acf^2 MVS

The Access Control Facility General Information Manual

acf2™

The Access Control Facility

GENERAL INFORMATION MANUAL

for

acf2/MVS Release 4.0 Installations

Base Manual Dated: January 15, 1985

Doc. Nr. AMG0002-03

SKK

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 GENERAL INFORMATION MANUAL

INTRODUCTION

This manual describes the basic functions and concepts of ACF2, the Access Control Facility. Other manuals provided in the ACF2 package are the Administrator's Guide, System Programmer's Guide, Utilities Manual, Implementation Planning Guide, Auditor's Guide, Messages Manual, CICS Support Manual, IMS Support Manual, IDMS Support Manual, Other Products Manual, and Composite Index. These manuals provide detailed explanations, examples, and instructions on the use of ACF2.

DESIGN PHILOSOPHY

ACF2 has been designed as an extension to the IBM OS/VS - MVS Operating System. It provides the ability to control access to the computer system and to control access to data residing on the direct access storage and tape devices of the system. Under ACF2, all data is protected by default and the purpose of the ACF2 system extensions is to control access to data. Protection levels of READ, WRITE, ALLOCate (which includes allocation, catalog, scratch, and rename functions), and EXECute-only are augmented by the concept of "environment" checking. That is, when the installation defines its access "rules" to ACF2 it may specify certain additional conditions that must be met before the access is allowed. For example, "environment checking" allows that a user of the system may be qualified for more powerful access to data if he is using a specified program or set of programs ("Program Pathing" feature), or if he is making his request from a particular input device ("Input Source" checking). These concepts provide for enhanced data integrity with controlled access to data.

ACF2 uses a highly efficient method for algorithmically specifying access rules for the controlled sharing of data. This results in a system that is easy to understand and use, is flexible, and is efficient in its use of computer resources such as CPU utilization and input/output requests.

ACF2 also provides a transition capability for making the change from an environment where no data is protected to one in which all data is protected. The system has the ability to log violations, warn the users of violations, or actually abort requests that would cause the violations. The level of control can be phased in.

SYSTEM INTEGRITY

Proper implementation of ACF2 will provide significant enhancements to data and resource security at an installation. It will also greatly reduce unauthorized accesses and similar exposures. Maximum protection and benefit is achieved when ACF2 is utilized as part of an overall approach to data processing security. Therefore, ACF2 is designed to interface with and support a number of other management controls. These include separation of function, individual responsibility and accountability, access to data on a need-to-know (job function) basis, and detailed auditing of system resource usage, data access, and internal controls. ACF2's design philosophy and its built-in features, options, and audit trails provide valuable assistance in implementing these additional controls. A sensible combination of ACF2 and these other management controls will minimize administrative requirements while maximizing the protection levels possible for a given installation and operating system.

The protection provided by ACF2 cannot be more absolute than the integrity of the operating system under which it is running. ACF2 by itself cannot prevent or protect against integrity exposures within the operating system. However, ACF2 will not knowingly introduce any exposures and will provide the maximum protection which is realistically possible under the given operating system.

TSO ACF COMMAND

ACF2 uses TSO as an interactive command system for the definition of users, resources, and their relationships. The ACF command may also be used in batch by the execution of the TSO Terminal Monitor Program or via the ACF2-supplied ACFBATCH utility. Additionally, ACF2 applications may be performed under SPF or ISPF. ISPF tutorials and screens are provided for ACF2 rule processing, Logonid maintenance, report generator execution, and ACF2 SHOW command displays.

DATASET PROTECTION (RESIDENT VOLUMES)

ACF2 provides protection at the dataset name level for all datasets residing on a set of direct access (DASD or MSS) volumes defined to ACF2 as "resident". (These volumes do not necessarily have to be physically resident on the system at all times.) In order to use the Program Pathing feature, the program libraries referenced must reside somewhere on this set of resident volumes.

VOLUME PROTECTION (SECURED VOLUMES)

ACF2 provides protection at the volume name level for all direct access storage (DASD or MSS) and tape volumes contained in a set of volumes defined as "secured". Tape volumes not defined as "secured" at the volume level can be protected at the dataset name level by specifying the system-wide option TAPEDSN=YES. Data on a volume is either protected at the volume level or the dataset level, but not both.

VTOC PROTECTION

ACF2 provides protection for the VTOCs (Volume Table of Contents) of the resident volumes by generating a pseudo dataset name of SYSVTOC.volser and then using a set of rules to determine the appropriate access. VTOCs on secured volumes are protected in the same manner as any other dataset on that volume.

TAPE VOLUMES

ACF2 provides the option of dataset protection on tape so that protection is equivalent for tape, disk, and MSS. Under this option, ACF2 considers the dataset name specified in the JCL as valid although MVS later verifies only the last seventeen characters. If the installation has installed a tape management system that validates the full dataset name specified in the JCL as matching its catalog information, then full dataset protection on tape can be accomplished without the use of ACF2 exits or local coding. Such a tape management system that falls into this category is UCC-1.

If the tape management system does not retain information describing each dataset on tape, then the installation must use the ACF2 Pseudo Dataset Name Generator Exit to interface with the tape management system and determine whether access should be allowed or whether the supplied dataset name should be validated. (See the description of the DSNGEN exit in the chapter "Installation Exits" of this manual.)

As a word of caution, tapes may be written by using EXCP (Execute Channel Program) access even though a tape dataset was opened for input only. The only true protection is to manually remove the write ring. Thus ACF2 provides the installation exits with the IBM Ring IN/OUT status indication, which could be used for additional security checking. Furthermore, once access has been gained to a tape dataset, any dataset residing on that tape volume may be accessed. Therefore datasets with differing security requirements should not be stored on the same volume.

OTHER RESOURCE PROTECTION

ACF2 also provides for the protection of various other system resources. These resources are explained in the functional overview in the next chapter.

MINIMUM MAINTENANCE LEVEL

ACF2 is designed to operate with either the IBM MVS, VS1, or VM operating systems and with various subsystems such as IMS, CICS, TSO, IDMS, JES, CMS, RSCS, etc. See the appropriate ACF2 System Programmer's Guide for detailed information on any required minimum system maintenance levels.

ACF2 CHARACTER DEPENDENCIES

Throughout this and other ACF2 manuals the special characters $, #, and @ are used to represent the hex equivalents 5B, 7B, and 7C respectively. Installations with different character representations for these hex values must take these differences into consideration locally (e.g., hex 5B may be a pound sterling sign in Great Britain).

FUNCTIONAL OVERVIEW

SYSTEM ACCESS CONTROL

All users are validated on entry to the system by ACF2. This validation takes place during TSO logon, JES2 JCL Conversion processing, JES3 Input Services processing (i.e., on the reader), or similar online system logon (e.g., TONE, ROSCOE). Optionally, this validation can take place during CICS, IMS, or IDMS sign-on. ACF2 facilities provide for a common user identification in all environments so that each system user needs only to be defined once regardless of which combinations of subsystems he is authorized to use (e.g. TSO, IMS, batch, etc). The validation consists of taking a user-supplied Logonid and password, and matching them with the appropriate Logonid record and password in the ACF2 Logonid Database.

The TSO ACF command and its subcommands are provided for the insertion of new Logonid records and the maintenance or deletion of existing records in the ACF2 Logonid Database. Optionally, the TSO UADS (User Attribute Dataset) dataset may be bypassed to avoid duplicate maintenance and definition requirements. Under this option, ACF2 will maintain equivalent user profile information in its own databases.

Other validations which may optionally take place at system access time include checking whether the physical input device (e.g., a terminal or RJE station) is an authorized job input source for that user, and whether the user is attempting to access the system within an allowable time period. There are also exit points provided in ACF2 at TSO logon pre- and post-processing time and at job entry time if special local validation or other processing is desired. General information on these and other ACF2 exits are included later in the chapter on installation exits.

In addition, a TSO user can be required to log on at a designated terminal where the insertion of an Operator Identification (OID) card is necessary.

DATA ACCESS CONTROL

ACF2 installs "intercepts" in the MVS Supervisor to gain control during new dataset allocation, open, scratch, rename, and catalog processing. These intercepts determine whether access should be allowed as a matter of course or whether further verification must be performed.

Accesses are automatically allowed for temporary datasets, and also for datasets which reside on the set of resident volumes and whose high-level index is owned by the current user. (However a validation of a user's accesses can be forced regardless of the dataset high level index.) If further verification is necessary, the appropriate set of access rules is obtained and used to determine the specified level of access.

If this set of rules does not exist, or the specified level of access is not sufficient, the user request is aborted. A record of this security violation is entered into both the System Management Facilities (SMF) log and the job log, and is displayed on the security console.

ACF2 also provides a transition mode, known as RULE mode, that can be selectively enforced for each individual rule set. Under this mode, each dataset access rule set may be set to QUIET, LOG, WARN, or ABORT modes by means of the $MODE control card within the rule set. Additionally, when RULE mode is set for the ACF2 system, a default system-wide mode for those dataset access rule sets lacking the $MODE control card may be specified.

An installation exit can optionally be invoked prior to all validation processing. This exit has the ability to whether a dataset name will be modified before ACF2 validation takes place. It can also be used as an alternative to RULE mode in providing a transition mode towards full ACF2 protection. ACF2 also provides pre- and post-processing exits for the retrieval and storage of records on the Access Rule database.

After ACF2 has made its determination as to whether or not an access is to be allowed, an installation dataset violation exit routine may be invoked. ACF2 will pass to this exit routine all pertinent information and the exit routine has the option of continuing with the abort, requesting that ACF2 allow the access, or allowing the access with a log record being generated. The exit routine can also take any specific action concerning an attempted security breach as required by the local installation.

In addition, locally-defined information can be used by the exit in determining the appropriate action to be taken. ACF2 is supplied with a sample exit that will allow transition to full security on an index by index basis using this locally-defined information.

GENERALIZED RESOURCE CONTROL

ACF2 provides for the protection of other resources. The following types of resources have been predefined to ACF2. They can be protected through installation-defined rules:

- CICS files
- CICS transactions
- CICS DL/I requests
- CICS programs
- CICS transient data
- CICS temporary storage
- IDMS areas
- IDMS protected programs
- IDMS non-protected programs
- IDMS subschema
- IDMS tasks
- IMS applications
- IMS transactions
- TSO accounts
- TSO procedures

An installation can define other resources and protect them through installation-defined rules. These resources can be protected individually or as part of a group of resources.

INFORMATION STORAGE

In addition to generalized resource definitions, the ACF2 Information Storage Database contains other ACF2-related records. These include the definition of scope lists for privileged users limiting the range of records (such as Logonids and access rules) that can be displayed or manipulated. Shift records which determine the allowable system, data, and resource access times for users are also defined on the Information Storage Database. Zone records defined on this database specify local time zones for system users. Input sources (devices) can also be defined here. Global System Option (GSO) records define system-wide options that affect the operation of ACF2 and can be dynamically changed by an authorized security administrator.

Installations may also use the ACF2 Information Storage database for storage and retrieval of records of up to 4000 bytes of installation-unique data to be associated with a key within a class of data storage. This data may be accessed from any authorized program or by authorized users within non-authorized programs under ACF2 pseudo dataset name control or via scope lists.

ACF2 services include direct access from any address space in any MVS system in a multi-cpu complex and complete data integrity for multi-cpu updating, as well as standard ACF2 backup and recovery facilities. ACF2 also provides pre- and post-processing exits for the retrieval and storage of records on the Information Storage database.

The ACF command provides support for a subclass of data that associates keys with data items or lists of data items. Examples of this are "physical input source to logical input source" translation and the definition of input source groups.

GENERALIZED PROGRAM PATH SUPPORT FOR TSO

Because certain TSO support subsystems, such as SPF and VSAPL, provide the ability to attach other TSO commands, it is important to be able to validate the control block structure and be able to determine which command is actually active. Additionally, some installations have written their own CALL commands to perform special pre-processing. ACF2 supports these different environments via a set of "structure" descriptions which can be modified by the installation to support their own unique environments.

RESTRICTED DUMP CONTROL

It is possible to compromise the security of a program that may only be accessed for execute or for data that may only be accessed via a particular program path by forcing an ABEND condition and thereby obtaining a dump. When one of these conditions exist, ACF2 will abort the dump request unless permission to obtain it was given to that user. ACF2 issues a WTP (Write-to-Programmer) message indicating the original ABEND code that has occurred.

ACF2 DATABASES

The ACF2 databases consist of three VSAM key-sequenced clusters:

1) The Logonid database contains an entry for each user or logon identifier (Logonid).

2) The Rules database contains a rule set for each dataset name high-level index or volume (volser). These rule sets each allow specific access to certain data.

3) The Information Storage database contains ACF2 generalized resource rule sets, entry records, scope records, zone/shift records, and Global System Option (GSO) records.

FIELD DEFINITION RECORD

The ACF2 Field Definition Record (the ACFFDR) is a module modified and assembled by each installation to customize ACF2 to meet its particular security requirements. The assembler macros used and the parameter specifications for each macro are provided in the acf2/MVS System Programmer's Guide. A general overview of these macros is provided in this manual in the chapter entitled "Field Definition Record Generation".

MODES OF THE ACF2 SYSTEM

Transition to full dataset-related security under ACF2 is facilitated by the five modes of the ACF2 system. The mode is specified in the GSO OPTS record maintained on the Infostorage database.

QUIET MODE

Under this mode, ACF2 performs user identification, as under all other modes. This validation is to check whether a user is entering the system under the correct Logonid and password, through a correct input source, during an allowable time period, and according to other privileges designated by the user's Logonid record. ACF2 does not check data access rules in QUIET mode. However, privileges specified in the Logonid record are checked and enforced.

LOG MODE

In LOG mode, ACF2 checks its rule databases as well as performing user validation. If an attempted access is determined invalid according to the rules (e.g., no rule is written yet) ACF2 allows but logs the access.

WARN MODE

This mode functions in the same manner as LOG mode, except that ACF2 now issues a warning message to the user when an access rule is violated and indicates to the user that, in the future, this access will be denied if the rules are not updated. These messages will appear on terminals and/or printed in the job log for batch environments. A logging report record is also created for each of these accesses.

ABORT MODE

This is the normal mode of operation under which accesses considered invalid will be denied. Access rules now determine and control data sharing and all violation attempts are aborted and logged.

RULE MODE

The RULE mode allows an installation to migrate access rules to full ABORT mode on a rule set basis. If access would ordinarily be prevented by an existing rule, RULE MODE can provide overrides depending on whether or not a rule set is found.

SYSTEM ACCESS CONTROL

One major ACF2 controlling factor for system access validation is the Logonid record. Each system user should be defined to ACF2 by a unique Logonid and Logonid record for individual accountability. A Logonid record contains numerous fields of information. These fields define the user's privileges (such as the ability to use CICS or TSO), the user's system history (such as the number of security violations to date), the user's TSO attributes, and other ACF2 system-related information.

THE LOGONID RECORD

The Logonid record is variable in length, with a maximum of 1,024 bytes. The ACF2 system itself reserves 640 bytes. The rest of this record can be used for installation-defined fields. The fields of the Logonid record, when displayed using the ACF command, are presented in 7 logical groups, such as the Identification Section, the Privileges Section, and the TSO Section. The group (or section) into which a particular field is placed can be determined locally. A detailed description of each field of the Logonid record and the organization into these groups is provided in the acf2/MVS Administrator's Guide. That manual also provides instructions on the creation and maintenance of Logonid records.

All Logonid record fields are defined to ACF2 via the ACF2 Field Definition Record (ACFFDR). For each field, a field name and its attributes are defined, along with information which indicates what authorization is required to modify or iist that field. This gives the system field level data control over each Logonid record field, whether predefined with the ACF2 package or locally added. Installation-defined fields are easily added. For further information see the chapter entitled "Field Definition Record Generation".

The ACF command and its subcommands are used to create, change, list, and delete Logonid records dynamically and under strict ACF2 controls. These commands can be issued via TSO and also via ACF2-supplied ISPF screens. The ACF subcommands can also be executed in a batch environment by use of ACF2 batch utilities.

USER IDENTIFICATION STRING

ACF2 uses a "pseudo field" called the User Identification (UID) string. This string, which defaults to the Logonid, can be defined by a set of ACF2-supplied or installation-defined fields of the Logonid record. The UID string allows the addition of pertinent information such as department, job responsibility, employee id number, or other data necessary for determining access privileges.

It is important to spend some time initially determining the appropriate set of fields to include in the UID String. After the users of the system have begun creating access rules, the definition of this field cannot be modified without impacting many of the existing rules.

Careful consideration should be given to both the length and content of the UID string. The format selected must apply to the entire installation, though selected fields within the string can be used to define different functional uses within different identifiable groups.

The UID string should be long enough to contain all meaningful data that will be required for the proper generation of access and resource rules. However, redundancy and/or use of excessive characters can make the UID string too long to use easily.

SYSTEM ACCESS VALIDATION

TSO Logon

Validation to the system via TSO is performed when the TSO LOGON command is issued. It has the following format:

```
LOGON logonid
```

at which point the system will respond:

```
ACF82004 ACF2, ENTER PASSWORD-
```

The installation may also allow a short, one line, LOGON command:

```
LOGON logonid/password
```

Wherever the password can be specified, a new password may also be specified (i.e., the password may be changed) by the following notation (unless this has been disallowed by local selection of ACF2 parameters):

```
old-password/new-password
```

The installation may also (optionally) direct ACF2 to check that any user attempting to LOGON to TSO has the "TSO" attribute in their Logonid record.

NOTE: A TSO logon can be aborted by entering a plus sign (+) in response to any ACF2 prompt.

TSO Fullscreen Logon

After entry of the Logonid and password, a logon screen can appear on an installation-wide basis or for individual users. This facility provides a convenient method for entering logon values if an installation chooses not to use UADS.

The installation can control which values each particular user can enter, which default values will exist, and which values will be retained from session to session.

This facility can operate even without TSO/E.

IMS, CICS, and IDMS Sign-ons

Validation to the system via IMS, CICS, and IDMS sign-ons may also be done with ACF2 via optional interfaces to the sign-on processing in each IMS, CICS, or IDMS address space for which ACF2 protection has been selected. Logonids and passwords are verified and, additionally, the user's Logonid record is checked to see whether the user has permission to use the particular IMS or CICS region or IDMS address space he is trying to sign on to.

Batch Jobs

Validation to the batch portion of the system is done via JES2 or JES3, which recognizes two new OS control cards of the format:

```
//*LOGONID logonid

//*PASSWORD password
```

or, the following format for specifying new passwords:

```
//*PASSWORD old-password/new-password
```

Jobs submitted by TSO users (via the SUBMIT command) inherit the validated Logonid and password entered at TSO logon. Therefore, the //*LOGONID and //*PASSWORD control cards would not be necessary.

Batch jobs that enter the system without the //*LOGONID and //*PASSWORD cards can be executed using an installation-specified default Logonid.

If MVS Selectable Unit 16 (SU816) has been installed, or the system is at a recent maintenance level, then the Logonid and password may alternately be specified as USER= and PASSWORD= parameters of the JOB card (specifying a maximum of 7 characters in the Logonid by means of the USER= parameter). To change the password via this method, specify:

```
USER=logonid,PASSWORD=(old-password,new-password)
```

The installation may optionally direct ACF2 to check that any user entering a batch/background job must have the appropriate authorization by means of his Logonid record.

All System Accesses

For all system accesses, ACF2 also validates that the Logonid specified: has been previously defined to ACF2; is currently active (neither cancelled nor suspended); and either is "restricted" (no password required) or provides the correct password which must not be expired. See the discussion that follows on Logonid password controls for further information on password verification.

If so specified in the Logonid record, then the actual input device being used is checked as being in the source group specified for that user. If it is not in the specified group, access to the system will be denied.

The Logonid record is also checked to determine if the user is attempting access during the allowable time specified. The installation may designate in the Logonid record that a user be allowed access outside the shift specified.

Under TSO, a user can be prompted to insert an Operator Identification (OID) card before entry is allowed to the system.

Whenever access to the system is denied, a textual error message will be returned to the user explaining the reason for denial.

LOGONID PASSWORD CONTROLS

ACF2 requires that every Logonid have a password associated with it (except for restricted Logonids, generally used for running production jobs). Before processing the password information, ACF2 irreversibly encrypts the password in a one-way methodology which cannot be deciphered. ACF2 never displays a password, either in its encoded or clear text form. An installation also has several options regarding passwords. The installation can determine:

1. How many times an incorrect password can be entered from TSO in a single day for a given Logonid before the Logonid is suspended.

2. Whether to count batch password violations in the total number of incorrect passwords under a given Logonid during a single day.

3. The maximum number of days allowed before a password must be changed to a different one.

4. The minimum number of days that a newly set password must by kept.

5. The number of days before password expiration when warnings to change the password will be issued on each system access.

6. The minimum acceptable number of characters for a new password.

7. Whether a user can modify his own password at system entry time.

8. Whether to use a password exit, to gain control when a new password is entered at system entry time or via the ACF command. This exit has the capability to deny the new password.

9. An expired password exit, which is given control when system access is attempted under a Logonid with an expired password. This exit can allow access, deny access, or allow access and set a new password. If a new password is set under TSO, this exit will cause ACF2 to inform the user and ask for reverification.

10. If a Logonid is not assigned a password when it is established, ACF2 will assign the password from the first system access for the Logonid. It will never allow a Logonid to be used without a password unless it is restricted.

11. The method of password encryption to be used (R221 or XDES). Use of the XDES encryption method provides not only an enhanced password encryption algorithm, but ensures that passwords transmitted over Network Job Entry (NJE) systems will never appear in clear text.

Additionally, a standard ACF2 feature is that jobs that submit other jobs will, by default, inherit the Logonid of the submitting job (except from Started Tasks or via the ACF2 JOBCOPY or ACFSUB utility. Refer to the section on Production Jobs). Therefore, if a production job submits other jobs, these jobs run under a production Logonid with no Logonid or password required in the submitted JCL. Similarly, if a TSO user submits a job via the SUBMIT command, no changes to the actual JCL are required. ACF2 will automatically transfer the user's Logonid and input source information to the submitted job.

PRODUCTION JOB CONTROL

Production jobs are very powerful and therefore require powerful Logonids to ensure proper control. The control of production jobs is a critical factor in establishing a thorough system of security.

Production jobs should run under production Logonids. Production Logonids should be clearly distinguishable from user Logonids.

WHAT SHOULD BE CONTROLLED?

The following controls should be considered for production jobs:

1. Access to production data files should be restricted to production programs. Test programs should access only test files. All allowed accesses to production data by programmers should be logged. Any non-production access of production data should be logged and periodically audited to determine if proper access control is being maintained.

2. Changes to production program libraries should be restricted to an individual (or small group of individuals) assigned the task of system librarian. Ideally, these individuals should be the only ones who have access to production libraries.

3. Changes to production JCL must be controlled. Cataloged procedures and the JCL for jobs submitted by production job streams must also be controlled. Production JCL should only come from a highly controlled library. If used, the production Logonids and passwords should not be stored with the JCL.

4. The submission of production jobs using a production Logonid must be controlled. The submitters, the source of submission, and the path used must be under ACF2 control. Only authorized persons should be able to modify and submit production jobs, and only from authorized locations or terminals.

JOB SUBMISSION

Consider the ways a job can be submitted and the source of that submission. Jobs may enter the system by means of local or remote card readers, as started tasks initiated by the operator, or from TSO, CICS, IMS, etc.

Remember that jobs or programs once inside the system can submit other jobs to the internal reader. Every job that is submitted to the system has a source identifiable to ACF2. Controlling all aspects of production processing requires the following:

1. Access to JCL libraries as well as cataloged procedure libraries should be controlled to prevent unauthorized additions, changes, or deletions.

2. Only a small, specific group of people located at specific terminals should be allowed to set up and/or submit production jobs.

3. Production jobs should be submitted through a specific program stored in a highly controlled library. This library should be accessible only by those authorized to submit production jobs.

4. Access to production data should be restricted to production Logonids and, on an exception basis, possibly to specific support programmers (with their accesses logged).

5. Keep in mind physical access control. No matter how thoroughly you have controlled your libraries and sources, allowing unauthorized access to paper printouts of production data, for example, defeats your control efforts.

ACF2 CONTROL MECHANISMS

JCL libraries, which are often partitioned datasets, can be protected via ACF2 access rules. Programmers can be allowed only to read these libraries. Writing to these libraries may be allowed but logged for a specific individual by means of a rule. In this way, a complete record can be kept of all changes and updates to production libraries. Thus, changes and updates can be monitored to ensure a controlled environment. Procedure libraries can be protected in the same way by means of access rules.

Production data access can be limited to production Logonids by means of rules that provide read, write, and allocate authority only to the production Logonids. Support programmers may be given the same privileges (temporarily or permanently) but with logging specified.

All production Logonids should be restricted. These restricted Logonids require no password, and are logged upon entry to the system. They cannot be used for online processing.

In addition, a Logonid record can contain the name of the program that must submit any JCL with this Logonid. Also, the submitting program can be required to come from an APF-authorized library and from an allowable submission source (such as a particular reader).

Thus, ACF2 provides several means of controlling production data. Access rules are used to determine allowable types of access, and the fields of the Logonid record are used to define production Logonid records. ACF2 provides 2 programs, ACFSUB and JOBCOPY, which can be used to control the submission of production jobs. The source code for these jobs is provided with ACF2, and they can be fit into your installation's present job scheduling package.

DATA ACCESS CONTROL

As mentioned at the beginning of this manual, ACF2 protects all data on the system by default. To allow a user to access particular data, access rules must be written either by an installation security officer or by the owner of the data. Access rules are maintained on the ACF2 Access Rules Database as rule sets which consist of control cards and rule entries.

CONTROL CARDS

The control cards of the rule set determine such attributes as:

1. The high-level index of the dataset for which the rule is written.

2. A prefix to be used instead of the high-level index when a rule set pertains to datasets of various high-level indexes, or when several rule sets pertain to datasets of a particular high-level index.

3. User information to be supplied to locally defined exits.

4. The "mode" of the rule set. Through this control card and the system option of RULE mode (discussed previously), a rule set can have its own mode.

5. The user or users who have the authority to change the entire rule set and/or users who may change rule entries.

6. Whether or not rule entries are to be sorted by ACF2.

ACCESS RULE ENTRIES

Following the control cards in a rule set are the actual rule entries that determine the environment under which access to particular data will be allowed. Among the items that are specified are the dataset name and the access permission (such as READ, WRITE, ALLOCATE, or EXECUTE) allowed. Additional parameters may be specified to further limit the use of the data. These include the volume on which the dataset must reside, the allowable input source from which a user can access the data, the program and library which must be used to access the data, the times and days (shifts) when access is allowed, etc. Also, a time limitation may be specified to allow only temporary access to this dataset.

Access rules for datasets of a particular high-level index are grouped together and compiled, much like a program, into a rule object record. The input to the ACF2 rule compiler consists of the access rule entries and control cards.

THE ACF COMMAND

The ACF command and its various subcommands are used to build, maintain, and delete access rules on the ACF2 Access Rules database. This command can be issued as a TSO READY command or in batch. Additionally, ACF2 provides ISPF screens for the processing of access rules. Special ACF2 privileges are required to issue these commands.

MASKS

Almost all character strings used in the ACF2 system may be replaced with a character string pattern. For example, in defining the data access rules, patterns may be used to define the dataset, volume, User Identification String (UID), program library, and program name fields. Two special symbols drive the character substitution process: the asterisk (*), and the minus sign (-). There are differences in the way ACF2 processes these symbols depending upon the type of character string, i.e., fixed length strings (such as the UID) versus variable length strings (such as dataset names, program names, etc.). The basic function of each symbol is described below.

Asterisk (*)

Indicates that any character will be considered "matched" on that position. Any number of asterisks may be placed in a string pattern. For example, AB*D will match any string containing AB in positions 1-2 and a D in position 4. Thus, ABCD, AB1D, ABXD, ABBD, etc. are considered matched with AB*D. An asterisk embedded anywhere within a string does not match a NULL character, e.g., AB*D does not match ABD.

However, if an asterisk is placed at the end of a string, the trailing asterisk matches a NULL character as well as every other character. For example, ABC* matches ABC, ABCD, ABC1, ABC2, etc.

Dash (-)

Indicates that asterisks are to be appended to create the maximum length of the character string. Maximum length varies according to the string type. A minus sign is only valid when placed at the end of a character string or when used by itself. It may not be imbedded between other characters.

If the minus sign is omitted from a variable length string specification, the remainder of the name is assumed to be blanks and will only match blanks. Also, when specifying dataset names, a minus sign may be specified as the only character of a dataset name index level to indicate that any number (including zero) of index levels may be present in the target dataset.

SAMPLE RULE SETS

```
$KEY(SYS1)
*ALLOW DP-MGR TO UPDATE RULE ENTRIES
%RCHANGE DPMGRUID
*PREVENT PEOPLE FROM LOOKING AT SYS1.PARMLIB
 PARMLIB READ(P)
*UNLESS THEY ARE INTERNAL USERS (UIDS BEGINNING WITH INT)
*AND ARE ACCESSING SYSTEM DURING NORMALLY-ASSIGNED SHIFT
 PARMLIB UID(INT) SHIFT(NORMAL) READ(A) WRITE(L)
*ALLOW SYSTEMS STAFF (UIDS INTS) TO CREATE NEW SYSTEM PACK
 - VOL(NEWRES) UID(INTS) READ(A) WRITE(A) ALLOC(A) EXEC(A)
*ALLOW EXECUTE ONLY ACCESS TO PROGRAM PRODUCT LIBRARY
 PPLIB READ(P) EXEC(A)


*RESTRICT ACCESS TO PANVALET SOURCE LIBRARIES (PAN.-)
*ONLY VIA THE PANVALET PROGRAMS (PAN#1, PAN#2, ...)
*WHICH ARE STORED IN THE SYSTEM LINKLIST
*ALLOW THIS FOR A PERIOD OF 90 DAYS
$KEY(PAN)
 - LIB('SYS1.LINKLIB') PGM(PAN**) FOR(90) READ(A) WRITE(A)


*APPL.PROD.- ARE THE PRODUCTION LIBRARIES
*APPL.TEST.- ARE THE TEST LIBRARIES
*UQC... ARE THE QUALITY CONTROL STAFF
*UAP... ARE THE APPLICATION PROGRAMMERS
*UPP... ARE THE PRODUCTION PERSONNEL
*$COPY IS A UTILITY WHICH COPIES LOAD MODULES, KEEPS AN
*ACCURATE HISTORY, AND ARCHIVES OLD COPIES OF THE PROGRAMS
*FOR RECOVERY AND BACKUP.
```

```
$KEY(APPL)
* ALLOW UAP... TO OBTAIN PRODUCTION CODE
 PROD.- UID(UAP) LIB('SYS1.LINKLIB') PGM($COPY) READ(L)
* ALLOW UAP... EXECUTE-ONLY ACCESS TO REST OF CODE
 PROD.- UID(UAP) EXEC(A)
* ALLOW UAP... TO MODIFY TEST LIBRARIES
 TEST.- UID(UAP) READ(A) WRITE(A)
* ALLOW UQC... TO TEST AND COPY CODE
 TEST.- UID(UQC) READ(A)
* ALLOW UQC... TO COPY INTO PRODUCTION LIBRARIES
 PROD.- UID(UQC) LIB('SYS1.LINKLIB') PGM($COPY) READ(A) WRITE(L)
* ALLOW UPP TO EXECUTE PRODUCTION CODE
 PROD.- UID(UPP) EXEC(A)
```

NOTE: Access rules are sorted by ACF2 alphabetically by dataset name and then from most specific to most general. In the $KEY(APPL) rule set, the PROD rules would precede the TEST rules. In the $KEY(SYS1) rule set, PARMLIB UID(INT) would precede PARMLIB READ(P) since it is more specific. Therefore, internal users would always match the PARMLIB UID(INT) rule and would not be prevented from reading SYS1.PARMLIB.

EXECUTION FLOW

The flow of control during execution of a job making a dataset access request is as follows:

1. If the access is for the Volume Table of Contents (VTOC) on a "resident" volume, then the dataset name is changed to SYSVTOC.volser.

2. The installation dataset pre-validation exit is taken. This exit can modify the dataset name or force a disposition for the request.

3. Non-Tape Datasets. If the dataset resides on the "resident" set of volumes, the actual dataset name is used in the search. If the dataset resides on the "secured" set of volumes, the dataset name @volser.VOLUME or VOLUME.@volser is used in the search. In this name, @volser is the actual volser, and VOLUME is literally the word VOLUME. (The exact name format is determined by the VOLRULE option of the GSO OPTS record). If the dataset resides on neither, the Pseudo Dataset Name Generator Exit (DSNGEN) is taken.

   Tape Datasets. Validation can follow one of these cases:

   a) The access attempt may request no tape bypass label processing (tape BLP). ACF2 then gets the volser from the JCL. If the JCL does not specify a volser, ACF2 will attempt to get the volser from the catalog.

b) The access attempt may request tape BLP while the user (Logonid) has the TAPE-BLP attribute. The label on the tape will then be ignored. As in the previous case, ACF2 will get the volser from the JCL or, alternatively, the catalog.

c) The access attempt may request tape BLP while the user (Logonid) has the TAPE-LBL attribute. If the tape has a standard label, then ACF2 will attempt to read the actual volser from the tape. If the tape has a nonstandard label, then ACF2 will get the volser from the JCL or, alternatively, the catalog.

d) The access attempt may request tape BLP while the program has been authorized for tape bypass label processing (via the GSO BLPPGM record). ACF2 will get the volser from the JCL or, alternatively, the catalog.

e) The access attempt may request tape BLP while no bypass label authority exists. The access request is aborted unless a post-validation exit changes this access recommendation. No further validation takes place.

After the above validation takes place, ACF2 determines whether the dataset resides on a secured volume, as follows:

a) If the dataset resides on a secured volume, then the volser is put in the format @volser.VOLUME or VOLUME.@volser, depending upon the installation option. Validation continues with the next step.

b) If the dataset does not reside on a secured volume and a DSNGEN exit exists, then the exit is taken and validation continues with the next step.

c) If the dataset does not reside on a secured volume and a DSNGEN exit does not exist, then ACF2 checks whether the system-wide option for tape dataset validation is in effect (via the GSO TAPESDN record). If so, then validation continues with the next step. Otherwise, the access is allowed (assuming that an unauthorized dump is not being taken).

4. If the dataset is temporary or begins with the owned prefix of the user, access is allowed.

5. If the Logonid is marked NON-CNCL, then the list of allowable maintenance Logonids,programs, and libraries is checked. (This list is in the GSO MAINT record.) If a match is found, access is allowed. If a match is not found, then normal validation continues. (However, if normal validation does not allow the access, then the access is allowed but logged.)

6. If the Logonid is marked MAINT, then the list of allowable maintenance Logonids, programs, and libraries is checked. If a match is found, then access is allowed. If a match is not found then normal validation continues.

7. If not allowed under one of the above conditions, the list of system-wide resident rules is searched for the correct rule set. If found, this set of rules is interpreted.

8. If the applicable rule is not in the system-wide resident set, the address space list is searched. If found, this set of rules is interpreted.

9. If the rule is not in the address-space list, a VSAM GET is performed out of the Globally Shared Resource Pool to obtain the record. If the record exists, it is chained to the address space list of resident rules and then interpreted. If it is not found, the request is aborted.

10. The appropriate access rule set is interpreted. If a matching access rule says allow or allow and log, then the appropriate action is taken.

11. If the request is to be aborted but the Logonid is marked NON-CNCL or is a security officer with the proper scope, the access is allowed but logged. Otherwise the request is aborted unless the installation supplied security violation exit requests that the access be allowed.

NOTES: Depending on the ACF2 mode, accesses may be allowed or allowed but logged instead of being aborted where indicated in the execution flow above. (For example, the system-wide MODE option can be set to LOG versus ABORT during a transition period.)

Optionally, an installation can specify in a user's Logonid record that all dataset accesses require authorization via an access rule. This requirement can override other attributes specified in the user's Logonid record. However, the NON-CNCL attribute can be used to override rule entries. The READALL attribute can allow for read-only access not allowed by rule entries.

For every job, an ACF2 Rules database access will be performed only once for each unique high-level index referenced. Furthermore, no access will be performed for those datasets which are resident, owned by the user, or temporary.

GENERALIZED RESOURCE CONTROL

In addition to providing access rule sets for the definition of access control to datasets, ACF2 provides generalized resource rule sets for the control of access to logical resources. Logical resources are grouped into types of resources, and resource names within a given type must be unique.

Resource types are from one to three characters in length and denote a logical class of resources such as transactions, account or billing numbers, etc. Installations can locally define any new logical class of resources as a resource type, and then write rules for it under a type code. However, ACF2 comes with two predefined resource types that cannot be reassigned locally. These are:

TAC	Time Sharing Option account rule sets
TPR	Time Sharing Option procedure rule sets

In addition, other resource types come predefined in ACF2. These relate to the optional CICS, IMS, and IDMS CICS interfaces and can be renamed locally. For further information, see the acf2/MVS CICS Support Manual, acf2/MVS IMS Support Manual, or the acf2/MVS IDMS Support Manual.

IAG	IMS application group name rule sets
ITR	IMS transaction rule sets
CFC	CICS file rule sets
CKC	CICS transaction rule sets
CPB	CICS DL/I request rule sets
CPC	CICS program control rule sets
CTD	CICS transient data rule sets
CTS	CICS temporary storage rule sets
PGM	IDMS program control rule sets

PGN	IDMS non-protected program control rule sets
SSC	IDMS subschema rule sets
TSK	IDMS task control rule sets

Resource names are from one to forty characters long and denote an individual resource within a given type. If a directory for a type of resource has been built, either by a Field Definition Record option or upon request by an application subsystem, then the resource names may contain asterisks; standard ACF2 masking operations will be used.

GENERALIZED RESOURCE RULE SET

Generalized resource rules for each name are grouped together and compiled, much like a program, into a rule object record. The input to the rule compiler consists of the individual generalized resource rules and the rule set control cards. These rule sets and control cards are similar to those used in the writing of access rules.

THE ACF COMMAND

The ACF command and its subcommands are used to create and maintain resource rules on the ACF2 Information Storage database. These are available under TSO or in batch (via the TMP in background or the ACF2 ACFBATCH utility). Additionally, ACF2 SPF screens are available for resource rule processing at MVS installations with TSO/SPF.

RESIDENT DIRECTORIES AND USING RESOURCE NAMES AS MASKS

In order to efficiently perform the masking operations on large numbers of resource name masks, it is necessary to form an in-storage directory of all resource names in a given resource type (specified via the GSO RESDIR record). An application program may also specifically request that a directory be built or that a list of rules be made resident at its initialization. If generalized resource rule sets have not been made resident, then they normally will be made resident in the address space queue of rule sets when they are needed. However, when building the directory or making the resource verification call, the generalized resource rule set can optionally be forced to remain transient (not retained in storage). Directories to be built globally should be specified to ACF2 via the GSO RESDIR record.

SAMPLE GENERALIZED RESOURCE RULES

```
* LIMIT ACCESS TO THE PAYROLL ADD TRANSACTION
* TO MANAGERS IN PERSONNEL DEPT (UID=PDM...).
* IF THEY USE THEIR OFFICE TERMINALS
* NO PASSWORD REVERIFICATION NECESSARY
$KEY(PAYADD) TYPE(ITR)
 UID(PDM) SOURCE(PDTRMS) ALLOW
 UID(PDM) VERIFY ALLOW

* LIMIT USE OF ACCOUNT (BILLING) NUMBERS
* CC9xxxx TO TECH SUPPORT PEOPLE (DPT)
* BUT ALLOW OPERATIONS (DPO) TO USE
* ALL OTHER CCnxxxx ACCOUNT NUMBERS
* ALTHOUGH THEIR USAGE IS TO BE LOGGED..
* THE xxxx ARE SUB-PROJECT NUMBERS
* THAT THEY ASSIGN AND THEREFORE
* NO CONTROLS ARE PLACED ON THEM..
* USING A MASK ASSUMES THAT A DIRECTORY HAS
* BEEN BUILT FOR THAT TYPE(TAC) VIA THE
* FIELD DEFINITION RECORD MACRO @RESDIR
$KEY(CC9****) TYPE(TAC)
 UID(DPT) ALLOW
$KEY(CC*****) TYPE(TAC)
 UID(DPO) LOG
```

NOTE: Rule records, like rule entries within a rule record, are sorted by ACF2 to most specific first. Thus the directory built for type code TAC in this example would recognize CC9**** as more specific than CC*****. Therefore, validation of a TSO account number beginning CC9 would always be dependent on the CC9**** rule only (assuming no other, more specific rule record existed), even though it could have also matched the CC***** rule.

EXECUTION FLOW

The flow of control for a resource authorization request is as follows:

1. The installation resource pre-validation exit is taken. This exit can modify the resource type or name and can force a disposition for the request.

2. The system-wide resident generalized resource rule set is searched, either for the directory for the type specified, or the exact resource rule set. If the directory is found, the directory is searched for the appropriate entry. If rules for that directory were made resident, then the address of the rule set is obtained from the directory entry and that set of rules is used for interpretation. If the rules were not made resident, then the search argument is changed to the mask instead of the resource name. The rest of the system-wide chain is searched for the mask entry rule set.

3. If the rule set is not on the system-wide set, the address space queue is searched for the directory or an exact match on the name, with procedures identical to those used in searching the system-wide set.

4. If the rule set is not found on either of the lists, a VSAM GET is performed out of the Globally Shared Resource Pool to obtain the record. If the record does not exist, the abort checking sequence is started (#5). If the record is found, then the rule set is interpreted. If the rule set is to remain resident, it is added to the address space rule list.

5. If the interpreter indicates that the request should be aborted, or the resource rule set was not found, a recommendation will be made to abort the request. If the Logonid is NON-CNCL or indicates a full scope security officer, the recommendation will be to allow but log the request.

6. The installation resource post-processing exit is now taken, and it is given the ACF2 recommendation for the disposition of the request. It has the ability to modify this disposition.

Both the $USERDATA and the rule DATA fields are passed to the installation post processing exit and the calling application subsystem so that unique requirements can be satisfied.

STARTED TASK CONTROL

If the Started Task (STC) Control option has been selected, ACF2 will provide control over dataset access by the STCs. If it has not been selected, ACF2 will ignore (allow) all STC access to data.

EXECUTION FLOW

1. Upon the first interface of an STC to ACF2 (dataset allocation, OPEN, etc.), the procedure name will be selected as the Logonid and a validation call will be made to the ACF2 support routines. In non-SE/SP systems, validation is done at the first open. SE and SP systems will validate STC's using $EFUI1 and IEFUSI SMF exits, same as in batch jobs.

2. Within the support routine, an installation STC validation exit will be called. This exit may instruct ACF2 to abort the STC, alter the Logonid to be used for validation, or alter the Logonid to be used as the STC default (in case the original Logonid did not exist).

   Logonids for use by STCs must have the STC attribute and any Logonid that has the STC attribute cannot be used by a normal user.

3. After the Logonid has been validated, standard ACF2 dataset controls will take effect except for the Master Scheduler, Job Entry Subsystem, and ACF2 address spaces.

4. Note that when STC=YES, validation calls are made for operator started tasks, but not for internal operating system tasks such as the Master Scheduler.

INFORMATION STORAGE

ACF2 provides, via its VSAM Global Shared Resource Pool and Shared DASD support, direct access from any address space in any CPU to a common information storage facility. ACF2 provides integrity for multiple concurrent updates, journalling of all changes, and its standard recovery facilities.

Creation and updating of records on the Information Storage database is limited to security officers. Other privileged users (e.g., auditors) may display records (within their scopes). The ACF command and subcommands for processing records on the Information Storage database are available under TSO and in batch. Access can also be performed by APF authorized programs, or under special ACF2 controls. For example, each input source entry record in Information Storage may have a pseudo dataset name associated with it. If read access is allowed to the pseudo dataset name, then read access will be allowed to that Information Storage record. Write access can be controlled similarly. Authority to change generalized resource rules can be given to non-security officers by means of special control cards within the generalized resource rule set.

There are 26 classes of Information Storage records (A-Z) reserved for standard ACF2 use. ACF2 currently uses classes C, E, R, S, and T. Additional classes could be assigned locally using non-alphabetic class codes. Within each class, specific three-character type codes may be used. Within a class and type, individual records are referenced via a 40-byte name. Up to approximately 4000 bytes of data can be associated with each key. Possible uses for Information Storage include any system-wide information such as accounting, volume control, etc.

ENTRY RECORDS

Entry records can be one of the following:

* Input source records identify individual input sources involved in ACF2 validation. (Input sources may be terminals, card readers, etc.)

* Source group records identify groups of input sources involved in ACF2 validation.

* OID records identify those users who are required to use Operator Identification (OID) cards when entering the system. These records also contain the character strings on the OID card.

Authorization to create and update entry records is normally allowed only for unrestricted security officers, while full scope auditors may display these records. This authorization can be extended to any other users through:

- * A pseudo dataset name, which identifies an existing or imaginary dataset. Those users who have access to the pseudo dataset (via access rules or Logonid privileges) can access the entry record.
- * ACF2 access rules that permit reading, writing, and deleting (allocating) of the Infostorage database.
- * APF authorized programs, which have create and update access.

Input Source Support

Every Logonid that is validated by ACF2 has an input source designation associated with it. For batch jobs this is the JES input device designation (e.g. READER1, RMT1.RDR1, INTRDR etc.) and for TSO this is the device address or the TCAM/VTAM Line Identifier (e.g. UCB-038, T0027, etc.). The source associated with a JOB read is determined as follows:

ONSITE Reader - source is READERn or RDRn, such as READER1 or RDR1.

REMOTE Reader - source is RMTn.RDRm, where RMTn is the terminal identification number and m is the device or reader number. For example, a job read from REMOTE 1, READER 2; the source would be RMT1.RDR2.

IMS Systems - IMS sign-on source restrictions will be based on either the VTAM Node Name or the relative line and terminal (stated as lll/ttt) for terminals connected directly to IMS through BTAM.

CICS Systems - CICS sign-on source restrictions will be based upon the Terminal Control Table (TCTTE) Terminal Identification Name which is left justified and padded with four (4) blanks for terminals connected to CICS.

| IDMS Systems - IDMS sign-on source restrictions will be based upon
| the logical source name defined by IDMS.

Often these device designations are not particularly accurate. An installation exit is provided that may modify the input source designation. For example, it may read the terminal serial number. This source designation is called the physical source identifier.

Physical source identifiers are subject to random changes (recabling of terminals, swapping of terminals for repairs, etc.). If serial numbers are used (or some other information generated in a local exit), these identifers may be longer than the eight character limit for source identifiers. This makes them a less than optimum choice for use in access and resource rules and for limiting Logonid usage. Therefore ACF2 translates the physical source identifier, which can be up to forty characters long, to a logical source identifier, which is up to eight characters long. It does this by looking up an entry in its information storage database with type code SRC and a key of the physical source identifier. It then uses the first data item as the logical source identifier.

The logical source identifier can be composed of location information such as "Terminal Room #1, Terminal #3" (designated as "TR1T3") or a building and office number designation (such as "BDG2RM32"). Thus, they can be oriented more the way a security officer views them than the way the operating system does.

It is often necessary to group logical source identifiers because a number of input devices are to be treated the same way in terms of access control; therefore, ACF2 provides the ability to define source groups. This is done by defining entries in ACF2's Information Storage Database under type code SGP with the Source Group Names as the key. The items that are associated with each entry are considered part of the group and can be either logical source identifiers or other source group names. A cross reference table between logical source identifiers and source group names is built at ACF2 Initialization time and must be rebuilt dynamically via a console operator command (F ACF2,NEWXREF) to take effect while ACF2 is active.

To limit a Logonid to specific input sources, set the SOURCE field of the Logonid record to either a logical source identifier or a source group name. To make the input source part of the environment specification for resource and/or access rules, use the SOURCE parameter.

Input source identifiers are inherited by all spun jobs, such as jobs submitted via the TSO SUBMIT command or from a normal batch program submitting a job directly to JES. Multi-user interactive systems must use the //*JOBFROM control card to specify the input source of the user.

Examples of source control might be to limit certain Logonids' access to the system only if they come from a specific set of remote batch terminals or a set of interactive terminals.

OID Card Support

Certain ACF2 users can be required to insert an Operator Identification (OID) card at logon time. These users insert their OID cards after entering their Logonids and passwords.

OID card support is useful in situations where a user must work under the supervision of another user. This support provides such benefits as:

* A supervisor can maintain possession of an OID card so that a user can log on only in this supervisor's presence.

* A user can be physically restricted to logging on only to terminals with an OID card reader.

Each user who must log on with an OID card is identified, by Logonid, in an entry record called an OID record. OID records are distinguished from other entry records by a type code of OID, as opposed to SRC for input source records and SGP for source group records.

In addition to the Logonid of the user, an OID record also contains the character string from the OID card. This string is one-way encrypted into the record by means of the same algorithm used to encrypt the password.

SCOPE RECORDS

Under ACF2, each user is granted certain privileges by means of a Logonid record. A scope record can define restrictions on the access granted by these privileges. Such restrictions can limit the user's ability to access:

* Datasets and the access rule sets for those datasets

* Logonid records

* Records on the Infostorage database, which include: generalized resource rule sets, entry records, scope records, shift/zone records, and GSO records.

A scope record does not become effective for a given user until the name of the scope record is specified in that user's Logonid record (in the SCPLIST field).

Alternatively, these various scopes can be specified through one of the following fields of the Logonid record:

Field Name	Description of Scope
DSNSCOPE	Datasets and corresponding access rule sets
UIDSCOPE	Logonid records (identified by UID string)
LIDSCOPE	Logonid records (identified by Logonid)

However, all installations should convert to using the SCPLIST field, since these fields will be dropped in a future release of acf2/MVS. During this conversion, the scopes specified through the SCPLIST field will override those specified in the DSNSCOPE, UIDSCOPE, and LIDSCOPE fields.

A restricted security officer or account manager has limited power as applies to performing specific ACF2 functions. Onl an account manager having no LIDSCOPE, UIDSCOPE, or SCPLIST is considered unrestricted, as is a security officer having no DSNSCOPE or SCPLIST. The SCPLIST is the overriding determinent of the user's restriction.

Note that the presence of any SCPLIST value, regardless of the actual entries which are to be found in the related scope list itself, is translated as indicating that the user is restricted. Also, if a user has SCPLIST specified, all desired matching for that user must be specified. Any scope definition (LID, UID, DSN, and/or INF) not present implies no matching of records.

A security officer, for example, whose scope list contains only entries governing Logonid records would be considered restricted, regardless of the fact that his scope list contains no entries regarding access rules. Additionally, a Logonid record must match both LID and UID SCPLIST entries in order for a user to modify the record.

The scope list record and its specific scope entries are created by use of the TSO ACF command and its subcommands under SCOPE mode in a way similar to the creation of other ACF2 records. The INSERT, LIST, CHANGE, and DELETE subcommands of the ACF command are used for the creation and maintenance of scope list records.

A scope list name is from one to eight characters long and is specified in the SCPLIST field of the Logonid record. This serves as a "pointer" to the associated record on the Information Storage database. A scope list entry is from one to forty-four characters long and contains specific information depending on the type of record to which the scope will apply. For example, a scope entry limiting a user's authority over Logonid records may contain the actual Logonid or UID string or masked Logonid or UID string.

Multiple entries of any type may be specified within a scope list. For example, a user's scope may be defined to include multiple Logonid records via multiple LIDSCOPE-type entries such as PAY***, ACT***, AUD***. Thus, the user has the authority over Payroll, Accounting, and Audit department Logonid records. Using the Logonid LIDSCOPE field would allow for the specification of only one of these Logonid groups.

In a decentralized environment, for example, the Director of the Financial Division for Company ABC may have the ACCOUNT privilege and may control the creation and maintenance of Logonid records for his division but only his division.

If the UID string for company ABC is defined as company (ABC), division (e.g., FIN), and Logonid, a scope list record for the Finance director would specify UID(ABCFIN-) to limit him to only the "FINancial" group of Logonid records. In a similar fashion, a SECURITY user can be limited to particular groups of dataset names for the purpose of rule writing.

SHIFT CONTROLS

ACF2 allows the specification of time/shift controls via the SHIFT field of the Logonid record and entries defining the specific shift on the Information Storage database. A user may be allowed system access, data access, and resource access for a specific time of day, days of the week, or actual dates (specified as mm/dd/yy, yy/mm/dd, or dd/mm/yy, depending on installation option).

Shift Records. The SHIFT field in the Logonid record contains the name of the shift record on the Information Storage database. Fields within the shift record itself allow for specific times and dates to be applied to allow or disallow access.

The DAYS field may contain days of the week (MO,TU,WE, etc.) and/or dates (10/10/81, 12/20/81) on which a user is allowed access. Additionally, the NDAYS parameter is used to indicate specific days or dates not allowed. For example, a shift record can be named NORMAL and be defined as DAYS(MO,TU,WE,TH,FR), allowing a user access during the regular work week. NDAYS can have 12/25/81 specified to disallow access on Christmas regardless of whether or not it falls on a weekday. Similarly, the TIME and NTIME fields can specify the allowed and prevented times (in five minute increments) during which access may occur.

For example, the same shift record NORMAL may specify TIME(0800-1700) to allow access during the standard business day, with NTIME(1200-1300) specified to disallow access during the lunch period. Or the same time frame could be defined using only the TIME field by specifying TIME(0800-1200,1300-1700). Note that DAYS must be specified in order to specify TIME or NTIME.

Shift records are located on the Information Storage database under record class 'T' and type code 'SFT'. These records are created by use of the TSO ACF subcommands.

The LOGSHIFT privilege field of the Logonid record allows a user system access outside the SHIFT specified in the Logonid record. If the LOGSHIFT field is set, access to the system is allowed but logged. If LOGSHIFT is on but no shift name has been specified, the LOGSHIFT field is ignored. A programmer whose shift is defined as NORMAL, for example, and who must access the system on a weekend, could be allowed access by use of the LOGSHIFT privilege. The LOGSHIFT privilege applies to a user's access to the system and not to shift controls within generalized resource rules or access rules.

Access rules may contain the SHIFT parameter to indicate the shift during which a particular rule applies. Thus, access to individual datasets can be limited to particular days and times. Resources are also protected by use of shift records. For example, specific IMS transactions within a banking environment can be governed by a SHIFT record designed to limit the hours of access to only morning or only afternoon.

Zone Records. Zone records located on the Infostorage database allow the specification of local time zones. These records contain an offset in hours and minutes to the local CPU time (i.e., ±hhmm). The ZONE field of the Logonid record contains the name of the zone record on the database that applies to that user.

For example, an installation in Chicago (running under local time) might define a ZONE called "EST" for Eastern time. Since Eastern time is one hour ahead of Chicago time, the zone record "EST" would contain an adjustment of (+0100) to "offset" the local time ahead by one hour. A zone record for Pacific time named "PST" might also be defined, and would contain an adjustment of (-0200), to "offset" the local time back by two hours.

Zone records have the record class of 'T' and type code 'ZON'. The ACF command and subcommands are used to create and maintain these records.

GSO RECORDS

Global System Option (GSO) records define most of the ACF2 system-wide options. These options include various operating parameters and installation-defined exits. For a description of each set of these options, see the chapter on system-wide options.

TSO ENHANCEMENTS

ELIMINATING THE UADS DATASET

ACF2 has the ability to optionally bypass the use of the SYS1.UADS dataset. This has the advantage that users need only be defined in one place. The disadvantage is that certain portions of TSO processing are modified and the terminal user may have to alter his procedures. The following areas need to be considered:

* If an installation decides not to use UADS, ACF2 provides a TSO fullscreen logon feature. This fullscreen display shows the values under which a user will log on. The user can change these values or retain them from session to session as the installation permits.

* If an installation decides not to use UADS, ACF2 can maintain a default TSO account number and procedure name for each TSO user. Furthermore, the installation can decide which users, if any, can specify the account number and procedure name at logon time.

* If an installation decides to use UADS, ACF2 provides for a default UADS password for each TSO user, which automatically invokes the appropriate procedure at logon time. The installation can decide which users, if any, can change their default password at logon time.

* If an installation decides not to use UADS, ACF2 can maintain the PROFILE information for each user in an individual Logonid record. Modifications via the PROFILE command are not reflected in the Logonid record. Therefore, the ACF command must be used for changes to PROFILE-related values stored in a user's Logonid record.

* In non-UADS mode, ACF2 provides for TSO/E Broadcast Performance (only to those installations using the expanded LIDREC). The following section describes these support facilities.

TSO/E BROADCAST PERFORMANCE SUPPORT

A TSO user receives two types of messages from TSO Broadcast during logon processing. These are NOTICES, which contain general messages and are sent to all users at logon, and MAIL messages, which are sent only to specific users.

NOTICES are stored as separate records on the TSO BRODCAST dataset, and one I/O operation is needed to send each NOTICE message. For MAIL messages, one mail-index-record (MIR) is maintained in the BRODCAST dataset for every nine TSO users. This record contains the MAIL message locations for this group of users. The TSO/E enhanced support reduces logon processing time by providing a direct pointer to each user's mail-index-record in the SYS1.UADS dataset, thus eliminating the previous TSO sequential search of these records. Direct access to the user's MAIL section of the BRODCAST dataset is then available during logon processing.

Under ACF2, with UADS=NO specified and if utilizing the expanded LIDREC, the TSORBA field (TSO Relative Block Address) of the Logonid record may be used to specify the MIR pointer for each TSO/E user. The TSORBA field is updated with the MIR pointer the first time each user issues a logon command. It is also updated when the BRODCAST dataset is synchronized by the SYNCH command or the ACFBSYNC utility.

TSO LOGON PROCESSING

Under ACF2. when the fullscreen option is not being used, the TSO LOGON Command has the following syntax, regardless of whether UADS is being used or being bypassed:

```
LOGON     logonid[/password[/newpassword]]

          [ACCT(account-number)]

          [PROC(procedure-name)]

          [SIZE(integer)]

          [PERFORM(integer)]

          [INDEX(password)]

          [TIME(integer)]

          [UNIT(unit-name)]

          [MSGCLASS(class)]

          [MAIL/NOMAIL]

          [NOTICES/NONOTICES]

          [RECONNECT]
```

The following notes decsribe the logon operands under ACF2:

1. Forced Password Prompt. If the NOQLOGON system option is specified in the TSO GSO record, then ACF2 will ignore any password provided on the ^irst line and will prompt for the password. The prompt allows the password to be entered in a nondisplay area.

2. Ability to Change the Password at Logon Time. If the NOPSWALT system option is specified in the GSO PSWD record, then ACF2 will not allow a user to specify a new password (i.e., change his password) at logon time.

3. Account Number Controls. If the user does not have the ACF2 LGN-ACCT attribute, the user cannot specify ACCT at logon. If the user has the PMT-ACCT attribute, ACF2 will prompt for account number, unless the user had already specified ACCT before the prompt, regardless if UADS is being bypassed. If no account number is specified at logon and PMT-ACCT is not on for that user, ACF2 will use the user's default account number (TSOACCT in the Logonid record) or, if none present for that user, the system default account. This default is specified in the ACCOUNT field of the GSO record named TSO. If none is specified and neither a user nor a system default is available, the user will be prompted for an account number in spite of the settings of LGN-ACCT and PMT-ACCT. The account number is then validated by ACF2 against the TAC type generalized resource rules if the user has the VLD-ACCT attribute on.

   Additionally, if UADS is still being used, then normal UADS account number validation will also take place. The account number provided must match one of the account numbers present in UADS for that user (when more than one is present) or TSO will prompt for a valid account number. ACF2 processing is completed before UADS processing, so even if VLD-ACCT is on for the user ACF2 will not revalidate here if TSO reprompts.

4. Procedure Name Controls. Under ACF2, procedure name processing is similar to account number processing, described above in #3. However, the applicable fields of the Logonid record are LGN-PROC, PMT-PROC, TSOPROC, and VLD-PROC as opposed to LGN-ACCT, PMT-ACCT, TSOACCT, and VLD-ACCT. Thus, the LGN-PROC attribute allows a user to specify a procedure name at logon time. The PMT-PROC attribute allows for prompting of the procedure name if the user has not entered one. The TSORPOC field of the GSO record named TSO contains the name of the system-wide default proc. The VLD-PROC attribute will cause ACF2 to validate the proc against generalized resource rules of type code TPR. Normal TSO UADS processing occurs if the UADS system option is specified in the GSO OPTS record.

5. User's Region Controls. At logon time, any TSO user may optionally specify a SIZE operand. The integer specified in this

operand represents the number of thousand bytes (e.g. 128K), and becomes the JCL REGION parameter. If the system-wide option of NOUADS is in effect, then the TSORGN, TSOIZE, and LGN-SIZE fields of the Logonid record become active.

6. Performance Group Controls. Performance group operands are processed without change if the system-wide option of UADS is in effect. If the NOUADS option is in effect, then the LGN-PERF and TSOPERF fields of the Logonid record become active.

7. Session Time Limit Controls. The TIME operand is not valid if the system-wide option of UADS is in effect. If the NOUADS option is in effect, then the LGN-TIME and TSOTIME fields of the Logonid record become active.

8. UADS Password Controls. The INDEX operand is not valid if the system-wide option of UADS is in effect. IF the NOUADS options is in effect, then the LGN-INDX and UADSINDX fields of the Logonid record become active.

9. Unit Name Controls. The UNIT operand is not valid if the system-wide option of UADS is in effect. If the NOUADS option is in effect, then the LGN-UNIT and TSOUNIT fields of the Logonid record become active.

10. Message Class Controls. The MSGCLASS operand is not valid if the system-wide option of UADS is in effect. If the NOUADS option is in effect, then the LGN-MSG field of the Logonid record becomes active. Specification of this field allows the user to change the TSO session message class, and to direct TSO session output to another class.

11. TSO Mail Controls. The MAIL operand retains its default of MAIL (as opposed to NOMAIL) if the system-wide option of UADS is in effect. If the NOUADS option is in effect, then the MAIL field of the Logonid record determines whether a user will receive personal messages at logon time by default. The value of the MAIL operand can be overridden at logon time.

12. TSO Notices Controls. The NOTICES operand retains its default of NOTICES (as opposed to NONOTICES) if the system-wide option of UADS is in effect. If the NOUADS option is in effect, then the NOTICES field of the Logonid record determines whether a user will receive general user messages at logon time by default. The value of the NOTICES operand can be overridden at logon time.

13. Session Reconnection. The RECONNECT operand is not effected by the UADS/NOUADS option.

ACF2 provides the capability for a site to request ACF2 to terminate incomplete TSO logon sessions after a specified elapsed time. The installation may specify the number of seconds it wishes to use via the WAITIME field of the GSO record named TSO.

TSO COMMAND LIMITING

Through the ACF2 TSO command limiting function, an installation can define a subset of available TSO commands for an individual user or for the installation as a whole. This command limiting applies to TSO commands entered under READY mode or under ISPF (Function Code 6 or the ISPF TSO command).

This feature is activated for an individual through the TSOCMDS field of the Logonid record. This feature is activated on a system-wide basis through the CMDLIST field of the GSO record named TSO. If the system-wide default is not specified and the TSOCMDS field value is blank, then TSO will operate without ACF2 command limiting. If ACF2 command limiting is used, the following fields within each user's Logonid record can be used:

TSOCMDS(module-name)
: Specifies the name of a module which contains a list of commands valid for this user. See ACF$CMDS in SYS1.ACFMAC for a sample list.

ALLCMDS
: Permission for that user to bypass command limiting. This bypassing can be accomplished by prefixing the command name with the character specified in the BYPASS field of the GSO TSO record.

CMD-LONG
: Requires that user to enter the complete command name or alias name if a command limiting list is active for the user.

Each TSOCMDS field or CMDLIST value specifies a load module in the Link Pack Area or system Linklist libraries that is loaded at logon time and stays resident throughout the life of the TSO session. This module is specified via ACF2 macros, and is assembled and link edited into one of these libraries. Basically, the input to the assembler consists of a list of valid command names and aliases.

The advantage of the command list, besides the ability to limit available commands, is that the Terminal Monitor Program will not automatically issue an ATTACH for the command name. Thus, invalid command names and implicit CLISTs (those defined via the SYSPROC DD card in the logon procedure) will not cause a search of directories of all system linklist libraries, which can save a considerable amount of channel time. IBM recommends that implicit CLISTs be entered with a percent sign (%) preceding the name to avoid this search.

Since ACF2 command lists will provide a significant performance improvement, they should also be used even for users with "unlimited" privileges. The ALLCMDS attribute can be assigned to these "unlimited" users, and, when necessary, they can bypass ACF2 command limiting by using an optional escape character. This character is specified through the BYPASS field of the GSO TSO record.

TSO COMMAND SMF RECORDS

Optionally, the ACF2 system can produce SMF records containing the information for each TSO command or CLIST issued in READY mode or under ISPF.

These records appear in the TSO Command Statistics Log (ACFRPTCR), explained in the acf2/MVS Utilities Manual.

SUBMIT COMMAND PROCESSING

The ACF2 PASSWORD control card need not be specified for a TSO submitted job if the Logonid on the ACF2 LOGONID control card is to be the same as the Logonid of the TSO user. This avoids having to specify the password during job submission, or worse yet, having it in the CNTL datasets.

This situation is applicable to jobs submitted by batch jobs in general. If there is no Logonid in the input control cards, the Logonid of the submitting job will automatically be inherited by the submitted job. For exceptions, see the sections on production jobs in this manual.

PATH CONTROL DESCRIPTIONS

For commands that will attach other commands or CALL user programs, ACF2 supplies definition macros that describe the TCB/RB structures. Properly specified, those "structure" macros will give ACF2 the information it needs to validate path control access to datasets. Supplied with ACF2 are structure descriptions for SPF1, SPF2, VSAPL, EDIT, QED and an installation written CALL command replacement.

BLDL INTERFACE

This interface allows a standard BLDL to be replaced by an ACF2 SVC call. (Replacement can be accomplished via superzap.) ACF2 will validate the Command Name against the Command Table specified from the user. If the command is authorized, an actual BLDL will be done to obtain the correct data for the caller. If the command is not authorized, a not-found indication will be returned (usually indicating that the command is a CLIST). This allows ACF2 command limiting to be easily interfaced with existing TSO commands that attach other commands.

If the original BLDL was issued only to determine the existence of a command and the BLDL data returned was not used (such is the case with SPF), then the actual BLDL can be eliminated. By setting the BLDL length to eight, only command validation will take place, without the overhead of the actual BLDL.

EXTENDABILITY TO NON-TSO INTERACTIVE SYSTEMS

A non-TSO ınteractive system which is APF-authorized or runs in supervisor key may access all the facilities of the ACF2 system. Briefly these are:

SYSTEM ACCESS VALIDATION

There are control block descriptor macros that will define the control blocks needed to validate a user onto the system. Textual error messages are returned so that no error code interpretation need be done by the interactive subsystem. Also, additional fields may be defined in the Logonid record to contain information needed by the subsystem.

USER CONTROL BLOCK

The ACF2 User Control Block is necessary to do dataset access validation. All fields that need to be filled in this control block are available after the user has been validated onto the system.

DATASET ACCESS VALIDATION

Dataset access validation is done by the subsystem via an SVC call pointing to a parameter list. This list includes the pointer to the User Control Block and the dataset name. Any logging necessary will be done by the SVC interface. This interface will also provide a return code and a textual message indicating whether access is to be allowed. It is the responsibility of the subsystem to issue the call in the appropriate places and take the correct action on return.

LOGONID RECORD DISPLAY AND ALTERATION

Logonid record display and alteration is done by the subsystem. Such a display or alteration is done via an SVC call which points to a parameter list and the User Control Block. Subroutines are provided to decode the user input operands and create the internal format text used for communicating with the central facility, and for converting the internal format text into the display format used by the LIST subcommand of the TSO ACF command.

JOB SUBMISSION

A new JES control card is available when an authorized interactive system wishes to submit a job on behalf of a user. The interactive system becomes authorized by issuing the ACFSET macro with the JOBFROM=YES operand, or by using the JOBFROM attribute in the Logonid record and the following card image immediately after the JOB card:

```
//*JOBFROM  logonid/source
```

If the submitted job has no Logonid specified, the one from the JOBFROM card will be used. If the supplied Logonid matched the one from the JOBFROM card, no password will be required. Finally, the supplied source will be propagated with the job to be used for data access control.

CLEANUP

The referenced access rules and resource rules are chained off the User Control Block in the subsystem's address space. When the user logs off the subsystem, the subsystem must issue a special cleanup call to the ACF2 SVC to release space occupied by these rules and control blocks.

PERFORMANCE

All ACF2 VSAM operations except for those done for JES are normally done in Step-Must-Complete (SMC) status. This means that all TCBs in the address space are quiesced except for the one that issued the ACF2 SVC. This method may have some negative performance implications for some heavily used interactive systems.

To bypass the SMC status, the ACFSET macro (SMC=NO operand) may be issued. However, if this is done, it is the interactive system's responsibility to insure that the address space is not terminated while an ACF2 VSAM operation is in progress. If the address space does terminate with ACF2 VSAM operations in progress, it may leave control blocks in an unknown state which may require a re-IPL to fix.

Since ACF2 SVC routines may generate SMF records, all multi-task systems that also generate SMF records (explicitly or implicitly) from other tasks must also run with the SMC=NO operand in effect.

SYSTEM AUTHORIZATION FACILITY (SAF)

ACF2 provides appropriate interface modules for integration with all products that make use of IBM's System Authorization Facility (SAF). For complete information about the provided SAF interface, see the acf2/MVS System Programmer's Guide. With this interface, ACF2 protection for other products that issue calls to SAF can easily be integrated.

BACKUP AND RECOVERY

BACKUP

Since ACF2 datasets are standard VSAM key-sequenced clusters, full support is available from the Access Methods Services utility (AMS) which is available for the backup, movement, and general support of VSAM datasets. However, ACF2 also contains some internal backup and recovery facilities.

The ACF2 database can be backed up daily at a prespecified time and/or on operator command. The three VSAM clusters of the database are initially copied to a scratch dataset. Updates to the clusters are inhibited during this operation.

The clusters are then copied from the scratch file to permanent non-VSAM datasets. Additional backup can be provided by copying these sequential datasets to generation dataset groups or by using them to establish an alternate set of ACF2 VSAM clusters for a "quick" restart procedure.

RECOVERY

As each record in any of the VSAM datasets is modified, a copy of the new record is written via SMF. A utility (ACFRECVR) is provided which utilizes recent backup copies of VSAM clusters and SMF records to forward-recover the clusters into an up-to-date status. See also the section on system access with ACF2 not active in the acf2/MVS Administrator's Guide.

SYSTEM-WIDE OPTIONS

Most ACF2 system-wide options are defined in GSO records, stored on the Infostorage database. Through these records, a security administrator can conveniently and dynamically make changes to these system-wide options.

Other system wide options, of a more technical nature, are defined in the macros of the ACF2 Field Definition Record (ACFFDR). See the acf2/MVS System Programmer's Guide for further information on these options.

SYSTEM OPTIONS IN GSO RECORDS

GSO records contain fields that each define a particular system-wide option. Options defined in GSO records can be dynamically changed through ACF subcommands. No reassembly is involved. The following list describes each of these options, which are further explained in the acf2/MVS Administrator's Guide:

Backup Definition Entry - GSO BACKUP Record
This defines the time of day when a sequential copy is to be made of the VSAM datasets that compose the ACF2 database. An optional string may be specified that will be issued as a command to the operating system (via SVC 34). For example, this command may be used to start a task or submit a job that will copy the contents of the ACF2 backup files to tape if further backup is desired.

Bypass Label Programs - GSO BLPPGM Record
This defines a set of environments (program and library names) for which ACF2 will allow bypass label processing on tape even if the user himself does not have authorization. This could be used for tape scanning and listing utilities or other "safe" programs in controlled libraries.

Local Exit Specification - GSO EXITS Record
This record specifies the modules name for each installation-written exit. These exits are described in the acf2/MVS System Programmer's Guide.

LINKLIST Extension - GSO LINKLST Record

When ACF2 determines the library from which a program came, this record defines the partitioned datasets or libraries that ACF2 will consider as part of the system LINKLIST (SYS1.LINKLIB). LINKLST provides flexibility in controlling a program's access to data and resources through JCL JOBLIB and STEPLIB statements.

Logged Programs - GSO LOGPGM MACRO

This is a list of program names that will cause an ACF2 logging record to be produced for each dataset that they access. Thus, an audit trail is provided for some programs that should be used on a limited basis because of installation standards.

Maintenance Programs and Logonids - GSO MAINT Record

This defines a set of environments in which ACF2 validation is to be bypassed, and is normally used for DASD maintenance programs. Up to 100 combinations of Logonid, program, and program library name may be specified. If the Logonid is marked NON-CNCL (non-cancellable by ACF2) or MAINT, and the environment is correct, ACF2 will bypass all further security checking. This avoids rule access overhead and the great numbers of SMF records being generated by these highly active programs.

Network Job Entry Validation - GSO NJE Record

This macro specifies the node at which ACF2 job validation is to be performed (i.e., origin or execution node) as applies to Network Job Entry Subsystems (NJE). When used in conjunction with the XDES password encryption method, this feature also ensures that clear text passwords are never transmitted over an NJE system.

Mode of ACF2 - GSO OPTS Record

The mode of the ACF2 system is specified in this macro. The allowable modes are QUIET, LOG, WARN, ABORT, and RULE. These modes allow security to be implemented in phases within an installation. The RULE mode allows for the transition of individual dataset rule modes, while the remaining modes determine system-wide access modes.

Protected Programs Specification - GSO PPGM Record

This record is used to define the set of programs that are authorized to bypass the system integrity features within the operating system. These programs can be executed only by unrestricted users with the SECURITY attribute or users (Logonids) with the NON-CNCL attribute.

Password Support Options - GSO PSWD Record

This record is used to specify various password options and controls. These options and controls include the password encryption method (R221 or XDES), maximum allowable password attempts in a session and in a day, minimum password length, and other options.

Resident Resource Directories - GSO RESDIR Record

This is a list of generalized resource types for which a resident directory is to be built at ACF2 initialization. Directories allow the resource names to be masks. Optionally, all of the resource rule sets for the type may also be made resident.

Resident Rules - GSO RESRULE Record

This is a list of access rule sets to be brought into storage at ACF2 initialization time. References to these indexes will not result in any I/O, not even for the first reference in a job. Resident rules would be used for frequently accessed indexes such as SYS1 where almost every job or session would reference some dataset under this index. The disadvantage of the resident rule is that the resident copy of the rules will not be updated when a new set of rules is compiled and changes will therefore not take place until the next IPL or until ACF2 is instructed to reload an access rule set by the console operator.

Resident Volumes Definition - GSO RESVOLS Record

This defines the "resident" set of volumes. Datasets on these volumes will be protected by dataset name. Also, the libraries used for program paths must reside on these volumes. These volumes do not necessarily have to be resident to the operating system at all times. The definition is a list of volume serial number masks where an asterisk (*) may be used to specify that any character, including a blank, in the actual volser may be present.

SAF Safelist - GSO SAFSAFE Record

This record defines the subsystem, control points, and classes that the System Authorization Facility (SAF) will consider to be safe and to not require ACF2 validation.

Secured Volumes Definition - GSO SECVOLS Record

This record defines the "secured" set of volumes. These volumes will be protected on a volume basis; for references to datasets on these volumes, the pseudo dataset name of @volser.VOLUME or VOLUME.@volser will be used to determine access. As in the "resident" volume definition, masks created by using asterisks may be specified.

Time Sharing Options and Defaults - GSO TSO Record

This record defines controls and operands to be in effect at TSO logon time and during a user's TSO session.

ASCII CRT Clear String - GSO TSOCRT Record

This record defines the clear string for ASCII CRT devices. This clear string is to ensure the secrecy of users' passwords at system access time.

User Logon Operand Keyword - GSO TSOKEYS Record

This record defines up to 256 keywords (of up to eight characters) that the installation wants to recognize as valid at TSO logon time. Although ACF2 does not validate user-defined keywords, it does pass them to other installation routines.

TWX X-Out String - GSO TSOTWX Record

This record defines the x-out string for ASCII and TWX devices. This x-out string is to ensure the secrecy of users' passwords at system access time.

2741 X-Out Mask - GSO TSO2741 Record

This record defines the x-out string from 2741 devices.

System WARN Mode Message - GSO WARN Record

This record specifies the text of a warning message to be displayed at the terminal or on the job log when the ACF2 system is in WARN mode and a security violation has taken place.

SYSTEM OPTIONS IN THE ACFFDR

The ACF2 Field Definition Record (ACFFDR) is an assembly of a module which defines the fields in the Logonid record and specifies the other options of the system. Detailed information on how to specify these options locally are included in the acf2/MVS System Programmer's Guide. The following list provides general information on the macros making up the ACFFDR:

Create Field Definition Entry - @CFDE Macro

There exists one Field Definition Entry (@CFDE macro in the ACFFDR) for each field that may be displayed or modified by the INSERT, CHANGE, or LIST subcommands of the ACF command as well as fields updated only by ACF2.

Create Supervisor Call- @CSVC Macro

This macro specifies the two SVCs required by the ACF2 system. These SVCs are enabled, non-lock holding, and either type three or four.

Dynamic Dataset Allocation - @DDSN Macro

This macro specifies the groups of names of the datasets to be dynamically allocated for use by ACF2 and for backup of the ACF2 clusters.

ACF2 Field Definition Record (ACFFDR) Generation - @GENFDR Macro
This macro creates the ACFFDR CSECT.

Group Definition Entry - @GROUP Macro This macro defines the group names for the formatted display of the Logonid record by the LIST subcommand.

Heading Definition Entry - @HEADER Macro
This entry defines the format of the first line of output when used by the LIST subcommand in VERBOSE mode and consists of a list of field names. The first field specified will be displayed in the group heading area of the display on the left of the line. The remaining fields will be displayed without the field names in the order specified on the first set of lines of output. Each field will be separated by a blank.

Logonid Record Compression - @MLID Macro
This macro specifies a compression algorithm for Logonid record. Such an algorithm is useful in a Multiple User, Single Address Space System (MUSASS) environment to conserve space. Logonid record compression eliminates, from the resident copy of the Logonid record, information that will be not be used by that MUSASS.

Multiple User Single Address Space System - @MUSASS Macro
This macro defines special processing to be performed by ACF2 on behalf of a Multiple User, Single Address Space System (MUSASS), such as CICS. The acf2/MVS System Programmer's Guide contains an explanation of the MUSASS environment.

DSECT Map Initialization - @SETUP Macro
This macro expands the DSECTs and the necessary equates for the ACFFDR.

System Management Facility (SMF) Record Numbers - @SMF Macro
This macro specifies the default SMF record number(s) to be used for ACF2 SMF records. These records are processed by the ACF2 report generators. For further information on SMF record number(s), see the acf2/MVS Utilities Manual and the acf2/MVS System Programmer's Guide.

User Identification String Definition Entry - @UID Macro
This macro defines the combination of fields, or portions of fields, from the Logonid record that will comprise the User Identification (UID) string. The UID string identifies users for the purpose of access and generalized resource rules. It is important that this definition be given a great deal of thought prior to installation since a good definition can give added flexibility to the access and generalized resource rules. In the ACF2-supplied ACFFDR, the UID string is defined as the LID field only. Installation-defined fields, such as department, job responsibility, product responsibility, etc. can be defined in the UID string.

Field Zeroing Definition Entry - @ZEROFLD Macro

When a new Logonid record is created from an existing, model record, all current values of the fields in that record may not be valid for the new record. This macro specifies which fields will not be copied from the model record to the new record.

UTILITIES

REPORT GENERATORS

ACF2 produces records in the SMF datasets under the following circumstances:

1. Each time a logon, sign-on, or JES reader (system access) attempt is made that is rejected for any reason. Each time system access is allowed but logged outside a specified shift by a LOGSHIFT privileged user.

2. Each time a restricted Logonid is used. (Logging will occur only on full Logonid validation; spun-off jobs are not logged.) Each time a Logonid with the TRACE attribute is used.

3. Each time a dataset or resource access attempt is made but is aborted by ACF2 security routines.

4. Each time a successful dataset or resource access attempt is made but logged upon the request of a system option (such as LOGPGM) or a rule.

5. Each addition, modification, or deletion made to the Logonid records via the TSO ACF command interface.

6. Each time an access rule, generalized resource rule set, or an information storage record is inserted, replaced, or deleted. (Either the new record or an indication of the deletion is logged.)

7. Each time a TSO command is issued because the TSO Command SMF Record option was selected or the user had the TSO-TRC keyword set in his Logonid record. TSO command statistics are buffered within a single SMF record and are written only when the statistics buffer has been filled.

8. Each record processed by the ACF2 recovery utility (with action taken indicated).

9. Each time an console operator ACF2 MODIFY (F ACF2) command is issued, each time ACF2 is stopped or started, and whenever SMF data is lost.

It is only by careful scanning of the reports produced by the ACF2 system that the computer system can remain secure. If there is no easy way to bypass the security of ACF2 or the integrity of MVS, look for a mistake in the specification of an access rule or attempt to guess the password of an authorized user.

The ACF2 report generators will produce information which will highlight these events, but someone must be looking at the reports to take appropriate action. The MONITOR and TRACE controls on a suspected violator's Logonid are methods of obtaining further evidence.

The following utilities and report generators are supplied with ACF2. They format and edit information from the produced SMF records and the ACF2 databases:

ACFRECVR - performs the recovery of the ACF2 VSAM databases.

ACFRPTCR - provides tracing and statistics for TSO command execution, which includes CLISTs and TSO commands executed within CLISTs.

ACFRPTDS - formats the dataset violation, logging, and trace records. It also formats the program violation and logging records.

ACFRPTEL - provides a report of modications made to generalized resource rule sets and ACF2 records on the Infostorage database. (These records include entry, scope, shift/zone, and GSO records.) This report can show before and after values for changed fields of ACF2 records.

ACFRPTIX - selects and edits information about high level indexes and their rules (including changes over an historical period).

ACFRPTLL - provides a report of modications made to Logonid records, including before and after values for changed fields.

ACFRPTNV - provides loggings of each ACF2 START, STOP (PURGE), and MODIFY operator command issued. This report generator also produces loggings of system IPLs and possible losses of SMF data.

ACFRPTPP - selects the ACF2-produced records from the SMF datasets and puts them into separate files for further processing by other report generators.

ACFRPTPW - provides a listing of password violations as well as LOGSHIFT system accesses.

ACFRPTRL - provides a report of modifications made to access rule sets.

ACFRPTRV - formats the resource violation, logging, and trace records which pertain to generalized resources.

ACFRPTRX - provides cross reference information linking users with associated access and generalized resource rules. This report is organized and ordered by Logonid.

ACFRPTSL - selects and edits information on users (Logonid records) per selection criteria.

ACFRPTXR - provides cross reference information tying users to access rules and generalized resource rules, such as listing which Logonids have access to which datasets. This report is organized and ordered by dataset or resource name.

Additional detailed information about the ACF2 reports and sample report output can be found in the acf2/MVS Utilities Manual.

BATCH PROGRAMS

Brief descriptions of various batch utilities included with ACF2 are provided below. Installations with TSO/SPF can optionally install a complete set of ACF2/SPF screens, which will perform ACF2 reports, utilities, and rule writing functions. A full set of online SPF tutorials are also supplied. For more detailed information on any of the utilities, see the acf2/MVS Utilities Manual.

ACFBATCH
Allows for execution, in a batch environment, of a sequence of ACF subcommands.

ACFBCOMP
Compiles access or generalized resource rule sets. The text of the rule sets can be in the jobstream itself or in a file.

ACFBDCMP
Decompiles the access or generalized resource rule set specified as a parameter.

ACFBSYNC
Synchronizes the Logonid database with the BRODCAST dataset. Alternatively, the ACF subcommand SYNCH produces a similar effect.

ACFNRULE
Adds single rules, and then recompiles the access or generalized resource rule set specified by the parameters. It can also delete rules that contain a particular character string. This batch program can also be executed as a TSO command.

ACFERASE
Removes data from a direct-access dataset or erase a tape volume. This batch program produces the same effect as the TSO command ACFDEL.

ACFRECVR
Is used for recovering the ACF2 databases by merging SMF update records with the backup ACF2 databases, and then building VSAM clusters.

JOBCOPY
Allows for submission of production and other special types of job streams the need to run under a Logonid other than that of the submitter. This batch program serves the same function as the TSO ACFSUB command.

TSO COMMANDS

In addition to the multi-function ACF command, the following TSO commands are supplied with ACF2:

ACFCOMP
Compiles access or generalized resource rule sets. Rule set text can be entered directly from the terminal or be input from a file. The ACF and ACFCOMP commands may be used in batch under the Terminal Monitor Program (TMP) IKJEFT01.

ACFDEL
Removes data from a direct-access dataset or erase a tape volume. This TSO command produces the same effect as the batch program ACFERASE.

ACFNRULE
Adds single rules, and then recompiles the access or generalized resource rule set specified by the parameters. It can also delete rules that contain a particular character string. This TSO command can also be executed as a batch program.

ACFSUB
Allows for submission of production and other special types of job streams the need to run under a Logonid other than that of the submitter. This TSO command serves the same function as the batch program JOBCOPY.

INSTALLATION EXITS

ACF2 installation exits allow for special processing and support that is normally not done by the ACF2 system. This processing and support is done through user-written programs. To define an exit, an installation must specify the user module name in the GSO EXITS record. The exit must also be linked into SYS1.LPALIB. The acf2/MVS Administrator's Guide explains the GSO EXITS record. The acf2/MVS System Programmer's Guide supplements the descriptions of the exits listed below:

LOGON AND PASSWORD VALIDATION EXITS

LGNTERM: Logon Terminal Exit
This exits allows an installation to identify different kinds of terminals, such as 2741, TWX, and ASCII CRT devices. This exit is entered in the problem state.

LGNIXIT: Logon Pre-Validation Exit
After the ACF2 TSO Logon Processor has obtained the Logonid and password and built the validation parameter list, this exit is invoked. It can allow the Logonid or password to be modified, or the logon attempt to be rejected.

LGNPARM: Logon Parameter Exit
This exit allows an installation to examine and alter logon operands during logon validation. Alterations may be to such values as the TSO account number and procedure name.

LGNPXIT: Logon Post-Validation Exit
This exit is invoked after the ACF2 TSO Logon Processor has successfully validated a Logonid and password, and has built all the necessary control blocks to pass back to TSO. Through this exit, information can be passed to TSO, or the logon attempt can be rejected.

USRFLD: Logon Pre-prompt User Exit
This exit allows an installation to perform further Logon Work Area (LWA) processing. This exit is entered in the problem state.

NEWPXIT: New Password Exit
This exit is entered when the password is altered at system entry time or via the ACF command. Through this exit, an installation can implement special criteria for validating a new password.

EXPPXIT: Expired Password Exit

This exit is invoked when a Logonid with an expired password is used. It can deny system access, allow system access, or assign a new password.

DATASET/PROGRAM VALIDATION EXITS

VLDEXIT: Dataset Pre-Validation Exit

This exit is taken prior to any access rule validation. This exit can make a determination of whether to deny access, allow access with or without logging, or allow ACF2 to control the access. In addition, this exit can alter the dataset name that ACF2 will use for validation, or alter the search key that ACF2 will use to obtain the rule set. As discussed in the acf2/MVS Administrator's Guide, the NEXTKEY parameter of an access rule can serve this same function.

DSNGEN: Pseudo-Dataset Name Generator Exit

This exit is taken for each access to a volume that is not in the "resident" or "secured" lists. This exit can make a determination of whether to deny access, allow access with or without a logging, or allow ACF2 to control access based on a dataset name generated by the exit. In addition, this exit can interface with a tape management system to create a dataset name or to validate the correct dataset name to be used.

DSNPOST: Dataset Post-Validation Exit

This exit is taken after ACF2 has determined whether an attempted dataset access is to be allowed. This exit overrides the Dataset Security Violation Exit (VIOEXIT).

VIOEXIT: Dataset Security Violation Exit

This exit is taken just before ACF2 is to abort an access request. Through this exit, an installation can implement specific actions. For example, the system can notify the correct personnel that a security violation has occurred. In addition, the exit can determine whether the access should be allowed, and can indicate its recommendation on its return to ACF2.

ACCESS RULES DATABASE EXITS

RULEPRE: Access Rule Pre-Processing Exit

This exit enables an installation to apply its own control in retrieving access rule records. This exit is taken before a read of the Access Rules database is performed. Through this exit, an installation can modify the exit parameter list, and refuse or allow the read when the request is completed.

RULEPST: Access Rule Post-Processing Exit

This exit allows an installation to apply its own control in the storage of access rule records. This exit is taken before a write to the Access Rules database is performed. Through this exit, the exit parameter list can be modified, and the write attempt can be allowed or rejected when the request is completed.

GENERALIZED

RSCXIT1: Resource Pre-Validation Exit

This exit can force ACF2 to allow access, allow and log access, deny access, or modify the resource name or type to be used in validation. rameter

RSCXIT2: Resource Post-Validation Exit

This exit receives ACF2's access recommendation and can modify it.

INFOSTORAGE DATABASE VALIDATION EXITS

INFOPRE: Infostorage Pre-Processing Exit

This exit allows an installation to apply its own control in the retrieval of Infostorage database records. The exit is taken before the read of the database is performed, allowing the installation to modify the exit parameter list. In addition, this exit can refuse or allow the read when the request is completed.

INFOPST: Infostorage Post-Processing Exit

This exit enables an installation to apply its own control in the storage of Infostorage database records. The exit is taken before the write to the database is performed, allowing the installation to modify the parameter list. In addition, this exit can refuse or allow the write when the request is completed.

OTHER EXITS AND EXIT-RELATED INTERFACE

ACF6OSNU: SEND Command Interface

This interface allows an installation exit to process the SN subcommand, the ACF2 form of the TSO SEND command. Each time the ACF SN subcommand is issued, this interface is invoked.

SRCXIT: Source Name Modification Exit

This exit is called during Logonid validation and can modify the physical input source name that ACF2 will translate into a logical name. For an explanation of input source names, refer to the discussion on input source support in the chapter on Information Storage.

STCXIT: Started Task Validation Exit

This exit is taken after ACF2 validates a dataset access request by a started task. Through this exit, an installation can determine whether the task should be aborted, can alter the Logonid to be used for validation, or can set the default Logonid for the verification call. Use of this exit requires that the STC option be specified in the GSO OPTS record, as explained in the acf2/MVS Administrator's Guide.

SVCIXIT: Supervisor Call Initialization Exit

This exit receives control before ACF2 Supervisor Call processing begins. This exit can alter the address of the ACUCB which ACF2 uses to validate access requests and is particularly useful within the MUSASS environment.

ACF2 INSTALLATION AND MAINTENANCE

Where possible, ACF2 has been designed to avoid the impact of IBM PTFs. This design is to accommodate most users, who want to keep as up-to-date as possible with IBM maintenance.

ACF2 is distributed in the IBM System Maintenance Program (SMP) format. Thus, all maintenance to the system will be done in SMP format, similar to the Program Update Tape (PUT) process now being used for the IBM System Control Program.

For further information on the installation and maintenance of ACF2, refer to the acf2/MVS System Programmer's Guide.

INDEX