CA ACF2™ for z/OS

Quick Reference Guide

r12

Third Edition

Contact CA

Contact Technical Support

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

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

Provide Feedback

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

If you would like to provide feedback about CA product documentation, complete our short customer survey, which is also available on the CA Support website, found at http://ca.com/docs.

Contents

Chapter 1: ACF Command

This section contains the following topics:

Using Common ACF Subcommands

The ACF Command and Subcommands let you create and maintain the major components of CA ACF2. There are also ISPF panels that perform the same functions as the TSO ACF command. For more on the ISPF panels, see the *Administrator Guide*.

The ACF command provides subcommands to process CA ACF2™ for z/OS (CA ACF2) rules and records.

Issue the ACF command and its subcommands:

- **Online under TSO.** After issuing the ACF command under TSO, enter individual subcommands.
- **In batch.** The ACFBATCH utility is the batch equivalent of the ACF command. Specify a sequence of ACF subcommands or execute the ACF command under TMP in background.
- **Online under CICS.** The ACFM transaction is equivalent to the ACF command. Enter individual subcommands using ACFM.
- **In special transactions.** Under IMS, use the ACF command and subcommands for logonid record processing.

To continue an ACF command to another line, place a blank space and a dash (-) at the end of the line. For example:

```
COMPILE MY.DATASET LIST STORE –
ALL
```

The default setting of the ACF command is ACF. All administrative functions available under the LID and RULE settings are available under ACF.

HELP Subcommand

You can issue the HELP subcommand at any time while the ACF command is active, except while compiling or testing a rule. The HELP subcommand provides online descriptions of the setting in which you are processing and the ACF subcommands under that setting.

HELP subcommand syntax:

```
Help [ACCess]
     [ACFRSrc]
     [ACFRUles]
     [CHAnge]
     [CHKcert]
     [COMpile]
     [CONnect]
     [DEComp]
     [DELete]
     [ENd]
     [EXport]
     [FIELDS]
     [F ACF2/F]
     [GENCert]
     [GENReq]
     [GSO]
     [Help]
     [INsert]
     [List]
     [MLSlabel]
     [MLWrite]
     [MODE]
     [REKey]
     [REMove]
     [ROllover]
     [SEt|T]
     [SHow]
     [SN]
     [STore]
     [SYNch]
     [TEst]
```

For example, to view a description of an ACF subcommand under the setting you are in, enter HELP and then the subcommand name.

```
help change
```

SN Subcommand

This subcommand interfaces with the TSO SEND command. It enables communication between user terminals or between a user terminal and a LAN operator console.

SN subcommand syntax

```
SN 'message'   [Operator(2|routecode)]
               [User(*|logonid,...,logonid)]
               [CN(consoleid)]
               [NOW|Logon|Save]
               [NOWait|Wait]
```

You can issue this subcommand under any setting of the ACF command. For a description of SN subcommand parameters, see the description of the TSO SEND command in the IBM *TSO Command Language Reference* guide.

SET Subcommand

The SET subcommand lets you establish the setting of the ACF subcommand. It can also modify the operation of other subcommands.

SET subcommand syntax

Syntax	Description
Process logonid records or access rule sets (default setting)	SEt|T {ACF}
Process CAC, CPF, GSO, NET, SMS, SSO, or TSO records	SEt|T Control(CAc|CPf|Gso|LDS|Net|SMs|SSo|Tso)
Group Infostorage records for a specific system (same as SYSID)	SEt|T {DIVision(div)| MDIVision(divmask)}
Process input source and source group records	SEt|T Entry{SRC|SGP}
Process field records	SEt|T Field(EXPressn|RECord)
Store rule sets even though they exist	SEt|T {FOrce|NOForce}
Process user identification device options	SEt|T Identity(Aut)
Process logonid records	SEt|T Lid
Specify CA ACF2 generated member name of a PDS during a decompile	SEt|T MEmber(nnnnnnn)
Reset the error indicator	SEt|T NOError
Clear locally resident rules	SEt|T NORule(jobname|ALL)
Process profile records	SEt|T PROFILE(profiletype)
Process resource rules	SEt|T Resource(typecode)

Syntax	Description
Process access rules	SEt|T Rule
Process scope records	SEt|T SCope(SCP)
Process security labels assigned to data sets	Set|T SECLABEL(DSN)
Process shift or zone records	SEt|T SHift(SFT|ZON)
Group Infostorage records for a specific system	SEt|T SYSid(?|sysid)| MSYSid(?|sysidmask)
Identify nodes where updates to the Logonid or Infostorage database can take place	SEt|T TArget(null|=|?|node,…,node)
Display short version of records and rule sets	SEt|T {TErse|Verbose}
Display all fields of logonid record	SEt|T {TRivia|NOTrivia}
Process cross-reference IMS DL/I records or source/resource group records	SEt|T Xref(IDS|Rgp|Sgp)

Displaying System Options—SHOW Subcommand

The SHOW subcommand lists information about CA ACF2 as it is currently running on your system. You cannot issue a SHOW subcommand to display data about a remote system.

SHOW Subcommand Syntax

Function	Command Syntax
Displays all ACF2 system parameters except FIELDS and MODE	SHow {ACF2|ALL}
Displays ACF2 intercepts	SHow ACTive
Displays all user-defined structured infostorage applications	SHow APpledef
Displays records as laid out in the internal CERTMAP table. The display first shows data from records that contain both IDNFILTER and SDNFILTER, followed by records with only SDNFILTER, and finally records with only IDNFILTER.	Show CErtmap

Function	Command Syntax
Displays internal and external CLASMAP records	SHow CLasmap
Displays information about CPF OPTIONS record and CPF network	SHow CPf
Displays information contained in CRITMAP records. The display shows the record id, SYSID, APPLID, USERID, and associated application variables.	SHow CRitmap
Displays startup ACF2/DB2 information	SHow DB2
List data set names used for Logonid, Rule, and Infostorage databases	SHow DDsn
Displays the CA ACF2 delegated resources currently active in the system.	Show DElrsrc
Show all logonid fields that a user can display or modify (the default), or show all structured infostorage record fields for a particular record	SHow Fields[(recid)]
Display the associated search sequences for a given logonid or logonid mask	SHow LIDMAP(lid|lidmask)
Displays the LDS status, the active LDAP records and the CA ACF2 logonid field information that is propagated to an LDAP server.	SHow LDS
List library names specified in SYS1.LINKLIB	SHow LINklst
Displays the Linux machine definitions currently in use.	SHow LINUx
Displays information about multilevel security (MLS) on a system including the following: the MLS status, the current system ID, the MLS options that are in effect, and the active levels, categories and security labels.	SHow MLS [STATUS|ALL|LEVELS| CATEGORIES|SECLABELS (ALPHA|LOW-HI|HI-LOW)]
Display current ACF command setting or mode	SHow Mode
Displays the GSO MLID records and the @MUSASS macros in the ACFFDR.	SHow Musass
Display the active nodes defined in NETNODE records	SHow NETNODE(nodemask)

Function	Command Syntax
Display the active DDB options defined in the NETOPTS record	SHow NETOPTS
Display all NJE records and options specified for each node	SHow NJe[(nodename)]
Display NODELIST records defined on the system	SHow NOdelist[ALL|LDS]
Displays OpenEdition MVS users and/or groups.	SHow OMVS[ALL|GROUPS(mmmm[-nnnn]) |SUPERUSERS| USERS(mmmm[-nnnn])
Displays programs or logonids permitted to bypass system integrity or access rule validation. Also, list the programs for which data set access is logged.	SHow {PROGrams|PGms}
Displays the GSO REALM records that are defined on the system.	SHow REAlm
List resident resource directories and access rules	SHow Resident
Display defined resource types	SHow RSRCTYPE
Display the words or prefixes not allowed in specification of a password	SHow RSVWORDS
Display SAFDEF records defined on the system	SHow SAFdef
Show the CA ACF2 system options in effect	SHow State
Display information pertinent to the SYSPLEX feature	SHow SYSPLEX
Display various system parameters, such as ACF2 SVC and SMF record numbers	SHow SYstems
Display the TNGNODEs on the system	SHow TNG
List default TSO options	SHow Tso
Displays UNIX options on the system.	SHow UNIXopts
Display logonid record fields that you cannot copy by using the ACF INSERT USING subcommand	SHow Zeroflds

END or QUIT Subcommand

The END or QUIT subcommand terminates the ACF command and displays the TSO READY message. Under the RULE and RESOURCE settings, you can also use the END subcommand to terminate the TEST and COMPILE subcommands.

END or Quit subcommand syntax:

Function	Command Syntax
End ACF processing	ENd|Quit

Chapter 2: Managing Logonid Records

This section contains the following topics:

Processing Logonid Database Records

Process Logonid database records under the ACF or LID setting by entering the following subcommands:

Function	Command Syntax
Begin ACF command processing	Set|T {ACF|Lid}
Add new logonid record	Insert [TArget(null|=|?|nodemask,…,nodemask)] {*|logonid field,…,field| Using(modellid) newlid field,…,field}
Display logonid record	List [TArget(null|=|?|nodemask,…,nodemask)] {*|logonid| Like(lidmask) [If(field,…,field)]| Uid(uidmask) [If(field,…,field)]| If(field,…,field)} [SECtion(name,…,name)] [PROFile(type,…,type)]
Change logonid or a group of logonids	CHAnge
Delete logonid record	DELete [TArget(null|=|?|nodemask,…,nodemask)] {*|logonid| LIke(lidmask) [If(field,…,field)]| Uid(uidmask) [If(field,…,field)]| If(field,…,field)} [NORULE]
Synchronize Logonid database with SYS1.BRODCAST	SYNCH {Like(lidmask) [If(field,…,field)]| Uid(uidmask) [If(field,…,field)]| If(field,…,field)}

Logonid record names are from one- to eight-characters in length.

To remove a bit field privilege, issue the CHANGE subcommand and prefix NO to the selected privilege. For example, specify NOACCOUNT to remove the ACCOUNT privilege. Extended IF support is available with the LIST command. All CA ACF2 Logonid record types can be tested (bit, character, packed, binary data, binary number.)

Describing Logonid Record Fields

An asterisk (*) after a field name indicates that this field is used only if UADS is bypassed. Fields listed without an asterisk are always used, regardless of UADS mode.

For more information about these fields, see the *Administrator* Guide.

ACC-CNT

This is the number of system accesses made by this logonid since it was created (4-byte binary).

ACC-DATE

This is the date of this user's last system access (4-byte binary).

ACC-SRCE

This is the logical or physical input source name where this logonid last accessed the system (8 characters).

ACC-TIME

This is the time of this user's last system access (4-byte binary).

ACCOUNT

The user can insert, delete, and change logonids, as limited by a scope (bit field).

ACCTPRIV*

This indicates the user has TSO accounting privileges (for UADS updates by the TSO ACCOUNT command) (bit field).

ACF2CICS

Indicates that CA ACF2 CICS security is to be initialized in any CTS 1.2 or higher region running with this address space logonid (bit field).

ACTIVE

The logonid is automatically activated one minute after midnight on the date contained in this field (4-byte binary).

ALLCMDS

The user can enter a special prefix character to bypass the CA ACF2 restricted command lists (bit field).

ATTR2*

The IBM program control faciltiy (PCF) uses the PSCBATR2 field for command limiting and data set protection (2 hexadecimal bytes).

AUDIT

With this privilege, a user can inspect, but not modify, the parameters of the CA ACF2 system. You can limit this privilege by a scope (bit field).

AUTHSUP1 through AUTHSUP8*

These fields can activate extended user authentication (EUA) for each designated system user (bit field).

AUTOALL

User can autolog any virtual machines without specifying a password. This field applies to VM sites only. (Bit field)

AUTODUMP

CA ACF2 takes an SVC dump when a data set or resource violation occurs—for debugging purposes only (bit field).

AUTONOPW

This virtual machine can be autologged without specifying a password (bit field).

AUTOONLY

This virtual machine cannot be logged on. It can be auto logged only (bit field).

BDT

This logonid's address space belongs to the Bulk Data Transfer (BDT) product. Certain privileges associated with BDT also extend to this address space (bit field).

CANCEL

The logonid is canceled and denied access to the system (bit field).

CHAR*

This indicates the TSO character-delete character for this user (1-byte binary).

CICS

The logonid has the authority to sign on to CICS (bit field).

CICSCL

This indicates the CICS operator class (3 hexadecimal bytes).

CICISID

This indicates the CICS operator ID (3 characters).

CICSOPT

Specifies the SYSID of the C-CIC records to use at initialization time. (eight-characters)

CICSPRI

This indicates the CICS operator priority (1-byte binary).

CICSRSL

This indicates the CICS resource access key (3 hexadecimal bytes).

CMD-LONG

This indicates that only the listed command and aliases are accepted when using TSO command lists. CA ACF2 does not accept an abbreviated character string (bit field).

CMD-PROP

This indicates that the user can override the global CPF target list by using the SET TARGET command or the TARGET parameter on the INSERT, CHANGE, LIST, or DELETE commands (bit field).

CONSOLE

The user access the TSO/E CONSOLE facility (bit field).

CONSULT

The user can display other logonids, as limited by a scope (bit field).

CRE-TOD

This is the date and time that this logonid record was created (8-byte binary).

CSDATE

This is the date when the CANCEL or SUSPEND field was set (mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on the DATE field of the GSO OPTS record) (4-byte binary).

CSWHO

This is the logonid that set the CANCEL, SUSPEND, or MONITOR field (8 characters).

DFT-DEST*

This is the default remote destination for TSO spun SYSOUT data sets (8 characters).

DFT-PFX*

This indicates the default TSO prefix that is set in the user's profile at logon time (8 characters—however, the last character is reserved).

DFT-SOUT

This is the default TSO SYSOUT class (1 character).

DFT-SUBC

This is the default TSO submit class (1 character).

DFT-SUBH*

This is the default TSO submit hold class (1 character).

DFT-SUBM

This is the default TSO submit message class (1 character).

DG84DIR

Permits a user to issue a diagnose 84 instruction. This field applies to VM sites only. (Bit field)

DIALBYP

Permits standard DIAL validation to be bypassed. This field applies to VM sites only. (Bit field)

DSNSCOPE

Specifies a logonid mask limiting the scope of SECURITY access. (Eight-byte character field).

Important! This is an old field from pre-4.0 releases of CA ACF2. This field should not be used. Instead, use the SCPLIST field in the Logonid record with Scope records to limit a user's administrative authority over the CA ACF2 Logonid, Rule, and Infostorage databases. For more information on how to create Scope records, see the chapter on "Maintaining Scope Records."

DUMPAUTH

This user can generate a dump even when the address space is in an execute-only or path control environment (bit field).

EXPIRE

This is the date when "temporary" logonids expire (4-byte binary).

GROUP

This field stores the group or project name associated with this user. The user enters this group at system access time (8 characters).

GRP-OPT

Designates an ID as an optional group ID. This field applies to VM sites only.

GRP-USER

Indicates the last user (logonid) to use the group virtual machine. This field applies to VM sites only.

Note: You cannot set this field. CA ACF2 displays and maintains this field.

GRPLOGON

Specifies that this virtual machine can be used by more than one person, while maintaining individual accountability. This field applies to VM sites only. (Bit field)

HOMENODE

CA ACF2 uses this field to keep track of the home node of a logonid that distributed database (DDB) support has acquired from a remote site. You cannot display or alter this field.

IDLE

This is the maximum number of minutes permitted between terminal transactions for this user (1-byte binary).

IMS

The logonid has the authority to sign on to IMS (bit field).

INTERCOM*

This user is willing to accept messages from other users through the TSO SEND command (bit field).

JCL*

This user can submit batch jobs from TSO and use the SUBMIT, STATUS, CANCEL, and OUTPUT commands (bit field).

JOB

The user can enter batch and background Terminal Monitor Program (TMP) jobs (bit field).

JOBFROM

Specifies that this user can use //*JOBFROM control statements. Specify this field in the record for all multiple-user single address space systems (MUSASSs) such as, CICS, IMS, and CA-ROSCOE or in batch production environments like CA-7. This control statement allows another logonid and source to be transmitted by the MUSASS for any jobs submitted by that MUSASS without knowing the password. This privilege is not limited to MUSASS environments. (Bit field)

KERB-VIO

Specifies the number of Kerberos key violations. This field is similar to the PSWD-VIO count and is used with the PSWD-VIO count to suspend the user's LID when the combined counts exceed the global PSWDLMT count field.

KERBCUR

Specifies the current Kerberos key. This field is not modifiable via the ACF command and is only updated when the password is modified.

KERBCURV

Specifies the Kerberos key version. This field is not modifiable via the ACF command and is only updated when the password is modified.

KERBPRE

Specifies the previous Kerberos key. This field is not modifiable via the ACF command and is only updated when the password is modified.

KERBPREV

Specifies the previous Kerberos key version. This field is not modifiable via the ACF command and is only updated when the password is modified.

LDEV

Specifies that this user can create logical devices using the IBM Pass-Through Virtual Machine (PVM) product. This privilege applies only when the optional CA ACF2 intercept is in place. This field applies to VM sites only.

LDS

Specifies whether Logonid administrative changes for this user are propagated to the active Lightweight Directory Access Protocol (LDAP) servers in the network. (Bit field)

LEADER

The user can display and alter certain fields of other logonids for other users, as limited by a scope (bit field).

LGN-ACCT*

This user can specify an account number at logon time (bit field).

LGN-DEST*

The user can specify a remote output destination at TSO logon that overrides the value specified in the DFT-DEST field (bit field).

LGN-MSG*

This user can specify message class at logon time (bit field).

LGN-PERF*

This user can specify a performance group at logon time (bit field).

LGN-PROC*

This user can specify the TSO procedure name at logon time (bit field).

LGN-RCVR*

This user can use the recover option of the TSO or TSO/E command package (bit field).

LGN-SIZE*

This user is authorized to specify any region size at logon time (overriding TSOSIZE) (bit field).

LGN-TIME*

This user can specify the TSO session time limit at logon time (bit field).

LGN-UNIT*

This user can specify the TSO unit name at logon time (bit field).

LID

This is the user's logonid that is also a key for indexing into the Logonid database (8 characters).

LIDSCOPE

Specifies a logonid mask limiting the scope of SECURITY/ACCOUNT/LEADER access. (Eight-byte character field)

Important! This is an old field from pre-4.0 releases of CA ACF2. This field should not be used. Instead, use the SCPLIST field in the Logonid record with Scope records to limit a user's administrative authority over the CA ACF2 Logonid, Rule, and Infostorage databases. For complete information on how to create Scope records, see the chapter on "Maintaining Scope Records."

LIDTEMP

Specifies that the current password is a temporary password. This bit will be set if the current password was set by a non-owner of the LOGONID, such as a security administrator or account manager, and the password was immediately expired. This bit is not modifiable via the ACF command.

LIDZMAX

Specifies that a zero value for the MAXDAYS field in the LIDREC will override the global PSWDMAX value in the GSO PSWD record.

LIDZMIN

Specifies that a zero value for the MINDAYS field in the LIDREC will override the global PSWDMIN value in the GSO PSWD record.

LINE*

This indicates the TSO line-delete character (1 character).

LOGSHIFT

A user can access the system outside the time period specified in the SHIFT field of the logonid record (bit field).

MAIL*

This user wants to receive mail messages from TSO at logon time (bit field).

MAINT

A user can use a specified program executed from a specified library to access resources without loggings or validation (bit field).

MAXDAYS

This is the maximum number of days permitted between password changes before the password expires. If the value is zero, no limit is enforced (1-byte binary).

MINDAYS

This is the minimum number of days that must elapse before the user can change 62the password (1-byte binary).

MODE*

This user wants to receive modal messages from TSO (bit field).

MON-LOG

CA ACF2 writes an SMF record each time this user enters the system (bit field).

MONITOR

CA ACF2 sends a message to the security console and to a designated person (CSWHO) each time this user enters the system (bit field).

MOUNT

This user can issue mounts for devices (bit field).

MSGID*

This user wants to prefix TSO message IDs. (bit field).

MULTSIGN

This user has has multiple sign-on privileges (CICS only).

MUSASS

This logonid is a multiple user single address space system (MUSASS) such as CICS, IMS, or CA-IDMS (bit field).

MUSDLID(*logonid*)

Specifies the default logonid for a multiple-user single address space system (MUSASS) address space. The logonid specified *does not* need the MUSASS attribute. Currently, this field is used only for SAF. (eight characters)

MUSID

Specifies a one- to eight-character ID you assign to a multiple-user single address space (MUSASS). (eight characters)

MUSIDINF

Indicates that the MUSID field should be used to restrict access to a MUSASS region for CA ACF2 Info type system entry calls. (Bit field).

MUSUPDT

A logonid with this privilege and the MUSASS privilege can update the CA ACF2 databases (bit field).

NAME

This is the user name displayed on logging and security violation reports (20 characters).

NO-INH

A network job cannot inherit this logonid from its submitter (bit field).

NO-OMVS

Specifies that this user cannot use any UNIX System Services (OpenEdition). NO-OMVS overrides any user OMVS profile record defined for the user. This user cannot use the defaults specified in the DFTUSER or DFTGROUP fields of the GSO UNIXOPTS record. (Bit field)

NO-SMC

Specifies that this user can bypass step-must-complete (SMC) controls. A job is considered noncancelable for the duration of the sensitive VSAM update operation. (Bit field)

NO-STATS

Specifies that the last access statistics on a successful full validation (ACVAMVAL) MUSASS signon request are bypassed. (Bit field)

Note: You must also specify MUSASS on the logonid for this privilege to be effective.

With a successful signon, CA ACF2 will bypass writing an SMF record for the signon request. In addition, no ACF01134 or ACF01137 messages are returned to the caller. If the signon fails, an SMF record is written and the logonid database is updated.

NO-STORE

This user is unauthorized to store or delete rule sets (that is, cannot effect rule changes) regardless of ownership (PREFIX values), SECURITY attribute, or delegation through %CHANGE or %RCHANGE (bit field).

NOMAXVIO

Prevents the user violation counter from incrementing and MAXVIO processing from occurring.

NON-CNCL

A user can access all data, even if a rule prohibits this access. The event log shows that the request was permitted because the user was non-cancelable (bit field).

NOSPOOL

Specifies how CA ACF2 reacts when a user with this field enters a command that results in a "spool file not found" condition. The values for this field are listed in the following:

- PREVENT—rejects and logs the command
- LOG—passes the command to CP for normal syntax checking and generates a logging record
- ALLOW—passes the command to CP for normal syntax checking
- null—removes the NOSPOOL field from the user's logonid record.

Null, the default value, causes the @VM NOSPOOL setting in the ACFFDR to take effect for "spool file not found" conditions. If you specify a value other than null, the setting for this field overrides the @VM NOSPOOL setting in the ACFFDR. This field applies to VM sites only.

NOTICES*

This user wants to receive TSO notices at logon time (bit field).

OPERATOR*

This user has TSO operator privileges (bit field).

PASSWORD

This is the user password CA ACF2 stores in a one-way encrypted format (8 to 128 characters).

PAUSE*

This user wants a program to pause when a command executed in a CLIST issues a multilevel message. This lets the user enter a question mark to receive the second-level messages (bit field).

PGM

The user must use the specified APF-authorized program to submit jobs for this logonid. The user must also have the RESTRICT logonid field. CA ACF2 assumes SUBAUTH if PROGRAM is specified. This field can also be specified by using the field name, PROGRAM, instead of the name PGM, except for reports that require the full form of the name (PROGRAM). (8 characters).

PHONE

This field contains the user's telephone number (12 characters).

PMT-ACCT*

This field forces this user to specify an account number at logon time (CA ACF2 prompts for one if none is provided). LGN-ACCT should also be specified (bit field).

PMT-PROC*

This field forces this user to specify a TSO procedure name at logon time. (CA ACF2 prompts for one if none is provided). LGN-PROC should also be specified (bit field).

PP-TRC

Specifies whether CA ACF2 creates SMF loggings that contain the Active Library List for all data set access attempts made by this logonid in a batch job. PP-TRC is a tool to help in the coding of program-pathed data set access rules. You can view these loggings in the Data Set Report Log (ACFRPTDS) report generator. For details on program pathing trace records collected for the report generator, see the *Reports and Utilities Guide*. For more information on program pathing and the Active Library List, see the chapter "Maintaining Access Rules." PP-TRC is only valid for Batch jobs. (Bit field)

PP-TRCV

Specifies whether CA ACF2 creates SMF loggings that contain the Active Library List for all data set access violations made by this logonid in a batch job. PP-TRCV is a tool to help in the coding of program-pathed data set access rules. You can view these violations in the Data Set Report Log (ACFRPTDS) report generator. For details on program pathing violation trace records collected for the report generator, see the *Reports and Utilities Guide*. For details on program pathing and the Active Library List, see the chapter "Maintaining Access Rules." PP-TRCV is only valid for Batch jobs. (Bit field)

PPGM

The user can execute those protected programs specified in the GSO PPGM record (bit field).

PREFIX

This field defines the high-level index of the data sets that this user owns and can access. CA ACF2 does not check the rule when this field matches the data set's high-level index. The PREFIX field also identifies the key of the access rule set that the user can store and decompile (8 characters).

PRIV-CTL

This field causes CA ACF2 to check privilege control resource rules when the user accesses the system to see what additional privileges and authorities the user has (bit field).

PROGRAM

The user must use the specified APF-authorized program to submit jobs for this logonid. The user must also have the RESTRICT logonid field. CA ACF2 assumes SUBAUTH if PROGRAM is specified. This field can also be specified by using the field name, PGM, instead of the name, PROGRAM, except for reports that require the full form of the name (PROGRAM). (8 characters).

PROMPT*

This user wants to be prompted for missing or incorrect parameters (bit field).

PRV-TOD1 through PRV-TOD4

MVS Note 12 uses these fields with the PRVSWD1 through PRVSWD4 fields to provide the date and time the passwords were changed.

PRVSWD1 through PRVPSWD4

MVS Note 12 uses these fields to provide password history processing. CA ACF2 uses these fields with the PRV-TOD1 through PRV-TOD4 fields.

PSWD-DAT

This is the date when this user made the last invalid password attempt (4-byte binary).

PSWD-EXP

This indicates that this user's password was manually expired (forced to expire). This attribute enables a security administrator to force users to change their passwords (bit field).

PSWD-INV

This is the number of password violations that occurred since the last successful logon (2-byte binary).

PSWD-MIX

Specifies that the current password is case sensitive. You cannot set this field. CA ACF2 maintains and displays it. (Bit field)

PSWD-SRC

This is the logical or physical input source name or source group name where the last invalid password for this logonid was received (8 characters).

PSWD-TIM

This is the time when the last invalid password for this logonid was received (4-byte binary).

PSWD-TOD

This is the date and time the password was last changed (8-byte binary).

PSWD-UPP

Specifies that any new password will be upper-case only (not case-sensitive).

PSWD-VIO

This is the number of password violations occurring on PSWD-DAT (2-byte binary).

PSWD-XTR

The password for this logonid is halfway-encrypted and can be extracted by an APF-authorized program (bit field).

PSWD-XTV

Specifies the halfway-encrypted value of a password when PSWD-XTR is set. (Eight characters)

Note: You cannot set this field. CA ACF2 maintains it, and never displays it. This field is valid only when PSWD-XTR is set.

PSWDCVIO

Indicates the number of cumulative invalid password attempts for a user that occurred since the logonid was created. (2-byte binary).

PTICKET

Specifies that a passticket can be used with a userid that has the RESTRICT attribute. (Bit field)

PWP-DATE

Indicates the date of the last invalid password phrase attempt (4 byte packed).

PWP-VIO

Indicates the number of password phrase violations occurring on PWP DATE (2-byte binary).

PWPALLOW

Indicates that the user is allowed to authenticate using a password phrase when the GSO PWPHRASE record indicates NOALLOW to disable all users (bit field).

READALL

The logonid has only read access to all data at the site. Also implies EXEC(A) (bit field).

RECOVER*

This user wants to use the recover option of the TSO or TSO/E command package. The PROFILE RECOVER option is set at logon time (bit field).

REFRESH

This user is authorized to issue the F ACF2,REFRESH operator command from the operator's console (bit field).

RESTRICT

This restricted logonid is for production use and does not require a password for user verification (bit field).

RSRCVLD

This field specifies that a resource rule must authorize any accesses that a user makes. This field applies even if the user has the SECURITY privilege (bit field).

RSTDACC

Specifies that this user has restricted access to UNIX directories and files based on owner or group permissions, not on other permissions, when the user does not have at least read access to the UNIXPRIV resource, RESTRICTED.FILESYS.ACCESS.

RULEVLD

An access rule must exist for all data this user accesses, regardless if the user has ownership of the data (PREFIX field) or has the SECURITY privilege (bit field).

SCPLIST

This field contains the name of the infostorage scope record (under type code SCP) that restricts accesses for this privileged user (8 characters).

SEC-VIO

This is the total number of security violations for this user (2-byte binary).

SECURITY

This user is a security administrator who, in the limits of his scope, can create, maintain, and delete access rules, resource rules, and infostorage records. A security administrator can update certain fields in logonid records, display logonid records, and access any data set (bit field).

SHIFT

This field identifies the shift record that defines when a user is permitted to log on to the system. The shift record, stored in the Infostorage database, can specify the times of day, dates, or days for access (8 characters).

SMSINFO

This field points to a CONTROL SMS Infostorage record that holds default storage management class values.

SOURCE

This is the logical or physical input source name or source group name where this logonid must access the system (8 characters).

SRF

Specifies that a user can use the system request facility (SRF) in a VM environment. This field is meaningful to VM sites. (Bit field)

STC

Only started tasks use this logonid (Bit field).

SUBAUTH

Only an APF-authorized program can submit jobs specifying this logonid. CA ACF2 uses this privilege with the PROGRAM and RESTRICT attributes (bit field).

SUSPEND

The logonid is suspended and denied access to the system (bit field).

SYNCNODE

This field contains the node name where the synchronized logonid for this logonid is found in the Logonid database (8 characters).

SYNERR

Specifies the action CA ACF2 takes when a user enters a command that results in a command syntax error.

SYSPEXCL|NOSYSPEXCL

Indicates that when the system is active in a sysplex environment, this logonid record should not be written to the structure. (Bit field)

TAPE-BLP

This user can use full bypass label processing (BLP) when accessing tape data sets (bit field).

TAPE-LBL

This user has limited bypass label processing (BLP) when accessing tape data sets (bit field).

TDISKVLD

Indicates that access rules must exist for all data on temporary disks that this user accesses. This field applies to VM users only. (Bit field)

TRACE

All data references by this user are traced and logged (bit field).

TSO

This user is authorized to sign on to TSO (bit field).

TSO-TRC

SMF traces all the TSO commands this user issues (bit field).

TSOACCT*

This is the user's default TSO logon account (40 characters).

TSOCMDS

This is the name of the TSO command list module that contains the list of the commands that this user is authorized to use (8 characters).

TSOFSCRN*

This user has the full-screen logon display (bit field).

TSOPERF*

This indicates the user's default TSO performance group (1-255). Zero indicates that no performance group is specified (1-byte binary).

TSOPROC*

This is the user's default TSO procedure name (8 characters).

TSORBA*

This is the mail index record pointer (MIRP) for this user (3 hexadecimal bytes).

TSORGN*

This is the user's default TSO region size (in K bytes) if the user does not specify a size at logon time (2-byte binary).

TSOSIZE*

This is the user's maximum TSO region size (in K bytes) unless the user has the LGS-SZE field specified (2-byte binary).

TSOTIME*

This indicates the user's default TSO time parameter (2-byte binary).

TSOUNIT*

This is the user's default TSO unit name (8 characters).

UID

This pseudo field is a concatenation of selected information from the Logonid record. You cannot change this field to modify a user's UID. The UID can include information from user-defined fields, such as company code, department, or job function (24 characters).

UIDSCOPE

Specifies a UID mask limiting logonid access. It can never by displayed. (24-byte character field)

Important! This is an old field from pre-4.0 release of CA ACF2. This field should not be used. Instead, use the SCPLIST field in the Logonid record with Scope records to limit a user's administrative authority over the CA ACF2 Logonid, Rule, and Infostorage databases. For more information on how to create Scope records, see the chapter on "Maintaining Scope Records."

UNICNTR

Indicates that this user also resides on the CA-Common Services platform. When set, CPF lets commands referring to this logonid be sent to nodes defined to CPF as UNINODEs. (Bit field)

UPD-TOD

This is the date and time that this logonid record was last updated (8-byte binary).

USER

All logonids defined to CA ACF2 are automatically designated as USERs. CA ACF2 never displays this field. Do not alter this field (bit field).

VLD-ACCT

This indicates that CA ACF2 is to validate the TSO account number (bit field).

VLD-PROC

This indicates that CA ACF2 is to validate the TSO procedure name (bit field).

VLDRSTCT

Turning on this field for a RESTRICT logonid indicates that PROGRAM and SUBAUTH are to be validated even when the logonid is inherited (bit field).

VLDVMACT

Specifies that a virtual machine must specify an account number if its VMACCT field contains blanks. This field applies to VM users only. (Bit field)

VM|NOVM

Indicates that a user can log on to VM. This field is meaningful for sites that have CA ACF2 VM and use synchronized database support. (Bit field)

VMACCT

Specifies an eight-byte logonid field that holds the default account number for a virtual machine. This field applies to VM users only and is available for VM/XA operating systems. (Character field)

VMD4AUTH

A user with this attribute can issue diagnose d4 to surrogate virtual machines with the VMD4TARG attribute. Use *extreme caution* when you assign this privilege. The VMD4TARG and VMD4AUTH privileges are very powerful. A typical class B user with both of these attributes could potentially surrogate itself to any user ID on the system and have access to anything on the system. In previous releases, this privilege was the VMBATMON privilege. This field applies to VM users only.

VMD4SEC

Specifies the Diagnose D4 CMS File Level Security attribute. It indicates that CA ACF2 should keep track of the surrogated ID that is in use when minidisks are linked. This saved information is then used to validate CMS file accesses using the surrogated ID, even if the CMS file accesses are done after the surrogation is no longer in place. This only applies to CMS file accesses through standard CMS interfaces, not through services such as the *BLOCKIO System Service.

For example, FTPSERVE uses Diagnose D4 to link minidisks under the authority of the user requesting FTP services, but resets surrogation before actually transferring the CMS files. With the VMD4FSEC attribute on in the FTPSERVE logonid, CA ACF2 will validate access to the CMS files on the minidisk using the authority of the user that was surrogated when the minidisk was linked. This field applies to VM users only. (Bit field)

VMD4RSET

Indicates that this user can be the target of the diagnose d4 reset after the logonid was surrogated to another ID. Use *extreme caution* when assigning this privilege. Never give this logonid attribute to a batch worker machine.

The combination of VMD4RSET, VMD4AUTH, and VMD4RSET lets products like TCP/IP and VMBACKUP function properly. To track the use of the diagnose d4, you can write a diagnose limiting rule to log each time the diagnose d4 is issued. In previous releases of CA ACF2 VM (releases 3.2 and below), this attribute was called VMRESET. This field applies to VM users only.

VMD4TARG

A user ID with this attribute can be the target of diagnose d4 (the alternate user diagnose). Use *extreme caution* when you assign this attribute. In previous releases, this privilege was the VMBATCH privilege. This field applies to VM users only.

VMESM

Indicates that this server can use the CA ACF2 security interface. This field applies to VM users only. (Bit field)

VMIDLEMN

Specifies the number of minutes (from 1 to 240) that this user can be idle on the system before the idle terminal processing begins. This value overrides the system-wide IDLEMN value defined in the OPTS VMO record. This field applies to VM users only.

VMIDLEOP

Specifies the type of idle terminal processing to perform when this user exceeds the idle time limit. This value overrides the system-wide IDLEOP VMO record. This field applies to VM users only.

VMSAF

Indicates this logonid can issue the diagnose code, DIAG 'A0' subfunction code '04'. This support lets users validate CA ACF2 passwords from their unique applications. This field applies to VM users only. (Bit field)

VMSFS

Indicates that this SFS server can use the CA ACF2 security interface. This field applies to VM users only. (Bit field)

VMXA

Permits this user to log on to the VM/ESA system. This attribute is required only if the VMCHK=VMXA operand has been specified in the @VM macro. This field applies only to VM users. (Bit field)

VSESRF

Indicates that this logonid can issue System Request Facility (SRF) requests to the CA ACF2 VM service machine from CA ACF2 VSE. These SRF requests can validate the accesses of users and perform direct maintenance of the CA ACF2 databases. This field applies only to VM and VSE users. (Bit field)

WTP*

This displays write-to-programmer (WTP) messages. CA ACF2 issues all violation and warning messages as WTPs, so that this attribute should be present in all TSO user logonid records to enable them to receive CA ACF2 messages (bit field).

ZONE

This specifies the name of the Infostorage Database zone record defining the time zone where this logonid normally accesses the system that is, the user's local time zone (3 characters).

Chapter 3: Managing Access Rule Sets

Todd process access rule sets, create an access rule set, compile it from a terminal, or a partitioned data set (PDS) follow these step-by-step instructions. Access rule syntax and parameter descriptions follow.

This section contains the following topics:

Processing Access Rules

Process access rule sets under the ACF command or RULE setting by entering these subcommands.

Function	Command Syntax
Begin ACF command processing	SEt|T {ACF|Rule}
Create a set of access rules from a terminal or batch	COMpile [*] [List|NOList] [Store|NOStore] [Force|NOForce] [Maxrule(250|nnn)] [ALL]
Create a set of access rules from a member of a partitioned data set (PDS)	COMpile datasetname [List|NOList] [Store|NOStore] [Force|NOForce] [Maxrule(250|nnn)] [ALL]
Displays access rule set or write into a data set to change	DEComp {*} [Into(datasetname)] {ruleid} {Like(ruleidmask)}
Delete access rule set	DELete {*} {ruleid} {Rule(ruleid)}
Decompile, then add or delete a rule entry, recompile and store a rule set	RECKEY {ruleid {ADD(rule-entry)|DELETE(rule-entry)| MOD(rule-entry)}

Function	Command Syntax
Save access rule set	STore
Test access rule set	TEst [*|ruleid] **Test Subcommand Keywords** DSname(datasetmask) [Access(READ|WRITE|EXEC|ALLOC] [Volume(volumemask)] [LID(logonidmask)|Uid(uidmask)] [LIBrary(librarymask)] [PGm(pgmmask)|PROGram(pgmmask)] [DAte(date)] [DDname(ddname)] [RESET] [SHift(shift)] [SOurce(sourcemask)] [Time(hhmm)] [ENd]

When compiling all members of a PDS, defaults are List, Store, and Force. Maxrule applies only if you specify NORULELONG.

CA ACF2 allows partitioned data sets (PDS) to be secured at the member level. MVS/EVA 4.3.0 or above is required. See the *Administrator Guide* for more information about PDS security.

Create access rules

The ACFCOMP, ACFDCMP, and ACFNRULE commands also process rules. See the *Reports and Utilities Guide* for detailed information.

Create access rules from the terminal

1. Type the **ACF** command.
2. Type the **SET RULE** subcommand if you changed settings at any time. If you have not changed settings since entering ACF, you do not have to enter **SET RULE**.
3. Type the **COMPILE** subcommand without parameters.
4. Type the **$KEY** control statement, followed by the other control statements.
5. Type all the rule entries, each beginning on a separate line.
6. Select **ENTER** or type **END** to end the rule set.
7. Type the **STORE** subcommand to save the rule set in the Rule database.

Create access rules from the a partitioned data set (PDS)

1. Build the rule set text in a PDS.
   - Type the **$KEY** control statement on the first line, followed by the other control statements.
   - Type all rule entries, each beginning on a separate line.
   - Save the data set.
2. From TSO, type the **ACF** command.
3. Type the **COMPILE** subcommand with the data set name that contains the rule set.

Writing Access Rules

Control statements and individual rule entry parameters of a rule set follow:

Control Statements	Character Length	Maskable?
$Key(ruleid)	8	No
[$Mode(Quiet|Log|Warn|Abort)]	-	N/A

Control Statements	Character Length	Maskable?
[$NORulelng]	-	N/A
[$NOSort]	-	N/A
[$Owner(ownerid)]	24	N/A
[$Prefix(prefix)]	24	No
[$Resowner(resourceowner)]	8	No
[$Userdata(userdata)]	64	N/A
[%Change uidmask,…,uidmask]	24	Yes
[%Rchange uidmask,…uidmask]	24	Yes
[*comment]	-	N/A
[$Member]	8	No

Rule Entry Parameters	Character Length	Maskable?
datasetmask	44	Yes
[Volume(volumemask)]	6	Yes
[UId(uidmask)]	24	Yes
[SOurce(sourcemask)]	8	Yes
[SHift(shift)]	8	No
[Library(libmask)]	44	Yes
[PGm(pgmmask)|PROGrams(pgmmask)]	8	Yes
[DDname(ddnamemask)]	8	Yes
[UNtil(date)|For(days)]	8	No
[ACTIVE(date)]	8	No
[DAta(text)]	64	No
[Nextkey(nextkey)]	8	No
[Read(Allow|Log|Prevent)]	-	No
[Write(Allow|Log|Prevent)]	-	No
[Allocate(Allow|Log|Prevent)]	-	No
[Execute(Allow|Log|Prevent)]	-	No

Use a ditto mark (") to repeat a parameter value from the previous rule entry. Use only one set of control statements per rule set. Multiple sets of rule entries are permitted. You can enter more than one control statement or parameter per line.

A sample access rule set follows:

```
*   Following are the control statements in this sample rule set.  Omit the
*   $ when more than one control statement is placed on a single line.
 $KEY(ruleid) OWNER(owner) NOSORT
$PREFIX(prefix)
 %RCHANGE uidmask
*   Following is one rule entry.  Each rule entry is indented one space in
*   case the dsnmask begins with an asterisk and is mistaken for a comment.
 Dsnmask1 VOL(volmask) UID(uidmaks) READ(ALLOW) WRITE(LOG)
*   Following is a second rule entry.  A dash continues the rule entry to
*   another line.
 Dsnamask2 VOL(volmask) SHIFT(shift) –
    UID(uidmask) READ(LOG) WRITE(LOG) EXECUTE(LOG)
```

Describing Access Rule Parameters

$KEY(*ruleid*)

Supply the high-level index of the data set name you are writing this rule for or the VSAM key of the rule set. This control statement usually contains your logonid when writing rules for your own data sets.

$MEMBER(membername)

Specify the member name to be used for a decompile into a partitioned data set (PDS) if one is not provided with the decompile request. When you specify a $MEMBER name that matches the $KEY value for the access rule, CA ACF2 issues a warning message and ignores the $MEMBER control statement. For details on how the $MEMBER control statement affects DECOMP subcommand processing, see the DECOMP subcommand section in the *Administrator Guide*.

$MODE(QUIET|LOG|WARN|ABORT)

Use this control statement with the GSO OPTS MODE parameter to phase in CA ACF2 data protection.

$NORULELNG

Overrides the use of the rulelong compiler when RULELONG is active. Normally, CA ACF2 uses the rulelong compiler to compile rules if the RULELONG option is set. The rulelong format is an expanded record format. If a ruleset is small and therefore does not require the rulelong format, specifying NORULELONG on a compile lets you compile a ruleset using a compact record format. This way, you can choose to compile rules with the format that is required for the ruleset.

Note: If the dynamic compile option (COMPDYN) is set in the GSO RULEOPTS record then the $NORULELNG control statement is not needed to compile rule sets of varying size.

$NOSORT

Specify this control statement to prevent standard CA ACF2 sorting of access rules. (You must also specify the GSO RULEOPTS $NOSORT parameter).

$OWNER(*ownerid*)

Indicate the logonid or owner of the rule set for tracking purposes.

$PREFIX(*prefix*)

Indicate a value that overrides the rule set key ($KEY) as a prefix to all data set names in this rule set. Use this field with the NEXTKEY parameter to indicate the true high-level index.

$RESOWNER(resourceowner)

Indicate the logonid to act as the resource owner of the data set. DFSMS uses $RESOWNER to verify whether a user has access to STORCLAS and MGMTCLAS.

$USERDATA(*userdata*)

Store user information with the rule set.

%CHANGE uidmask,...,uidmask

Indicate who, besides the high-level index owner or security administrator, can replace or delete a set of rules. You can specify more than one of these control statements. This user can further delegate this authority.

%RCHANGE uidmask,...,uidmask

Indicate who has restricted change authority over this rule set. These users can change individual rule entries, but not control statements. You can specify more than one of these control statements.

***comment**

The asterisk in column one lets you place comments inside an uncompiled rule set. These are lost during the compile and decompile sequence.

Datasetmask

Specify the name of the data set or data sets the rule pertains to; however, the high-level index is omitted. (The high-level index is specified on the $KEY or $PREFIX control statement.)

VOLUME(volumemask)

Specify the volume or volumes where the data set must reside. If omitted, all volumes are considered.

UID(uidmask)

Specify the set of users this rule entry applies to. If omitted, this entry applies to all users.

SOURCE(sourcemask)

Specify a physical or logical input source or source group name this rule applies to. If omitted, any source is valid.

SHIFT(*shift*)

Specify the name of an infostorage shift record that defines the allowable days, date, and times for access. If omitted, no shift limitations exist.

LIBRARY(*libmask*)

Identify the library or libraries where a program must execute for the access rule to apply.

PGM(*pgmmask*|PROGRAM(*pgmmask*)

Define the program or programs (in the set of libraries specified by the LIB keyword) this rule applies to.

DDNAME(ddnamemask)

Identify the specific ddnames that must be used for this rule to apply. If omitted, any ddname is permitted.

UNTIL(*date*)

Define the last date when this rule is considered valid. Specify as mm/dd/yy, yy/mm/dd, or dd/mm/yy as defined in the DATE field of the GSO OPTS record.

ACTIVE(*date*)

Define the first date on which this rule is considered valid. Specify as mm/dd/yy, yy/mm/dd, or dd/mm/yy as defined in the DATE field of the GSO OPTS record. ACTIVE applies only when RULELONG in GSO(RULEOPTS) is specified.

FOR(*days*)

Specify the number of days (0 to 365), starting from the compilation date of this rule set, when this rule is considered valid.

DATA(*text*)

Specify any character string to be retained with the rule set and formatted when the rule set is decompiled.

NEXTKEY(*nextkey*)

Specify the key of the alternate rule set to check if access to this data set is denied based on this rule entry.

READ(ALLOW|LOG|PREVENT)

Specify the READ access permission.

WRITE(ALLOW|LOG|PREVENT)

Specify the WRITE access permission.

ALLOCATE(ALLOW|LOG|PREVENT)

Specify the ALLOCATE (create, delete, rename, and catalog authority) access permission.

EXECUTE(ALLOW|LOG|PREVENT)

Specify the EXECUTE access permission.

Chapter 4: Managing Resource Rule Sets

To process resource rule sets in the Infostorage database, create a resource rule set, compile it from a terminal, or a partitioned data set (PDS) follow the step-by-step instructions. Resource rule syntax and parameter descriptions follow.

This section contains the following topics:

Processing Resource Rule Sets

After issuing the ACF command, enter the following subcommands to process resource rule sets in the Infostorage database:

Function	Command Syntax
Begin resource rule processing	SEt|T Resource(typecode)
Create a set of resource rules from a terminal or batch	COMpile [*] [List|NOList] [Store|NOStore] [Force|NOForce] [Maxrsrc(250|nnn)] [ALL]
Create a set of resource rules from a member of a partitioned data set (PDS)	COMpile datasetname [List|NOList] [Store|NOStore] [Force|NOForce] [Maxrsrc(250|nnn)] [ALL]
Displays resource rule set or write into a data set to change	DEComp|List {*} {ruleid} {Like(ruleidmask)} {Into(datasetname)}
Delete resource rule set	DELete {*|ruleid|Rule(ruleid)}

Function	Command Syntax
Decompile, then add or delete a rule entry, recompile and store a rule set	RECKEY {ruleid {ADD(rule-entry)|DELETE(rule-entry)}
Save resource rule set	STore
Test resource rule set	TEst [*|ruleid] Test Subcommand Keywords [Rsrcname(resourcemask)] [LID(logonidmask)|Uid(uidmask)] [Date(date)] [RESET] [SOurce(sourcemask)] [Time(hhmm)] {SErvice(Read,Update,Add,Delete)] [ENd]

An asterisk (*) in column one of a rule set begins a comment line, used primarily in batch or background Terminal Monitor Program (TMP) environments for documentation purposes.

When compiling all members of a PDS, defaults are List, Store, and Force.

Maxrsrc applies only when NORULELONG is specified.

CA ACF2 allows partitioned data sets (PDS) to be secured at the member level, MVS/EVA 4.3.0 or above is required. See the *Administrator Guide* for more information

To Create Resource Rules

The ACFCOMP, ACFDCMP, and ACFNRULE commands also process rules. See the *Administrator Guide* for detailed information.

From the Terminal

1. Type the **ACF** command.
2. Type the **RESOURCE** setting.
3. Type **SET RESOURCE(typecode)**.

   The type code is three characters.
4. Type the **COMPILE** subcommand without parameters.
5. Type the **$KEY** control statement, followed by the other control statements.
6. Type all rule entries, each beginning on a separate line.
7. Select **ENTER** or type **END** to end the rule set.
8. Type the **STORE** subcommand to save the rule set in the Infostorage database

From a partitioned data set (PDS)

1. Build the rule set text in a PDS.
   - Type the **$KEY** control statement on the first line, followed by the other control statements.
   - Type all rule entries, each beginning on a separate line.
   - Save the data set.
2. From TSO, enter the **ACF** command.

   Establish the **RESOURCE** setting.
3. Type **SET RESOURCE(typecode)**. The type code is three characters.
4. Type the **COMPILE** subcommand with the data set name that contains the rule set.

Writing Resource Rules

Control statements and individual rule entry parameters of a rule set follow:

Control Statements	Character Length	Masakble?
$Key(ruleid)	40	Yes
$Type(typecode)	3	No
[$NORulelng]	-	N/A
[$NOSort]	-	N/A
[$Prefix(prefix)]	24 (40 for HFS file resources)	No
[$Recname(recordname)]	24	No
[$Userdata(userdata)]	64	No
[%Change uidmask,…,uidmask]	24	Yes
[%RChange uidmask,…uidmask]	24	Yes
[*comment]	-	N/A
[$MEMBER]	8	No

Rule Entry Parameters	Character Length	Masakble?
[resourcemask]	256	Yes
[Uid(uidmask)]	24	Yes
[SOurce(sourcemask)]	8	No
[SHift(shift)]	8	No
[UNtil(date)|For(days)]	8	No
[ACTIVE(date)]	8	No
[Nextkey(nextkey)]	40	No
[Reccheck(recname)]	24	No
[SErvice(Read,Add,Update,Delete)]	-	No
[DAta(text)]	64	N/A
[Verify]	-	N/A
[Allow|Log|Prevent]	-	N/A

If directory is resident.

Use a ditto mark (") to repeat the parameter value from the previous rule entry.

Use only one set of control statements per rule set. Multiple sets of rule entries are permitted.

A sample resource rule set follows:

```
*   Control statements for a sample rule set follow.  Omit the $ to place
*   more than one control statement on a single line.
 $KEY(ruleid) TYPE(typecode) NOSORT
%CHANGE uidmask
*   A rule entry follows.  Each entry is indented for readability.
 UID(uidmask) ALLOW
*   This second rule entry uses a dash to continue to another line.
 UID(uidmask) SOURCE(sourcemask) SHIFT(shift) –
    SERVICE(ADD) LOG
```

Describing Resource Rule Parameters

$KEY(*ruleid*)

Supply the high-level index that is being protected. You can mask it if the directory for the specified TYPE is resident.

$TYPE(typecode)

Identify the type of resource being protected. CA ACF2 predefines some type codes.

$MEMBER(membername)

Specify the member name to be used for a decompile into a partitioned data set (PDS) if one is not provided with the decompile request. When you specify a $MEMBER name that matches the $KEY value for the access rule, CA ACF2 issues a warning message and ignores the $MEMBER control statement. For details on how the $MEMBER control statement affects DECOMP subcommand processing, see the DECOMP subcommand section in the *Administrator Guide*.

$NORULELNG

Overrides the use of the rulelong compiler when RULELONG is active. Normally, CA ACF2 uses the rulelong compiler to compile rules if the RULELONG option is set. The rulelong format is an expanded record format. If a ruleset is small and therefore does not require the rulelong format, specifying NORULELONG on a compile lets you compile a ruleset using a compact record format. This way, you can choose to compile rules with the format that is required for the ruleset.

Note: If the dynamic compile option (COMPDYN) is set in the GSO RULEOPTS record then the $NORULELNG control statement is not needed to compile rule sets of varying size.

$NOSORT

Specify this control statement to prevent standard CA ACF2 sorting of resource rules. (The GSO RULEOPTS $NOSORT parameter must also be specifed.)

$PREFIX(*prefix*)

Specify a value to override the rule set key as a prefix to all resource names in this rule set.

$RECNAME(*recordname*)

Specify the name of a RECORD definition that you want CA ACF2 to validate with this rule.

$USERDATA(*userdata*)

Store user information with the rule set.

%CHANGE uidmask,...,uidmask

Indicate who can store and delete this rule set. You can specify more than one of these control statements.

%RCHANGE uidmask,...,uidmask

Indicate who has restricted change authority over this rule set. These users can change individual rule entries, but not control statements. You can specify more than one of these control statements.

***comment**

The asterisk in column one lets you place comments inside an uncompiled rule set. These are lost during the compile and decompile sequence.

resourcemask

Specify additional qualifiers in the resource name.

UID(uidmask)

Specify the set of users this rule entry applies to. If omitted, this entry applies to all users.

SOURCE(sourcemask)

Specify a physical or logical input source or source group name this rule applies to. If omitted, any source is valid.

SHIFT(*shift*)

Specify the name of an infostorage shift record that defines the allowable days, date, and times for access. If omitted, no shift limitations exist.

UNTIL(*date*)

Define the last date when this rule is considered valid. Specify as mm/dd/yy, yy/mm/dd, or dd/mm/yy as defined in the DATE field of the GSO OPTS record.

ACTIVE(*date*)

Define the first date on which this rule is considered valid. Specify as mm/dd/yy, yy/mm/dd, or dd/mm/yy as defined in the DATE field of the GSO OPTS record. ACTIVE applies only when RULELONG in GSO(RULEOPTS) is specified.

FOR(*days*)

Specify the number of days (0 to 365), starting from the compilation date of this rule set, when this rule is considered valid.

NEXTKEY(*nextkey*)

Specify the key of the alternate rule set to check if access to this data set is denied based on this rule entry.

SERVICE(READ,ADD,UPDATE,DELETE)

Specify the type of access to be associated with the access attempt. You can specify one or more access keywords.

DATA(*text*)

Specify any character string to be retained with the rule set and formatted when the rule set is decompiled.

VERIFY

Request password validation for any access attempt made under this rule.

ALLOW|LOG|PREVENT

Specify the type of access permitted to the resource. ALLOW permits access; LOG permits access but logs the access; and PREVENT denies access.

Chapter 5: Managing Cross-Reference Records

After issuing the ACF command, enter the following subcommands to process source group and resource group records in the Infostorage database.

Function	Command Syntax
Begin source group or resource group record processing	SEt|T TArget(null|=|?|nodemask,…,nodemask)] Xref(Sgp|Rgp) [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)]
Create source group record	Insert [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {USING(mode|recid) newrecid} [SYSid(?|sysid)|DIVision(?|div)] [USYSid(?|sysid)|UDIVision(?|div)] [Source|Group] [Include(entry,…,entry)] [Exclude(entry,…entry)] [ADD|REP|DEL]
Create resource group record	Insert [TArget(null|=|?|nodemask,…nodemask)] [*] {recid}[Type(typelist)]} {USING(mode|recid) newrecid[Type(typelist)]} [SYSid(?|sysid)|DIVision(?|div)] [USYSid(?|sysid)|UDIVision(?|div)] [Resource|Group] [Include(entry,…,entry)] [Exclude(entry,…entry)] [ADD|REP|DEL]
Displays cross reference records	List [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {LIKE(recidmask)} [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)]

Function	Command Syntax
Change cross-reference records	CHAnge [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {LIKE(recidmask)[Type(typelist)]} [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)] [Source|Group] [Resource|Group] [Include(entry,…,entry)] [Exclude(entry,…entry)] [ADD|REP|DEL]
Delete cross-reference records	DELete [TArget(null|=|?|nodemask,…nodemask)] {recid} {LIke(recidmask)} [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)]

Cross-reference record names are from one- to eight-characters in length.

Chapter 6: Managing Scope Records

After issuing the ACF command, enter the following subcommands to process scope records in the Infostorage database:

Function	Command Syntax
Begin scope record processing	SEt|T SCope(SCP)
Create scope record	Insert [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {USing(mode|scope) newscope} [Dsn(entry,…,entry)] [Inf(entry,…entry)] [Lid(entry,…entry)] [Uid(entry,…entry)] [ADD|REP|DEL]
Display scope record	List [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} [ALL|DSN,INF,LID,UID]
Change specific entries in scope record fields	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)} [Dsn(entry,…,entry)] [Inf(entry,…entry)] [Lid(entry,…entry)] [Uid(entry,…entry)] [ADD|REP|DEL]
Delete scope record	DELete [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {LIke(recidmask)}

Chapter 7: Managing Shift Records

After issuing the ACF command, enter the following subcommands to process shift records in the Infostorage database:

Function	Command Syntax
Begin shift record processing	SEt|T SHift(SFT)
Create shift record	Insert [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {[Days(MO,TU,WE,TH,FR,SA,SU,mm/dd/yy,…,mm/dd/yy)]} {[Time(hhmm-hhmm,…,hhmm-hhmm)]} {[NTime(hhmm-hhmm,…,hhmm-hhmm)]} {[NDays(MO,TU,WE,TH,FR,SA,SU,mm/dd/yy,…,mm/dd/yy)]} [Include(recid,…,recid)]
Display shift record	List [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)}
Change shift record	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)} [ADD|DEL|REP] {[Days(MO,TU,WE,TH,FR,SA,SU,mm/dd/yy,…,mm/dd/yy)]} {[Time(hhmm-hhmm,…,hhmm-hhmm)]} {[NDays(MO,TU,WE,TH,FR,SA,SU,mm/dd/yy,…,mm/dd/yy)]} {[NTime(hhmm-hhmm,…,hhmm-hhmm)]} {[Include(recid,…,recid)]}
Delete shift record	DELete [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {LIke(recidmask)}

Shift record names are from one- to eight-characters in length.

Chapter 8: Managing Zone Records

After issuing the ACF command, enter the following subcommands to process zone records in the Infostorage database:

Function	Command Syntax
Begin zone record processing	SEt|T SHift(ZON)
Insert zone record	Insert [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} Adjust(+hhmm|-hhmm)
Display zone record	List [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)}
Change zone record	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)} Adjust(+hhmm|-hhmm)
Delete zone record	DELete [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {LIke(recidmask)}

Zone record names are from one to three characters in length.

This section contains the following topics:

Zone Record Syntax (see page 60)

Zone Record Syntax

After issuing the ACF command, enter the following subcommands to process zone records in the Infostorage database:

Function	Command Syntax
Begin zone record processing	SEt|T SHift(ZON)
Insert zone record	Insert [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} Adjust(+hhmm|-hhmm)
Display zone record	List [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)}
Change zone record	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)} Adjust(+hhmm|-hhmm)
Delete zone record	DELete [TArget(null|=|?|nodemask,…nodemask)] [*] {recid} {LIke(recidmask)}

Chapter 9: Managing Source Records

After issuing the ACF command, enter the following subcommands to process source records in the Infostorage database. We recommend that you use the cross-reference (XREF) setting to group source names instead of the ENTRY setting for SGP records.

Function	Command Syntax
Begin source record processing	SEt|T Entry(SRC|SGP)
Create record that translates a physical input source to a logical name	Insert [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Using(modelrecid) newrecid – [Type(SRC|SGP)] [Clear] – [Newdata(newdata)] [Dsn(pseudodsn)]
List source group records	List TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)}
Add to or change source records in a source group	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {Like(recidmask)} {[0]ddata(olddata)] [Newdata(newdata)] {Clear]} [Verdata(verdata)] [Dsn(datasetname)]

Function	Command Syntax
Delete source group record entry	DELete [TArget(null|=|?|nodemask, …,nodemask)] [*] {recid} {LIke(recidmask)}

Source record names are from one- to eight-characters in length.

Chapter 10: Managing Control Records

You can process any structured Infostorage database record after establishing the proper CONTROL setting. To process CAC, CPF, GSO, NET, SMS, SSO, or TSO records, enter any of the following subcommands:

Note: SSO records require a user exit.

Function	Command Syntax
Begin control record processing	SEt|T [TArget(null|=|?|nodemask,…,nodemask)] Control(CAc|CPf|Gso|Net|SMs|Sso|Tso) [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)]
Create control record	Insert [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {USING(modelrecid) newrecid} [SYSid(?|sysid)|DIVision(?|div)] [USYSid(?|sysid)|UDIVision(?|div)] [field,…field] [ADD|REP|DEL]
Display control record	List [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {LIKE(recidmask)} [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)]
Change control record	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] [*] {recid} {LIKE(recidmask)} [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)] [field,…field] [ADD|REP|DEL]

Function	Command Syntax
Delete control record	DELete [TArget(null|=|?|nodemask,…node mask)] [*] {recid} {LIke(recidmask)} [SYSid(?|sysid)|DIVision(?|div)] [MSYSid(sysidmask)|MDIVision(divmask)]

Valid GSO Record Names:

- APPLEDEF*
- AUTHEXIT*
- AUTOERAS
- AUTOIDLX
- AUTOIDOM
- BACKUP
- BLPPGM
- CACHESAV
- CERTMAP
- CLASMAP*
- CRITMAP
- EIM
- ETAUDIT
- EXITS
- INFODIRLINKLIST
- LOGPGM
- LINUX
- MAINT*
- MLID
- MLSOPTS
- MUSASS
- NJE*
- OPTS
- PDS*
- PPGM
- PROXY
- PSWD
- PWPHRASE
- REALM
- RESDIR
- RESRULE
- RESVOLS

- RESWORD
- RULEOPTS
- SAFDEF*
- SECVOLS
- STC
- SYNCOPTS
- SYSPLEX
- TNGNODE*
- TSO
- TSOCRT
- TSOKEYS
- TSOTWX
- TSO2741
- WARN
- UNIXOPTS

Record names with an asterisk might append an optional one- to eight-character qualifier.

This section contains the following topics:

CACHE (CAC) Records

Records for the CA ACF2 database cache facility include the following:

- CACHOPTS
- INFOEXCL*
- INFOPRIM*
- LIDSEXCL*
- LIDSPRIM*
- RULEEXCL*
- RULEPRIM*

Record names with an asterisk might append an optional one- to eight-character qualifier.

Command Propagation Facility (CPF) Records

Records for the CA ACF2 CPF include the following:

- NODEDEF*
- OPTIONS

Record names with an asterisk might append an optional one- to eight-character qualifier.

LDAP Directory Services (LDS) Records

Records for CA ACF2 Security Control LDS S records include the following:

- OPTIONS
- LDAP*
- XREFLDAP*

Note: Record names indicating an * require a qualifier.

Storage Management Subsystem (SMS) Records

See the *Administrator Guide* for information about records for DFP Version 3 Storage Management Subsystem (SMS).

Chapter 11: Managing Profile Records

CA ACF2 supports the following profile and segment information:

Profile	Segement
USER	CERTDATA, CICS, DCE, EIM, KERB, KERBLINK, KEYRING, LANGUAGE, LINUX, LNOTES, NETVIEW, NDS, OMVS, OPERPARM, PASSWORD, PROXY, PWPHRASE, SECLABEL, WORKATTR
GROUP	OMVS, LINUX
DATASET	DFP
DLFCLASS	DLFDATA
APPCLU	SESSION
KEYSMSTR	SSIGNON
PTKTDATA	SSIGNON
SECLABEL	SECLEVEL
SECLABEL	CATEGORY
SECLABEL	SECLABEL
SYSMVIEW	SVFMR

You can use the following subcommands to process user profile records or profile data records after establishing the proper PROFILE setting.

Function	Command Syntax
Begin profile record processing	SEt|T PROFILE(profiletype] DIVISION(PROFILE|profile segment_name)
Create profile data record	Insert [*] {recid} {USING(urecid) recid}
Change profile data record	CHAnge {recid} {LIKE(urecid)
Display user profile record	List [*] {recid} {LIKE(recidmask)}
Delete user profile record	DELete [*] {recid} {LIke(recidmask)}

Chapter 12: Managing Field Records

You can process field records after establishing the FIELD setting of the ACF command.

Function	Command Syntax
Begin field record processing	SEt|T Field(EXPressn|RECord)
Create field record	COMpile [*] [datasetname] [List|NOList] [Store|NOStore] [Force|NOForce] [ALL]
Display field record	DEComp|List {*} {recid} {LIKE(recidmask) [Into(datasetname)]
Delete field record	DELete [*] {recid} {Rule(recidmask)}
Store field record	STore

Chapter 13: Managing Identity Records

After issuing the ACF command, enter the following subcommand to process extended user authentication records in the Infostorage database:

Function	Command Syntax
Begin identity record processing	SEt|T [Target(null|=|?|nodemask,…,nodemask)] Identity(Aut) [DIVision(?|div)|SYSid(?|sysid)] [MDIVision(?|divmask)|MSYSid(?|sysidmask)]
Create identity record	Insert [TArget(null|=|?|nodemask,…,nodemask)] {*} {recid} {USING(modelrecid) newrecid} [DIVision(?|div)|SYSid(?|sysid)] [MDIVision(?|divmask)|MSYSid(?|sysidmask)] [field,…,field] [ADD|REP|DEL]
Display identity record	List [TArget(null|=|?|nodemask,…,nodemask)] {*} {recid} {LIKE(recidmask)} [DIVision(?|div)|SYSid(?|sysid)] [MDIVision(divmask)|MSYSid(sysidmask)]
Change identity record	CHAnge [TArget(null|=|?|nodemask,…,nodemask)] {*} {recid} {LIKE(recidmask)} [DIVision(?|div)|SYSid(?|sysid)] [MDIVision(divmask)|MSYSid(sysidmask)] [field,…,field] [ADD|REP|DEL]
Delete identity record	DELete [TArget(null|=|?|nodemask,…,nodemask)] {*} {recid} {LIke(recidmask)} [DIVision(?|div)|SYSid(?|sysid)] [MDIVision(divmask)|MSYSid(sysidmask)]

Identity record names are from one- to eight-characters in length.

Chapter 14: Managing SECLABEL (DSN) Records

When MLS is active and write-down is protected on a system, after issuing the ACF command, enter the following subcommands to process SECLABEL(DSN) records in the Infostorage database.

Function	Command Syntax
Begin SECLABEL(DSN) record processing	**Set|T Seclabel(DSN)**
Display SECLABEL(DSN) record	**List{*recid*}**
Delete SECLABEL(DSN) record	**DELete{*recid*}**

Note: SECLABEL(DSN) record IDs are data set names that can be from 1 to 44 characters in length. Only the LIST and DELETE commands can be used to administer these MLS-related records.

Chapter 15: Running Reports

You can create some CA ACF2 reports through JCL statements or ISPF panels. To create reports through JCL, CA ACF2 provides a prototype JCL procedure on the distribution tape. Modify this JCL or create your own. (See the REPORTS member of the CAIJCL data set for the prototype JCL.) You can also create reports using ISPF panels. The *Reports and Utilities* Guide provides more information about running reports and utilities.

This section contains the following topics:

To Create an CA ACF2 Report

1. Use the following ddnames for the input/output files that are common to most CA ACF2 reports. See the following specific reports for additional files:
   - RECxxxxx (input files containing SMF records)
   - SYSIN (input file containing additional parameters)
   - SYSPRINT (output file)
2. Specify the following parameters in either or both of the following ways:

   Through the PARM parameter of the EXEC statement in the JCL:

   ```
   //DSLOGS   EXEC PGM=ACFRPTDS,REGION=128K,
   //     PARM=('TITLE(DATA SET LOGGING RECORDS)'
   //     'MASK(SYS1.-)','SDATE(97170)','EDATE(97.174)')
   ```

   Through the SYSIN file:

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

Using Parameters Common to All Reports

Parameters common to all ACF2 reports include:

```
[JOBMASK(********|jobmask,…,jobmask)]
 [LINECNT(60|nnnnnnnnn)]
 [TITLE(first 35 characters of PARM parameter|string)]
 [SDATE(00000|yyddd)]
 [EDATE(99365|yyddd)]
 [STIME(0000|hhmm)]
 [ETIME(2359|hhmm)]
 [SELECT(ACFFDR @SMF ACF2|nn,…,nnn|NOSELect]
 [SYSID(********|SYSEMID)]
 [HEX|NOHEX]
```

Using Parameters Specific to Each Report

Optional parameters specific to each report include:

ACFRPTCR—TSO Command Statistics Log

```
[BUFFER|NOBUFFER]
[MASK(********|logonidmask)]
[UID( |uidmask)]
[UPPER|NOUPPER]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTDA—MLS DIRAUTH Event Log

```
[SUMMARY|DETAIL]
[EXCLUDE(nnnnnnnn,…,nnnnnnnn|********)]
[INCLUDE(nnnnnnnn,…,nnnnnnnn|********)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTDS—Data Set/Program Event Log

Optional File: HEXDUMP

```
[NLIDMASK(logonid mask,…,logonid mask)]
[MASK( |datasetmask)]
[NMASK(datasetmask)]
[LIDMASK(********|logonidmask)]
[NLIDMASK(logonidmask)]
[UID( |uidmask)]
[SIZE(2500|nnnnn)]
{SHORT]
[LOGGING|VIO|TRACE|PGMNAME|TAPE|INSTALL|UNKNOWN|ALL]
[PRINTER|SUMMARY|TERMINAL]
[EXTEND|NOEXTEND]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRTPEL—Infostorage Update Log

```
[SUMMARY|DETAIL]
[UID( |uidmask)]
[MASK(********|logonidmask)]
[TYPE{ |typemask)]
[CLASS(R|class)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTIX—Data Set Index Report

Optional File: DETAIL

```
[DETAIL|NODETAIL]
[PREFIX(********|mask)]
[SELLID(ACFFDR @SMF LID|nn,…,nn)]
[SELRULE(ACFFDR @SMF RULE|nn,…,nn)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTJL—Restricted Logonid Job Log

```
[MASK(********|logonidmask)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTLL—Logonid Modification Log

```
[SUMMARY|DETAIL]
[MASK(********|logonidmask)]
[UPDATE|NOUPDATE]
[LIDFLDS(nnnnnnnn,…,nnnnnnnn | ********)]
[CHANGER(changermask | ********)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTNV—The Environment Report

Optional Files: DISK, TAPE, or VSAM

```
[CPUID(****|cpuidmask)]
[DBLSPC|NODBLSPC]
[HEADER]
[TRACE]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTOM—Open Edition MVS Report

```
[SUMMARY|DETAIL]
[UID(value)]
[GID(value)]
[USER(logonid)]
[GROUP(groupname)]
[SERVICE(service)|{INCLUDE(service,…,service)}{EXCLUDE(service,…,service)}]
```

ACFRPTPP—The Preprocessor

```
[MASK(********|logonidmask)]
[SMFxx(nnn,…,nnn)]
[SMF$x(nnn,…,nnn)]
[SMFxxxxx(nnn|x,nnn|x,”description”)]
[SELECT(nnn,…,nnn)]
```

ACFRPTPW—Invalid Password/Authority Log

```
[MASK(********|logonidmask)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTRL—Rule ID Modification Log

```
[SUMMARY|DETAIL]
[MASK(********|rulemask)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTRV—Resource Event Log

```
[LOG|VIO|TRACE|ALL]
[MASK(********|rulemask)]
[PRINTER]
[TYPE(***|typemask)]
[UID( |uidmask)]
[SUMMARY|DETAIL]
[CLASS(R|class)]
[COND(4|n)]
[TIME(M|S|H)]
```

ACFRPTRX—The Logonid Access Report

Optional Files: SYSUT1, SYSUT2, SYSIDLST, LOGONIDS, RULES, INFOSTG

```
[ACF2|NOACF2]
[DSET|RSRC]
[LID(********|logonidmask)]
[RMASK(accessrulemask|resourcerulemask)]
[TYPE(type)]
[UID( |uidmask)]
[CLASS(R|class)]
```

ACFRPTSL—Selected Logonid List

Optional Files: BACKUP, SAVREC, SYSLIB

```
[DTCFIELD(YES|NO)]
{INPUT(SMF|BKUP|ACF2)]}
[IF(fieldnameoperators)]
[MASK(********|logonidmask)]
{REPORT(SHORT|FULL|NONE)}
[SFLDS(fieldlist)]
[UPDATE|NOUPDATE]
```

ACFRPTST—The Sectrace Report

```
[ASID]
[DETAIL]
[POSTLOG]
[PRELOG]
[TRACEID]
[COND(4|n)]
```

ACFRPTXR—The Cross Reference Report

Optional Files: SYSUT1, SYSUT2, SYSDLST, SYSRSLST, LOGONIDS, RULES, INFOSTG

```
[ACF2|NOACF2]
[CLASS(R|class)]
[DSET|RSRC]
[DSN(datasetname)]
[LID|NOLID]
[NAME(name)]
[RKEY(rulekey)]
[RRSUM|NORRSUM]
[TYPE(type)]
[VOL(volumeserial)]
```

Using the New Rule Utility (ACFNRULE)

The ACFNRULE utility provides a simple means of adding rules to access and resource rule sets. It can also be used to delete unwanted rules.

ACFNRULE can be executed as a TSO command, or via batch JCL.

DDNAMES for use include:

SYSPRINT

Output file.

SYSIN

Contains input ACFNRULE parameter statements. Used when multiple input parameters are specified or when the size of the input parameters exceeds 100 bytes.

TSO users can specify ACFNRULE parameter statements on the command line. Batch users can specify parameters from the JCL PARM field or SYSIN.

Parameters for this utility include:

```
{[ADD(rule)] [DELETE(string)]}
 [CLASS(class)]
 [KEY(name)]
 [LIST|NOLIST]
 [SYSID(sysid)]
 [TYPE(type)]
 [VERIFY|NOVERIFY]
```

Using the XREF Cleanup Utility (ACFXREF)

The ACFXREF utility provides an efficient method of identifying invalid INCLUDE or EXCLUDE values contained in cross-reference records. ACFXREF is executed via batch JCL.

DDNAMES for use include:

SYSPRINT

Specifies a file for all output including error messages. This DDName must be specified.

SYSIN

Specifies where all ACFXREF parameter statements are supplied. A SYSIN DDName statement is required.

ACFXREF parameters are entered in the SYSIN file. Parameters for this utility include:

```
[CLTYPE(XSGP|XRGP)]
[RECID(recidmask1,…,recidmaskn)]
[RULETYPE(ruletype[,ruletype])]
```

Optional parameters include:

```
[SRCETYPE(XSGP|XRGP)]
 [RECID(-|[recid1],[recid2][recid3],…,[recidN])
```

Chapter 16: Using Console Operator Commands

Enter the following console commands at an operator console:

Command Syntax	Function
F ACF2,BACKUP	Initiate backup of CA ACF2 databases
F ACF2,CACHE(option)	Activate and apply changes to cache facility Options: DEMAND|NODEMAND HELP MONITOR|NOMONITOR START STATUS STOP SYNCRESET
F ACF2,CACHESYN(START|STOP)	Start or stop the cache synchronization process.
F ACF2,CPF(START|STOP)[SYSID(systemid]	Start or stop CA ACF2 CPF
F ACF2,CPF(JOURNAL|NOJOURNAL)	Start or stop CPF journal processing.
F ACF2,DDB(START|STOP)	Reactivate and deactivate DDB
F ACF2,LDS(ACTIVE|NOACTIVE),LDAPNODE(LDAP.xxxxxxxx)	Activate or deactivate the specified LDS LDAP node.
F ACF2,LDS(REMOVE),[LDAPNODE(LDAP.xxxxxxxx)],[UNTIL(date)]	Delete LDS Recovery Records from the LDS Recovery File by ldapnode or a given date. At least one of LDAPNODE or UNTIL parameters must be indicated.
F ACF2,LDS(JOURNAL|NOJOURNAL)	Start or stop LDS journal processing.
F ACF2,MLS	Rebuilds multilevel security (MLS) classification tables
F ACF2,LDS(START|STOP),[SYSID(systemid)]	Start or stop CA ACF2 LDS.

Command Syntax	Function
F ACF2,LDS(DEBUG|NODEBUG),LDAPNODE(LDAP.xxxxxxxx)	Enable or disable the DEBUG trace option for the specified LDS LDAP node.
F ACF2,NEWMOD(ACFFDR)	Reloads the ACFFDR
F ACF2,NEWMOD(ACF00SVA)	Reloads the module that services SVC-A
F ACF2,NEWMOD(ACF9C000)	Reloads the CA ACF2 SAF processor module
F ACF2,NEWMOD(ACF99SVC)	Reloads the module that services SVC-S
F ACF2,NEWMOD(MACPCxxx)	Reloads a CA-MAC PC routine
F ACF2,NEWMOD(SAFxxxxx)	Reloads requested SAF module
F ACF2,NEWUID	Rebuild LID/UID XREF table
F ACF2,NODE(nodeid),Active|Inactive	Activate and deactivate nodes
F ACF2,OMVS	Rebuild OpenEdition MVS tables Options: CERTDATA DCE GID KERB KERBLINK LINUX LNOTES NDS UID
F ACF2,NOMLACTIVE	Deactivate Multilevel Security
F ACF2,LMPCHECK	Have CA LMP verify the LMP execution keys
F ACF2,NEWSHIFT	Reload resident shift matrix tables
F ACF2,NEWXREF[,TYPE(RGP|SGP)]	Rebuild input resource and source cross-reference tables
F ACF2,REBUILD(typecode) [,CLASS(class)]	Re-create directory of resident resource rule sets
F ACF2,REFRESH(recid,…,recid|ALL) [,SYSID(sysid)] [,CLASS(class)]	Apply changes dynamically to GSO records

Command Syntax	Function
[,TYPE(type)]	
F ACF2,RELOAD(ruleid)	Reload resident data set access rule sets
F ACF2,RESET(logonid)	Reduce password violation count for specified logonid
F ACF2,SETCLASS(class)	Set current infostorage class
F ACF2,SETNORUL(jobname|ALL)	Reset and clear locally resident rule chain in user's address space
F ACF2,SETSYS(sysid)	Set current system ID or division
F ACF2,SETTYPE(type)	Set current infostorage type
F ACF2,SHOWCLAS	Display current infostorage class at console
F ACF2,SHOWGSO	Display current GSO trace option at console
F ACF2,SHOWSEBE	Display DDB session block info
F ACF2,SHOWSYS	Display startup and current SYSIDs at console
F ACF2,SHOWTYPE	Display current infostorage type at console
F ACF2,STATS(START|STOP)	Start or stop CA ACF2 STATS gathering
F ACF2,SWITCH(DDSN list name)	Switches databases as defined in the ACFFDR from the active DDSN to those specified by the user. This command must be executed from the MVS console. **Note:** This command must be executed from the MVS console. Issuing this modify command may result in the CA ACF2 databases becoming out of synch. See the *Systems* Programmer *Guide* for more information.
F ACF2,SYSPLEX(START|STOP|CLEAR)	Start, stop, or clear CA ACF2 Coupling Facility

Command Syntax	Function
F ACF2,TRACEGO(option)	Set destination of trace records for GSO events Options: ALL CONSOLE SECURITY SMF SYSLOG
P ACF2	Stop CA ACF2
S ACF2[,00E][,PARM='option']	Start CA ACF2 with specified optional parameters Options: BACKUP|NOBACKUP COMMAND(string) DDSNS(dsngroup) NOCPF NOLDS NOMLACTIVE NOSTATS SYSID(systemid) TRACEGO(destination)
CASF DISPLAY(APPL|POE|USER|GROUP|SECLABEL)	Display Signed_On_From list for LU
CASF SIGNOFF APPL POE USER (GROUP|SECLABEL)	Remove User from Signed_On_From list

This section contains the following topics:

Using DDB Console Commands

Command Syntax	Description
F ACF2,DDB(START)	Reactivate DDB after it has stopped
F ACF2,DDB(STOP)	Deactivate DDB support without affecting CA ACF2 for z/OS and OS/930
F ACF2,NODE(nodeid),Active	Place individual logical node into service

Command Syntax	Description
F ACF2,NODE(nodeid),Inactive	Remove individual logical node or suspend from service

Adding [,XCF,(*)] to any F ACF2 command when the CA ACF2 Cross Coupling Facility (XCF) SYSPLEX support is active broadcasts that command to all CA ACF2 systems connected to the sysplex.

Using SECTRACE Commands

Command Syntax	Description
SecTrace {ENable|DISable|DELete|[D|DISPlay]},ID=trapid|ALL	
SecTrace{[SET|T]MODify|F]},ID=trapid|ALL [,TYPE=SAF|SAFP]	Trap Identification
[,ASID,=nn] [,JOBname=mask] [,USERid=mask]	TYPE Group TRACING Group
[,NZERO|ZERO] [,RB=mask] [,ProGraM=mask] [,RETcode=nn] [,RSNcode=nn]	ENVIRONS Group
[,ENable|DISAble]	STATUS Group
[,ACTION=IGNORE|TRACE] [,TRACE=[PRE|BEFORE][POST|AFTER][ALL]]	EVENTS Group
[,MATCHLIM=0|nn]	ROUTE TO Group
[,DEST=[CONSOLE|JOBLOG|SMF|SYSLOG|TSOUSER| ALL]|DATASET] [,CONSid=nn] [,DSName=dsn] [,TSoUser=id] [,LINELEN=nn] [,Member=name] [,ForMaT=[DUMP|LABEL|NOPACK|PACK]] [,MSGid|NOMSGid] [,WAIT|NOWAIT] {END|CANCEL}	Other Operands

Command Syntax	Description
F ACF2,NODE(nodeid),Active	Place individual logical node into service
F ACF2,NODE(nodeid),Inactive	Remove individual logical node or suspend from service

Chapter 17: Digital Certificate Commands

The following Table summarizes all of the CA ACF2 commands that can be issued to generate, install and maintain digital certificates, key rings, and digital certificate mappings. For more information about Digital Certificates, see the chapter "Digital Certificate Support" in the *Administrator* Guide.

Command	Function	ACF Setting/ Component	Syntax
CHKCERT	CA ACF2 displays information about an X.509 certificate in a CERTDATA profile record or a z/OS data set (including whether it is registered with CA ACF2).	ACF COMMON SUBCOMMAND	CHKcert {*logonid* Label(*label*)| logonid.*suffix*|Dsname(*data-set-name*)} [Password(*password*)] [Nolist] [Dump]
CONNECT	CA ACF2 associates a certificate with a key ring.	ACF COMMON SUBCOMMAND	CONnect Certdata(*userid1.suffix*) Keyring(*userid2.suffix*) [Ringname(*ringname*)] [Label(*label*)] [Usage(PERSONAL|CERTAUTH|SITE)] [DEFAULT]
EXPORT	CA ACF2 exports an X.509 digital certificate from the CA ACF2 database and puts it into a z/OS data set.	ACF COMMON SUBCOMMAND	Export {*logonid*|*logonid.suffix*| Dsname(*data-set-name*)} [Label(label)] [Format(CERTDER|CERTB64| PKCS12DER|PKCS12B64)| PKCS7DER|PKCS7B64] [Password(*password*)]
GENCERT	CA ACF2 generates a digital certificate and inserts a CERTDATA profile record into the CA ACF2 infostorage	ACF COMMON SUBCOMMAND	GENcert {*logonid*|*logonid.suffix*| CERTAUTH|CERTAUTH.*suffix*| SITECERT|SITECERT.*suffix* } [Label(*label*)] [Dsname(*data-set-name*)] [SUbjsdn([CN=*common name*] [T=*title*] [OU=*organizational-unit-name*]

Command	Function	ACF Setting/ Component	Syntax
	database.		[O=*organization-name*] [L=*locality*] [S=*state-or-province*| SP=*state-or-province*| ST=*state-or-province*] [C=*country*])] [SIZe({*key-size*|1024})] [ICSF|PCICC|DSA] [ACtive({*date-or-date-time*| current-date-000000| current-date-time})] [Expire({*date-or-date-time*| current-date-000000| current-date-time})] [SIGnwith({CERTAUTH Label(*label-name*)| SITECERT Label(*label-name*)| CERTAUTH.*suffix*|SITECERT.*suffix*)| Label(*label-name*)}] [Keyusage([HANDSHAKE] [DATAENCRYPT][DOCSIGN][CERTSIGN])] ALtname([IP=*numeric-ip-address*] [DOMAIN=*internet-domain-name*] [EMAIL=*email-address*] [URI=*universal-resource-identifier*])
GENREQ	CA ACF2 generates a certificate request (PKCS #10) to be sent to a Certification Authority.	ACF COMMON SUBCOMMAND	GENReq {logonid|logonid.suffix} Dsname(data-set-name) [Label(label)]

Command	Function	ACF Setting/ Component	Syntax
P11TOKEN	Manager PKCS 11 tokens	ACF COMMON SUBCOMMAND	P11token Add Token(token-name) P11token DELete Token(token-name) [Force] P11token List Token(token-name) | Mtoken(token-mask) P11token Bind Token(token-name)| Certdata({logonid | logonid.suffix}) [Label(label)] [Usage(PERSONAL| CERTAUTH|SITE)] [DEFault] P11token Unbind Token(token-name) Certdata(logonid | logonid.suffix) [Label(label)] Seqnum(sequence #) IMport Token(token-name) Seqnum(sequence #) Certdata(logonid | logonid.suffix) [Label(label)][ICSF][PCICC][Pkdslbl (pkds-label)]
REMOVE	CA ACF2 disassociates a certificate from a key ring.	ACF COMMON SUBCOMMAND	REMove Certdata(*userid1.suffix*) Keyring(*userid2.suffix*) [Ringname(*ringname*)] [Label(*label*)]
REKEY	CA ACF2 generates a new certificate with a new key pair using the contents of an existing certificate	ACF COMMON SUBCOMMAND	REKey{logonid|logonid.suffix|CERTAUTH| CERTAUTH.suffix|SITECERT|SITECERT.suffix} [Label(existing-certificate-label)] [WITHLbl(new-certificate-label)] [WITHSfx(new-certificate-suffix)] [SIZe({key-size|1024})] [ICSF|PCICC] [ACtive({date-or-date-time| current-date-000000| current-date-time})] [Expire({date-or-date-time| current-date-000000| current-date-time})]
ROLLOVER	CA ACF2 rolls over a certificate by removing the old private key, reconnecting the new certificate to the old key rings and updates the serial number base	ACF COMMON SUBCOMMAND	ROllover{logonid|logonid.suffix|CERTAUTH| CERTAUTH.suffix|SITECERT| SITECERT.suffix} [Label(old-certificate-label)] [NEWLabel(new-certificate-Label)] [NEWSufx(new-certificate-suffix)] [Force]

Command	Function	ACF Setting/ Component	Syntax
INSERT	CA ACF2 reads an X.509 digital certificate from a z/OS data set and inserts it, along with data from the command input line, into a CERTDATA profile record, which associates a user with a certificate.	PROFILE USER RECORD (CERTDATA)	Insert {*logonid*|*logonid.suffix*| CERTAUTH|CERTAUTH.*suffix*|SITECERT| SITECERT.*suffix* } [Active(*date*)] Dsn(*data-set-name*) [Expire(*date*)] [Label(*label*)] [Password(*password*)] [HITRUST|TRUST|NOTRUST] [ICSF]
CHANGE	CA ACF2 accepts data from the command input line and, accordingly, changes the CERTDATA profile record(s), which associates a user(s) with a certificate(s).	PROFILE USER RECORD (CERTDATA)	CHAnge {*logonid*|*logonid.suffix*| CERTAUTH|CERTAUTH.*suffix*|SITECERT| SITECERT.*suffix* } [Active(*date*)] [Expire(*date*)] [NEWLABEL(*label*)] [HITRUST|TRUST|NOTRUST] CHAnge *userid* ISSUERDN(*dn*) SERIAL#(*serial-number*) [Active(*date*)] [Expire(*date*)] [NEWLABEL(*label*)] [HITRUST|TRUST|NOTRUST]
DELETE	CA ACF2 deletes the CERTDATA profile record(s), which associates a user(s) with a certificate(s).	PROFILE USER RECORD (CERTDATA)	DELete {*logonid* LABEL(*label*)| *logonid.suffix*| CERTAUTH LABEL(*label*)|CERTAUTH.*suffix*| SITECERT LABEL(*label*)|SITECERT.*suffix*} DELete *userid* ISSUERDN(*dn*) SERIAL#(*serial-number*)
LIST	CA ACF2 displays the CERTDATA profile record(s), which associates a user(s) with a certificate(s).	PROFILE USER RECORD (CERTDATA)	List {*logonid* LABEL(*label*)|*logonid.suffix*| CERTAUTH LABEL(*label*)|CERTAUTH.*suffix*| SITECERT LABEL(*label*)|SITECERT.*suffix*} List *userid* ISSUERDN(*dn*) SERIAL#(*serial-number*)

Command	Function	ACF Setting/ Component	Syntax
INSERT	CA ACF2 inserts a KEYRING profile record, which associates one or more certificates with a single user (logonid).	PROFILE USER RECORD (KEYRING)	Insert {recid|recid.suffix } [Default(userid.suffix)] Ringname(ringname)
CHANGE	CA ACF2 accepts data from the command input line and, accordingly, changes the KEYRING profile record, which associates one or more certificates with a single user (logonid).	PROFILE USER RECORD (KEYRING)	CHAnge {*recid*|*recid.suffix* } [Default(*userid.suffix*)] [Ringname(*ringname*)]
DELETE	CA ACF2 deletes the KEYRING profile record, which associates one or more certificates with a single user (logonid).	PROFILE USER RECORD (KEYRING)	DELete {recid|recid.suffix }
LIST	CA ACF2 displays the KEYRING profile record, which associates one or more certificates with a single user (logonid).	PROFILE USER RECORD (KEYRING)	List {recid |recid.suffix }

Command	Function	ACF Setting/ Component	Syntax
INSERT	CA ACF2 inserts a CERTMAP GSO record, which defines the IDN (issuer's distinguished name) or SDN (subject's distinguished name) filters used to assign a specific logonid to a group of certificates.	CONTROL GSO RECORD (CERTMAP)	Insert CERTMAP.*recid* [SDNFILTR(*subject's-dist-name-filter*)] [IDNFILTR(*issuer's-dist-name-filter*)] [DSN(*data-set-name*)] [CRITERIA(*criteria-name-template*)] [LABEL(*label*)][TRUST|NOTRUST] [USERID(*userid-to-map-to*) [MULTIID|NOMULTIID]
CHANGE	CA ACF2 accepts data from the command input line and, accordingly, changes the CERTMAP GSO record, which defines the IDN (issuer's distinguished name) or SDN (subject's distinguished name) filters used to assign a specific logonid to a group of certificates.	CONTROL GSO RECORD (CERTMAP)	CHAnge CERTMAP.*recid* [SDNFILTR(*subject's-dist-name-filter*)] [IDNFILTR(*issuer's-dist-name-filter*)] [DSN(*data-set-name*)] [CRITERIA(*criteria-name-template*)] [LABEL(*label*)][TRUST|NOTRUST] [USERID(*userid-to-map-to*)] [MULTIID|NOMULTIID]

Command	Function	ACF Setting/ Component	Syntax
DELETE	CA ACF2 deletes the CERTMAP GSO record, which defines the IDN (issuer's distinguished name) or SDN (subject's distinguished name) filters used to assign a specific logonid to a group of certificates.	CONTROL GSO RECORD (CERTMAP)	DELete CERTMAP.*recid*
LIST	CA ACF2 displays the CERTMAP GSO record, which defines the IDN (issuer's distinguished name) or SDN (subject's distinguished name) filters used to assign a specific logonid to a group of certificates.	CONTROL GSO RECORD (CERTMAP)	List CERTMAP.*recid*
SHOW	CA ACF2 displays information contained in CERTMAP records as laid out in the internal CERTMAP table.	ACF COMMON SUBCOMMAND	SHow CERTMAP

Command	Function	ACF Setting/ Component	Syntax
INSERT	CA ACF2 inserts a CRITMAP GSO record, which is used with the CRITERIA parameter of the CERTMAP GSO record, to assign a specific logonid to a group of certificates based on the system ID, application ID, or application-defined variables specified in the CRITMAP GSO record.	CONTROL GSO RECORD (CRITMAP)	Insert CRITMAP.*recid* [APPLID(*application-name*)] [SYSTEMID(*sysid*)] [APPLVAR(*site-variable-list*)] USERID(*userid-to-map-to*)
CHANGE	CA ACF2 accepts data from the command input line and, accordingly, changes the CRITMAP GSO record, which is used with the CRITERIA parameter of the CERTMAP GSO record, to assign a specific logonid to a group of certificates based on the system ID, application ID, or application-defined variables specified in the CRITMAP GSO record.	CONTROL GSO RECORD (CRITMAP)	CHAnge CRITMAP.*recid* [APPLID(*application-name*)] [SYSTEMID(*sysid*)] [APPLVAR(*site-variable-list*)] USERID(*userid-to-map-to*)

Command	Function	ACF Setting/ Component	Syntax
DELETE	CA ACF2 deletes the CRITMAP GSO record, which is used with the CRITERIA parameter of the CERTMAP GSO record, to assign a specific logonid to a group of certificates based on the system ID, application ID, or application-defined variables specified in the CRITMAP GSO record.	CONTROL GSO RECORD (CRITMAP)	DELete CRITMAP.*recid*
LIST	CA ACF2 displays the CRITMAP GSO record, which is used with the CRITERIA parameter of the CERTMAP GSO record, to assign a specific logonid to a group of certificates based on the system ID, application ID, or application-defined variables specified in the CRITMAP GSO record.	CONTROL GSO RECORD (CRITMAP)	List CRITMAP.*recid*

Command	Function	ACF Setting/ Component	Syntax
SHOW	CA ACF2 displays information contained in CRITMAP records as laid out in the internal CRITMAP table.	ACF COMMON SUBCOMMAND	SHow CRITMAP

Chapter 18: Multilevel Security Commands

The following Table summarizes all of the CA ACF2 commands that are specific to a multilevel security (MLS) environment. For more information about other commands used to administer MLS, see the *Multilevel Security Planning Guide* and the *Administrator Guide*.

Function	Command Syntax
Displays your active session security labe	Mlslabel
Rebuilds the multilevel security classifications tables	F ACF2,MLS
Deactivates MLS	F ACF2,NOMLACTIVE
Starts CA ACF2 with multilevel security deactivated regardless of whether MLACTIVE is set in the CONTROL GSO MLSOPTS record	Start ACF2,PARM='NOMLACTIVE'
Activates, deactivates, or queries the write-down setting for the user who issues the command	MLWRITE[STATUS|ENALBE|DISABLE | RESET]