provides a high-speed batch utility to synchronize the SYS1.BROADCAST dataset with the ACF2 Logonid database for TSO users. ACFBSYNC will accept either a newly allocated BROADCAST dataset or an old BROADCAST dataset. The entire dataset will be rebuilt for processing, including both the Notices section and User Mail Directory. The INSERT command will add new Logonids to the dataset. To limit the users added, set the TSO record in GSO to NOLOGONCK. This triggers checking for only the TSO attribute in the Logonid before adding it to the BROADCAST dataset.

Full ACCOUNT privileges are required in order to execute ACFBSYNC.

Files

SYSLBC

This file identifies the BROADCAST dataset that is to be built. If this DD card points to an old BROADCAST dataset, only the user mail section is rebuilt. If the DD card points to a newly allocated BROADCAST dataset, that dataset is built from scratch. ACFBSYNC issues an ENQ for integrity so the active system BROADCAST dataset may be rebuilt. It is advisable that such a process be undertaken during low usage hours.

SYSUT1

If not specified, ACFBSYNC will use the online ACF2 Logonid database (implies ACF2 is active) as input. If the SYSUT1 file is specified, it must define a variable format dataset containing ACF2 Logonid records. This file may be produced either from the daily ACF2 backups or as the SAVEREC output from the ACFRPTSL report generator.

Parameters

Parameters are valid only when the online ACF2 Logonid database is being processed. These parameters are not valid if the SYSUT1 input file is specified.

LIKE(logonid-mask) - DEFAULT=***** (all)

Specifies that only those Logonid records that match the specified logonid-mask should be created in the SYS1.BROADCAST dataset.

UID(uid-mask) - DEFAULT=-

Specifies that only those Logonid records whose UID string matches the specified uid-mask should be created in the SYS1.BROADCAST dataset.

IF(attribute-name-1,attribute-name-2,...attribute-name-n) -
DEFAULT=NONE

Specifies that only those Logonid records which have the specified attribute fields "on" in their Logonid record should be added to the SYS1.BROADCAST dataset. The IF parameter may be specified as one attribute-name or with multiple attribute-names separated by commas. Multiple attribute-names are processed using AND logic; meaning that an ACFBSYNC parameter of "IF(TSO,ACCOUNT)" causes Logonid records that have both the TSO and ACCOUNT attributes to be added to the SYS1.BROADCAST dataset. Any field defined as a bit field in the Logonid record may be specified in the IF parameter.

The IF parameter may be used in conjunction with the LIKE and UID parameters. This allows the installation to further define the ACFBSYNC selection criteria.

Example

SAMPLE JOBSTREAMS TO RUN ACFBSYNC -

```
//SYNCH   JOB1      (ADD ALL ACF2 LOGONIDS TO BROADCAST)
//SYNCH   EXEC PGM=ACFBSYNC
//SYSLBC  DD   DSN=SYS1.BROADCAST,DISP=OLD

//SYNCH   JOB1      (ADD ONLY LOGONIDS WITH TSO ATTRIBUTES TO BROADCAST)
//SYNCH   EXEC PGM=ACFBSYNC,PARM='IF(TSO)'
//SYSLBC  DD   DSN=SYS1.BROADCAST,DISP=OLD

//SYNCH   JOB1      (ADD LOGONIDS BASED ON INPUT FILE)
//SYNCH   EXEC PGM=ACFBSYNC
//SYSLBC  DD   DSN=SYS1.BROADCAST,DISP=OLD
//SYSUT1  DD   DSN=INPUT.LOGONIDS,DISP=SHR
```

- NOTE: 1. For ACFBSYNC processing the USERID is restricted to be no more than seven characters in length. Any Logonid in the input to ACFBSYNC that is eight characters in length will be bypassed.
2. All messages produced by ACFBSYNC will appear in the executing user's JOBLOG. Consult the acf2/MVS Messages Manual for an explanation of the messages.
3. Upon completion, ACFBSYNC will issue termination messages identifying the number of Logonids input and the number placed in the BROADCAST dataset.
4. Parameters are valid only when the online ACF2 Logonid database is being used as input.

OTHER ACF2 UTILITIES

This chapter discusses the following ACF2 utilities:

<u>Utility Name</u>	<u>Type of Utility</u>	<u>Purpose</u>
ACFSUB	TSO command	Controlled submission of batch jobs.
JOBCOPY	Batch program	Controlled submission of batch jobs.
ACFERASE	Batch program	File erasure.
ACFDEL	TSO command	File erasure.

ACFSUB - TSO PRODUCTION JOB SUBMITTER

ACFSUB is a TSO command issued from TSO ready mode. This utility may be used to submit controlled production-type and other special job streams under a Logonid other than the one of the TSO operator. ACFSUB verifies that the TSO operator submitting the job has the authority to submit job streams via ACFSUB out of the referenced JCL library. In addition, this utility dynamically creates the Logonid that the new job will run under. The default version of ACFSUB provided with ACF2 creates a Logonid based on the first JOB card accounting field parameter. This utility can be modified locally to assign or build the Logonid with any other formula desired.

Note that ACFSUB is similar to ACF2's sample JOBCOPY utility for batch production job submissions.

Files

INTRDR

The Internal Reader is dynamically allocated/deallocated when the INTRDR option is selected.

jcllib1 - jcllibn

These files are the libraries containing the controlled job streams (JCL) to be submitted from TSO via ACFSUB. The full library dataset names must be pre-defined to ACFSUB. For instructions on defining these libraries, refer to the explanation of ACFSUB in the acf2/MVS System Programmer's Guide.

Parameters

'member-list' - DEFAULT=none

This is a required, positional parameter (always the first one specified). It indicates the name or list of names of PDS members that contain the JCL to be submitted.

LIB('pds-id') - DEFAULT=Primary JCL Library Defined

This optional parameter specifies a 1-8 character id of the PDS that will be accessed for the indicated members. This PDS id must be pre-defined to ACFSUB. Note that the full JCL library dataset name is not used here. If no PDS id is specified, ACFSUB will use the primary (or first) PDS id that has been pre-defined.

TERMINAL/INTRDR - DEFAULT=INTRDR

Specifies whether the requested job streams are to be submitted to the job entry system via the internal reader (INTRDR) or just displayed on the terminal (TERMINAL) for review.

Syntax Examples

The full syntax of the ACFSUB TSO command is:

```
ACFSUB 'member-list' LIB('pds-id') TERMINAL/INTRDR
```

For example, suppose a user has the authority to use either the test or production library. This user issues the ACFSUB command to submit a jobstream from the member ACCTG1 of the JCL library TEST:

```
ACFSUB ACCTG1 LIB(TEST)
```

Before this user can execute ACFSUB, the installation must do the following:

1. First pre-define which libraries the ACFSUB command can reference. To do so, the installation modifies the module ACFSCMD, which resides in the library SYS1.ACFMAC. In a table within this module, the installation creates an entry for each JCL library to be pre-defined.

For example, assume that an entry specifies a pds-id of PROD for a library named SYS1.JCLLIB.CNTL. An authorized user can issue the following command to submit a jobstream at the terminal from the ACCTG1 member of SYS1.JCLLIB.CNTL: terminal:

```
ACFSUB ACCTG1 LIB(PROD) TERMINAL
```

The entry in the module ACFSCMD also specifies the attribute that a user must have in his Logonid record in order to issue ACFSUB. This attribute may be specified through an installation-defined Logonid record field.

2. Update the restricted TSO command list (IKJEFTE2) so that it contains the module name ACFSUB.

Installations may wish to change the method used to create special Logonids for the submitted jobs (i.e., use something other than the first job accounting field to create the Logonid).

Also, all Logonids associated with a jobstream submitted via this command processor should be defined with the following attributes:

```
RESTRICT SUBAUTH PROGRAM(ACFSUB)
```

For further information on ACFSUB and the restricted TSO command list, refer to the section on ACF2/TSO facilities in the acf2/MVS System Programmer's Guide.

JOBCOPY UTILITY - BATCH PRODUCTION JOB SUBMISSION

The JOBCOPY utility can be run as a batch program or started task.

It allows for submission of production and other special types of job streams that need to run under a Logonid other than that of the TSO operator. This utility verifies that the user submitting the job stream has the authority to submit job streams via JOBCOPY from the referenced JCL library.

JOBCOPY also creates a Logonid under which the job can run. Under the default version of JOBCOPY supplied with ACF2, this Logonid is based on the account-number parameter of the first JOB card encountered. This default can be modified locally to build the Logonid by a method other than the default one.

Files

The JOBCOPY utility uses these files with the following DD names and descriptions:

SYSUT1

This file contains the production job stream to be processed by JOBCOPY.

SYSUT2

This file will contain the processed job stream that will be submitted through JOBCOPY to the internal reader. Thus, this file should be assigned to the internal reader.

Parameters

The JOBCOPY utility takes the following required parameter:

'member-name' - DEFAULT=none (Required parameter)

This parameter indicates the names of PDS members from which jobs will be submitted.

Examples

When JOBCOPY is submitted in batch, the jobstream might look like:

```
//PROD EXEC PGM=JOBCOPY  
//SYSUT1 DD DSN=file.name.here,DISP=SHR  
//SYSUT2 DD SYSOUT=(A,INTRDR)
```

When JOBCOPY is run as a started task from the operator console, the operator command might look like:

S JOBCOPY,JOB=member-name

The member name refers to the file that contains the production job to be submitted.

Note that in batch or as a started task, JOBCOPY can submit only one jobstream at a time.

Prerequisites for the Use of JOBCOPY

Before using JOBCOPY, an installation must define any libraries containing jobstreams that will be submitted through JOBCOPY. For instructions on defining these libraries, refer to the explanation of JOBCOPY in the acf2/MVS System Programmer's Guide.

ACFERASE - THE DATA DISPOSAL UTILITY

The ACFERASE utility will remove all data from an allocated, non-VSAM, direct access dataset or erase an entire tape volume. This is an important function since new dataset allocation can overlay an old data space, and thereby accidentally provide users access to the sensitive data that was previously stored there. For VSAM datasets, the Access Method Services utility (IDCAMS) should be used to provide this function when the VSAM dataset is defined.

For direct access datasets, ACFERASE erases each track allocated to the dataset. ACFERASE processes the entire data space regardless of logical end of dataset indicators.

For tape datasets, the ACFERASE utility uses the data security erase command to erase the entire volume from the load point (ACFERASE specifically rewinds the tape to the load point). Note that use of the data security erase operation does not inhibit the control unit from processing requests for other tape drives while it is in progress. For 2400 series tape drives which do not support DSE, ACFERASE will write tape marks across the entire tape volume.

Files

DDxxxxxx

ACFERASE selects all DD cards with the first two characters of 'DD'. Any number of DD cards may be coded; all will be processed.

SYSIN

The SYSIN file may be used to specify input parameter information for all of the ACFERASE parameters. Note that ACFERASE can accept parameter input from either the SYSIN file or the JCL "PARM=" field. If the SYSIN file is not defined and ACFERASE requires parameter specifications via the SYSIN file, then an error is generated.

The following specifications apply to the SYSIN file. The SYSIN input file must have a record format of F, FB, or VB. For fixed format records the last 8 columns are assumed to contain sequence numbers and are ignored. For variable format records, the first 8 characters are assumed to be the sequence field and are ignored. The hyphen character ('-') may be specified as the last non-blank character of an input record to indicate that the next input record is to be considered a continuation of the current record. The concatenation process proceeds such that the first character of the next record takes the position of the hyphen and all subsequent characters will follow thereafter; blanks after the hyphen on the record to be continued are ignored.

If ACFERASE is executed under an online system (such as TSO at MVS sites via the CALL command), one of the following processing modes may be invoked:

1. SYSIN input is required by ACFERASE: input will be obtained from either the dataset to which the SYSIN file is allocated or, if the SYSIN file is not allocated, input will be obtained from the usage terminal.
2. SYSIN file is an optional extension of JCL parm field: input is accepted only if the last non-blank character of the CALL program parameter operand is a hyphen:
 - This requires coding the CALL parameter operand in quotes.
 - If no hyphen is coded, any SYSIN allocation is ignored.
 - If the hyphen is coded, input is taken from the dataset to which the SYSIN file is allocated, or from the terminal.
 - The hyphen specified in the "PARM=" field is not used to indicate continuation of the input parm field, only to flag the usage of the SYSIN file.

Any messages from ACFERASE will appear in the user's JOB LOG.

Parameters

In batch operation ACFERASE accepts the parameters listed below. The DISP= parameter may be used to uncatalog and/or scratch the dataset after ACFERASE processing completes; however, since JCL DISP processing will occur regardless of ACFERASE processing, it is suggested that datasets be scratched in a separate job step that will test the condition code set by ACFERASE. If no parameters are specified for ACFERASE the default is 'ERASE(ALL)'.

ERASE[(ddname) or ALL] - DEFAULT=ALL

Requests the erasure of all data and all labels. For tape datasets, erasure will occur from the load point to the end-of-volume; for 2400 series tape drives, ACFERASE will write tape-marks across the entire volume. For direct access datasets, ACFERASE will erase all data allocated to the data space regardless of logical end of dataset indicator. NOTE: ERASE is the only parameter allowed for processing direct access datasets (DASD).

SLERASE[(ddname) or ALL] - DEFAULT=NONE

Requests that ACFERASE retain the VOL1 label, create a new HDR1 label, and erase all other data on the tape. SLERASE is valid only for tape datasets.

INIT[(ddname) or ALL,(new-volser),('owner-id')] - DEFAULT=NONE
Requests that ACFERASE erase all data on the tape, re-initialize the VOL1 label, and create a new HDR1 label. All parameters must be specified. If ALL is specified, each tape processed will be initialized with identical VOL1 and HDR1 labels. The INIT parameter is valid only for tape datasets.

RELABEL[(ddname),(old-volser),(new-volser),('owner-id')]
Requests that ACFERASE validate the old-volser, erase all data and labels, create a new VOL1 label using the specified information, and create a new HDR1 label. All parameters must be specified (ALL is not valid). RELABEL is valid only for tape datasets.

Examples

Erasing disk datasets:

```
//DISK EXEC PGM=ACFERASE,PARM='ERASE(ALL)'  
//DD1 DD DSN=SOME.DATA,DISP=(OLD,DELETE)  
//DD2 DD DSN=SOME.DATA.SECND,DISP=(OLD,DELETE)
```

Retaining label on tape volume:

```
//SLTAPE EXEC PGM=ACFERASE,PARM='SLERASE(ALL)'  
//DD3 DD UNIT=TAPE,VOL=SER=VOLO01,LABEL=(,BLP)  
//DD4 DD UNIT=TAPE,VOL=SER=VOLO02,LABEL=(,BLP)
```

Initializing tape volume:

```
//INTAPE EXEC PGM=ACFERASE,PARM='INIT(DD5,TO1005,'USER123')'  
//DD5 DD UNIT=TAPE,VOL=SER=TO1005,LABEL=(,BLP)
```

Relabeling tape volume:

```
//RELTAPE EXEC PGM=ACFERASE,  
// PARM='RELABEL(DD6,VOLO03,VOL103,'USER456')'  
//DD6 DD UNIT=TAPE,VOL=SER=VOLO03,LABEL=(,BLP)
```

If condition codes are to be tested:

```
//ERASE EXEC PGM=ACFERASE  
//DD7 DD DSN=PRIVATE.DATA,DISP=(OLD,KEEP)  
//SCRTCH EXEC PGM=IEFBR14,COND=(0,NE,ERASE)  
//DD8 DD DSN=PRIVATE.DATA,DISP=(OLD,DELETE)
```

Note: If no parameters are specified as in the above example the default is 'ERASE(ALL)'.

ACFDEL - TSO DATA DISPOSAL COMMAND

The ACFDEL command can remove all data from allocated, non-VSAM, direct-access datasets, or from an entire volume.

Syntax

This command, issued from TSO READY mode, has the following syntax:

```
ACFDEL dsn [SCRATCH/NOSCRATCH] [CATALOG/UNCATALOG]
          [ERASE/NOERASE] [VOLUME(volser)]
```

Parameters

The ACFDEL command takes the following parameters:

dsn

Positional parameter, the name of the dataset to be processed by the ACFDEL command.

SCRATCH/NOSCRATCH - DEFAULT=SCRATCH

Specify SCRATCH if the dataset is to be scratched from disk after any ACFDEL processing. The dataset will be uncataloged based on the UNCATALOG parameter.

UNCATALOG/NOUNCATALOG - DEFAULT=UNCATALOG

Specifies that the dataset is to be uncataloged after any ACFDEL processing. The UNCATALOG and SCRATCH parameters operate independently so a dataset may be uncataloged and not scratched or scratched and not uncataloged if the user wishes.

ERASE/NOERASE - DEFAULT=ERASE

The ERASE parameter requests the ACFDEL command erase the entire data space occupied by the dataset being processed. This is the same function described under the ACFERASE utility above.

VOLUME(volser) - DEFAULT=as cataloged

Specifies the volume serial on which the dataset to be processed resides. If the dataset is not cataloged the VOLUME parameter must be specified. If the dataset is cataloged and the VOLUME parameter is specified, the catalog information is ignored.

Example

ACFDEL SOME:DATA - Will erase, scratch, and uncatalog.
ACFDEL SOME.DATA NOUNCAT NOSCRATCH
- Will erase the data space only.

INDEX

- @BACKUP macro ... 132
- @DDSN macro ... 132
- Access codes
 - ACFRPTRX ... 102
 - ACFRPTXR ... 116
- Access rule
 - ACFBATCH ... 151
 - batch compilation ... 155
 - TSO individual rule processing ... 159
 - TSO rule compilation ... 152
- ACF Command
 - operator ... 132-134
- ACFBATCH utility
 - examples ... 151
 - files ... 150
 - parameters ... 151
- ACFBCOMP command ... 155-156
 - examples ... 155
 - files ... 155
- ACFBDCMP utility ... 158
 - examples ... 157
 - files ... 157
- ACFBKUP utility ... 147
 - execution ... 134
- ACFBSYNC utility ... 162
 - examples ... 162
 - files ... 161
 - parameters ... 161
- ACFCOMP command ... 152-154
 - ACFBATCH ... 151
 - methods ... 152
 - parameters ... 153
 - syntax ... 152
- ACFDEL command ... 171
 - examples ... 172
 - parameters ... 171
 - syntax ... 171
- ACFERASE utility ... 170
 - examples ... 170
 - files ... 168
 - parameters ... 169
- ACFNRULE utility
 - examples ... 160
 - files ... 159
 - parameters ... 159
- ACFRLCVR utility
 - execution ... 133
 - field descriptions ... 137
 - files ... 135
 - parameters ... 135
 - sample jobstream ... 143
 - sample output ... 137
- ACFRPTCR report generator
 - examples ... 19
 - field descriptions ... 22
 - files ... 17
 - parameters ... 17
 - sort sequence ... 18
- ACFRPTDS report generator
 - field descriptions ... 35
 - files ... 25
 - NEXTKEY reporting ... 33
 - parameters ... 25
 - sample output ... 30
 - sort sequence ... 29
- ACFRPTEL report generator
 - files ... 44
 - parameters ... 44
 - sample output ... 47
 - sort sequence ... 45
- ACFRPTIX report generator
 - files ... 52
 - parameters ... 52
 - sample output ... 53
- ACFRPTJL report generator
 - files ... 56
 - parameters ... 56
 - sample output ... 58
 - sort sequence ... 57
- ACFRPTLL report generator
 - files ... 61
 - parameters ... 61
 - sample output ... 62
- ACFRPTNV report generator
 - field descriptions ... 70
 - files ... 67
 - parameters ... 67
 - sample output ... 69
 - sort sequence ... 68
- ACFRPTPP report generator
 - field descriptions ... 79
 - files ... 72
 - parameters ... 73
 - sample output ... 77

- ACFRPTPW report generator
 - files ... 80
 - parameters ... 80
 - sample output ... 82
 - sort sequence ... 81
- ACFRPTRL report generator
 - files ... 85
 - sample output ... 86
- ACFRPTRV report generator
 - field descriptions ... 91
 - files ... 87
 - parameters ... 87
 - sample output ... 89-90
 - sort sequence ... 88
- ACFRPTRX report generator
 - field descriptions ... 99
 - files ... 96
 - Logonid access report ... 95
 - message lines ... 102
 - parameters ... 98
 - reason codes ... 102
 - sample output ... 104-105
- ACFRPTSL report generator
 - ACF2 inactive ... 16
 - field descriptions ... 114
 - files ... 107
 - parameters ... 108
 - sample output ... 112
- ACFRPTXR report generator
 - access reason codes ... 116
 - ACF2 inactive ... 16
 - exit considerations ... 115
 - field descriptions ... 122
 - files ... 117
 - parameters ... 120
 - sample output ... 125
- ACFSUB command
 - examples ... 164
 - files ... 164
 - parameters ... 164
- ACF2 parameter
 - ACFRPTRX ... 98
 - ACFRPTXR ... 120
- ADD parameter
 - ACFNRULE ... 159
- ALL parameter
 - ACFRPTDS ... 28
- Alternate VSAM clusters
 - recovery ... 133
- Backup
 - alternate clusters ... 133
 - automatic facility ... 132
 - general information ... 132
 - restore primary
 - clusters ... 134
- Batch utilities
 - ACFBATCH ... 150
 - ACFBCOMP ... 155
 - ACFBDCMP ... 157
 - ACFBSYNC ... 161
 - ACFERASE ... 168
 - ACFNRULE ... 159-160
 - JOBCOPY ... 166
- BUFFER parameter
 - ACFRPTCR ... 18
 - examples ... 19
- Clusters
 - alternate ... 133
 - restoring primary ... 134
- COMPILE subcommand
 - batching ... 156
- CPUID
 - ACFRPTNV ... 67
- Cross-Reference Report
 - see ACFRPTXR ... 115
- Data disposal
 - TSO command ... 171
 - Utility program ... 168
- Databases
 - recovery ... 131
- Dataset
 - cross-reference report ... 115
 - event log ... 25
 - index report ... 52
- DBLSPC parameter
 - ACFRPTNV ... 68
- DELETE parameter
 - ACFNRULE ... 159
- DETAIL parameter
 - ACFRPTTEL ... 45
 - ACFRPTIX ... 53
 - ACFRPTLL ... 62
- DSET parameter
 - ACFRPTRX ... 98
 - ACFRPTXR ... 121
- DSN parameter
 - ACFDEL ... 171
 - ACFRPTXR ... 121

- ERASE parameter
 - ACFDEL ... 171
 - ACFERASE ... 169
- Execution of other ACF2 utilities ... 11
 - TSO CALL command ... 13
- Execution of report generators
 - ACF2 inactive ... 16
 - JCL ... 2
- Exits
 - entry points ... 130
 - standard ... 95, 115
- EXTEND parameter
 - ACFRPTDS ... 29
- Generalized Resource Event Log
 - see ACFRPTRV ... 87
- HEADER parameter
 - ACFRPTNV ... 68
- ID parameter
 - ACFRPTTEL ... 45
 - ACFRPTRV ... 88
- IDCAMS utility
 - ACFERASE ... 168
- IF parameter
 - ACFRPTSL ... 109
- Information Storage Update Log
 - see ACFRPTTEL ... 44
- INIT parameter
 - ACFERASE ... 169
- INPUT parameter
 - ACFRPTSL ... 108
- Input/output files
 - report generators ... 3
- INSTALL parameter
 - ACFRPTDS ... 28
- Installation exits
 - report generators ... 129
- INTRDR parameter
 - ACFSUB ... 164
- Invalid Password/Authority Log
 - see ACFRPTPW ... 80
- ISPF
 - report generator
 - execution ... 8
 - utility execution ... 11
- JCL
 - batch submission
 - utility ... 166
 - database recovery ... 143
 - executing report
 - generators ... 2
 - utility execution ... 11
- JOBCOPY utility
 - examples ... 166
 - files ... 166
 - parameters ... 166
- JOBMASK parameter
 - ACFRPTCR ... 18
 - ACFRPTDS ... 26
 - ACFRPTTEL ... 45
 - ACFRPTIX ... 53
 - ACFRPTJL ... 56
 - ACFRPTLL ... 61
 - ACFRPTPP ... 74
 - ACFRPTPW ... 80
 - ACFRPTRL ... 85
- KEY parameter
 - ACFNRULE ... 159
- KEYEXCES
 - error reporting ... 33
- LAB field
 - ACFRECVR ... 137
 - ACFRPTJL ... 60
 - ACFRPTLL ... 65
 - ACFRPTPW ... 84
- LIB parameter
 - ACFSUB ... 164
- LID field
 - ACFRPTXR ... 121
- LID parameter
 - ACFRPTRX ... 98
- LIDMASK parameter
 - ACFRPTDS ... 26
- LIST parameter
 - ACFNRULE ... 160
- LOG parameter
 - ACFRPTRV ... 87
- LOGGING parameter
 - ACFRPTDS ... 27
- Logonid Access Report
 - see ACFRPTRX ... 95
- Logonid database
 - alternate ... 97, 120
 - recovery ... 134
 - synchronization utility ... 161
- Logonid Modification Log
 - see ACFRPTLL ... 61

- LOGSHIFT field
 - access reporting ... 80
- MASK parameter
 - ACFRPTCR ... 18
 - ACFRPTDS ... 26
 - ACFRPTTEL ... 45
 - ACFRPTJL ... 56
 - ACFRPTLL ... 61
 - ACFRPTPP ... 73
 - ACFRPTPW ... 80
 - ACFRPTRL ... 85
 - ACFRPTRV ... 88
 - ACFRPTSL ... 108
- MAXRULE parameter
 - ACFCOMP command ... 152
- Member-list parameter
 - ACFSUB ... 164
- NAME parameter
 - ACFRPTXR ... 122
- New Rule Utility
 - see ACFNRULE ... 159
- NEXTKEY
 - excessive condition ... 33
 - looping error ... 33
 - sample trace report ... 33
- NKEYLOOP
 - looping error ... 33
- NLIDMASK parameter
 - ACFRPTDS ... 26
- NMASK parameter
 - ACFRPTDS ... 26
- NOACF2 parameter
 - ACFRPTRX ... 98
 - ACFRPTXR ... 120
- NOBUFFER parameter
 - ACFRPTCR ... 18
- NODBLSPC parameter
 - ACFRPTNV ... 68
- NOEXTEND parameter
 - ACFRPTDS ... 29
- NOLAB parameter
 - ACFRECVR ... 137
- NOLID parameter
 - ACFRPTXR ... 121
- NORRSUM parameter
 - ACFRPTXR ... 121
- NOUPDATE parameter
 - ACFRPTSL ... 108
- NOUPPER parameter
 - ACFRPTCR ... 18
- Operator commands
 - backup ... 134
 - backup initiation ... 132
 - recovery ... 133
- Parameters
 - continuation ... 4
 - report generators ... 5
- PARM parameter
 - ACFBATCH ... 151
- Password
 - violation report ... 80
- Planning
 - recovery ... 131
- Pre-Processor utility
 - see ACFRPTPP ... 71
- PREFIX parameter
 - ACFRPTIX ... 53
- Primary VSAM clusters
 - backup procedure ... 132
- PRINTER parameter
 - ACFRPTDS ... 28
 - ACFRPTRV ... 88
- Production jobs
 - batch submission ... 166
 - TSO submission ... 164
- Programs
 - event log ... 25
- Recovery of ACF2 databases
 - ACFBKUP ... 147
 - ACFRECVR ... 134
 - general information ... 131
 - planning ... 131
 - procedure ... 133
 - sample jobstream ... 143
- RECxxxxx files
 - ACFRPTPP ... 72
 - records ... 14
 - report generators ... 3
- RELABEL parameter
 - ACFERASE ... 170

- Report generators
 ACFRPTCR ... 17-18
 ACFRPTDS ... 25-26, 28, 30-32, 35-37, 39-41
 ACFRPTTEL ... 44, 47
 ACFRPTIX ... 52-55
 ACFRPTJL ... 56, 58-59
 ACFRPTLL ... 61
 ACFRPTNV ... 67-68, 70
 ACFRPTPP ... 71, 73
 ACFRPTPW ... 80, 82, 84
 ACFRPTRL ... 85-86, 86.2
 ACFRPTRV ... 87-94
 ACFRPTRX ... 95-98, 104-106
 ACFRPTSL ... 107-114
 ACFRPTXR ... 115-116, 119-127
 execution authorization ... 5
 input records ... 14
 ISPF screens ... 8
 sort sequences ... 15
- REPORT parameter
 ACFRPTSL ... 109
- Restricted Logonid Job Log
 see ACFRPTJL ... 56
- RKEY parameter
 ACFRPTXR ... 121
- RMASK parameter
 ACFRPTRX ... 99
- RRSUM parameter
 ACFRPTXR ... 121
- RSRC parameter
 ACFRPTRX ... 98
 ACFRPTXR ... 121
- Rule-id Modification Log
 see ACFRPTRL ... 85
- SCRATCH parameter
 ACFDEL ... 171
- SELECT parameter
 ACFRPTPP ... 76
- Selected Logonid List
 see ACFRPTSL ... 107
- SELLID parameter
 ACFRPTIX ... 52
- SELRULE parameter
 ACFRPTIX ... 53
- SFLDS parameter
 ACFRPTSL ... 109
- SHORT parameter
 ACFRPTDS ... 27
- SIZE parameter
 ACFRPTDS ... 26
- SLERASE parameter
 ACFERASE ... 169
- SMFxx parameter
 ACFRPTPP ... 74
- SMFxxxxx files
 ACFRPTPP ... 72
- Sort sequence
 ACFRPTCR ... 18
 ACFRPTDS ... 29
 ACFRPTTEL ... 45
 ACFRPTJL ... 57
 ACFRPTNV ... 68
 ACFRPTPW ... 81
 ACFRPTRV ... 88
 report generators ... 15
- SPF
 see ISPF ... 8
- Subcommands
 ACFBATCH ... 151
- SUMMARY parameter
 ACFRPTDS ... 28
 ACFRPTTEL ... 45
 ACFRPTLL ... 62
- SYSHELP file
 ACFBATCH ... 150
- SYSIN file
 ACFBATCH ... 150
- SYSLBC file
 ACFBSYNC ... 161
- SYSPRINT file
 ACFBATCH ... 150
 ACFRPTPP ... 72
- System Management Facilities (SMF)
 records ... 72
- SYSTSIN file
 ACFBATCH ... 150
- SYSTSPRT file
 ACFBATCH ... 150
- TERMINAL parameter
 ACFRPTDS ... 28
- TRACE parameter
 ACFRPTDS ... 27
 ACFRPTNV ... 68
 ACFRPTRV ... 87
- TSO
 access rule commands ... 152
 ACFDEL command ... 171
 ACFNRULE command ... 159
 ACFSUB command ... 164
 CALL command utility
 execution ... 13
 command statistics log ... 16

TYPE parameter

ACFBCOMP ... 155
ACFNRULE ... 159
ACFRPTEL ... 44
ACFRPTRV ... 88
ACFRPTRX ... 99
ACFRPTXR ... 122

UID parameter

ACFRPTCR ... 18
ACFRPTDS ... 26
ACFRPTRX ... 99

UNCATALOG parameter

ACFDEL ... 171

UNKNOWN parameter

ACFRPTDS ... 28

UPDATE keyword

ACFRPTLL ... 61
ACFRPTSL ... 108

UPPER parameter

ACFRPTCR ... 18

Utilities of ACF2

ACFBATCH ... 150-151
ACFBCOMP ... 155
ACFBDCMP ... 157-158
ACFBKUP ... 147-148
ACFBSYNC ... 161-162
ACFCOMP ... 152
ACFDEL ... 171
ACFERASE ... 168-170
ACFNRULE ... 159-160
ACFRECVR ... 134-135, 137,
140-141, 143-146
ACFSUB ... 164
JOBCOPY ... 166
security administration ... 149

VERIFY parameter

ACFNRULE ... 160

VIO parameter

ACFRPTDS ... 27
ACFRPTRV ... 87

VOL parameter

ACFRPTXR ... 122

VOLUME parameter

ACFDEL ... 171