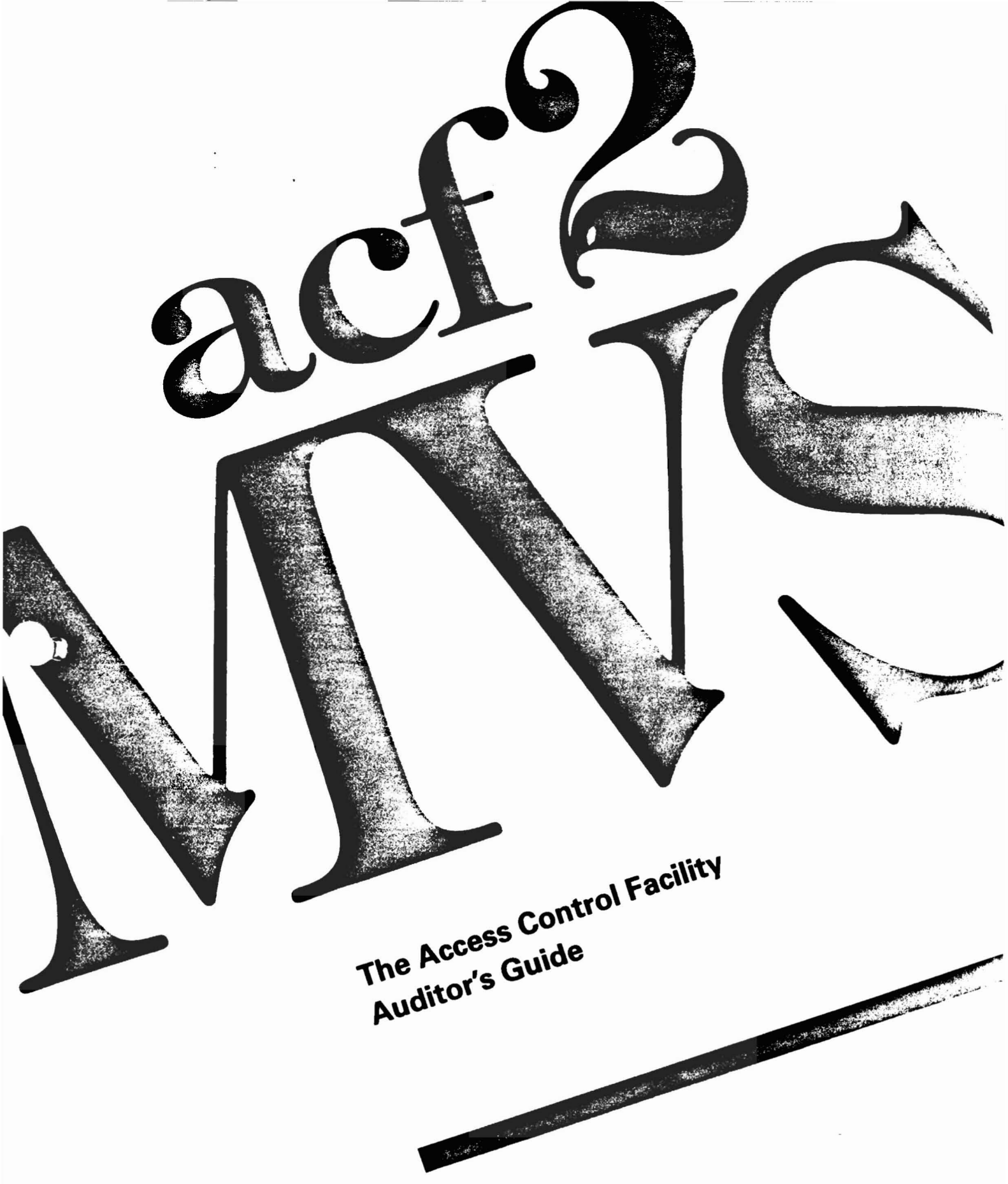
acf2
MVS
The Access Control Facility
Auditor's Guide

acf2™

The Access Control Facility

AUDITOR'S GUIDE

for

acf2/MVS Release 4.0 Installations

Base Manual Dated: January 15, 1985

Doc. Nr. ABP0006-02

Printed in U.S.A.

ACF2 is a proprietary product developed and maintained by:

SKK, Inc.
10400 West Higgins Road
Rosemont, Illinois 60018-9990

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

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

ACF2 AUDITOR'S GUIDE

INTRODUCTION

An important managerial responsibility is the review and evaluation of an organization's automated systems. Because sensitive organizational information is frequently stored on these systems, sound auditing practices must necessarily be implemented to maximize security.

Several professional associations have recommended that all organizations have effective internal controls, especially those with automated systems -- the Independent Commission on Auditors' Responsibilities, the Financial Executives Institute, the Institute of Internal Auditors, and the American Institute of Certified Public Accountants (AICPA). Also, several laws have been implemented in many countries that require organizations to establish accounting and auditing procedures.

For example, in the United States, the Security and Exchange Commission (SEC) has stated that public companies must "review their accounting procedures, systems of internal accounting controls and business practices" in order to comply with the requirements of the Foreign Corrupt Practices Act of 1978. This act applies not only to foreign transactions, but also to the assessment and verification of existing domestic accounting and data processing controls. If the controls are insufficient, then an organization must promptly implement a plan that affords effective security with auditing capability.

ACF2 SECURITY PHILOSOPHY

The authors of ACF2 define data security as the protection of data against unauthorized disclosure, modification, or destruction. ACF2 protects all data by default, and it shares data only on explicit action by the data owner or security officer. Therefore, ACF2 is not only a data protection system, but a system that provides for the controlled sharing of data in accordance with the authorizations defined to it by the installation.

COMPONENTS OF ACF2

In ACF2, an algorithmic methodology is used to determine whether access to a given dataset or other defined resource by an individual user under a specific environment should be allowed. The ACF2 algorithms, called rule sets, are composed of access or resource rules. An authorized ACF2 user (perhaps a dataset owner or a security officer) first defines a rule set, then the rule set is transformed into object records. These object records can then be stored in the ACF2 access rule database. When ACF2 needs to consult rules to determine whether or not a service should

be performed (for example, the opening of a dataset), ACF2 translates the related rule set.

ACF2 has placed intercepts in the MVS operating system components and in various subsystem components such as TSO, JES, IDMS, IMS, and CICS. Using these intercepts, ACF2 gains control the service is performed and decides if the request should be processed in one of three ways:

- allow
- allow but log to SMF
- deny and log to SMF

ACF2 makes this decision based on the total environment of the request and whether a rule specifies that an access under those conditions should be allowed. The "total environment" might include (for a dataset access) aspects such as the user making the request, the dataset name, the volume it is on, where the job came from, the program making the request and the library it came from, the DDNAME specified in the job's JCL, the date of the request, and additional local items if specified.

ACF2 includes other special control features, such as specific controls over certain programs, over the use of various TSO commands or BLP (Bypass Label Processing), over use of terminals and readers, and in many other areas. It also includes numerous report generators, TSO commands, and other aids to assist in the administration and audit of the system.

ACF2 DOCUMENTATION

There are also manuals provided with ACF2 that explain the all aspects of the system. This guide is designed to help the internal or external EDP auditor to audit the use of ACF2 controls at the installation. It should also be useful to security officers and administrators at the installation.

AUDIT PLANNING

Auditing ACF2 is only part of a larger systems audit effort, which will be different at each site. The technical expertise of the auditors and their experience with ACF2 will also differ from site to site. This manual will cover only the areas relating directly to the implementation and use of ACF2. It highlights the sensitive or critical areas of ACF2-related controls that should be reviewed. This is not a complete audit approach for an installation, but rather a guide to ACF2-related aspects that should be considered. Portions of this can be modified and incorporated into the installation's overall audit plan as appropriate. Throughout the manual, ACF2-provided audit aids will also be highlighted to help explain how an ACF2 audit could be approached.

Also, list of sample audit survey questions has been included as Appendix E to this manual. This not a complete list of questions. It is included here to provide some ideas and examples of how an organization's policies and requirements may be incorporated into part of the audit plan at an ACF2 site.

SCOPE OF ACF2

One of the most important aspects of ACF2 security that the auditor should address is identifying the scope of ACF2's controls at that installation.

ACF2 can be easily tailored to different needs and surrounding environments. A number of ACF2 options are available and the installation's choices in the use of these options can have a dramatic effect on how much control ACF2 does or does not provide at that site. The three means of defining system options to ACF2 are:

1. Via the ACF2 Field Definition Record Generation (ACFFDR) macros. See the acf2/MVS System Programmers' Guide for a complete description of the ACFFDR macros and their function.

2. Global System Options (GSO) records which are stored on the ACF2 Infostorage Database. See the acf2/MVS Administrator's Guide for complete information about each record and associated options.

3. Optionally, ACF2 can also be used to protect resources available through CICS (IBM's Customer Information Control System), IMS (IBM's Information Management System), and IDMS (Cullinet's Integrated Database Management System). The scope of ACF2 controls for these products are defined on an individual basis for each CICS, IMS, and IDMS address space. Control options for each of these product interfaces are described in the appropriate ACF2 support manual (e.g., the ACF2/IDMS Support Manual describes the options available for IDMS).

The detailed description of every possible option should be reviewed by referring to the appropriate manuals. Review of the values cited in all of the ACFFDR, GSO records, IMS, CICS, and IDMS systems should be frequent and thorough. This step should also be taken very early during an ACF2 audit.

Some of the more critical control options will be reviewed here to help identify their importance and how they can significantly alter the level of ACF2 protection at a site.

ACF2 BOUNDARIES SET VIA THE ACFFDR AND GSO

A number of values defined via the ACFFDR macros and the GSO records effect the boundaries of the ACF2 controls. The current settings of these values at the site should be checked to determine what portions of ACF2 controls are active. The ACFFDR values can be checked by reviewing the input to the ACFFDR assembly processing or the output of this process, as long as adequate controls are in place so that you are confident you are reviewing the appropriate (active) copy.

Displaying ACF2 Control Options

Most of the ACFFDR and GSO values can be verified online using the SHOW subcommand on TSO, such as the SHOW ACF2, SHOW SYSTEM, SHOW STATE, etc. ACF2 commands or subcommands can also be submitted via batch jobs by running the TMP in batch or by running the utility ACFBATCH (see appendix a for sample JCL). In addition, SPF/ISPF (System Productivity Facility) screens are provided to display the ACF2 system parameters, Logonid records, GSO records, access rules, generalized resource rules, etc.

Samples of the outputs from the ACF SHOW subcommand are included in appendix B. The SHOW subcommand allows any auditor or security administrator (any ACF2 user with the AUDIT or SECURITY attribute, regardless of any scope field values) to display the system options currently active at that site as generated via the latest ACFFDR assembly and GSO entries.

Important GSO Record Control Options

Some of the GSO values displayed by SHOW commands that are of particular interest in determining the active ACF2 boundaries are:

OPTS MODE - Displayed on SHOW STATE as:

MODE = ABORT/WARN/LOG/QUIET/RULE,no-rule,no-$mode

This identifies the mode the main ACF2 system is in. This applies to all ACF2 processing, except each IMS and CICS region (which each have their own individual MODE values), or any portion of ACF2 processing whose mode is affected by local ACF2 exit coding. If this system mode is set to anything other than ABORT or RULE minimal protection is being offered by ACF2 since even access attempts specifically prevented by the rules will be allowed (though logged if in LOG or WARN mode). QUIET, LOG, RULE, and WARN modes are only provided to assist in the transition to full ACF2 security and the installation should proceed to ABORT mode as soon as possible. Also, if the mode is RULE or ABORT, use of various system exits must be carefully audited to determine if any local coding is being used to allow accesses that ACF2 would consider violations, and

are thus circumventing the rules and the ABORT mode. Refer to the discussion on Local Exits later in this manual.

> Note: When MODE=RULE has been specified, the SHOW STATE output will include 3 mode specifications, such as MODE=RULE,ABORT,ABORT. In this example, the second parameter of ABORT is taken when no rule record is found, and the third parameter of ABORT is taken when no $MODE control card is included in the rule set.

OPTS NOSORT - Displayed on SHOW STATE as:

$NOSORT/NO$NOSORT

If NOSORT is in effect and a $NOSORT control card is used in a rule, the normal ACF2 sorting of rules from most specific to most general is suppressed. Therefore, all rule sets with the $NOSORT control card should be carefully reviewed to ascertain that rule entries are in the proper sequence and that no general rule placed early in the rule set inadvertently supersedes a more specific rule appearing later in the rule set. Use of $NOSORT should be very limited and any use should be justifiable by the data owner.

OPTS NOTIFY - Displayed on SHOW SYSTEM as:

NOTIFY/NONOTIFY

When this option is in effect, an informational message is produced for users whenever they logon/signon to the system. The message indicates the date and time of the last system access; users should be instructed to use the information displayed to verify that there has not been any unauthorized use of their Logonid since their last legitimate session.

OPTS STC - Displayed on SHOW STATE as:

STC/NOSTC

This indicates if ACF2 is to validate any access requests made by any system (started) tasks. If NOSTC is in effect, any system task can access any dataset regardless of ACF2 access rules. Current STCs on the system, as well as the procedures and controls for adding new STCs to the system, should be reviewed carefully if NOSTC is in effect.

OPTS UADS - Displayed on SHOW STATE as:

UADS/NOUADS

Indicates if the UADS (User Attribute Dataset) is being used for TSO Logonids. If it is being used, the majority of the fields in the TSO section (Group 5) of the ACF2 Logonid record

are not active (i.e., not used by ACF2) and the related value is obtained from the UADS file rather than from ACF2's Logonid records. The only fields in the TSO section which are not affected by this option are: ALLCMDS, CMD-LONG, VLD-ACCT, VLD-PROC, and WTP. These fields are always active. All other TSO fields in the ACF2 Logonid record are only active when NOUADS is in effect. Therefore, if UADS is in effect, procedures for the control and maintenance of the UADS dataset should also be reviewed.

PPGM - Displayed on SHOW PROGRAMS or SHOW PGMS as a list under the heading:

RESTRICTED PROGRAM NAMES

This is used to identify each specific program name or program name pattern that ACF2 is to control based on the execution of programs by that name, regardless of the library it is from or what accesses it is attempting. Programs matching this list may be executed only by users with the SECURITY or the NON-CNCL attribute, and ACF2 logs the occurrence of each execution. The programs which should be identified on this list are those which do not use standard system services (such as the standard Open SVC) and thus could bypass system security by avoiding the ACF2 system intercept points. Some programs of this type would be Innovation Data Processing's Fast Dump and Restore (FDR and FDRDSF) and the IBM utilities IEHDASDR and DRWDASDR.

For those programs specified in PPGM, each should also be stored in an ACF2-protected library with close control over who can read (copy) programs from that library so that the program cannot be easily copied, renamed, and executed under an uncontrolled name.

Also see the section entitled "Program Controls" later in this manual for further discussion of PPGM and for discussion of other records (such as MAINT, BLPPGM, LINKLST, and LOGPGM) and ACF2 controls which relate to monitoring program usage.

RESVOLS - Displayed on SHOW STATE as a list under the heading:

DSNAME PROTECTED VOLUMES

SECVOLS - Displayed on SHOW STATE as a list under the heading:

VOLSER PROTECTED VOLUMES

OPTS TAPEDSN - Displayed on SHOW STATE as:

TAPEDSN/NOTAPEDSN

The RESVOLS, SECVOLS, and OPTS TAPEDSN specifications must be considered jointly, as together they determine on which media and to what extent access to datasets will be controlled by ACF2's access rules.

For accesses to datasets on DASD or MSS, ACF2 always first checks the DASD or MSS volume name against the list specified by RESVOLS. If the volume name matches a volume name or name pattern on this list (RESVOLS could be six asterisks, in which case all DASD/MSS volumes would be considered as matching), then ACF2 will check the access request against the dataset name access rules in the normal manner. If a match is found on RESVOLS, then SECVOLS is not checked.

If the volume name does not match the RESVOLS list, then ACF2 will check the SECVOLS list for a matching volume name or name pattern. If there is a match on SECVOLS, then ACF2 will check the access request against a special access rule under the volume's name (an access rule under the pseudo dataset name @volser.VOLUME or VOLUME.@volser depending on installation option). If the DASD or MSS volume name does not match either the RESVOLS list or the SECVOLS list, accesses to it are not controlled by ACF2 (i.e., no access rules will be checked and any access attempt, including allocate or scratch, would be unconditionally allowed - unless otherwise checked by a local exit).

For accesses to datasets on tape, ACF2 does not check the RESVOLS list, but does check the SECVOLS list. If the tape volume name matches a name or name pattern on the SECVOLS list, then the applicable volume name rule is used to validate the request. If the tape volume name does not match any SECVOLS entries, then the OPTS TAPEDSN field value is checked. If this is set to NOTAPEDSN, then no rule checking is done by ACF2 and accesses to that tape volume will not be controlled by ACF2. However, note that use of BLP is controlled by ACF2 and is active independent of the SECVOLS and OPTS TAPEDSN values. If OPTS TAPEDSN is in effect, then ACF2 will validate access requests for datasets on tape at the dataset name level, checking the normal dataset access rules.

A good configuration for an installation may be an RESVOLS value of "******", no SECVOLS entries, and OPTS TAPEDSN. This would ensure that accesses to all present and future datasets, regardless of storage media, would be validated (at the dsn level) by ACF2. Of course, other considerations, such as non-standard labeled tapes, the availability and use of a tape management system, multiple datasets on a single tape volume, etc. must be taken into account when reviewing the full scope of the security afforded and the best approach for an installation.

ACF2 AND IMS

Optionally, ACF2 can interface with and help control usage of IMS systems, but this protection is optional and must be specifically requested by the way ACF2 and each IMS Control Region is generated into the system. Two actions are required for each IMS system, the specification of IMS-related options to ACF2 via ACF2's @IMS macro, and the inclusion of ACF2-related factors in the generation of the IMS system via the @IMSGEN macro. If these actions have not been taken or inappropriate parameter values were used, then ACF2 will not be providing control over IMS users, terminals or transactions.

The IMS SECURITY macro parameters are required to generate the ACF2 interface to IMS. The optional @IMSGEN macro can be used to point to an alternate ACF2 @IMS parameter list whose values are active for that IMS Control Region. The values in the ACF2 @IMS macro that are significant to the bounds of ACF2 IMS controls are:

MODE=LOG/ABORT - This parameter works the same as the ACF2 system MODE parameter except that here it applies to the related IMS system(s) only. Again, the installation should use LOG mode as a transition step only and must be in ABORT mode to have true protection active.

Resource Type Specifications AGN, CDL, LCK, PTP, TLK, and TRN - These parameters indicate 1) which types of IMS processing are to be controlled by ACF2 resource rules and 2) which ACF2 generalized resource rule type code is used to store the rules for each aspect being protected. If any of these parameters equals "IGNORE", than those IMS activities will not be within the scope of ACF2 controls.

The parameters are for Application Group Name authorization checking (AGN), Change DL/I calls (CDL), lock and unlock operator commands (LCK), program to program switches (PTP), transactions entered from IMS MSC links (TLK), and transactions entered from IMS terminals (TRN). If any of these parameters equals an ACF2 resource type (e.g., IAG or ITR), then there should be ACF2 generalized resource rules under these types which cite the authorizations for use of each resource name (or name pattern) under each type code.

SPECIAL CONSIDERATIONS FOR AUDITING ACF2/CICS SYSTEMS

A completely updated ACF2/CICS interface was introduced in acf2/MVS Release 4.0. However, the ACF2/CICS interface distributed with pre-Release 4.0 systems will still function under Release 4.0. The auditor must be aware of this fact when auditing an ACF2/CICS system. The particular audit concerns for both ACF2/CICS interfaces are discussed below.

Regardless of which version of the ACF2/CICS interface is used, ACF2 support must be defined for each CICS region. Therefore, separate control parameters must be built for each region. The installation specifies the resources that ACF2/CICS will protect. The installation can also specify a three-character generalized-resource type code for each resource to be protected by ACF2.

ACF2/CICS Interface Introduced in Release 4.0

The ACF2/CICS interface introduced in Release 4.0 includes generalized resource validation for the following CICS resources:

* transactions
* programs
* files
* transient data
* DL/I calls
* temporary storage
* MRO requests (Multiple Region Operation)

ACF2 must be explicitly directed to validate access to all of the above resources via the ACF2/CICS "System Initialization Parameters". The parameters are defined in a sequential dataset by the system programmer who installs the ACF2/CICS interface. For this reason, it is important that adequate access controls are in place for the dataset that contains these parameters.

In addition, a transaction named ACFM is provided that allows these ACF2/CICS control parameters to be dynamically updated from a CICS terminal. Strict controls should be placed on who can use the ACFM transaction and it might be desirable to log all use of ACFM. ACFM also allows all curent ACF2/CICS control parameters to be displayed at a CICS terminal, which is a convenient tool for auditors.

Control parameters that the auditor will be particularly concerned with are:

CICSKEY - there can be any number of CICSKEY parameters for each CICS region. Each one describes a particular CICS resource and directs ACF2 to validate requests to the resource or to ignore (always allow) accesses to the resource. The CICSKEY parameters also define the three character ACF2 resource type code to be associated with a particular type of CICS resource. By default, ACF2 uses the following type codes:

* CKC for transactions
* CPC for programs
* CFC for files
* CTD for transient data
* CPB for DL/I calls
* CTS for temporary storage

MODE - specifies the security mode of the ACF2/CICS system. Options are QUIET, LOG, and ABORT. Unless ABORT is specified, ACF2 will not deny a user access to CICS resources. The QUIET and LOG options are provided as transition aids to be used during initial implementation of ACF2/CICS. LOG will cause resource validation to be performed (based on CICSKEY parameters), but users will be granted access to the resource (with an SMF logging) even if a rule does not specifically grant the access. When in QUIET mode, ACF2/CICS does no resource validation.

SAFELIST - these parameters define those CICS resources that all CICS users may access. No ACF2/CICS validation is performed when a user requests access to a "safe" resource. The auditor will want to ensure that the SAFELIST parameters do not inadvertantly allow users access to a sensitive resource.

ACF2/CICS Interface Available in Pre-Release 4.0 Systems

In the ACF2/CICS interface distributed in pre-Release 4.0 systems, the ACF2/CICS control parameters are specified via macros that require assembly and placement in the system Link Pack Area (LPA). These macros are @CICSOPT, @CICSKEY, and @CICSAFE.

@CICSKEY - there can be any number of @CICSKEY macros for each CICS region. Each macro describes a particular CICS resource and directs ACF2 to validate requests to the resource or to ignore (always allow) accesses to the resource. The @CICSKEY macros also define the three character ACF2 resource type code to be associated with a particular type of CICS resource.

By default, ACF2 uses the following type codes:

* CKC for transactions
* CPC for programs
* CFC for files

@CICSOPT: MODE parameter - specifies the security mode of the ACF2/CICS system. Options are LOG, and ABORT. Unless ABORT is specified, ACF2 will not deny a user access to CICS resources. The LOG option is provided as transition aid to be used during initial implementation of ACF2/CICS. LOG will cause resource validation to be performed (based on @CICSKEY macros), but users will be granted access to the resource (with an SMF logging) even if a rule does not specifically grant the access.

@CICSAFE - this macro defines those CICS resources that all CICS users may access. No ACF2/CICS validation is performed when a user requests access to a "safe" resource. The auditor will want to ensure that the @CICSAFE macros do not inadvertantly allow users access to a sensitive resource.

ACF2 AND IDMS

An optional ACF2 interface for Cullinet's Integrated Database Management System (IDMS) is provided.

ACF2/IDMS support must be defined for each IDMS region. Therefore, separate control parameters must be built for each region. The installation specifies the resources that ACF2/IDMS will protect. The installation can also specify a three-character generalized-resource type code for each resource to be protected by ACF2.

The ACF2/IDMS interface includes generalized resource validation for the following IDMS resources:

* tasks
* protected programs
* non-protected programs
* data areas
* subschemas

ACF2 must be explicitly directed to validate access to all of the above resources via the ACF2/IDMS @GRCE macros. Also, the ACF2/IDMS interface requires that a macro named @MOPTS be assembled. Control parameters in the @MOPTS macro determine other ACF2 options for the IDMS system. These macros are assembled into a load module which is obtained by ACF2 when the IDMS system is started.

Control parameters that the auditor will be particularly concerned with are:

@GRCE macro: INTTYPE and INSTYPE parameters - there can be any number of @GRCE macros for each IDMS region. Each one describes a particular IDMS resource and directs ACF2 to validate requests to the resource or to ignore (always allow) accesses to the resource. The @GRCE parameters also define the three character ACF2 resource type code to be associated with a particular type of IDMS resource. The installation must define these type codes as ACF2 supplies no defaults.

@MOPTS macro: MODE parameter - specifies the security mode of the ACF2/IDMS system. Options are LOG, and ABORT. Unless ABORT is specified, ACF2 will not deny a user access to IDMS resources. The LOG option is provided as transition aid to be used during initial implementation of ACF2/IDMS. LOG will cause resource validation to be performed (based on @GRCE macros), but users will be granted access to the resource (with an SMF logging) even if a rule does not specifically grant the access.

@GRCE macro: SAFE parameter - - this parameter defines those IDMS resources that all IDMS users may access. No ACF2/IDMS validation is performed when a user requests access to a "safe" resource. The auditor will want to ensure that the SAFE parameters do not inadvertantly allow users access to a sensitive resource.

USE OF LOCAL EXITS

A major significant factor which affects the total scope of ACF2 protection is the use of exits to insert local code and thus alter (or circumvent) the ACF2 decision processes. There are currently a number of local exits provided in the standard ACF2 product. Additionally, exits in other subsystems could also be used to alter the results of ACF2 controls (e.g., Logon Pre-prompt User Exit, SMF exits, or JESx exits). One of the very first things that should be checked when reviewing an ACF2 installation is the usage of these exits. If these exits are being used, the actual code placed in each one should be carefully reviewed to determine exactly what takes place.

The need to review these exits early in the system review stems from two important considerations. First, the exits should be carefully checked to ascertain that only authorized activities take place there. Procedures and controls over changes to these exits should also be reviewed. Second, it is necessary that the authorized activities that do take place be understood by the reviewer(s) before the rest of the ACF2 parameters, rules, and records are checked, since these exits could have a profound impact on how ACF2 processing will proceed. For

example, an exit could affect which access rule ACF2 will actually use to check a dataset access request, or it could decide to bypass the ACF2 check altogether. Keep in mind that the effect of local exits is not apparent from a review of decompiled rules or via the ACF TEST subcommand. Therefore, the exits themselves must be checked carefully.

The SHOW ACTIVE subcommand assists in the review of ACF2 exits by indicating the existence of any exits. Each of the ACF2 exits which has active code at that installation will be identified, along with the module name used for the code at that installation, in the SHOW ACTIVE output.

The various ACF2 exits and some of the effects they could have on ACF2 processing include:

Access Rule Authorization Pre-Processing Exit (the applicable GSO EXITS record field is "RULEPRE") - This exit allows installation control of the retrieval of access rule records.

Access Rule Authorization Post-Processing Exit (the applicable GSO EXITS record field is "RULEPST") - This exit allows installation control of the storing of access rule records.

Dataset/Program Pre-Validation Exit (GSO EXITS record field is "VLDEXIT"; SHOW ACTIVE text is "DSN VALD EXIT") - This exit could alter the dsn or rule set key ACF2 will use for validation, or instruct ACF2 to unconditionally allow, log, warn, or abort a request without rule checking.

Dataset/Program Post-Validation Exit (GSO EXITS record field is "DSNPOST") - This exit receives control after ACF2 has determined if an attempted dataset access is to be allowed.

Dataset/Program Violation Exit (GSO EXITS record field is "VIOEXIT", SHOW ACTIVE text is "DSN VIO EXIT") - This exit may also alter the ACF2 recommendations (e.g., allow the access versus abort).

Expired Password Exit (GSO EXITS record field is "EXPPXIT", SHOW ACTIVE text is "EXP PSWD EXIT") - This exit may allow system access even when the user's password is expired and/or may set a new password for the user (and display it to him).

Infostorage Authorization Pre-Processing Exit (the applicable GSO EXITS record field is "INFOPRE") - This exit is similar to the access rule exit described above and allows installation control of the retrieval of records on the Infostorage database.

Infostorage Authorization Post-Processing Exit (the applicable GSO EXITS record field is "INFOPST") - This exit allows installation control of the storing of records on the Infostorage database.

JESx USER01 Exits - Local code in these exits could also affect ACF2 processing by altering the Logonid or the source name, or by having the system bypass password validation requirements.

Logon Parameter Exit - (GSO EXITS record field is "LGNPARM", SHOW ACTIVE text is "TSO LOGON PARM")- This exit allows installations to alter certain selected parameters (e.g., account and procname before ACF2 builds the JCL to complete logon validation process.)

Logon Terminal Exit- (GSO EXITS record field is "LGNTERM", SHOW ACTIVE text is "TSO LOGON TERM TYPE"). This exit allows installation to identify special printing devices.

New Password validation Exit (GSO EXITS record field is "NEWPXIT", SHOW ACTIVE text is "NEW PSWD EXIT") - This exit may enforce certain syntax or formats on user's passwords. If these formats are too stringent or the permissible patterns are too widely known, this could lead to easy guessing of passwords.

Pseudo DSN Generator Exit (GSO EXITS record field is "DSNGEN", SHOW ACTIVE text is "PSEUDO DSN GEN") - This exit may alter the dsn ACF2 will use for validation against the rules. It may also select (point ACF2 to) the rule set key to be used. This exit can also instruct ACF2 to unconditionally allow, log, warn, or abort a request without ACF2 doing any rule checking at all.

Resource Pre-Validation Exit (GSO EXITS record field is "RSCXIT1", SHOW ACTIVE text is "RSRC PRE EXIT") - This exit can do everything the Pseudo DSN or dataset Pre-Validation exits could do, except that it applies to generalized resource rules instead of dataset access rules.

Resource Post-Validation Exit (GSO EXITS record field is "RSCXIT2", SHOW ACTIVE text is "RSRC POST EXIT") - This exit may alter the ACF2 recommendations, e.g., it could change the ACF2 decision to abort the request to "allow, no log".

Source Name Modification Exit (GSO EXITS record field is "SRCXIT", SHOW ACTIVE text is "SRC MOD EXIT") - This exit may modify the source name ACF2 will use for validating the user's access to the system or for matching rules. This could make the access request appear as if it were from a different entry point than it really was.

Supervisor Call Initialization Exit (GSO EXITS record field is "SVCIXIT") - This exit receives control before ACF2 Supervisor Call processing begins. It is particularly useful within a MUSASS environment.

System Task Validation Exit (GSO EXITS record field is "STCXIT", SHOW ACTIVE text is "STC VALD EXIT") - This exit could modify the Logonid used to validate accesses by started tasks.

TSO Logon Pre-Validation Exit (the applicable GSO EXITS record field is "LGNIXIT", and the SHOW ACTIVE output text which will appear with the module name if this exit is active is "LOGON PRE EXIT") - This exit may modify the Logonid (or the password) that ACF2 will use for checking.

TSO Logon Post-Validation Exit (GSO EXITS record field is "LGNPXIT", SHOW ACTIVE text is "LOGON PST EXIT") - This exit may modify information sent to TSO or used to build the TSO JCL.

Again, any of these exits present at your installation should be carefully reviewed for usage, and their effect on the rest of ACF2 processing. Most of this manual (as well as other ACF2 manuals) is written assuming no exits are present to alter ACF2 processing. Thus when these exits are being used, you must consider the consequences when reading any other section of the manual as well as when auditing the system. This could even mean trying to manually simulate the exit processing when reviewing rules or output reports.

EXTERNAL ENVIRONMENT

As mentioned earlier, the purpose of this manual is to address the management, administration, and audit of the ACF2 portion of your installation's approach to data processing security. There are many other aspects of security outside the scope of direct ACF2 control, and thus outside the scope of this manual. However, these related aspects are still very important, and no security implementation or EDP audit is complete without giving these areas careful consideration. In short, the controls provided by ACF2 are only as secure as the foundation on which the operating system is built and the environment in which it exits.

To ensure that all internal and ACF2 controls are operating properly, system maintenance procedures must be carefully reviewed. Some of these areas are referred to in this manual - for example, comments on the necessity to check the procedures and controls related to the local establishment and modification of ACF2 exits. Others are not discussed in this manual but still should not be overlooked because of their impact on the overall validity of your system controls. Some examples of these other areas for investigation are:

- The procedures/controls that are in place for adding or changing
  - SVCs
  - STCs
  - PPT (Program Properties Table)
  - APF Authorized Programs/Libraries, PARMLIB, etc.
  - System/Subsystem code (vendor provided and/or locally generated)
- The procedures/controls that exist for
  - reassembling the ACFFDR
  - altering GSO entries
  - re-linking ACF2
  - modifying/adding ACF2 exit code
  - changing the TSO Command Limiting load modules
  - changing the ACF2 parameters for CICS, IMS, and IDMS interfaces or other product interfaces.
- The method in which operating system integrity problems are reported, documented, tested, and fixed.
- The procedures/controls that are in place for hands-on (physical) access to the computer system (e.g., operator consoles) and data (e.g., disk packs and tape volumes).

SYSTEM ACCESS CONTROL

Since ACF2 determines whether or not an individual user should be allowed access to a resource, it must be able to associate a user's identity with each job or time-sharing session. No job will run on an ACF2-controlled system unless it can first be identified with a valid, predefined user. Thus ACF2 is also protecting the resources of the computer system itself. No one can use processing time on your system unless running under a Logonid you have previously defined to ACF2.

Each user should be assigned a unique Logonid for ACF2 control. Use of individual Logonids for each individual user should be encouraged and the procedures for assigning, changing, and using Logonids should be reviewed.

The reports should also be reviewed carefully to see how the Logonids are being used, with special attention to powerful Logonids and production jobs. ACF2 default Logonids are provided as ACF2 implementation aids, and should appear only infrequently after the initial transition period. Since the default ids are only assigned to jobs for which valid regular Logonids were not available, there is no control of the actual submitting user. Therefore, no rules should be written authorizing default ids to access data or granting default ids special privileges.

Since the key to ACF2's control of system access is the Logonid, most of the details on user controls will be discussed in the following section, The Logonid Record.

THE LOGONID RECORD

The ACF2 Logonid record contains the information about each user that ACF2 needs in order to make decisions regarding access to resources and the user's authority to update ACF2 databases. It may also contain some installation-specified fields. A number of these fields are merely informational or for efficiency and ease, while others are very critical to the operation of ACF2 and the definition of controls at that installation.

SEPARATION OF FUNCTION

An auditor must ascertain if there is adequate separation of function between different activities and between users with different powers. Use of ACF2 is no exception. There are a number of special privileges and special powers which may be granted to users via ACF2 Logonid and system options. There are also related activities (such as system maintenance functions) which take place at an installation and which should be considered due to their possible impact on ACF2 controls.

SPECIAL USERS

One important area concerns the assignment of the ACF2 attributes of SECURITY, ACCOUNT, AUDIT, LEADER, and CONSULT. Each of these attributes carries with it special powers which should not be available to the normal ACF2 user. Any given user should not have all of these attributes at once. In fact, SECURITY, ACCOUNT, and AUDIT should normally be mutually exclusive to help enforce separation of function. However, many of the exact privileges that each of these attributes entails is variable (changeable by the installation itself), so policies on the usage of each of these may be different for each installation.

As these special attributes are predefined in ACF2, the following general definitions would apply:

SECURITY attribute - Users with this attribute are normally the data security administrators or officers for the installation. Normally their duties would include the maintenance of all access rule and generalized resource rule records, input source entry lists, and scope and shift records. An unrestricted security officer (one with SECURITY and no SCPLIST or DSNSCOPE limits) can create, change, list, or delete any rule record or ACF2 Infostorage record (such as entry lists, scope lists, and shift definitions). He or she can also access any resource since even if a permitting rule did not exist, he or she has the authority to create or change

such a rule. However, any accesses that security officers are permitted outside the rules are logged and flagged. Users with the SECURITY attribute but restricted with a DSNSCOPE or SCPLIST entry can still modify any rules and access any resource within their scope. Refer to "Centralization and Decentralization" for more information regarding SCOPE fields.

The unrestricted security officer can also execute any program on the restricted programs list (PPGM). He can also change and display certain fields in user's Logonid records which no other users (including restricted security officers) can change. However, the privileges related to changing selected Logonid record fields are modifiable by the installation. Note that a person with the SECURITY attribute cannot establish or delete Logonid records unless he or she also has the ACCOUNT attribute.

A restricted security officer can also change or display various fields in existing Logonid records (which ones are modifiable by the installation). A security officer can be limited to having access authority to Logonid records for only certain users by use of the LIDSCOPE, UIDSCOPE, or SCPLIST field of the Logonid record.

Any security officer (restricted or unrestricted) can utilize various TSO ACF subcommands not available to the normal user (such as SHOW STATE, SHOW ACTIVE, and SHOW TSO).

ACCOUNT attribute - Users with this attribute are normally assigned the responsibility to establish, maintain, and delete Logonid records. The ACCOUNT attribute grants no privileges relative to rule writing or resource accessing.

Persons with ACCOUNT can also utilize the various SHOW subcommands. They can also change and display a large number of individual Logonid fields (which fields can be defined by the installation).

Unrestricted account managers (users with ACCOUNT and with no LIDSCOPE, UIDSCOPE, or SCPLIST limits) are the only ones who can execute the SYNCH command to synchronize the ACF2 Logonid database with the TSO BRODCAST dataset.

Note: A user with both the SECURITY and the ACCOUNT attributes is considered more powerful than one with only one of these attributes. Thus a user with only one cannot modify the Logonid record of a user with both.

AUDIT attribute - A user with this attribute can normally display all Logonid records, all rules and source entry lists, and all system control options (e.g., use the SHOW subcommands). However, an auditor should not have the authority to update or delete any of these records or to access any resources except

those specifically authorized to him via access or resource rules. An auditor can be restricted to only certain rules via DSNSCOPE or SCPLIST, to certain Logonid records via LIDSCOPE, UIDSCOPE, or SCPLIST, and to certain Infostorage records via SCPLIST. The READALL privilege (defined in the Logonid record) may be set to allow an auditor or any other privileged user the ability to access all datasets at the installation, regardless of the access rules. This is similar to the NON-CNCL attribute, but grants READ access only and continues to enforce the existing rules for any other type of access.

LEADER attribute - This attribute does not grant any special powers relative to rules, entry lists, or ACF subcommands. Leaders also may not create or delete Logonid records (unless they also have ACCOUNT). Leaders can, however, change or display a limited number of fields in existing Logonid records. Which fields they have access to can be controlled at the field level at the installation, and which Logonid records they have access to can be controlled via LIDSCOPE, UIDSCOPE, and SCPLIST. Leaders are normally not too powerful and normally have very limited scopes.

CONSULT attribute - This attribute is normally given to users who assist other users in using the computer system. Normally consultants cannot update anything significant in Logonid records but can display some of the less sensitive fields in order to help answer questions. Which fields they can display or alter are assignable by the installation, and which Logonid records the consultant can access can be controlled by the LIDSCOPE, UIDSCOPE, or SCPLIST.

A combination of system options (set in the ACFFDR and GSO) and individual Logonid record values determine the extent of some of the powers associated with each of these ACF2 special privileges. These must be reviewed together as there are a number of interrelationships.

One area to review is the assignment of ALTER, LIST, AUTH and FLAGS values for each field of the Logonid record. This requires a review of the ACFFDR @CFDE entries (there is one @CFDE entry for each ACF2 standard or locally defined Logonid record field). Those keywords listed after "ALTER=" in each @CFDE identify which special ACF2 privileges (SECURITY, ACCOUNT, AUDIT, LEADER, CONSULT, or USER) a user must have in order to alter the value of that field in his (or anyone else's) Logonid record.

Additionally, the "AUTH=" parameter in the @CFDE macro may be used to indicate that a locally defined attribute in a requestor's Logonid record will be checked for this field prior to passing control to "ALTER=" processing. Some fields have no ALTER values; these fields should be updated by ACF2 only (internally) and no users should be authorized to change them. The authority to change a field can also be affected by the FLAGS parameter. If ALTER=SECURITY but FLAGS=RESTRICT (or RESTRICT plus any other parameters), then the ability to alter the

field is limited to unrestricted security officers (no DSNSCOPE or SCPLIST).

The "LIST" values in the @CFDE record indicate what user privileges are needed in order to display the values of that field. The auditor should ensure that the AUDIT attribute is included as one of the LIST= attributes on every field (except PASSWORD, which nobody can list) so that he has the authority to see values of all Logonid data elements for a user. Similarly, no field (other than PASSWORD) should have FLAGS=NEVER, as this would also prohibit an auditor or anyone else from displaying that field's value, regardless of the LIST= parameters.

In the case of both the ALTER and LIST values, a user is only required to have one of the attributes indicated in order to have the necessary authority, even though the format of these lists requires plus signs ("+") to be used. For example, if a Logonid record field has ALTER = SECURITY + ACCOUNT + LEADER in its @CFDE entry, any user with either the SECURITY, ACCOUNT, or LEADER attribute turned on could change that particular field (assuming the Logonid record being changed belonged to a user within his "scope" - see discussion on scope limitations under Centralization and Decentralization).

A listing of all the Logonid record fields that come predefined with the ACF2 package (have @CFDEs in the default ACFFDR) is provided in Appendix C to this manual. They are arranged by the ACF2 display groups. For each field, ACF2 special user attributes which are authorized to alter and/or list that field are identified. These are the default values (the way ACF2 comes defined for you when you unload the distribution tape) and are probably not exactly the way any given installation has them defined. The actual @CFDE ACFFDR values your installation is running with should be checked. Appendix C should be useful as a guideline for that review.

Another control over the definitions of SECURITY, ACCOUNT, and similar ACF2 authorities is the GSO OPTS record DECOMP field. This is used to define which types of users are authorized to decompile and display rule records on a general basis. Again, see the section on Centralization and Decentralization for additional related controls. Also, how DECOMP is set for your installation can be determined by looking at the SHOW STATE output for the "DECOMP AUTHORITY = " value. The default value (as ACF2 is distributed) allows users with SECURITY and/or AUDIT to do decompiles.

CENTRALIZATION AND DECENTRALIZATION

There are various facilities built into ACF2 to assist in centralizing or decentralizing different aspects of ACF2 administration. These need to be reviewed to determine who has the authority to add or change user Logonid records or rules, who can further delegate these authorities, and what separation of function is in effect. For example, scope fields within each Logonid record provide the ability to limit an individual's administrative authority.

LID, UID, and DSN SCOPES

The SECURITY attribute powers can be limited by also assigning a value to the LIDSCOPE, UIDSCOPE, DSNSCOPE, or SCPLIST fields in the user's Logonid record. DSNSCOPE limits which rule sets the Security Officer can compile or decompile. LID and UID scopes limit which user Logonid records the security officer can list and/or alter fields in. A security officer with DSNSCOPE or SCPLIST fields set to other than blank (i.e., having any limits set) is considered a limited security officer.

The ACCOUNT, AUDIT, LEADER, and CONSULT powers can also be limited by citing either a LIDSCOPE or UIDSCOPE to identify over which user Logonid records their privileges extend. Additionally, if the ACFFDR @CFDE entry for a particular field such as NAME shows ALTER=ACCOUNT and a special user with ACCOUNT privileges also has a LIDSCOPE or UIDSCOPE value (LID or UID pattern), then that account manager can only alter the NAME field of Logonid records whose LID or UID matches the applicable LID or UID pattern in the scope record.

Support for DSNSCOPE, LIDSCOPE, and UIDSCOPE will be removed in a future release of the acf2/MVS product. The function of these fields has been replaced by the more flexible scope list feature (see below).

Scope Lists

Limitations may also be imposed on the special users described above by means of the ACF2 scope list feature. This feature allows a multiple selection of Logonid records, rules, and Infostorage records to be placed within the authority of a user. A LIDSCOPE, for example, assigns a single group of Logonid records to be within the range or scope of a user. Similarly, the DSNSCOPE is limited to one high-level index group. By defining a scope list to a user (via the SCPLIST field of the Logonid record) multiple dataset indexes, Logonid records, and UID strings can be placed within the user's authority.

It should be noted that the presence of the SCPLIST field takes precedence over any other scope defined to that user (LID, DSN, or UID). Furthermore, the scope list is a determinant of whether a security officer or account manager is restricted or unrestricted. The presence of the SCPLIST field in a user's Logonid record defines him as restricted regardless of the presence or absence of any other UID, LID, or DSN scopes and regardless of the actual values or limits specified within the scope list. If no SCPLIST field is specified in the Logonid record, ACF2 will then check for the presence of other scopes.

Careful audit considerations should be given to use of scope lists. Scope records that are very long (e.g., numerous entries for the same type) and the overuse of masking within such a record should be of concern. It is important to display the records and regularly review their use. Also, care should be taken to determine whether a user has authority to change his own scope list (e.g., the user's SCPLIST name matches the Infostorage mask specified). The ACF command may be used in SCOPE mode to display these records:

```
SET SCOPE(SCP)
LIST {*/scope-name/LIKE(scope-name-mask)} -
     {ALL,DSN,INF,LID,UID}
```

The SET subcommand places the ACF command in the proper mode for processing or displaying scope list records. The LIST subcommand may be used to display:

1. A single record (by entering one scope-name).

2. The previous scope list record referenced in this session (by entering '*').

3. A group of records (by entering the LIKE operand and a mask).

Rule Compilation and Decompilation Authorities

In addition to the users with attributes listed in DECOMP, those users who have the authority to change a rule can always decompile that rule. The following options determine who can change a rule:

SECURITY attribute - Security officers (users with the SECURITY attribute in their Logonid record) can write or change any rule set with a key (high-level index) which matches their DSNSCOPE or SCPLIST name pattern (or any rule set if they have no DSN-type limit).

NOCENTRAL- If the GSO OPTS record CENTRAL field is set to NOCENTRAL, that means ACF2 rule writing authority is not centralized (for security officers only) but rather is decentralized. This means that each user has the authority to change the rule set or sets which match his owned dataset prefix (which is in the Logonid record field named PREFIX). For TSO users, PREFIX would normally be equal to the user's TSO Logonid. Thus the user could write or change the rule set whose key (high-level index) is his Logonid. Note that since PREFIX can contain a mask (pattern), it could match multiple high-level indexes. A user with a PREFIX field of all asterisks under NOCENTRAL could change any rule set, even though he is not a security officer. The CENTRAL field is a system-wide option.

%CHANGE - A "%CHANGE uid1,uid2,...,uidn" entry may be present in any access or generalized resource rule set. When a

%CHANGE entry is present, the users who match the UID strings listed after the %CHANGE entry are normally allowed to decompile, compile, and store the rule set, to change rule entries and control cards within the set, and to further delegate the %CHANGE authority (by changing the %CHANGE control card within the set).

%RCHANGE - A "%RCHANGE uid1,uid2,...,uidn" entry indicates "restricted" %CHANGE authority. The users who match the UID strings listed in %RCHANGE can change only rule entries within the rule set. They cannot change control cards, and therefore cannot further delegate %CHANGE or %RCHANGE authority. However, there are two more fields which affect how %CHANGE and %RCHANGE operate at an installation (see NOCHANGE and NO-STORE below).

NOCHANGE - If the GSO OPTS record CHANGE field is set to NO CHANGE, then the %CHANGE and %RCHANGE authorities are not operational at that installation. This is a system-wide field.

NO-STORE - If a user has NO-STORE turned on, he or she is not authorized to store (change) any rule sets. Thus, even if NOCENTRAL is stipulated or a %CHANGE OR %RCHANGE option gives the user authority to change a rule, "NO-STORE" allows the user to only decompile (display) or compile and test the rule, but not actually change it (store a new rule).

DISPLAYING AND CHANGING INFOSTORAGE RECORDS

Users with the unrestricted SECURITY attribute are allowed to list, delete, insert, and change any record that resides on the ACF2 Infostorage Database. These records include generalized resource rules, entry lists, GSO records, scope lists, and shift and zone records.

As previously discussed, the authority of the SECURITY user can be restricted by the presence of a scope list (SCPLIST) specification in the Logonid record. However, there is an option in the OPTS record called INFOLIST that can be used to allow scoped users to list (read-only access) all records residing on the Infostorage Database.

The INFOLIST option specifies the Logonid attributes (such as ACCOUNT, SECURITY, AUDIT, etc.) required to list records on the Infostorage database.

CRITICAL LOGONID RECORD FIELDS

There are a number of Logonid record fields which, as ACF2 is distributed, cannot be changed by any user except an unrestricted security officer (i.e., ALTER=SECURITY only and FLAGS=RESTRICT in the @CFDE entry the ACFFDR). These fields represent extremely powerful attributes which must be carefully assigned. These fields are:

ACCOUNT	AUDIT	AUTODUMP	DSNSCOPE
DUMPAUTH	EXPIRE	IDMS	IMS
JOB	JOBFROM	LIDSCOPE	LOGSHIFT
MAINT	MUSASS	NON-CNCL	NO-SMC
REFRESH	RULEVLD	SCPLIST	SECURITY
STC	TAPE-BLP	TAPE-LBL	TSO
UIDSCOPE			

Descriptions of each field can be obtained from the acf2/MVS Administrator's Guide or via the ACF command HELP FIELDS subcommand. Some of the other Logonid fields which authorize powerful privileges and which are discussed elsewhere are:

RESTRICT/SUBAUTH/PROGRAM
LEADER and CONSULT (powers are installation-variable)
PREFIX

Additional critical Logonid record fields are the PASSWORD and any field which makes up part of the UID string for that installation. Special care should be given as to who can alter any of these fields on any user's record.

Two things are important when reviewing the use of these fields. First of all, who can alter and display the contents of each of these fields should be carefully checked (by reviewing the ALTER, LIST, AUTH, and FLAGS parameters of each related @CFDE entry in the ACFFDR). Second, which Logonid records (which users) have been granted the special privileges should be checked for appropriateness. This can be done for each of the "bit" type fields listed above (each one that represents a privilege, either turned on or off) by using the ACF LIST IF(field-name) subcommand. The ACFRPTSL (Super Logonid List) report generator can also be used to select and list groups of users with special privileges.

SENSITIVE LOGONID RECORD FIELDS

Users who are given the authority to alter or list critical and sensitive fields in the Logonid records should be selected carefully, since they can influence the overall effectiveness of the security system. Sensitive fields in the Logonid record include:

NAME - The name of the user assigned that Logonid. This is displayed on logging and security reports.

CANCEL, SUSPEND, MONITOR, PSWD-EXP, TRACE, and TSO-TRC - These fields are used to temporarily or permanently remove a user's privilege to access the system and to produce special audit trails (reports) of his activities.

IDLE - This field is used to control IMS and CICS operations by specifying the number of minutes an operator may leave his terminal inactive (no transactions submitted) before being forced to re-validate his password (IMS) or to re-signon (CICS). If it is set too high or not set at all, operators can leave their terminals for extended periods of time without signing off.

MAXDAYS - This field identifies the maximum number of days allowed between password changes by the user before ACF2 expires the password and forces a change. This value can be different for each user, and should be a smaller number for those users with more powers (such as security officers) to help prevent their passwords from being discovered by others.

MINDAYS - This field defines the minimum number of days that a newly-changed password must remain active before the user may change it. One use may be to keep users from changing their password to a temporary password and then immediately changing back to the old one again.

SOURCE, CICS,,IDMS, IMS, JOB, TSO - These fields identify the input sources from which the user can submit jobs or enter the system. They can help control the usage of various Logonids. These fields can be used independent of, or in conjunction with, the SOURCE fields in the rule records.

PSWD-DAT, PSWD-VIO, SEC-VIO, SHIFT, ZONE - These fields include information on the violations incurred by the user, allowable access time periods, and time zone definitions. They are useful for monitoring users' activities, and are vital to certain ACF2 routines in determining when to suspend the user's privileges.

TSOCMDS - - This field indicates the name of the list of TSO commands the user is authorized to use. It can help control usage of various powerful commands, including the ACF command itself if desired. Which list is assigned to each user, as well as the contents of each list, should be reviewed.

Most of the Logonid record fields mentioned above come defined in the standard ACF2 package (via the ACFFDR @CFDE entries) as modifiable only by a security officer or an account manager. These fields should normally be controlled at that level, and not be alterable by the user himself or leaders, consultants, or auditors. However, the circumstances at any given installation may differ, and again the @CFDE entries for your site must be reviewed and considered in light of local conditions.

Additionally, no one should have the authority to alter Logonid fields which ACF2 maintains internally. Again, the local @CFDE entries should be checked to see who (if anyone) is authorized to ALTER these fields. As ACF2 comes predefined, no one is authorized to alter the following fields: ACC-CNT, ACC-DATE, ACC-TIME, CSDATE, CSWHO, PSWD-TOD, UPD-TOD. Correct operation of ACF2 could be jeopardized by allowing users, even security officers, to change these fields. For example, the PSWD-TOD value is used in the password encryption algorithm. If it were changed, the password would no longer match. However, under special circumstances, an installation may wish to grant special user authority to change one of these fields.

For example, it might be appropriate to allow a security officer to reset some or all ACC-CNT fields to zero and use the field to help identify which users were the heaviest system users, or whether or not some special Logonid (like a batch production Logonid or a default id) was being overused.

Which Logonid record fields can be displayed or changed by a user can also be determined via the ACF SHOW FIELDS subcommand. Examples of outputs from this subcommand are in Appendix B to this manual. The SHOW subcommands can be executed via TSO, batch utilities, and also via ACF2-supplied SPF/ISPF screens.

DATASET ACCESS CONTROL

ACF2 protects all data by default from everyone except the data owner. Each Logonid on the system has an "owned dataset prefix" which is displayed as the value of the PREFIX field in the Logonid record. During an attempted access to a dataset, the high-level index (the characters in the dataset name before the first period) is compared to the value of the user's PREFIX field. If they match, access is automatically allowed. Otherwise, a search is made for an access rule set that governs access to that index. If no access rule set for that index exists, then access is denied. The PREFIX field could be set to blanks, which would effectively require all accesses by that user to check the rules. Similarly, the RULEVLD field of the Logonid record could be turned on for this user, which also requires that the rules be checked for all dataset accesses.

After the access rule set is obtained, the access rules are interpreted to locate one that matches the environment that currently exists. If no access rule matches the environment, then access to the dataset is denied. On the other hand, after an access rule is located that matches the current environment, the purpose of the access request is compared against the READ, WRITE, ALLOCate and EXECute-only permission values specified in that access rule. In accordance with what is specified for these privileges, the request is handled in one of three ways:

- allowed
- allowed, with an informational journal entry made (logged)
- prevented, with a "violation attempt" journal entry made.

Access rules also refer to a pseudo-field of the Logonid record called the User Identification String (UID). The UID string is constructed dynamically at job initiation by concatenating specific character fields from the user's Logonid record. The fields, which are concatenated to form the UID string, are determined by the ACFFDR @UID specification, as defined by the installation. For example, the UID string for a site might be composed of the department, job responsibility code, and the user's Logonid, such as:

```
dpjlogonid
Where:
     dp = department
     j = Job responsibility code
     logonid = Logonid
```

As a specific example of the above, take an Accounting department with its personnel split between Receivables and Payables. Then the UID string would look like:

ACRxxxxxxxx for the Accounting department employees authorized to work o Receivables

ACPxxxxxxxx for the Accounting department employees authorized to work o Payables

where xxxxxxxx is the individual's Logonid.

When an access rule is being set up, ACF2 allows an asterisk (*) to be used anywhere in the specification of the UID string to indicate that any character in the user's UID string is to be considered valid (or "matched"). In addition, UID specifications are padded on the right with asterisks to the maximum length of the UID string, so that only the left hand portion need be specified. Using the previous example, some valid UID string specifications are:

UID(AC) refers to anyone in the Accounting department

UID(ACR) refers to anyone in Receivables

UID(ACP) refers to anyone in Payables

UID(AC*T1234) refers to users with Logonids beginning with T1234 and who are employed by the Accounting department.

THE ACCESS RULE SET

RULE SET ELEMENTS AND SYNTAX

Access rule sets consist of control cards, optional comments, and rules. The rule set syntax is as follows:

1. Control cards are denoted by a dollar sign ($) or a percent sign (%) and must begin in column 1. The $ control cards must appear before any % control cards or any rules.

2. Comments are denoted by an asterisk (*) in column 1. Comment lines may be placed anywhere in a rule set, except between continuation lines of a single rule entry. Note that comments are not preserved during compilation so that when the rule is later decompiled (via the DECOMP subcommand), they will not appear in the output.

3. Rule entries should start in column 2 (although they may start in column 1 if they do not start with a $, %, or *) and may be continued by specifying the last character on the line as a dash (-).

Example:

```
$KEY(key)
$MODE(QUIET/LOG/WARN/ABORT)
$NOSORT
$OWNER(owner-id)
$PREFIX
$USERDATA
%CHANGE uid1,uid2,...,uidn
%RCHANGE uid1,uid2,...,uidn
* sample comment
 rule # 1 -
 continued rule # 1
* sample comment
 rule # 2
 rule # 3
```

The $KEY control card contains the high-level index of the dataset name which this rule set will govern (users who have no authority to write rules for data other than what they own, as specified in the PREFIX field of the Logonid record, will enter their Logonid in the $KEY card). The $MODE card indicates the mode for this individual rule set (will be used only if MODE(RULE) as been set in the GSO OPTS record). This control card is provided as a transition tool to be used to "ease" the system into ABORT mode. $NOSORT indicates that this rule set will not be sorted by ACF2 and thus may not be in most specific to most general

order. Auditors should carefully review the rule entries for any rule set that contains $NOSORT, ascertaining that a valid reason exists for the use of this control card within the set and that the resultant rule entry sequence is proper and desired. $OWNER is simply an informational field of up to 24 characters to be used for local tracking or reporting purposes.

The presence of a $PREFIX card may indicate that local exit coding (Dataset Pre-Validation Exit) is being used to alter ACF2's rule selection process. Thus extra care must be taken to ensure that the rule you are reviewing is really the one ACF2 (as modified by the exit) would use. Additionally, the $PREFIX card will be specified when the NEXTKEY feature is used to point to the "alternate" rule set to be checked. Care should be taken to ensure that proper control of NEXTKEY is maintained, since up to 25 alternate rule sets can be referenced. It is also important, when reviewing a rule entry with a NEXTKEY, to make sure the correct alternate rule set is checked. (See the NEXTKEY description later in this chapter.) The $USERDATA card contains up to 64 characters of comments which will be stored with the rule set (unlike comment cards, which will be discarded at compilation time).

The %CHANGE control cards lists the UIDs that, in addition to security officers or those whose owned dataset prefix is equal to the index, may update the access rule set. Therefore the %CHANGE control cards delegate authority for rule replacement. %RCHANGE indicates users with 'restricted' %CHANGE authority. They may change rule entries only, and may not further delegate the %RCHANGE authority. If a UID matches both %CHANGE and %RCHANGE control cards, the higher authority (%CHANGE) would apply. The UID(s) may be fully specified UIDs or UID patterns (may contain asterisks (*) to indicate that any character may be present). A list of UIDs or patterns may also be specified. Multiple users could therefore have the authority to change or delete the same rule set.

In ACF2 access rules, dataset references may also be specified as either fully-qualified dataset names or as patterns. An asterisk may be used to indicate that any character may be present and a minus sign (-) as the only character in an index level may be used to indicate that any number of index levels may be substituted. Some examples of dataset patterns are:

```
A*.DATA which matches A.DATA
                      AB.DATA
                      A1.DATA

A*******.DATA which matches A.DATA
                            A1234567.DATA

A.- which matches A.DATA
                  A.B.C.D.DATA
                  A
```

```
-.LOAD which matches A.LOAD
                     A.B.C.D.LOAD
                     LOAD
```

In addition, a minus sign can also be used as a shorthand notation for padding the index level with asterisks. Thus, in the above example:

A*******.DATA is equivalent to A-.DATA

The full format of an access rule is:

```
dsn VOL(vol) UID(uid) SOURCE(source) SHIFT(shift) LIB(lib) -
PGM/PROG(pgm) DDN(ddname) UNTIL(date)/FOR(days) -
READ(r) WRITE(w) ALLOC(a) EXEC(e) DATA(data) -
NEXTKEY(next-key)
```

which says that a dataset which matches the pattern specified by "dsn" and resides on volume "vol" being accessed by a user "uid" submitting a job from input source "source" during the time/date specified by "shift" while executing program "pgm" which resides in a library "lib" via the ddname "ddname", may open the dataset for input "r" or output "w" or for program loading (execute-only) "e". "a" gives permission for allocation, deletion, renaming and cataloging of the dataset. These parameters are described more fully below. Note also that a hyphen or minus sign used as the last non-blank character on a line indicates that the rule is continued on the next line.

The specification of the parameters are:

dsn - (required) - a dataset name pattern. The compiler/interpreter will always prefix it with the high-level index specified by the KEY control card (unless a PREFIX field is also present for that rule set, in which case the value of the PREFIX field will be used to prefix any dataset name in the rule set).

vol - (optional) - a pattern specifying the specific set of volumes on which the dataset must reside in order to match this rule. If omitted, any volume will be allowed.

uid - (optional) - a pattern specifying the set of users to which this rule should apply. If omitted, all users of the system will be considered as "matched".

source - (optional) - the job input source group name for which this rule should apply. If omitted, any input source will be valid. Contact your security officer for a list of valid source group names or list entry types SRC and SGP under SET ENTRY mode.

shift - (optional) - the name of the shift record on the Infostorage database that applied to this rule entry. It defines the days, dates, and times that this rule line is in force.

lib - (optional) - a dataset pattern specifying the set of libraries where the currently executing program must reside in

order for this rule to apply. If this pattern is not specified in quotation marks, the compiler will prefix it with the index specified by the KEY control card (or the PREFIX control card, if one is present). The library name 'SYS1.LINKLIB' is used to specify all the libraries in the system link list and Link Pack Area. If omitted, any library name will be considered as matching the rule.

pgm - (optional) - a program name or a pattern defining the set of program names (within the set of libraries specified by the LIB keyword) which must be the executing program for this rule to apply. If omitted, any program will be considered matched. May be specified as PROG or PGM.

ddn - (optional) - a pattern specifying the DDNAME that must be used in the JCL for the applicable DD card for this rule to apply. If omitted, any DDNAME will be considered matched.

date - (optional) - a Gregorian date specified as mm/dd/yy (or dd/mm/yy or yy/mm/dd, depending on an installation option) which will be the last date on which this rule will be considered valid.

days - (optional) - a number of days from the day the access rule set was compiled that this rule will be considered valid. The minimum number that may be specified is zero (meaning today) and the maximum number is 365.

r - (optional) - a letter A, L, or P. This is used to specify the read access (opening the dataset for input) permission to be applied if there is a successful match of the DSN, VOL, UID, SOURCE, LIB, PGM, and DDN parameters. "A" indicates that access should be allowed, "L" indicates that access should be allowed and journalled (logged), and "P" indicates that access should be prevented. The default value is "P" for prevent.

w - (optional) - the same as r except that it applies to write access (opening the dataset for output).

a - (optional) - the same as r except that it applies to new dataset allocation, deletion, rename or catalog functions.

e - (optional) - the same as r except that it applies to access by the initiator or TSO CALL command for program loading. If omitted, the value will be set to the READ value.

data - (optional) - any character string up to 64 characters. This string will be retained with the rule set and formatted when the rule set is decompiled. Your installation may have standards concerning the format and use of this string.

next-key - (optional) - the next or alternate rule key to be checked by ACF2. This option is frequently used to split large rule sets,

such as SYS1, into smaller ones. NEXTKEY can also be used to merge groups of datasets that have identical access requirements. NOTE that NEXTKEY is only used when an ABORT condition is detected (if a rule entry allows access, then the NEXTKEY is not searched). See the acf2/MVS Administrator's Guide for detailed examples of the use of this feature.

When the ACF2 compiler is invoked for a rule set, it re-orders access rules so that the rule defining the most specific environment will be first. This is the order that rules will be searched at execution time when the rule set must be consulted to determine if access should be permitted. This order can be displayed by first compiling and then decompiling (via the DECOMP subcommand) an access rule set. The order specified in the output of the DECOMP subcommand will be the order ACF2 will use in its search. Rules become active as soon as they are stored via the STORE subcommand, the ACFCOMP command, or the ACFBCOMP program. However, since an access rule set is only obtained from the ACF2 database once per job or online session, a new or revised rule may not affect jobs or sessions already in progress when the access rule set was stored.

REVIEWING ACCESS RULES

Once the mechanics of rules usage are understood, it is easy to recognize that there are a number of aspects to review. First of all, each existing high-level index should have a rule set (with the possible exceptions of items such as TSO Logonid datasets not shared by their owners). Once all the rules have been identified, each rule set should be reviewed in some detail. When reviewing each rule set, the security policies of the company and the installation must be kept in mind as well as standard good business practices. The ACF2 rules are an attempt to automate the enforcement of these policies and practices. Thus some of the common areas which should be checked include:

1. Ensuring that rule entries are correct - check for the wrong number of asterisks in a name mask pattern, for contradictory entries, etc.

2. Ensuring that rule entries are not too general for adequate controls, like "-" (everything) allowed, or the allocate authority used inappropriately.

3. Ensuring that "log" versus "allow" is used where access is authorized but audit trails are desired.

4. Ensuring that local naming conventions, need-to-know, and similar policies are being enforced by the rules as desired.

5. Ensuring that authority to modify the rules is not being inappropirately delegated (via %CHANGE, %RCHANGE, and/or PREFIX entries and/or multiple security officers).

6. Ensure that desired mode is enforced - if $MODE is used in individual rule sets, then MODE(RULE) should be set in the ACFFDR. Default modes for no-$mode and no-rule set conditions should also be specified. Be aware that the $MODE value can override a rule entry read, write, allocate, or execute access permission.

7. Ensure that, if the $NOSORT control card is used, the rule entries are in the desired sequence.

8. Ensure that, if the NEXTKEY feature is used, the correct alternate rule set is referenced.

Of course, these types of questions are in addition to the obvious topics of review such as whether or not the sensitive/secret/critical data is being appropriately protected by rules which correctly authorize only those with need-to-know to access the data. Sensitive datasets, such as system or production datasets (like the master payroll file) should receive early and careful attention. In addition, the ACF2 control datasets and their alternate and backup datasets should be protected by appropriate rules as well as the logging (SMF) datasets.

One tool to help check that sensitive and critical datasets are appropriately protected is the ACFRPTXR (Cross Reference) report generator. Given a specific dataset name, list of dataset names, or resource name, this utility will identify each and every user who has access to that dataset or resource and/or has the authority to change the rules. Additionally, the Logonid Access Report (ACFRPTRX) provides this information in order by Logonid, and matches users with dataset and resource rules.

In addition to the rules themselves, a number of system options and user Logonid record fields should also be reviewed as these can impact the way the system is running. These related elements are the GSO OPTS record CENTRAL, CHANGE, DECOMP, and MODE fields (see the earlier section on Centralization and Decentralization), and Logonid Record fields DSNSCOPE, NO-STORE, PREFIX, SCPLIST, and SECURITY (see Logonid Record section).

A rule set can be decompiled, or displayed, either online via TSO or ACF2 SPF screens (DECOMP subcommand of ACF) or in batch (ACFBDCMP batch decompiler or DECOMP command via batch TMP, see Appendix A or the Utilities Manual). This output will indicate who last changed and stored this rule set, when that was done, and the percentage of available space used.

The output of the rule sets as displayed by decompiling will also be sequenced the way ACF2 would check the rules. This is always "from most specific to most general" unless the $NOSORT control card has been specified and is active (GSO OPTS record $NOSORT field). $NOSORT indicates that no sorting of the rule set should be done by ACF2. Furthermore, the ACF TEST subcommand can also be used to test rule interpretations. This is particularly useful for large or complicated

rule sets to test that the rules are operating as designed and that accesses are authorized or prevented under varying circumstances as desired.

In some cases, it may be appropriate to review the past versions of rules (and who changed them and when). This can be done via batch utilities and report generators provided with the ACF2 package. See the acf2/MVS Utilities Manual, specifically the ACFRPTIX (Index) report and the ACFRPTXR (Cross Reference) report. Last but not least, when reviewing rules you must be aware of any local exit code on the system which may affect rule selection, interpretation, or effectiveness (see the section on the Use of Local Exits). Refer to Appendix D of this manual for examples of rule writing techniques and additional hints on what to look for when reviewing or auditing access rules.

THE GENERALIZED RESOURCE RULE SET

Generalized resource rule sets consist of control cards, optional comments, and rule entries. Resource rule set syntax is as follows:

1. Control cards are denoted by a dollar sign ($) or a percent sign (%) in column 1. The $ control cards must appear before any % control cards or any rule entries.

2. Comments are denoted by an asterisk (*) in column 1. Comment lines may be placed anywhere in a rule set except between continuation lines of a single rule entry. Note that comments are not preserved during rule compilation, so that when the rule set is later decompiled they will not show in the output.

3. Rule entries should start in column 2 (although they may start in column 1 if they do not start with a $, %, or *) and may be continued on another line by specifying a dash (-) as the last character on the first line.

Example:

```
$KEY(resource-name) TYPE(type-code)
$NOSORT
$USERDATA
%CHANGE uid1 uid2 .... uidn
* sample comment
  rule entry #1 -
  continued rule #1
* sample comment
  rule entry #2
  rule entry #3
```

The $USERDATA field is an informational field for local use. The $NOSORT card, when specified, indicates that no sorting of the rule set is to be done by ACF2. The %CHANGE cards list the UIDs that, in addition to the security officer, may update the rule set. This allows the security officer to delegate authority.

The full format of individual resource rule entries is:

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

which says a user whose UID string matches the pattern specified as "uid", entering from the input source "source" during a date/time specified in the given "shift" definition and prior to the expiration date specified via UNTIL or FOR, has the permission specified (ALLOW, LOG, or PREVENT) to use the resource named in the $KEY field (which is of the generalized resource type defined by the "type-code"). The SERVICE keyword specified the type of file access valid for this rule (for CICS file access rules only).

The descriptions for these fields are the same as given previously in the section on the access rule set, with the exception of the SHIFT field. The SHIFT name indicates the shift record on the Infostorage database which applies to this resource. It defines the days, dates, and times that this resource may or may not be accessed in accordance with this particular rule entry.

The $KEY field "resource-name", unlike the $KEY field in access rules, can be a pattern. The installation must use the GSO RESDIR record to create a directory for those generalized resource rule type codes which use masks in the $KEY field. Any audit or review of the resource rule sets must take these patterns into account when determining which rules apply to which resources.

For example, the installation could specify the use of a directory for type code TAC (TSO account number validations) and use masks in the $KEY field of TYPE(TAC) rule sets. If their TSO account numbers followed a naming convention of two alpha characters followed by three numerics, some possible $KEY values might be AB123, AB***, A*12*, *B***, **123, **1**, etc.

When reviewing the decompiled rule sets to determine who has the authority to use a given TSO account number under certain conditions, be careful to review the appropriate rule set. In the above example of $KEY values, the samples listed are in the sequence they would be checked by ACF2 for a match on the account number being requested. Note that ACF2 would still process the most specific record first, and that it proceeds only until it finds the first match.

Most of the other comments made earlier about access rule sets also apply to resource rule sets. For example, the %CHANGE entry, the system options CHANGE and DECOMP, and the use of the SECURITY and NO-STORE attributes all affect who can list and change rules.

For examples of some generalized resource rules and some comments on their use and interpretation, see Appendix D.

PROGRAM CONTROLS

A number of options and facilities in ACF2 are associated with program controls and privileges. Of course, all program source and load libraries are datasets which can be protected by ACF2 access rules to help control who can change programs (WRITE authority), who can look at or copy programs (READ authority) and who can execute programs (EXEC-only authority). It is highly recommended that programs (including commands) be carefully segregated into multiple libraries with users granted only the minimum access authorities needed to do their jobs. Special care should be taken in securing powerful commands and utilities and in protecting critical and/or proprietary software from unauthorized or unnecessary destruction, disclosure (including copying), or modification.

In addition to using access rules for library protection, there are other system and user options which relate to programs. For example, the authority to use BLP (tape bypass label processing) can be specified at the program name level in addition to or instead of granting BLP authority to users. This is done via the GSO BLPPGM record and displayed in SHOW PROGRAMS as a list of program/library names under the heading "TAPE BYPASS LABEL PROGRAMS/LIBRARIES". In addition, the GSO OPTS record BLPLOG/NOBLPLOG field can be used to log all usages of BLP, either by a program authorized in the GSO BLPPGM record or by a user authorized via the TAPE-BLP or TAPE-LBL attribute in their Logonid record. The value active for your installation for the BLP parameter is displayed in SHOW STATE as TAPE BLP=LOG or NOLOG.

Another option related to programs is the GSO LOGPGM record. The names of programs for which the installation wishes to maintain audit trails of their dataset accesses should be specified here. These program names are displayed in SHOW PROGRAMS under the title "LOGGED PROGRAMS". Suggested candidates for this list might be Superzap under its various names.

Although the usage of these programs is still controlled by the access rules and other options, the LOGPGM record does provide a facility to produce audit trails which indicate any datasets accessed by any of these selected programs. This audit trail is edited as the fourth part of the ACFRPTDS report (see the "Reports and Audit Trails" section of this manual and the acf2/MVS Utilities Manual).

The GSO PPGM record allows the installation to specify a list of program names (and their libraries) which can only be executed by a user who has the NON-CNCL Logonid attribute or who is an unrestricted security officer. Programs put on this list are displayed in SHOW PROGRAMS as "RESTRICTED PROGRAM NAMES". Examples of programs which should probably be on this list are IEHDASDR, FDRDSF, etc. which do not use standard open SVCs for dataset processing and thus could bypass other ACF2 controls. If these types of programs are not on this list, review

alternate controls in place for these programs (such as putting them in special libraries with very limited access, etc.). See also the discussion on PPGM under "ACF2 Boundaries".

The GSO MAINT record is used to designate to ACF2 special combinations of users (Logonids), program names, and program library names which are authorized to access and process any dataset without checking the access rules and without creating any ACF2 logging records. The specified combinations are listed in SHOW PROGRAMS under "MAINTENANCE LOGONIDS/PROGRAMS/LIBRARIES". The Logonid(s) included in these entries must have either the NON-CNCL or MAINT attribute. Since no rules will be enforced for these combinations and no logging records/audit trails created, the authorized combinations on this list should be kept to the minimum necessary to allow for reasonable operations of the installation (e.g., standard archiving and disk compression utilities run by a lead operator or shift supervisor).

The GSO LINKLST record defines one or more libraries that ACF2 will consider to be a logical extension to the system LINKLIST. This macro provides the installation with added flexibility in terms of ACF2's program pathing facility. The SHOW LINKLST subcommand displays the GSO LINKLST record specifications. User libraries should not be defined in the LINKLST; a large number of entries would similarly be suspect.

REPORTS AND AUDIT TRAILS

ACF2 provides numerous report generators and sample JCL to create various reports for audit trail purposes. Any attempted violation detected by ACF2 will appear on a report. In addition, each update to any of the ACF2 three control databases (Logonids, Access Rules, or Information Storage datasets) are displayed in standard reports. Thus any addition, change, or deletion of any of ACF2's user or rule control information is visible to a person reviewing these reports. Records produced from these three databases as well as SMF records provide information presented in the various ACF2 reports. (It should be noted that attempts to input concatenated SMF files will result in an ACF2 error message, as only the first file is passed to the report generator.) Last but not least, numerous other reports are available at the option of the installation to record occurrences or produce audit trails of authorized activities. These are of various types and each type can be produced independently of the others. These are explained in the following sections:

DATASET, VOLUME, OR GENERALIZED RESOURCE LOGGINGS

Access and resource rules basically control production of logging reports as well as audit trails of accesses or usage of resources controlled by generalized resource rules. In all cases, a rule authorizing an access or resource usage can be specified in one of two ways: ALLOW (which allows it and creates no report records), or LOG (which considers the access "authorized" and allows it, but creates a logging report record). These permissions in rule records either create logging records (LOG) or do not (ALLOW) regardless of whether or not ACF2 is in LOG, WARN, RULE, or ABORT mode (no records are created in QUIET mode). In other words, an access matching a rule with ALLOW permission would not create a logging record even in LOG mode. Since rules can be written at various levels of detail and can specify various conditions, logging records can be created for almost any set of circumstances desired, and can be readily changed (since rules can be modified dynamically by authorized people via ACF2 TSO commands). For example,

```
$KEY(PROD)
 DATASET.NR1  UID(ABC) R(A) W(L)
 DATASET.NR1  UID(AB)  R(A) W(A)
```

would allow all users whose UID string begin AB to read and write the dataset PROD.DATASET.NR1. However, all updates (writes) by those users who also have 'C' as the third character of their UID string would be logged. Similarly,

```
$KEY(PROD)
 DATASET.NR2  UID(ABC) FOR(30) R(A) W(L)
 DATASET.NR2  UID(AB)  R(A) W(A)
```

would allow all users beginning AB to read and write PROD.DATASET.NR2, but users with ABC would have their updates logged for the first 30 days. After 30 days, the first entry would expire and disappear from the rule and ABC users would continue to be allowed access under the second entry (and the logging of their updates would cease). Another example,

```
$KEY(PROD)
 DATASET.NR3   UID(AB) PGM(PROD01) LIB(LOADLIB) R(A) W(A) A(A)
 DATASET.NR3   UID(AB) R(L) W(L) A(L)
```

would allow any user whose UID string begins with AB to read, write, or allocate PROD.DATASET.NR3 with any program from any library, but if any program other than program name PROD01 from library PROD.LOADLIB is used, ACF2 will log the access.

USER LOGGINGS

Besides being able to log user accesses to specific datasets or resources by writing rules specifying LOG versus ALLOW, ACF2 also provides options to create a logging record of all activities done by a given user regardless of the rules. The attribute TRACE, when turned on in a user's Logonid record, tells ACF2 to create a logging record for all data and resource accesses the user attempts. These will print on the reports as TRACE records.

Under MVS, the Logonid record attribute TSO-TRC will create logging records for each TSO command or CLIST issued by the user while on TSO. These are printed on a special TSO commands report, sorted by user. An ACF2 systems option under the GSO OPTS record CMDREC field, when set to YES, creates TSO command report records for all TSO users (works the same as turning on TSO-TRC for all users). These options and this report (the TSO Command Log) are only available at MVS/TSO sites.

When the MONITOR attribute is turned on in a Logonid record, each time the user enters the system (e.g., TSO logon or a batch job submission) a message will be immediately sent to the security console identifying this activity (this does not create a logging record). This could be used to identify a person in the act of entering the system. ACF2 also automatically creates logging records each time a Logonid with the RESTRICT attribute (no password) enters the system. These records are printed on the Restricted Logonid Job Log report.

All system accesses made by a user with the LOGSHIFT privilege, when outside the dates/times specified via the SHIFT field of the Logonid record, are logged. These loggings are listed on the Invalid Password/Authority Log produced by ACFRPTPW.

COMBINED SMF RECORDS

In acf2/MVS Release 4.0 (and above), a single System Management Facility (SMF) record will be produced for all ACF2 security loggings. The ACF2 report generators will use the record whose number is specified in the ACF2 parameter of the @SMF macro. Refer to the "Field Definition Record" chapter of the acf2/MVS System Programmer's Guide for more information.

If you have SMF tapes created under a previous release of acf2/MVS, please note that the old record parameters are supported. Those parameters are PSWD, DSN, LID, RULE, JTRACE, COMMAND, INFO, and RSRC. SKK has extended support for these parameters so that auditors may generate reports using pre-Release 4.0 records. The report generators are able to internally convert the old records into a the new form. Refer to the "Field Definition Record" chapter of the acf2/MVS System Programmer's Guide for more information.

ACF2 REPORTS

Detailed information on the contents and format of each ACF2 report (including sample outputs) is contained in the acf2/MVS Utilities Manual. Each of these reports is also described briefly here to identify what is generally available.

Dataset Cross Reference Report (ACFRPTXR) -

This report was primarily created for auditors, security administrators, and management to identify which users could access which datasets and generalized resources. Either historical information (by using backup copies of the ACF2 datasets taken at a previous point in time) or current status (by using the online databases) can be obtained. The normal usage of the utility would be to provide a set of dataset names (such as critical system and production datasets or all DSNs from the master catalog) and resource names as input to the report. The report would then display a list of every system user who has any access to each dataset and resource specified. The utility will identify users with access via the access rules, and also users with access because of other ACF2 attributes (such as security officers, matching PREFIX field, or non-cancellable).

Dataset Index Report (ACFRPTIX) -

This is a special report normally produced by request only and potentially very useful to auditors. This report will identify all changes to the access rules affecting any specified high-level index (or pattern) over any period of time (assuming the input SMF records are available). For example, all changes to the ACF2 databases which could have affected how the access rule for high-level index "PAYROLL" could be requested, and previous versions of the PAYROLL rules

would be identified, decompiled, and displayed, plus any changes to Logonid records with PREFIX or security officer privileges which could affect the PAYROLL high-level index would also be identified.

Dataset/Program Event Log (ACFRPTDS) -
This report has four parts and includes all dataset/volume access requests which were created due to: 1) logging options (such as rule entries specifying allow but log, because of logging or warning mode versus abort mode, because of BLP=LOG, or because the access did not match a rule saying allow but was still allowed since it was a security officer or NON-CNCL user), 2) trace requests (Logonids with TRACE set on), 3) access violation attempts, where the access request was denied by ACF2 and recorded as a violation attempt, or 4) accesses allowed but journalled due to some program usage logging request (e.g., LOGPGM record).

Generalized Resource Event Log (ACFRPTRV) -
This report is in three parts similar to the first three mentioned for the 'DS' report above, but is for accesses to resources protected via ACF2's generalized resource rules.

Information Storage Update Log (ACFRPTEL) -
This report displays all changes, additions, or deletions which have occurred to any entry lists, generalized resource rules, or other records in the ACF2 infostorage database. This report also includes detailed information (using a before/after image format) showing all modifications made to Global System Option (GSO) records.

The Environment Report (ACFRPTNV) -
This report generator produces a report that notes the use of the following commands: START (S ACF2), STOP (P ACF2), and MODIFY (F ACF2). The report shows the date and the time that each command was used. In addition, this report includes the date/time of each system IPL and highlights possible losses of SMF data.

Invalid Password/Authority Log (ACFRPTPW) -
This report contains an entry for each attempt to access the system which ACF2 denied for any reason. The reason for denial is also identified. Some common reasons are: invalid password given, Logonid not found or found but cancelled or suspended, invalid input source or submitting program being used, password expired and no new valid password provided, or an invalid OID card. This report also includes any SHIFT violations related to system access attempts, and all LOGSHIFT loggings.

Logonid Access Report (ACFRPTRX) -
This report is a reverse of the Dataset Cross Reference Report produced by ACFRPTXR. This report is ordered by Logonid record and matches users with dataset and resource rules.

Logonid Modification Log (ACFRPTLL) -
This report contains an entry for each occurrence of an update to ACF2's Logonid Records database. This would include any change, insert (add a new record), or delete Logonid activity. Optionally, the report can be produced in before/after image format which allows the auditor to carefully review all Logonid record changes.

Restricted Logonid Job Log (ACFRPTJL) -
This report contains an entry for each time a Logonid with an activated RESTRICT attribute enters the system.

Rule-id Modification Log (ACFRPTRL) -
This report has an entry for each change, addition, or deletion of any access rule record.

Selected Logonids Report (ACFRPTSL) -
This report allows for the flexible selection and display of Logonid information. Because Logonid records for the report can be selected via an IF statement with multiple criteria, almost any combination of users can be readily identified. For example, reports can be produced listing all users who have not changed their passwords in the last 90 days, or who have not accessed the system in the last 30 days, or who have a particular default TSO account number. Thus almost any subsection of users can be quickly identified.

TSO Command Statistics Log (ACFRPTCR) -
This report contains a record for each TSO command or CLIST issued during any TSO sessions by users with TSO-TRC set on in their Logonid records, or for all TSO users if the system option CMDREC is equal to YES. This applies to MVS TSO sites only.

Other Reports -
Reports are also generated by the ACF2 report Pre-Processor (ACFRPTPP) and by the ACF2 recovery program (ACFRECVR). Descriptions of the SMF records ACF2 produces are also provided so that additional report generators can be written locally, if desired.

In all cases, the records included in a given ACF2 report can be affected by 1) the report generator JCL, which has parameter fields allowing the specification of various options and various selection criteria, 2) the actual SMF datasets used as input and 3) the authorities of the user who ran the report (e.g., a security officer or auditor with a UIDSCOPE or SCPLIST limitation running the SL report would only see Logonid records for users who matched the IF statement criteria and who also were within his UIDSCOPE or SCPLIST).

In reviewing reports care must be taken that all proper inputs were included so that SMF records from some time period or for one of the

CPUs are not missing. Also make sure that the selection parameters have not inappropriately excluded important records, such as records from a certain time period or for certain dataset names or Logonids. You must also remember that various system options and the use of exits can affect what data is or is not on the reports. Part of the ACF2 audit should be directed at reviewing the normal processing of the ACF2 reports: if they are produced regularly and include all appropriate records; if (and by whom) they are reviewed regularly; what actions are taken when attempted violations or abuses are identified, etc.

The timely and proper usage of the ACF2 reports is an important aspect of internal controls and should be carefully reviewed. The ACF2 report generators may also be executed at MVS/TSO installations by means of ACF2-supplied SPF screens. Tutorials for these screens (which also include Logonid and rule processing and SHOW commands) are also provided.

CONVERSION TO ACF2

Because your data processing installation probably had very little data protected when it installed ACF2, a sudden conversion to total data protection could be disruptive to the continued smooth functioning of your company. Although each installation may accomplish this conversion differently, the way it is usually done is as follows:

1. Put the ACF2 system into LOG mode. This means that ACF2 will make all the decisions it is supposed to make concerning dataset access, except that it will not really deny any of them. The ACF2 reports will show the accesses that it would have denied and the Installation Security Officer (ISO) will make up access rules which will appropriately reduce the number of "violations". Either during this process or shortly before or afterwards, the ISO should consult with the data owner to determine whether the accesses are legitimate (i.e., should be allowed).

2. Put the ACF2 system into WARN mode. This means that for every dataset access that ACF2 would have prevented, a logging report record is created plus an ACF2 message is displayed on the terminal or printed in the job log for batch. Along with the ACF2 message, an installation-supplied message indicating the date on which the access will no longer be allowed is also displayed.

3. Finally, put the ACF2 system into ABORT mode. This is its normal mode of operation, and accesses considered invalid by the ACF2 rules will be denied.

RULE mode can be used as an aid in the transition to full security by allowing the phasing-in of protection at the rule set level. If access would ordinarily be prevented by an existing rule, RULE mode can provide overrides. Contact your security officer for further details on conversion plans and status at your site.

APPENDIX A - ACF COMMAND IN BATCH

At MVS sites, use of the TSO Terminal Monitor Program (TMP) in batch requires the creation of JCL (Job Control Language) similar to the following example:

```
//TMP EXEC PGM=IKJEFT01,DYNAMNBR=25
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
ACF
any ACF subcommands
END
/*
```

The ACF2 utility ACFBATCH may be used to execute the ACF command in the batch environment. A sample of the required JCL is as follows:

```
//ACFJOB    EXEC  PGM=ACFBATCH
//SYSPRINT  DD    SYSOUT=A
//SYSHELP   DD    DSN=SYS1.HELP,DISP=SHR
//SYSIN     DD    *
any ACF subcommands
/*
```

APPENDIX B - SAMPLE SHOW OUTPUTS

SAMPLE "SHOW ACTIVE" OUTPUT:

```
show active
-- ACF2 INTERCEPTS THAT HAVE RECEIVED CONTROL --
DASD-OPEN(YES)          DASD-EOV(NO)            VSAM-OPEN(YES)
TAPE-OPEN(YES)          TAPE-EOV(NO)            CATALOG(YES)
DASD-ALOC(YES)          DASD-RENAME(YES)        DASD-SCRATCH(NO)
USER CALL(NO)           EXTERNAL CALL(NO)       PROGRAM CALL(YES)
JOB INIT(YES)           JOB/STEP TERM(YES)      TSO-MVS(YES)
CAT-CVOL(NO)            READER-VS1(NO)          INTERP-VS1(NO)

-- LOCAL EXITS SPECIFIED ON THIS SYSTEM --
DSN PRE-VALIDATE=ABCVALD(INACTIVE)   DSN POST-VALIDATE=POSTVLD(INACTIVE)
DSN VIOLATION=NONE                   PSEUDO DSN GENERATE=NONE
RSRC PRE-VALIDATE=NONE               RSRC POST-VALIDATE=NONE
STC VALIDATE=NONE                    SOURCE MODIFICATION=NONE
LOGON PRE-VALIDATE=NONE              LOGON POST-VALIDATE=NONE
EXPIRATION=NONE                      NEW PASSWORD=NONE
RULE DB PRE-PROCESS=NONE             RULE DB PST-PROCESS=NONE
INFO DB PRE-PROCESS=NONE             INFO DB PST-PROCESS=NONE
SVC INITIALIZATION=NONE              TSO LOGIN TERM TYPE=NONE
TSO LOGON PARM =NONE
```

SAMPLE "SHOW FIELDS" OUTPUT

```
acf
show fields
-- IDENTIFICATION --
 LID          *NAME        *PASSWORD     *PHONE        UID
-- CANCEL/SUSPEND --
*CANCEL       CSDATE       CSWHO         *MONITOR      *MON-LOG      *PSWD-EXP
*SUSPEND      *TRACE       *TSO-TRC
-- PRIVILEGES --
 ACCOUNT      AUDIT        AUTODUMP      CICS          *CONSULT      DSNSCOPE
 DUMPAUTH     EXPIRE       *IDMS         *IMS          *JOB          JOBFROM
*LEADER       LIDSCOPE     LOGSHIFT      MAINT         MUSASS        NO-SMC
 NO-STORE     NON-CNCL     *PGM          *PROGRAM      READALL       REFRESH
*RESTRICT     RULEVLD      SCPLIST       SECURITY      STC           *SUBAUTH
 TAPE-BLP     TAPE-LBL     *TSO          UIDSCOPE      *USER
-- ACCESS --
 ACC-CNT      ACC-DATE     ACC-SRCE      ACC-TIME
-- MISCELLANEOUS --
*CICSCL       *CICSID      *CICSKEY      *CICSKEYX     *CICSPRI      *CICSRSL
*IDLE         *MAXDAYS     *MINDAYS      *MUSOPT       *MUSPGM
PREFIX        *SHIFT       *SOURCE       *ZONE
-- TSO --
*ACCTPRIV          *ALLCMDS            *ATTR2              *CHAR
*CMD-LONG          *DFT-DEST           *DFT-PFX            *DFT-SOUT
*DFT-SUBC          *DFT-SUBH           *DFT-SUBM           *INTERCOM
*JCL               *LGN-ACCT           *LGN-MSG            *LGN-PERF
*LGN-PROC          *LGN-RCVR           *LGN-SIZE           *LGN-TIME
*LGN-UNIT          *LINE               *MAIL               *MODE
*MOUNT             *MSGID              *NOTICES            *OID
*OID-ALL           *OPERATOR           *PAUSE              *PMT-ACCT
*PMT-PROC          *PROMPT             *RECOVER            *TSOACCT
*TSOCMDS           *TSOFSCRN           *TSOPERF            *TSOPROC
*TSORBA            *TSORGN             *TSOSIZE            *TSOTIME
*TSOUNIT           *VLD-ACCT           *VLD-PROC           *WTP
-- STATISTICS --
*PSWD-DAT           PSWD-TOD            *PSWD-VIO           *SEC-VIO
 UPD-TOD
```

SAMPLE "SHOW LINKLST" OUTPUT:

```
acf
show linklst
-- DATASETS INCLUDED IN THE LINKLIST --
SYS1.IMS.LOAD
```

SAMPLE "SHOW PROGRAMS" OUTPUT:

```
acf
show programs
-- RESTRICTED PROGRAM NAMES --
IEHD**** DRWD**** ICKDSF**

-- MAINTENANCE LOGONIDS/PROGRAMS/LIBRARIES --
```

```
MAINTLID $ARCHIVE SYS1.LINKLIB
MAINTLID $ASMBMON SYS1.LINKLIB
MAINTLID $BACKUP  SYS1.LINKLIB
MAINTLID $DASDMNT SYS1.LINKLIB
MAINTLIB $DEFRAG  SYS1.LINKLIB
MAINTLIB $MIGRATE SYS1.LINKLIB

-- TAPE BYPASS LABEL PROGRAMS/LIBRARIES --
COPYEDIT SYS1.LINKLIB

-- LOGGED PROGRAMS --
AMASPZAP
IMASPZAP
INCORZAP
```

SAMPLE "SHOW STATE" OUTPUT

```
show state
RUNNING ACF2 VERSION 4.0 WITH MODE = ABORT
USING ACFFDR ASSEMBLY:  14.14  05/30/85

OPTIONS IN EFFECT:
TAPE BLP=LOG              CONTROL=DECENTRALIZED    %CHANGE=ALLOWED
CPUTIME=LOCAL             DATE FORMAT=MM/DD/YY     STC DFLT LID=ACFSTCID
DEFAULT LID=ACFDFLT       JOB CHECK=NO             MAX VIO PER JOB=10
STC OPTION=ON             TAPE DSN=NO              UADS=BYPASS
NOSORT=NO

PASSWORD OPTIONS IN EFFECT:
LOGON RETRY COUNT=3       MIN PSWD LENGTH=5        MAX PSWD ATTEMPTS=5

-- TSO --
*ACCTPRIV    *ALLCMDS     *ATTR2       *CHAR        *CMD-LONG    *DFT-DEST
*DFT-PFX     *DFT-SOUT    *DFT-SUBC    *DFT-SUBH    *DFT-SUBM    *INTERCOM
*JCL         *LGN-ACCT    *LGN-INDX    *LGN-MSG     *LGN-PERF    *LGN-PROC
*LGN-RCVR    *LGN-SIZE    *LGN-TIME    *LGN-UNIT    *LINE        *MAIL
*MODE        *OID         *OID-ALL     *OPERATOR    *PAUSE       *PMT-ACCT
*PMT-PROC    *PROMPT      *RECOVER     *TSOACCT     *TSOCMDS     *TSOFSCRN
*TSOPERF     *TSOPROC     *TSORBA      *TSORGN      *TSOSIZE     *TSOTIME
*TSOUNIT     *UADSINDX    *VSD-ACCT    *VLDPROG     *WTP

-- STATISTICS --
*PSWD-DAT    PSWD-TOD     *PSWD-VIO    *SEC-VIO     *SEC-VIO     UPD-TOD
```

SAMPLE "SHOW SYSTEM" OUTPUT

```
acf
show system
-- SYSTEM PARAMETERS IN EFFECT --

SVCS:
ALTER SVC=222             VALIDATE SVC=221
```

```
SMP RECORD NUMBERS:
PASSWORD=220            DATASET VIO=221         LID JOURNAL=222
RULE JOURNAL=223        LID TRACE=224           TSO COMMAND=225
INFO JOURNAL=226        RESORCE VIO=227         ACF2 COMMOM=230

BACKUP:
AUTO BACKUP TIME=03.30  CPU-ID=SKK1

COMMAND STRING=S BKUPCOPY

NJE OPTIONS IN EFFECT:
VALIDATE OUT=NO         VALIDATE IN=YES         INHERIT=YES

OTHER:
CONSOLE MSGS=ROLL       SHR-DASD=SUPPORTED      SMF LOGONID STAMP=NO
JES2-XBM=NO VALIDATE    LOGONID LENGTH=1024     LAB NUMBER=5
LABEXP= 01:05:00        NOTIFY=YES              CURRENT SYSID=ABC1
STARTUP SYSID=ABC1      BUILT ACCVT=ABC1
```

SAMPLE "SHOW TSO" OUTPUT:

```
acf
show tso
-- TSO RELATED DEFAULTS ACTIVE --
LOGON ACCOUNT STRING=1
CMD LIST BYPASS CHAR=#  CHAR DELETE CHAR=NONE   TSO CMD LIST=NONE
COMMAND SMF RECORDS=NO  LINE DELETE CHAR=NONE   LOGON CHECK=NO
PERFORMANCE GROUP=NONE  TSO LOGON PROC=IKJACCNT QUICK LOGON=YES
TSO REGIONSIZE=1024     SUBMIT CLASS=NONE       SUBMIT HOLD CLASS=NONE
SUBMIT MSGCLASS=NONE    SESSION TIME=0          SYSOUT CLASS=A
TSO UNITNAME=SYSDA      LOGON WAIT TIME=60      FSRETAIN=YES
```

Note: The actual fields listed and how they are grouped is dependent upon who made the request (the privileges such as SECURITY, AUDIT, etc. that the user has), and how each field has been defined at that installation (the ALTER, AUTH, LIST, and GROUP parameters for each field in its ACFFDR @CFDE field description macro). Fields added locally will also be displayed, as appropriate.

Additional SHOW subcommands are described and sample output provided in the acf2/MVS Administrator's Guide. Note that all SHOW subcommands may be issued in batch or through ACF2-supplied SPF screens.

APPENDIX C - LOGONID RECORD FIELDS

The Logonid record fields listed below come predefined with ACF2. They are arranged by the group number assigned to each in the default copy of the ACFFDR that comes with the system. ACF2 attributes required to display and/or alter each field are indicated; these attributes are defined in the default ACFFDR. They may not necessarily be the ones currently active for your installation. The latest ACFFDR assembly values must be checked to determine if there are any changes, and also to identify any additional fields which are defined (added to ACF2) locally. Deviations from these ACF2 defaults should be examined to see if the change is appropriate. Each valid Logonid record field will have a corresponding @CFDE macro entry in the ACFFDR.

Key to attributes required, as noted on the following lists:

S	Security Officer (any)
S*	Unrestricted Security Officer Only (@CFDE FLAGS=RESTRICT)
Ac	Account Manager
Au	Auditor
L	Leader
C	Consultant
U	Normal User

FIELD NAME	CONTENT	ALTER S	Ac	Au	L	C	U	DISPLAY S	Ac	Au	L	C	U
(0) IDENTIFICATION SECTION													
LID	Logonid		(Ac)					S	Ac	Au	L	C	U
UID	User Id String (pseudo field)		See individual fields					S	Ac	Au	L	C	U
NAME	User's Name	S	Ac					S	Ac	Au	L	C	U
PASSWORD	Password	S	Ac				U			NONE			
PHONE	Phone Number	S	Ac		L			S	Ac	Au	L	C	U
(1) CANCEL/SUSPEND SECTION													
CANCEL	Cancel Status	S	Ac					S	Ac	Au	L	C	U
CSDATE	Cancel/Suspend Date							S	Ac	Au			
CSWHO	Who Cancelled/ Suspended							S	Ac	Au			
MONITOR	Monitor Status	S						S		Au			
MON-LOG	Log System Access	S						S		Au			
PSWD-EXP	Password is Expired	S	Ac					S	Ac	Au	L	C	U
SUSPEND	Suspend Status	S	Ac		L			S	Ac	Au	L	C	U
TRACE	Trace Status	S						S		Au			
TSO-TRC	TSO Trace Status	S						S		Au			

FIELD NAME	CONTENT	ALTER S	Ac	Au	L	C	U	DISPLAY S	Ac	Au	L	C	U
(2) PRIVILEGES SECTION													
ACCOUNT	Account Mgr. Indicator	S*						S	Ac	Au			
AUDIT	Auditor Indicator	S*						S	Ac	Au			
AUTODUMP	ACF2 SVC Dump	S*						S	Ac	Au	L	C	U
CICS	Auth to use CICS	S	Ac					S	Ac	Au	L	C	U
CONSULT	Consultant Indicator	S	Ac					S	Ac	Au		C	
DSNSCOPE	DSN Scope for Security/Account	S*						S	Ac	Au	L	C	
DUMPAUTH	User Dump Authorization	S*						S	Ac	Au	L	C	U
EXPIRE	Date LID Suspends	S	Ac					S	Ac	Au	L	C	U
IDMS	Auth. to use IDMS	S						S	Ac	Au	L	C	U
IMS	Auth. to use IMS	S	Ac					S	Ac	Au	L	C	U
JOB	Auth. to enter batch jobs	S	Ac					S	Ac	Au	L	C	U
JOBFROM	Auth. to use JOBFROM	S*						S	Ac	Au	L	C	U
LEADER	Leader indicator	S	Ac					S	Ac	Au	L	C	
LIDSCOPE	LID Scope for S/Ac/Au/L/C	S*						S	Ac	Au	L	C	
LOGSHIFT	Auth outside SHIFT	S*						S	Ac	Au	L	C	U
MAINT	Authority to Access Maintenance Programs	S						S	Ac	Au	L	C	U
MUSASS	Multi. User Single Addr. Space System	S*						S	Ac	Au	L	C	U
NO-SMC	Auth to bypass SMC	S*						S	Ac	Au	L	C	U

FIELD NAME	CONTENT	ALTER S	Ac	Au	L	C	U	DISPLAY S	Ac	Au	L	C	U
NO-STORE	Cannot store/ change rules	S*						S	Ac	Au	L	C	U
NON-CNCL	ACF2 not to cancel jobs	S*						S	Ac	Au	L	C	U
PROGRAM	APF-Auth program for submits	S	Ac					S	Ac	Au	L	C	U
READALL	READ access auth.	S*						S	Ac	Au	L	C	U
REFRESH	Auth. to use REFRESH command	S						S	Ac	Au			
RESTRICT	Cannot use passwords	S	Ac					S	Ac	Au	L	C	U
RULEVLD	Must use rule	S*						S	Ac	Au	L	C	U
SCPLIST	Scope list name	S*						S	Ac	Au	L	C	
SECURITY	Security Officer Indicator	S*						S	Ac	Au			
STC	Logonid for STCs only	S*						S	Ac	Au	L	C	U
SUBAUTH	Must be submitted via APF Auth Program	S	Ac					S	Ac	AU	L	C	U
TAPE-BLP	May use BLP	S*						S	Ac	Au	L	C	U
TAPE-LBL	May use limited BLP	S*						S	Ac	Au	L	C	U
TSO	Auth to use TSO	S	Ac					S	Ac	Au	L	C	U
UIDSCOPE	UID Scope for S/Ac/Au/L/C	S*						S	Ac	Au	L	C	
USER	System User	S	Ac					None					

FIELD NAME	CONTENT	ALTER S	Ac	Au	L	C	U	DISPLAY S	Ac	Au	L	C	U
(3) ACCESS SECTION													
ACC-CNT	Count of system accesses							S	Ac	Au	L	C	U
ACC-DATE	Date of last access							S	Ac	Au	L	C	U
ACC-SRCE	Last access source							S	Ac	Au	L	C	U
ACC-TIME	Time of last access							S	Ac	Au	L	C	U
(4) MISCELLANEOUS SECTION													
CICSCL	CICS Oper. Class	S	Ac					S	Ac	Au	L	C	U
CICSID	CICS Oper Id	S	Ac					S	Ac	Au	L	C	U
CICSKEY	CICS Key	S	Ac					S	Ac	Au	L	C	U
CICSKEYX	CICS Key Ext.	S	Ac					S	Ac	Au	L	C	U
CICSPRI	CICS Oper. Pri.	S	Ac					S	Ac	Au	L	C	U
CICSRSL	CICS Resource Key	S	Ac					S	Ac	Au	L	C	U
IDLE	Max. Idle Minutes (IMS/CICS)	S	Ac					S	Ac	Au	L	C	U
MAXDAYS	Max. days before password change required	S	Ac					S	Ac	Au	L	C	U
MINDAYS	Min. days before password change allowed	S	Ac					S	Ac	Au	L	C	U
PREFIX	Owned dataset prefix	S*						S	Ac	Au	L	C	U
SHIFT	Shift record name	S	Ac					S	Ac	Au	L	C	U
SOURCE	Authorized input source of system access	S	Ac					S	Ac	Au	L	C	U

FIELD NAME	CONTENT	ALTER S	Ac	Au	L	C	U	DISPLAY S	Ac	Au	L	C	U
TSOCMDS	Module name for Command list	S	Ac					S	Ac	Au	L	C	U
UADSINDX	INDEX name in UADS tree structure	S	Ac					S	Ac	Au	L	C	U
ZONE	Time zone name	S	Ac					S	Ac	Au	L	C	U
(5) TSO SECTION													
ACCTPRIV	TSO Account Privileges	S						S	Ac	Au	L	C	U
ALLCMDS	Ability to bypass command limiting	S	Ac					S	Ac	Au	L	C	U
ATTR2	PCF control field	S	Ac					S	Ac	Au	L	C	U
CHAR	TSO delete character	S	Ac		L	C	U	S	Ac	Au	L	C	U
CMD-LONG	Command name aliases not accepted	S	Ac					S	Ac	Au	L	C	U
DFT-DEST	Default remote for output	S	Ac					S	Ac	Au	L	C	U
DFT-PFX	Default TSO profile prefix	S	Ac		L	C	U	S	Ac	Au	L	C	U
DFT-SOUT	Default TSO SYSOUT Class	S	Ac		L	C	U	S	Ac	Au	L	C	U
DFT-SUBC	Default TSO submit class	S	Ac		L	C	U	S	Ac	Au	L	C	U
DFT-SUBH	Default TSO Submit Hold Class	S	Ac		L	C	U	S	Ac	Au	L	C	U

FIELD NAME	CONTENT	ALTER						DISPLAY					
		S	Ac	Au	L	C	U	S	Ac	Au	L	C	U
DFT-SUBM	Default TSO Submit Msg Class	S	Ac		L	C	U	S	Ac	Au	L	C	U
INTERCOM	Accepts msgs from others	S	Ac		L	C	U	S	Ac	Au	L	C	U
JCL	Can submit jobs from TSO	S	Ac					S	Ac	Au	L	C	U
LGN-ACCT	May specify acct at Logon	S	Ac					S	Ac	Au	L	C	U
LGN-INDX	May specify indx parameter	S	Ac					S	Ac	Au	L	C	U
LGN-MSG	May specify msg class	S	Ac					S	Ac	Au	L	C	U
LGN-PERF	May specify perform group	S	Ac					S	Ac	Au	L	C	U
LGN-PROC	May specify TSO procedure	S	Ac					S	Ac	Au	L	C	U
LGN-RCVR	May use TSO rerecover option	S	Ac					S	Ac	Au	L	C	U
LGN-SIZE	May specify any region size	S	Ac					S	Ac	Au	L	C	U
LGN-TIME	May specify time limit	S	Ac					S	Ac	Au	L	C	U
LGN-UNIT	May specify unit name	S	Ac					S	Ac	Au	L	C	U
LINE	TSO line delete character	S	Ac		L	C	U	S	Ac	Au	L	C	U
MAIL	Accepts mail at Logon	S	Ac		L	C	U	S	Ac	Au	L	C	U

FIELD NAME	CONTENT	ALTER S	Ac	Au	L	C	U	DISPLAY S	Ac	Au	L	C	U
MODE	Receives TSO modal msgs	S	Ac		L	C	U	S	Ac	Au	L	C	U
MOUNT	May issue mounts	S	Ac					S	Ac	Au	L	C	U
MSGID	Prefix on TSO msgs(id)	S	Ac		L	C	U	S	Ac	Au	L	C	U
NOTICES	Receives TSO notices at Logon	S	Ac		L	C	U	S	Ac	Au	L	C	U
OID	Must insert card	S	Ac					S	Ac	Au	L	C	U
OPERATOR	TSO Operator Privileges	S						S	Ac	Au	L	C	U
PAUSE	Pause during CLIST errors	S	Ac		L	C	U	S	Ac	Au	L	C	U
PMT-ACCT	Forced to specify account at logon	S	Ac					S	Ac	Au	L	C	U
PMT-PROC	Forced to specify Procedure Name	S	Ac					S	Ac	Au	L	C	U
PROMPT	To be prompted for missing parameters	S	Ac		L	C	U	S	Ac	Au	L	C	U
RECOVER	Uses Recover option (TSO Cmd Package)	S	Ac		L	C	U	S	Ac	Au	L	C	U
TSOACCT	Default Logon Account Number	S	Ac					S	Ac	Au	L	C	U
TSOFSCRN	Will get full-screen logon	S	Ac					S	Ac	Au	L	C	U
TSOPERF	Default Logon Performance Grp.	S	Ac					S	Ac	Au	L	C	U
TSOPROC	Default Logon Procedure Name	S	Ac					S	Ac	Au	L	C	U

FIELD NAME	CONTENT	S	Ac	Au	L	C	U	S	Ac	Au	L	C	U
TSORBA	Mail Index Record Pointer	S						S					
TSORGN	Default Logon Region Size	S	Ac					S	Ac	Au	L	C	U
TSOSIZE	Maximum TSO Region Size	S	Ac					S	Ac	Au	L	C	U
TSOTIME	Default TSO Time Parameter	S	Ac					S	Ac	Au	L	C	U
TSOUNIT	Default TSO Unit Name	S	Ac					S	Ac	Au	L	C	U
VLD-ACCT	Account is to be validated	S	Ac					S	Ac	Au	L	C	U
VLD-PROC	Proc Name is to be validated	S	Ac					S	Ac	Au	L	C	U
WTP	WTP Msgs are displayed	S	Ac		L	C	U	S	Ac	Au	L	C	U

(6) STATISTICS SECTION

PSWD-DAT	Date of last invalid pswd	S						S	Ac	Au	L	C	U
PSWD-TOD	Date/time last Pswd change							S	Ac	Au	L	C	U
PSWD-VIO	Nr. of pswd violations (1 day)	S						S	Ac	Au	L	C	U
SEC-VIO	Nr. of security violations	S						S	Ac	Au	L	C	U
UPD-TOD	Date/time last update (this record)							S	Ac	Au	L	C	U

APPENDIX D - RULE WRITING EXAMPLES

DATASET ACCESS RULE WRITING EXAMPLE:

A common mistake a number of sites have made when writing access rules is to be too general in writing a rule. Take the following example of a rule set for the SYS1 datasets:

```
(1)  $KEY(SYS1)
(2)   UADS UID(SYSPRG1) R(A) W(L) A(L)
(3)   PARMLIB UID(SYSPRG1) R(A) W(L) A(L)
(4)   PARMLIB UID(SYSPRG*) R(A)
(5)   MAN*    UID(SYSPRG1) R(A) W(L) A(L)
(6)   MAN*    UID(SYSPRG*) R(A)
(7)   -      R(A) E(A)
```

The site was attempting to 1) allow only the lead system programmer ("SYSPRG1") to read/write/allocate the sensitive datasets SYS1.UADS, SYS1.PARMLIB, and SYS1.MANX/MANY; 2) allow only the other system programmers ("SYSPRG*") to read SYS1.PARMLIB and SYS1.MANX/MANY; and 3) allow all other users to read or execute programs out of all the other SYS1 datasets.

However, this rule, as written, does not provide the level of control desired. This is because line (7) in the rule set says that all access requests not specifically matching an "environment" described by any prior rule would have the authorizations associated with line (7) apply - i.e., read and execute allowed.

An "environment" is a description of access conditions as defined by the rule elements DSN, VOL, UID, SOURCE, LIB, PGM, DDN, and UNTIL/FOR. ACF2 always checks the conditions or "environment" of the actual access attempt against all elements specified in a given rule, until it finds a set of conditions that completely match. At that point the authorization cited in the matching rule entry is the one ACF2 will enforce (e.g., read/write/allocate/execute only, allow/allow-but-log/prevent). Each access request is ultimately governed by only one rule, the first one which matches the "environment".

With the example rule set above, any user would be allowed to read SYS1.UADS! This is because accesses by user "SYSPRG1" would match the environment created by the rule in line (2) and be specifically allowed. Meanwhile accesses by other users would skip over the rule in (2) because the UID string would not match the UID pattern "SYSPRG1" specified. They would also skip over the entries in (3) - (6), because the DSN would not match. They would ultimately match and be governed by the entry on line (7), which would allow any UID to read any SYS1 dataset.

If line (7) had not been added to this rule set, then only the users (system programmers) specified in the entries line (2) - (6) would have access authority to any SYS1 datasets. Remember, ACF2's default is that any access not specifically allowed by a rule is prevented. But overly liberal use of the "dash/all-users-allowed" general rule entry can inappropriately negate ACF2's default protection, such as in this example.

The addition of just three more rule entries to the above example would restrict users other than those authorized in lines (2) - (6) from accessing the cited special datasets. The three additional lines would be:

(2a) UADS
(4a) PARMLIB
(6a) MAN*

Note: Since ACF2's defaults are R(P), W(P), A(P), E(P), they do not have to be explicitly entered on these lines. Also the actual placement of these rule entries in the existing rule record is not important, as ACF2 will resequence all entries as necessary for its checking. The new lines (2a), (4a), and (6a) would prevent users not specifically authorized in lines (2) - (6) from accessing SYS1.UADS, SYS1.PARMLIB, or SYS1.MAN* datasets. Line (7) would then allow all users read and execute access only to other SYS1 datasets.

Obviously, extreme care must be taken when adding a broad, generalized rule entry such as the one in line (7). One way to avoid trouble would be for the rule writer to always enter the corresponding "negative" rule at the same time as the "positive" rule whenever he is trying to limit access to datasets in a rule record with a general access entry like line (7) present (or at the time he is adding the general access entry if there was not one earlier).

For example, as a rule entry line like (2) is added, also add one line like (2a). Of course, as long as you do not give blanket authorizations by writing rule entries like line (7), these additional "negative" rule entries will not be necessary (again, ACF2 "prevent" defaults would remain active). When reviewing existing rules you want to be particularly watchful for the inappropriate usage of overly generalized rules.

GENERALIZED RESOURCE RULE WRITING EXAMPLE:

Generalized resource rules are similar to access rules but do not have as many variables. Also, since there are no "levels" of permission values (i.e., READ, WRITE, ALLOC, EXEC), only the specific choices of ALLOW, LOG, or PREVENT are necessary. The %CHANGE and $USERDATA fields are still available at the rule set level, and UID, UNTIL/FOR, SOURCE, and DATA are still available at the individual rule entry level. When the keyword VERIFY is used in a resource rule, the user must reverify his identity by entering the correct password before ACF2 will permit the transaction to proceed. One other important difference in generalized resource rules is that masks (resource name patterns) can be used in the $KEY field as long as the related TYPE directory is also built.

The following example shows a number of resource rule sets for type code ITR (IMS Transactions). Since masks are used in the $KEY field, a directory would have to be built by ACF2 for type code ITR rules. For the example, the following IMS transaction naming conventions are assumed:

```
APIxxx Accounts Payable Inquiry Transactions
APUxxx Accounts Payable Update Transactions
ARIxxx Accounts Receivable Inquiry Transactions
ARUxxx Accounts Receivable Update Transactions
ARU123 A particular Accounts Receivable update transaction
AxIxxx Other Accounting department inquiries
```

The following UID String naming conventions are assumed:

```
APSxxxxx Accounts Payable Supervisors
APCxxxxx Accounts Payable Clerks
ARSxxxxx Accounts Receivable Supervisors
ARCxxxxx Accounts Receivable Clerks
ACMSTEVE Accounting Department Manager
DBAKAREN Database Administrator
SHPxxxxx Materials Shipping Clerks
```

And the following SOURCE group names are assumed:

```
ACCTSPAY Accounts Payable Terminal Room
ACCTSRCV Accounts Receivable Terminal Room
ACTGDEPT Combined Group (ACCTSPAY + ACCTSRCV)
WAREHSE  Warehouse/Shipping Dock
```

A possible collection of ITR rule sets for this installation might be:

```
(1)     $KEY(API***) TYPE(ITR)
        *ALLOW DBA OR ACCTG DEPT MGR TO CHANGE RULE
        %CHANGE ACMSTEVE DBAKAREN
        *ALLOW AP SUPR TO DO AP INQUIRIES ANYWHERE
   (a)   UID(APS) ALLOW
        *ALLOW AP PERSONNEL TO DO AP INQUIRIES
```

```
          *FROM AP TERMINALS
     (b)   UID(AP) SOURCE(ACCTSPAY) ALLOW

(2)       $KEY(APU***) TYPE(ITR)
          *ALLOW ANY ACCTG DEPT MGR TO CHANGE RULE
          %CHANGE ACM*****
          *ALLOW AP SUPR TO DO UPDATES ANYWHERE IN
          *ACCTG DEPT, BUT LOG IF OUTSIDE OF ACCTS PAY
          *AND VERIFY PASSWORD AND LOG IF OUTSIDE OF ACCTG DEPT
     (a)   UID(APS) SOURCE(ACCTSPAY) ALLOW
     (b)   UID(APS) SOURCE(ACTGDEPT) LOG
     (c)   UID(APS) VERIFY LOG
          *ALLOW AP PERSONNEL TO DO UPDATES
          *FROM AP TERMINALS
     (d)   UID(AP) SOURCE(ACCTSPAY) ALLOW

(3)       $KEY(ARI***) TYPE(ITR)
          *ALLOW AR SUPR TO DO INQUIRIES ANYWHERE
     (a)   UID(ARS) ALLOW
          *ALLOW AR PERSONNEL TO DO AR INQUIRIES
          *FROM AR TERMINALS
     (b)   UID(AR) SOURCE(ACCTSRCV) ALLOW
          *ALLOW SHIPPING CLERKS TO DO AR INQUIRIES
          *FROM WAREHOUSE, BUT LOG
     (c)   UID(SHP) SOURCE(WAREHSE) LOG

(4)       $KEY(ARU123) TYPE(ITR)
          *ALLOW AR CLERKS TO DO FROM AR
          *TERMINALS, BUT REVERIFY PASSWORD
     (a)   UID(ARC) SOURCE(ACCTRCV) VERIFY ALLOW
          *ALLOW AR SUPR TO DO FROM ACCTS RECV TERMINALS
     (b)   UID(ARS) SOURCE(ACCTSRCV) ALLOW
          *ALLOW AR SUPR TO DO FROM
          *OTHER ACCTG DEPT TERMINALS, BUT REVERIFY
          *PASSWORD AND LOG
     (c)   UID(ARS) SOURCE(ACTGDEPT) VERIFY LOG

(5)       $KEY(ARU***) TYPE(ITR)
          *ALLOW AR PERSONNEL TO DO FROM AR TERMINALS,
          *BUT LOG THOSE DONE BY NEW CLERK BETTY
          *FOR FIRST 60 DAYS
     (a)   UID(ARCBETTY) FOR(60) SOURCE(ACCTSRCV) LOG
     (b)   UID(ARC) SOURCE(ACCTSRCV) ALLOW
          *ALLOW AR SUPR TO DO FROM ANY ACCTG TERM
     (c)   UID(ARS) SOURCE(ACTGDEPT) ALLOW

(6)       $KEY(A*I***) TYPE(ITR)
          *ALLOW ACCTG DEPT MGR AND SUPRS TO DO
          *FROM ACCTG DEPT TERMINALS
     (a)   UID(ACM) SOURCE(ACTGDEPT) ALLOW
     (b)   UID(A*S) SOURCE(ACTGDEPT) ALLOW
```

In the examples on the previous page, the rule sets and the individual rule entries within each set are ordered in the sequence ACF2 would order them. However, the comment cards would not be carried through a compilation nor stored on the ACF2 databases. It is important when writing rules to ensure that the checking sequence desired is the one that will be used by ACF2. This can be done by examining output from the decompiler (DECOMP).

When patterns are used in the $KEY field, rule sets are ordered by a direct alphanumeric sort on the $KEY values, with an asterisk (*) in any position placed last. For the 6 rule sets above, since they all have "A" their first character, the APxxxx rules are first (API before APU), the AR rules are next (ARI*** then ARU123 before ARU***), and A*I*** is last.

Within a resource rule set, the rule entries are sorted by UID first, then SOURCE, then SHIFT, then date (UNTIL/FOR, with the earliest expiration date first).

UID patterns are sorted the same as $KEY patterns, so entry (a) in rule (1) with the more specific UID(APS) sorts before entry (b) with its more general UID(AP). Note that UID(AP) equates to UID(AP******), as ACF2 automatically pads out the rest of any rule UID value with asterisks.

In rule set (2), entries (a), (b), and (c) have identical UID values, so these lines are sorted by the SOURCE field. No SOURCE is always more general than any specific SOURCE value, so entry (c) is last. The SOURCE field cannot be a pattern or mask, so is sorted strictly alphanumerically. Thus entry (a) with SOURCE(ACCTSPAY) sorts before entry (b) with SOURCE(ACTGDEPT).

Note that this means special care must be taken when assigning input source names and source group names so that the more specific groupings have "earlier" names than the more general or combined group names. For example, if the larger combined group for the accounting department was named ACCTDPET instead of ACTGDEPT, it would always sort before ACCTSPAY and ACCTSRCV, which are subsets of the larger group.

In rules where both levels of source names are to be used (like in rule set (2) above), the wrong results could occur; if entry (b) had SOURCE(ACCTDEPT), it would sort ahead of entry (a). In that case any use of resource APU*** which should match the (a) entry would first match the (b) entry (allowed but logged) and would never check entry (a). (ACF2 always applies the permission of the first rule entry that matches the environment of the request and stops checking there.)

Shift-names specified in the SHIFT field are sorted the same way as source names in the SOURCE field (alphanumerically) and cannot be a mask.

GENERAL RULE WRITING COMMENTS:

The previous comments on the sorting on UID string, SOURCE, and date fields apply to access rules as well as to generalized resource rules. The full sequence used in access rules is DSN value first, then VOL, UID, LIB, PGM, DDN, SOURCE, SHIFT, and date (UNTIL/FOR), in that order. Output from the rule decompiler will be listed in ACF2's checking sequence unless $NOSORT has been specified within the rule set.

Whenever you are in doubt as to how a specific rule may be interpreted by ACF2, you can use the ACF2 "TEST" subcommand to test different possibilities. This can be done even before a rule change is stored and actually made effective, or to test an existing rule.

APPENDIX E - SAMPLE ACF2 AUDIT SURVEY QUESTIONS

The following list of survey questions is provided here as a sample of items that an internal EDP auditor might examine during the ACF2 portion of his audit. It is not intended to be a complete list nor to represent the correct approach for any given installation. This list is provided here to show examples of how a hypothetical ACF2 site's policies might be translated into audit items. It is assumed here for illustration that this is an MVS site with TSO available. If items similar to these were included in a site's audit plan, the auditor would review each item and conclude whether that activity was satisfactory, unsatisfactory, or in progress, or that that item did not apply at that time. Other tests and audit work papers would also be completed, reviewed, and signed off.

We would like to take this opportunity to thank the General Motors Corporation and their corporate audit staff for their assistance and comments during the preparation of this document.

Sample Audit Items

Before you begin your survey: (a) Obtain a TSO terminal and a Logonid with the AUDIT privilege. (b) Determine the names of the ACF2 system files and obtain read-access to the "SYS1" prefix files. (c) Ask for SYSOUT listings of the ACF2 Field Definition Record (ACFFDR) and TSO command list generations.

Components of ACF2

1. Determine if each of the VSAM clusters for ACF2 is uniquely named and is of adequate size. Note if each cluster is VSAM-password protected. Utilize the TSO LISTCAT command to list the data and index components of VSAM clusters SYS1.ACF.RULES, SYS1.ACF.LOGONIDS, and SYS1.ACF.INFOSTG (if used).

2. Repeat the above test using the three alternate clusters: SYS1.ACF.ALTLIDS, SYS1.ACF.ALTRULES, and SYS1.ACF.ALTINFO.

3. Use the DECOMP command or the XR report to check the rules for the ACF2 system files. Verify that the rules allow only full scoped security officers to access the primary ACF2 files, and that write access is not allowed.

4. Examine the SYS1 rule set to determine that the ACF2 distribution libraries may only be accessed by the system programmer assigned to ACF2 support. Files included are SYS1.ACFMAC, SYS1.ACFMOD, and SYS1.ACFOBJ.

5. Examine the SYS1 rule set to determine if the three non-VSAM ACF2 backup datasets (SYS1.ACF.BKLIDS, SYS1.ACF.BKRULES, and SYS1.ACF.BKINFO) are accessible only to ACF2 and the ACFRECVR recovery job (which should be controlled by Operations).

6. Determine that the libraries which contain ACF2 load modules, such as SYS1.LPALIB and SYS1.LINKLIB, are adequately protected. Write and allocate permission should be logged and restricted to key system programmers only.

7. Examine the ACF2 command limiting load modules, such as ACF$CMDO, in SYS1.LINKLIB. Use AMBLIST or SPF Hex Browse to examine the patch area and verify that no new commands have been added to the list after assembly.

8. Determine that ACF2 PTFs and other maintenance is performed via SMP (System Modification Program) and that all ACF2 maintenance is carefully reviewed.

9. Verify that the ACF2 distribution/maintenance tapes are designated as critical and are protected with adequate physical security.

Other Products

10. If ACF2 is being used for TSO command limiting, then examine the $CMDS assembly listing for the $TSOCMD macro and determine that the appropriate commands have been limited.

11. Using the ACF2 "SHOW PROGRAMS" command, determine that maintenance programs (i.e., those which bypass security checking) are specified as usable only out of a controlled library by a specific Logonid.

ACFFDR and GSO Options

12. Examine the ACF2 Field Definition Record (ACFFDR) and GSO records comparing options selected there with those shown to be in actual use by various ACF commands such as "SHOW STATE", "SHOW TSO", "SHOW ZEROFLDS", "SHOW SYSTEMS", and "SHOW FIELDS". Note and investigate any discrepancies.

13. Using the "SHOW STATE" command, verify that started tasks are controlled by ACF2, i.e., STC=ON.

14. Using "SHOW STATE", verify that access to tape datasets is controlled by ACF2, i.e., TAPEDSN and TAPEBLP in the OPTS GSO record.

15. Using "SHOW STATE", determine that all disk dataset names are protected by ACF2, i.e., specified as ******.

16. Using "SHOW ACTIVE", determine which of the following user exits are used by the unit. Request and examine the source code for each. Cross reference compile data and load module size with SYS1.LPALIB contents. Note any discrepancies.

1. DSNGEN	11. RSCXIT1
2. DSNPOST	12. RSCXIT2
3. EXPPXIT	13. RULEPRE
4. INFOPRE	14. RULEPST
5. INFOPST	15. STCXIT
6. LGNIXIT	16. SVCIXIT
7. LGNPARM	17. USREFLD
8. LGNPXIT	18. VIOEXIT
9 LGNTERM	19. VLDEXIT
10. NEWXPXIT	

17. Determine that ACF2 exit usage is well documented as to its purpose and its effect on the system.

18. Using the "SHOW SMFXIT", verify that the ACF2-required SMF exit modules are receiving control. The ACF2 required modules are ACF9BUJI for JOB INIT, ACF9AUSI for STEP INIT, and ACF87TRT for TERMINATION.

19. Using "SHOW STATE", check the appropriate GSO option to determine if passwords are adequately controlled by ACF2. For example, MINPSWD(5) to enforce that all Logonid passwords have a minimum of 5 characters.

20. Carefully examine the ACFFDR, comparing the @CFDE entries to the SKK-supplied defaults. Determine that any local modifications do not materially weaken security or control.

21. Examine @UID in the ACFFDR to determine which fields make up the installation's UID string. Verify that the pertinent fields in the Logonid may be changed only by authorized security and/or account personnel, never by users themselves.

Logonid Records

22. Using either the LIST command or the Super List (SL) report generator, display a sample of Logonid records. Verify that the password expiration parameter - MAXDAYS - of the miscellaneous group is used and is no greater than 30.

23. Using the "LIST IF" command or the SL report, determine who has the SECURITY or ACCOUNT privilege. These powers should be restricted to 2 or 3 persons, or else limited by the person's DSNSCOPE, UIDSCOPE, or SCPLIST if the unit uses decentralized administration. Systems programmers should never have either attribute.

24. Using the "LIST IF" command or the SL report, determine which Logonids have the NON-CNCL attribute. No more than 3 or 4 such Logonids should be found, and these should be for emergency use or for special purposes (e.g., started task ids) only. In addition, their usage should be reviewed.

25. Using the "LIST IF" command or the SL report, determine which Logonids have the RESTRICT attribute. Verify that these Logonids have SUBAUTH specified, as well as the PGM and LIB parameters, to ensure that their usage is via an APF-authorized program from a controlled library.

26. Using the "LIST IF" command or the SL report, examine all Logonid records to determine that no user has a 'SYS1' prefix, as this would allow complete access to all system files, including the ACF2 files. Similarly, determine that no user has all asterisks specified.

27. Using the "LIST IF" command or the SL report, examine all Logonid records to determine which users have the REFRESH attribute. These users are allowed to dynamically active GSO options.

28. Using the "LIST IF" command or the SL report, examine all Logonid records to determine which users have the MAINT attribute. These users are allowed to execute any program defined in MAINT GSO record.

Rule Records

29. Use the DECOMP command to decompile the "SYS1" rule set. Determine that any %CHANGE or %RCHANGE statements to permit rule modification are appropriate and justified.

30. Determine that SMF (System Management Facility) files (for example, SYS1.MANX and SYS1.MANY) which ACF2 uses for logging are adequately protected. Write permission should never be given for these SMF files. ALLOC permission should be logged and restricted to the systems programmer responsible for SYSGENs.

31. Check the GSO OPTS record, NOSORT field. If NOSORT is in effect, verify that all rule sets containing a $NOSORT control card accurately reflect access permissions. If NONOSORT is in effect, there should not be any $NOSORT entries.

Regular Review of ACF2 Reports

32. ACF2 offers a comprehensive set of violation and logging reports. Determine that security personnel are reviewing such reports regularly and are actively following up on potential problems.

33. Examine the cross reference reports (ACFRPTXR and ACFRPTRX). Determine if ACF2 rules grant access according to the "need to know" doctrine. Is the authority to change rules adequately controlled?

INDEX