Rocket

Rocket Model 204 SirSafe

Reference Manual

July 2013
SAF-0704-RM-01

Notices

Edition

Publication date: July 2013

Book number: SAF-0704-RM-01

Product version: Rocket Model 204 SirSafe

Copyright

Trademarks

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

Note

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

Contact information

Website: www.rocketsoftware.com

Rocket Software, Inc. Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451–1468
USA
Tel: +1 781 577 4321
Fax: +1 617 630 7100

Contacting Global Technical Support

If you have current support and maintenance agreements with Rocket Software and CCA, contact Global Technical Support by email or by telephone:

Email: m204support@rocketsoftware.com

Telephone:

North America	+1 800 755 4222
United Kingdom/Europe	+44 (0) 20 8867 6153

Alternatively, you can access the Rocket Customer Portal and report a problem, download an update, or read answers to FAQs. You will be prompted to log in with the credentials supplied as part of your product maintenance agreement.

To log in to the Rocket Customer Portal, go to:

www.rocketsoftware.com/support

Contents

Proprietary Notices . . . . . **ii**

Contents . . . . . **iii**

Summary of Changes . . . . . **v**
Sirius Mods Version 6.8 v
Sirius Mods Version 6.7 v
Sirius Mods Version 6.4 v
Sirius Mods Version 6.3 vi
Sirius Mods Version 6.2 vi
Sirius Mods Version 6.0 vi
Sirius Mods Version 5.5 vii
Sirius Mods Version 5.4 vii

Chapter 1: Overview **1**

Model 204 Database Security 2
Model 204 Password Table (CCASTAT) 3
SECURE and DESECURE commands 4
Physical OPEN of Model 204 Files 5
Shared DASD Enqueueing 5

Chapter 2: Controlling Access to File and Group Passwords **9**

Model 204 Security Environments 10
Mapping CCASTAT Entries to Datasets 10
SirSafe Modes for CCASTAT 11
Support of “Visible” Passwords 12
Activating and Deactivating SirSafe 13
Identifying file/group CCASTAT Entries 13
Moving file/group CCASTAT Entries 14
Enhanced SECURE Command 15

Chapter 3: Read-Only Files under MVS **17**

Monitoring and Debugging 18

Chapter 4: Enhanced Shared DASD Enqueueing **19**

Global Resource Serialization (GRS) 19
SIRENQ 20
Modified Shared DASD List Maintenance 21

Controlling Shared DASD Enqueueing 21

Chapter 5: $SIR_CHECK_ACCESS function 23

Chapter 6: Command Reference 25

AUTHCTL 25
AUTHCTL A SIRSAFE 25
AUTHCTL C SIRSAFE 27
AUTHCTL D SIRSAFE 27
AUTHCTL LIST SIRSAFE 27
AUTHCTL REFRESH 28
AUTHCTL TEST 28
AUTHCTL VIEW 30
LOGCTL Enhancements for Visible File/Group Entries 30
LOGCTL R 31
LOGFILE and LOGGRP Enhancements 32
Viewing visible entries in sorted display 32
Selecting entries by password 32
SECURE command enhancements 33

Chapter 7: Parameter Reference 35

APSYSEC 35

Appendix A: SIRENQ Utility Reference 37

Installation 37
Executing SIRENQ 38
Operation 39

Index 41

Summary of Changes

This section describes significant changes to the documentation. In most cases these changes correspond to enhancements made to the underlying product.

Sirius Mods Version 6.8

Changes in version 6.8 include:

- Documentation of the APSYSEC parameter (“APSYSEC” on page 35), actually available in version 6.7, which allows system managers to STOP, START, TEST, or DEBUG subsystems without being defined to the subsystems.

Sirius Mods Version 6.7

Changes in version 6.7 include:

- The information displayed by `AUTHCTL VIEW` has been enhanced. See “AUTHCTL VIEW” on page 30.
- A system manager may now start and stop any application subsystem without requiring “subsystem manager” authority for the subsystem. This enables users to eliminate SCLASSes for many susbsystems, which typically improves performance. See “APSYSEC” on page 35.

Sirius Mods Version 6.4

Message MSIR.0550 is produced when a *Model 204* password table is marked as **requiring** *SirSafe*, but the current security environment is not allowed by the rules in `CCASTAT`. The message has been enhanced to list the key information about the *current* security environment.

Sirius Mods Version 6.3

Changes in version 6.3 include:

- *SirSafe* simplifies the sharing of *Model 204* database files between machines and LPARs that are members of a Sysplex or participants in a GRS ring. A standalone utility program, *SIRENQ*, enables *SirSafe* to detect when an MVS configuration provides enhanced enqueue support, as well as determines the other systems affected by the enhanced support. This mode of operation allows for automated clearing of obsolete shared dasd enqueue list entries in certain conditions.

- The `$SIR_CHECK_ACCESS` function provides a simple mechanism for *Model 204* applications to use the services of an MVS System Authorization Facility (like RACF or ACF2) to test access to files. The function may also be used to control access to application features.

Sirius Mods Version 6.2

Support is added for a “standard” zap to reactivate the “noise” logging for *SirSafe* security checks that was removed in version 6.0.

Sirius Mods Version 6.0

Changes in version 6.0 include:

- Optional support for read-only files under MVS environments is added. This removes the requirement for batch jobs to always have write access to *Model 204* database files.

- *SirSafe* uses a system security interface (for example, RACF) to determine if access to certain facilities is allowed. In many cases a denial of access does not represent an error, per se.

- This release of *SirSafe* eliminates the “noise” logging of security violations that do not actually represent errors.

Sirius Mods Version 5.5

The debugging facilities activated by the `AUTHCTL TEST` command are improved:

- Information is added to clarify the operating system user ID and group ID used for validating access.
- The `MSIR.Ø557` message is added to indicate access was allowed.

Sirius Mods Version 5.4

First release of *SirSafe*. Support for Security Server (formally RACF) has been tested under OS/390 and z/OS. Support for CA-ACF2 and CA-Top Secret is present, but has not been verified.

CHAPTER 1 Overview

SirSafe integrates the file security and access control mechanisms employed by *Model 204* with the native facilities of MVS (OS/390 and z/OS). *SirSafe* addresses several potential security exposures, while providing significant improvements in usability for large scale environments. *SirSafe* allows *Model 204* to continue to use its proprietary security mechanisms to manage access to the information contained in database files, preserving its rich set of security controls. In addition to improving security, the tight integration of *SirSafe* with modern MVS facilities enables support of *read-only* files, and it simplifies file sharing among members of a sysplex.

SirSafe functions as a layer on top of *Model 204*, using the System Authorization Facility (SAF) to control access to individual *Model 204* file and group passwords. SAF provides a standardized interface for accessing the services of a system **security manager** such as RACF, ACF2, or Top Secret. The *Model 204* System Manager is still responsible for maintaining file and group entries in a *Model 204* password table (`CCASTAT`). However, each user's access to particular file and group **entries** in the password table is controlled by a system security manager, using rules entered by a system security officer. Thus, even if a password is known to a user, that user may be prevented from gaining the access rights conferred by the password.

SirSafe can significantly improve the behavior of *Model 204* in multi-system or sysplex environments. In addition, *SirSafe* extends the functionality of *Model 204* database file security in four important ways:

- The security of data contained in *Model 204* databases no longer depends upon maintaining the secrecy of file and group passwords.
- *SirSafe* can be configured to support read-only access to database files, preventing unintended updates while reducing the number of users requiring update access to *Model 204* file datasets.
- A given file or group password can confer different privileges, depending upon the identity of the current user and the current online, *without* the confusion of multiple password tables.
- Enhancements to the *Model 204* `SECURE` command make it easier to manage security in complex, multiple online environments, while also eliminating certain security exposures.

By providing control over who can use *Model 204* file and group passwords, *SirSafe* allows the passwords to be freely shared, without the need for periodic changes. Straightforward file and group passwords that are easy to remember, such as `read`, `update`, or `sysprog`, can be freely distributed and embedded in batch job streams. Security is not compromised, because *SirSafe* will verify access to the passwords for each user and for the submittor of batch jobs.

This is especially useful for large programming teams, where members may come and go. Without *SirSafe*, either an ever-expanding circle of programmers become aware of key *Model 204* file and group passwords, or the passwords need to be changed frequently and redistributed to those users still authorized. With *SirSafe* the system security administrator would simply change the access rules for the affected user in order to grant or revoke access to the required *Model 204* file and group passwords.

1.1 Model 204 Database Security

Model 204 enforces a sophisticated array of controls to enforce security and integrity policies for its database files. The access to each individual file is determined by fourteen independent capability bits, four separate access levels for field-level security, and a procedure class designation. The unique "file group" feature of *Model 204* adds an additional two capability bits for each database file that is accessed as a group member. All of these controls are documented in the ***Model 204 File Manager's Guide***.

Most end users access applications that are formally defined to *Model 204* as application subsystems, using the `SUBSYSMGMT` administrative tool described in the *Rocket Model 204 System Manager's Guide*. Application subsystems are invoked from the *Model 204* command level with a simple command, and then the so-called **"APSY"** runtime uses information saved in the `CCASYS` file to determine the set of files and groups that need to be opened for the application along with their appropriate access rights. Users of application subsystems are not prompted for file or group passwords: the subsystem definition identifies the access granted for each user, with some users getting potentially different privileges based upon their user ID. *SirSafe* does not affect how APSY assigns file or group privileges.

Programmers and DataBase Administrators, on the other hand, frequently use the *Model 204* `OPEN` command to access files and groups while they are developing applications and maintaining files. The file or group access privileges granted as a result of an OPEN command depend upon the *type* of the file and upon the user-provided password, as follows:

Public The end-user is not prompted for a password, and the privileges are those that were specified as the default when the file or group was created.

Semipublic The end-user is prompted for a password. The *Model 204* password table is searched for a corresponding password (for the file or group being opened), and if a match is found, the user is given the privileges from the matching entry. If a matching password is not found, the user is given the default privileges for the file, as specified when the file or group was created.

Private The end-user is prompted for a password. The *Model 204* password table is searched for a corresponding password (for the file or group

being opened), and if a match is found, the user is given the privileges from the matching entry. Otherwise the OPEN is rejected.

The *Model 204* password table provides the ability to store multiple passwords for each file or group. A distinct set of access rights is associated with each password, so in effect the passwords become substitutes for named roles. It is the development and the file maintenance roles that present the greatest challenges, since they confer the strongest (update) privileges, and they are frequently embedded in maintenance job streams.

1.2 Model 204 Password Table (CCASTAT)

The *Model 204* password table (maintained in the sequential file `CCASTAT`) is divided into three physical sections. The `user ID` section contains one entry for each ten-character login ID that is directly managed by *Model 204* (as opposed to the IDs managed by a "security manager" like RACF or ACF2). Each user ID entry holds a one-way encrypted password and a set of login privileges. Most users of *SirSafe* do not use CCASTAT to manage login security; instead they typically use a formal security manager like RACF or ACF2. For more information, refer to the *Rocket Model 204 Security Interfaces Manual*.

The `file` and `group` sections of CCASTAT contain the various one-way encrypted passwords and the privileges they confer for *Model 204* database files and groups. The two sections are physically distinct so that a file and group of the same name can have separately managed passwords. A particular file or group may have multiple entries in CCASTAT, each conferring a set of (possibly different) access rights and privileges.

The `LOGCTL` command, documented in the ***Model 204 Command Reference Manual*** is used to maintain entries in the *Model 204* password table. Variants of the LOGCTL command are used to **A**dd, **D**elete or **C**hange entries in CCASTAT. The basic form of the command is:

```
LOGCTL {A | D | C} key
```

The first argument indicates the type of operation being performed. *key* indicates both the type of entry as well as the specific entry:

- For a file entry, *key* must begin with a colon (`:`), immediately followed by a one- to eight-character file or group name, then one or more optional blanks and a single `index character`. If no index character is present, a blank is assumed.
- For group entries, the format is the same as for a file entry, except the colon character is replaced by a comma (`,`).
- For a login entry, a key contains from one to ten alphanumeric characters.

The following example deletes the password table entry for group GARY with an index character of “C,” and then it adds an entry for file ALEX with index character “2”:

```
LOGCTL D ,GARY C
LOGCTL A :ALEX 2
```

After the second command, the end user would be prompted to enter a password, privileges, and other access information, and a terminal mask, if any.

The three sections of the password table are maintained in the order of their ten-byte key strings. Thus all of the entries for a particular file follow each other, sorted in the order of their index character (with blank coming first, of course). The LOGFILE command is used to display the file entries in a CCASTAT, producing output like:

```
:ALANPROC   X'0201'   0,   0,   0,   0,   0,    ALL
:ALANPROC A X'BFFF'   0,   0,   0,   0,   0,    ALL
:ALANPROC 1 X'0CCC'   0,   0,   0,   0,   0,    ALL
:ASDF       X'BFFF'   0,   0,   0,   0,   0,    ALL
:BACKUP     X'8761'   0,   0,   0,   0,   0,    ALL
```

When a Semipublic or Private file or group is OPENed, the end user is prompted for a password, which is immediately one-way encrypted. Then the file or group section of CCASTAT is searched for a match on the “middle” eight characters of the key. For each file or group name match, the one-way encrypted password in the entry is compared to the one-way encrypted value of the password provided by the end-user. If they match, the privileges contained in the entry are granted to the OPEN request. Otherwise, the scan continues until all entries for the file/group have been checked.

You can see that if two or more CCASTAT entries for a file or group contain the same password, the one with the lowest index character value will be the only one ever used. The possibility of duplicate passwords, coupled with the fact that LOGFILE and LOGGRP don't display password values, can cause a great deal of confusion.

1.3 SECURE and DESECURE commands

Each password table contains an encrypted eight-character key value. The LOGKEY command changes the key value from its default. The SECURE command copies the current CCASTAT key value into the current default file. Once this has been done, that file can only be opened by a copy of *Model 204* using a CCASTAT with the same key value. This feature can be used to prevent a malicious user from counterfeiting a password table and using it to obtain inappropriate access to a private or semipublic *Model 204* database file. Once a file has been successfully opened, the DESECURE command can be used to clear the key value from the current default file so that it can once again be opened by an online with any CCASTAT.

1.4 Physical OPEN of Model 204 Files

Under MVS, *Model 204* physically opens all database files with the `INOUT` option. using the access profile of the current job. Because INOUT implies WRITE access to the file, the job's profile must allow WRITE access, or the OPEN will be rejected with an IEC150I message. The INOUT option is used because subsequent OPEN commands for the same file will use the already opened DCB, so the initial OPEN must allow for READ or WRITE access. While this is not typically a problem for **ONLINE** jobs, it means that users running batch jobs require WRITE access to the datasets of the *Model 204* database files, even if the files are opened with access privileges that do not allow updating. This can raise concerns with Sarbanes/Oxley reporting requirements.

1.5 Shared DASD Enqueueing

A shared DASD configuration allows a particular DASD device to be accessed by more than one computer system or operating system instance. Shared DASD has become far more common with the advent of Logical PARtition (LPAR) support, fiber-optic (ESCON) channels with EMIF sharing between LPARs, and Sysplex clusters.

Model 204 utilizes a variety of mechanisms to serialize the access and update of database files among the various users in an online instance and between various instances of *Model 204*. The primary mechanism serializing database file accesses between various instances of *Model 204* is the operating system ENQ and DEQ facility.

When a database file is opened, *Model 204* enqueues upon a resource name comprising the database file name, the volume serial for the first (or only) dataset of the file, and the dataset name for the first dataset:

```
filename.volser.fully_qualified_dataset_name
```

The strength of enqueue (share or exclusive) depends upon the access intent for the file. In order to support deadlock-free reduction in access strength, *Model 204* uses three different “queue names”: `IFAMQA`, `IFAMQB`, and `IFAMQC`.

This mechanism works well for cataloged and uncataloged datasets referenced only from one operating system instance. However, it breaks down when two instances of *Model 204* running on separate systems access the same file via shared DASD. By default, the two operating system instances will not share information about the enqueues used by *Model 204*. Thus, two separate instances of *Model 204*, each running on different systems, can simultaneously hold exclusive enqueues for the same resource.

The shared DASD support in *Model 204* is based upon the concept of a “Shared DASD Enqueue List.” This enqueue list is a structure contained in the first page of the first (or only) dataset of each database file, the `File Parameter List (FPL)` page. An entry is placed in a file's enqueue list for every *Model 204* instance that opens the file.

The entry is deleted when the file is physically closed by the instance, and the entry is updated if the instance changes its access intent for the file.

The shared DASD enqueue list is used during the *Model 204* database file open process to identify conflicts with *Model 204* instances running under other operating system instances. Each entry in the enqueue list contains the following information:

System name A four-character name that identifies the system that is hosting the *Model 204* instance, identified as the first field in message M204.0061. For instances running under MVS, this ID will be the SMF system ID, set by the parameter SID in a SMFPRMxx parmlib member. For instances running under CMS, this ID will be "`CMS `."

Access intent This is `EXCL` when a file is opened for update; otherwise, it is `SHR`.

Jobname For most operating systems, this is the job name of the *Model 204* instance. For instances of *Model 204* running under CMS, it is the ID of the *Model 204* virtual machine.

Stepname For most operating systems, this is the step name of the *Model 204* instance. For instances of *Model 204* running under CMS, it is the six-character CPU serial number padded on the right with two blanks.

Date/Time The date and time when the entry was added or last updated.

The open logic also removes obsolete enqueue entries that were established, but no longer needed, by *Model 204* instances executing under the current operating system instance. Obsolete entries can result from system crashes, x22 ABENDs, and other infrequent (but ordinary) events. The following logic is used:

1. Standard operating system ENQ/DEQ logic is used to enqueue upon the specific database file. If the enqueue fails, the open is rejected with *file is in use*.

2. A hardware `RESERVE` is placed upon the volume containing the FPL page. This prevents other systems from accessing the device and potentially changing the enqueue list.

3. The FPL is read and the shared DASD enqueue list is scanned for a conflict with the access being requested by the current open.

    a. If an entry is found that conflicts with our access, but the entry was established by a *Model 204* instance under our system, delete the entry because it is obsolete. We know the entry is obsolete because the standard ENQ/DEQ mechanism under our operating system gave us the requested access.
    b. If a conflicting entry was established under a **different** operating system instance, reject the open with M204.0582 *access prevented by* and M204.0584 *file is in use* messages.

c. If all entries are processed with no current conflict, then add an entry for our requested access, and write the FPL back out to disk.

4. Release the hardware RESERVE on the volume containing the FPL page.

If an OPEN request is prevented by an obsolete entry from a different operating system instance, the *Model 204* `ENQCTL` command can be used to clear out the offending entry. If ENQCTL is used to delete a non-obsolete entry, then an M204.0585 *list overlaid* message will result when the file is closed on the system whose entry was deleted.

CHAPTER 2 Controlling Access to File and Group Passwords

The *Model 204* system manager uses the LOGCTL command to maintain database passwords in CCASTAT. Then *SirSafe* maps the individual file and group entries in CCASTAT into resources that may be controlled by a system security manager, such as RACF. Judicious use of naming standards simplifies the division of responsibility between the *Model 204* system manager and a system security officer.

When *SirSafe* is active, it alters the process of verifying passwords for Private and Semipublic files and groups. When CCASTAT is scanned for a matching password during the file or group open process, an additional step is added for each entry that matches the password entered by the end-user. Before the access rights associated with the entry are granted, a system security manager is used to verify that the user has READ access to that entry in CCASTAT. If the user doesn't have READ access, the entry is skipped, and CCASTAT processing continues as if the passwords did not match. Thus, an end user could know a password, but be denied its use.

File or group password entries with the same password and different privileges can be used to implement very flexible security schemes. Password entries conveying “strong” access rights should be entered into CCASTAT with index characters that collate low, such as blank or “A.” An entry with the same password and weaker privileges (like read-only) could follow with a higher collating index, such as “1.” Then the same password could give two different users different access rights, depending upon rules enforced by a system security manager.

SirSafe also enhances control over when an end user is allowed to change the password for a particular file or group CCASTAT entry. Whenever *Model 204* prompts for a password, the end-user may enter the password value, followed by a colon (:) and a replacement password value. If the password is matched, then the replacement password value may be used to overlay the password value in the CCASTAT entry. Without *SirSafe*, a particular end-user must be authorized to change **all** file or group passwords or to change **none**.

SirSafe adds another level of checking before end-users are allowed to change a file or group password. The end-user must first have READ access to the particular CCASTAT entry, then if a replacement password value was provided, *SirSafe* checks for WRITE access to the CCASTAT entry. If the end-user has WRITE access, the password is updated. Otherwise, the update request is rejected. This facility can prevent the accidental updating of a password shared by many people.

2.1 Model 204 Security Environments

Use of *SirSafe* requires an active *Model 204* `Security Environment`. A Security Environment consists of:

- An `interface` between *Model 204* and a particular security manager
- Certain `security parameters` that are specific to the interface

Detailed information about how to install and configure a security interface for *Model 204* can be found in the ***Model 204 Security Interfaces Manual***.

Each of the security manager interfaces supported by *Model 204* implements a default set of parameters, and it also provides a facility for customizing parameters that can be selected by the `SECPLIST` user zero parameter. In order to determine if a particular online is operating under the control of a security manager, and to determine the specific parameters in effect, you can login as a system manager and execute the following command:

```
AUTHCTL VIEW
```

If an interface is active, AUTHCTL VIEW ("AUTHCTL VIEW" on page 30) identifies it and list its current parameters. *SirSafe* adapts a parameter from each type of interface to form the High Level Qualifier (HLQ) used for mapping CCASTAT entries into virtual dataset names. The parameter used for each interface and the interface defaults are as follows:

Interface Type	*Description and HLQ source*
RACF	Now known as the IBM Security Server, RACF is an IBM Program product. The HLQ parameter is `GROUP`, which has a default value of "M204RACF".
TOPSECRET	CA-Top Secret is marketed by Computer Associates. The HLQ parameter is `ACID`, which has a default value of "M204TOPS".
ACF2	CA-ACF2 is marketed by Computer Associates. The HLQ is formed by appending the value of the `RESOURCE` field to the constant `R`. Thus, the default is "R204".

2.2 Mapping CCASTAT Entries to Datasets

SirSafe maps each file or group entry in CCASTAT to a corresponding dataset name. When an end-user needs to access a particular CCASTAT entry (for example, the entry contains a match for a file or group open password entered by the user), the active *Model 204* security interface is used to determine if the dataset corresponding to that CCASTAT entry could be read (or written) by the user. Note that no attempt is made to open the particular dataset, and the dataset does not need to exist.

The dataset names used by *SirSafe* for verifying CCASTAT access have four levels:

- The High Level Qualifier is determined by the active *Model 204* security interface as previously described.
- The second qualifier is the string `FILE` or `GROUP`, depending upon whether a file or group is being opened.
- The third level is the name of the file or group.
- The final level is determined by the index character for the current CCASTAT entry. It will contain the constant string `INDEX`, followed by the actual index character, if it is alphanumeric, or else by the two-character hexadecimal representation of the index character.

The following example shows the dataset names used by *SirSafe* to check access for some corresponding file password entries, assuming that the RACF interface is active with the default RACF Control Group Name (M204RACF):

```
 file    index     corresponding “dataset” name

:ALANPROC    ...   M204RACF.FILE.ALANPROC.INDEX40
:ALANPROC A  ...   M204RACF.FILE.ALANPROC.INDEXA
:ALANPROC 1  ...   M204RACF.FILE.ALANPROC.INDEX1
:ASDF        ...   M204RACF.FILE.ASDF.INDEX40
:BACKUP      ...   M204RACF.FILE.BACKUP.INDEX40
```

2.3 SirSafe Modes for CCASTAT

SirSafe is controlled by parameters contained in a special CCASTAT entry maintained by the `AUTHCTL` system manager command (see “AUTHCTL A SIRSAFE” on page 25). The special entry includes a list of allowed `security environments` and a *SirSafe* `mode` specification as follows:

OPTIONAL A CCASTAT that is set to OPTIONAL mode may be used by any *Model 204* load module, with or without *SirSafe* support and regardless of the current security environment. However, *SirSafe* will only control access to file and group entries in an optional CCASTAT when the current security environment matches one of those specified in the special *SirSafe* entry.

Note: OPTIONAL mode only activates a subset of the *SirSafe* functionality.

REQUIRED A CCASTAT that is set to REQUIRED mode may only be opened by a *Model 204* load module with *SirSafe* installed and with a security

environment that matches one of those specified in the special *SirSafe* entry. The REQUIRED mode activates additional features of *SirSafe*.

The REQUIRED attribute can be used to ensure that a specific security environment is used to control access to the file and group entries in CCASTAT. This is especially important when the value of passwords is widely known and *SirSafe* provides the security instead of relying on secrecy.

2.4 Support of "Visible" Passwords

When *SirSafe* is REQUIRED for a CCASTAT, then no file or group password can be used unless the end-user is allowed access to the CCASTAT entry containing the password. As explained earlier in this section, one benefit of this is that different end-users can be given different privileges when using the same password to open the same file or group. Another benefit is that passwords themselves can be freely shared and distributed, that is, they do not need to be kept a secret.

When *SirSafe* is REQUIRED for a CCASTAT, it supports so-called **visible** file and group passwords. Extensions to the `LOGCTL`, `LOGFILE`, and `LOGGRP` commands allow visible passwords to be entered, maintained, and displayed in clear text. This can greatly simplify management of multiple passwords for a particular file or group, since there is no guessing about the password value.

Ordinary (invisible) file or group passwords are maintained by the LOGCTL command, using either a colon (`:`) to indicate a file entry or a comma (`,`) to indicate a group entry. Visible entries are indicated by a different pair of special characters: The "greater than" symbol (`>`) indicates a visible file entry, and the "plus sign" symbol (`+`) indicates a visible group entry.

The LOGFILE and LOGGRP commands are extended to display the password value for visible entries, else a field of asterisks:

```
LOGFILE PROCFILE
>PROCFILE A THEMAN    X'BFFF'   Ø,   Ø,   Ø,   Ø,   Ø,    ALL
:PROCFILE B ********  X'Ø761'   Ø,   Ø,   Ø,   Ø,   Ø,    ALL
>PROCFILE 4 THEMAN    X'Ø221'   Ø,   Ø,   Ø,   Ø,   Ø,    ALL
```

In the example above, there are three password table entries for the file PROCFILE. Two of them, for the same password, are visible. In this example, a user in the "file managers" group could get access to the slot associated with index character `A`, while everyone else could get access to the slot associated with index character `4`.

2.5 Activating and Deactivating SirSafe

Until version 7.5 of Model 204, *SirSafe* was distributed as a component of the *Sirius Mods*, whose installation is described in the *Sirius Mods Installation Guide*.

Once *SirSafe* is installed, the `AUTHCTL A SIRSAFE` command ("AUTHCTL A SIRSAFE" on page 25) may be used to activate *SirSafe* for the current CCASTAT.

Activation adds a special control entry that contains the execution parameters for *SirSafe*. If the REQUIRED keyword is present, the version number of CCASTAT will be altered. This prevents the CCASTAT from being opened by *Model 204* load modules without *SirSafe* support or without the proper security environment.

For example, the following command would activate *SirSafe* as REQUIRED and usable only with RACF, using the default value for the GROUP parameter:

```
AUTHCTL A SIRSAFE REQUIRED MVSRW RACF=M204RACF
```

The contents of the *SirSafe* special entry may be displayed by the `AUTHCTL LIST SIRSAFE` command ("AUTHCTL LIST SIRSAFE" on page 27). The current *SirSafe* parameters can be replaced using the `AUTHCTL C SIRSAFE` command ("AUTHCTL C SIRSAFE" on page 27), or deleted using the `AUTHCTL D SIRSAFE` command ("AUTHCTL D SIRSAFE" on page 27).

Note: If any visible passwords have been stored, they must all be deleted before the *SirSafe* environment can be deleted or changed from REQUIRED to OPTIONAL.

2.6 Identifying file/group CCASTAT Entries

Most *Model 204* password tables contain a jumble of entries that have accumulated over time. Frequently a system manager just adds a new password when emergency access is required for a file. Without visible passwords, it is very easy to lose track of which password corresponds to a particular index character. Confusion is especially likely when a password is added that has the same value as one that occurs earlier in the collating sequence.

SirSafe implements an extension to the LOGFILE and LOGGRP commands that allows the *Model 204* system manager to create a map of the relationship between password values and index characters. It can also be used to identify password entries that have duplicate password values.

The `PWDLOCATE` keyword can be used with the LOGFILE or LOGGRP command to cause the system to prompt the user for a password value to be "ANDed" with the other

search conditions. The PWDLOCATE option could be used to diagnose a problem concerning a failure to achieve the desired access: Suppose a System Manager added a password with the value WRITE with index character A, but the user reports the password “didn't work”. LOGCTL shows the following:

```
logfile alanproc
:ALANPROC   ******** X'0201'   0,   0,   0,   0,   0,    ALL
:ALANPROC A ******** X'BFFF'   0,   0,   0,   0,   0,    ALL
:ALANPROC 1 ******** X'0CCC'   0,   0,   0,   0,   0,    ALL
```

You could use the PWDLOCATE option to identify all of the password entries that have the password value WRITE:

```
logfile pwdlocate alanproc
*** M204.0347: PASSWORD
:ALANPROC   ******** X'0201'   0,   0,   0,   0,   0,    ALL
:ALANPROC A ******** X'BFFF'   0,   0,   0,   0,   0,    ALL
```

This example shows that the CCASTAT entry for file ALANPROC with the blank index character also has the password value WRITE, and because it occurs first in the collating sequence, it is being used.

For more information about the PWDLOCATE option, see “Selecting entries by password” on page 32.

2.7 Moving file/group CCASTAT Entries

Because *SirSafe* controls access to individual file or group entries in CCASTAT, the index character for a password entry is very important. Naming conventions should be used to enable a few generic dataset rules to cover many files and groups.

A good convention to start with includes the following:

1. Reserve the blank character for system manager emergency use.
2. Reserve a few other low-collating characters (like A through E) for mapping unrecognized passwords, so “warning rules” can be used to identify their users.
3. Reserve the next few characters (like F through H) for all high-power file management passwords.
4. Reserve index characters that collate high, like numeric digits, for less-powerful, “public” passwords.

Most *Model 204* password tables contain entries that were allocated in a haphazard fashion with no particular order. In order to assist with a migration to a more orderly structure, *SirSafe* implements a facility for copying a file or group password entry from its

current slot to a slot with a different index character. The LOGCTL R command ("LOGCTL R" on page 31) is used to copy the identified file or group CCASTAT entry. If the specified entry is located, the user is prompted for the index character to be used for the copy:

```
logctl r :procfile
***  M204.0374: ENTER INDEX CHARACTER FOR REPLICATE
4
>PROCFILE 4 ******** X'BFFF'   0,   0,   0,   0,   0,    ALL
***  M204.0376: PARAMETERS ACCEPTED
***  M204.0345: CCASTAT UPDATED
```

Note: The sequence of a LOGCTL R followed by a LOGCTL D may be used to move a file or group entry in CCASTAT.

2.8 Enhanced SECURE Command

SirSafe extends the SECURE command so that a file or group can be set to open only when *SirSafe* is active (that is, the CCASTAT mode may be OPTIONAL or REQUIRED, but there must be a valid security environment). This provides an easier-to-manage facility for helping to avoid exposures to counterfeited password tables. This facility is activated with the following command:

```
SECURE FILE SIRSAFE
```

The CCASTAT modes are described in "SirSafe Modes for CCASTAT" on page 11, and the security environment is described in "Model 204 Security Environments" on page 10.

CHAPTER 3 Read-Only Files under MVS

SirSafe can be configured to provide support for read-only files under MVS environments — which can be quite useful for Sarbanes/Oxley auditing. By default this support is deactivated. In order to take advantage of read-only files, the system manager must explicitly activate `MVSRO` mode with the AUTHCTL command.

When *SirSafe* is active in MVSRO mode, additional checks are performed whenever a *Model 204* database file is **physically** opened. For each dataset comprising the *Model 204* database file, the current security interface is used to determine if the *Model 204* **job** is running under a profile that allows WRITE access. If so, the dataset will be opened for output, else an attempt will be made to open the dataset for input.

If any of the datasets for a *Model 204* database file are opened just for input, then the *Model 204* database file will be forced into read-only mode. Whatever privileges would have been granted to the opening user will be logically AND'ed with `X'8763'` and the *Model 204* message M204.0620 will be produced. If the first (or only) dataset for a *Model 204* database file is opened just for input, then the *Model 204* message M204.0590 will be produced and shared DASD enqueueing will be deactivated.

In order to activate read-only file support, the System Manager must use the AUTHCTL command. If *SirSafe* is already active, use the `AUTHCTL LIST` command to display the current *SirSafe* configuration:

```
authctl list sirsafe
AUTHCTL A SIRSAFE REQUIRED MVSRW -
               RACF=M204*
```

The keyword MVSRW indicates that read-only support is **not** active. Because *SirSafe* is running in REQUIRED mode, visible password entries may exist in CCASTAT. In this example, read-only processing could be enabled with the following command:

```
AUTHCTL C SIRSAFE REQUIRED MVSRO RACF=M204*
```

The keyword MVSRO indicates that read-only support is active.

For most jobs, the overhead of read-only support should be insignificant, because most commonly used *Model 204* database files tend to remain physically open for the life of a job. However, certain kinds of unusual jobs could experience degradation. An example could be an IFAM host language job that performs many IFOPEN and IFCLOSE calls.

If *Model 204* attempts to open a database file without *SirSafe* MVSRO active, and the job has only read access to one or more of the datasets comprising the file, an IEC150I message is produced, indicating that a 913 abend occurred. *Model 204* will intercept the open, and the open will be rejected with an M204.0454 error message.

Note: As shown below, it is still possible to receive an IEC150I message when *SirSafe* MVSRO is active, because *SirSafe* MVSRO processing just checks for **update** access to each dataset of a *Model 204* database file. An open in read-only mode is always attempted, even if a *Model 204* job has **no** access to a dataset.

```
AUTHCTL TEST ON
OPEN PROCFIL2
***  2  M204.0454: UNABLE TO OPEN FILE DATASET PROCFIL2
***  3  M204.0630: FILE OPEN COMMAND REJECTED

VIEW ERRORS
13.39.55 1 3: MSIR.0598: SirSafe: R/W access denied
13.39.55 1 3: MSIR.0597: SirSafe: (TOM,SYS1) checking R/W to
M204.GARY.PROCFIL2 on MVS204

JOB05308  ICH408I USER(TOM   ) GROUP(SYS1   ) NAME(TOM SWIFT   )  961
   961               M204.GARY.PROCFIL2 CL(DATASET ) VOL(MVS204)
   961               INSUFFICIENT ACCESS AUTHORITY
   961               FROM M204.GARY.PROCFIL2 (G)
   961               ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
JOB05308  IEC150I
913-38,IFG0194E,ONLINE,TEST,PROCFIL2,0705,MVS204,M204.GARY.PROCFIL2
```

3.1 Monitoring and Debugging

The `AUTHCTL TEST ON` command can be used to activate the display of debugging information for *SirSafe*. This lasts just for the current job, and it can be cancelled with an `AUTHCTL TEST OFF` command.

When AUTHCTL TEST is activated, two new messages track the *SirSafe* dataset access checking for read-only support: Message MSIR.0597 indicates the dataset being checked and the user ID and group for the access. MSIR.0598 indicates the failure or success of the check.

The following example shows *SirSafe* MVSRO processing forcing a file to open in read-only mode. (Remember that `VIEW ERRORS` output displays in reverse chronological order, and note that timestamps are removed from the example to save space.)

```
LOGFILE PROCFIL2
>PROCFIL2 A WRITE     X'BFFF'   0,   0,   0,   0,   0,    ALL
AUTHCTL TEST ON
O PROCFIL2
***  M204.0347: PASSWORD
***  M204.0590: SHARE-DASD ENQUEUEING INACTIVATED, FPL OF FILE PROCFIL2
                MVS204.M204.GARY.PROCFIL2 IS ON A READ-ONLY DEVICE
***  M204.0620: FILE PROCFIL2 OPENED -- NO UPDATES ALLOWED
V CURPRIV,ERRORS
CURPRIV   X'8763'     PRIVS FOR CURRENT FILE/GROUP
MSIR.0557: SirSafe approved password access
MSIR.0553: GARY (M204USR,M204GRP) read to M204RACF.FILE.PROCFIL2.INDEXA tried by
MSIR.0598: SirSafe: R/W access denied
MSIR.0597: SirSafe: (GARY,SYS1) checking R/W to M204.GARY.PROCFIL2 on MVS204
```

CHAPTER 4 Enhanced Shared DASD Enqueueing

SirSafe enhances the shared DASD enqueue mechanisms of *Model 204* to better exploit Sysplex and multiple LPAR environments. The *SIRENQ* stand-alone utility allows members of a Sysplex or GRS ring to "see" each other, providing automatic recovery from a variety of shared DASD enqueueing conflicts that arise when *Model 204* jobs are scheduled across multiple systems. Extended monitoring and administration facilities within the *Model 204* environment simplify administration of Sysplex environments.

4.1 Global Resource Serialization (GRS)

Global Resource Serialization (GRS) allows two or more operating system instances to share information about serialized resources. GRS can be used to support ENQ/DEQ serialization between address spaces executing under different instances of an MVS operating system. GRS is mandatory between members of a Sysplex, and it is optional between systems that are not members of the same Sysplex.

Once a GRS ring is established, ENQ/DEQ requests between all members of the ring can be serialized as if the entire ring were a single operating system image. The ENQ/DEQ services accept an argument that specifies the scope of the resource being enqueued:

SYSTEM The current ENQ/DEQ applies just to the current operating system instance. ENQ/DEQ requests with a scope of SYSTEM are not visible to other members of a GRS ring.

SYSTEMS The current ENQ/DEQ applies to the GRS ring of the current system. ENQ/DEQ requests with a scope of SYSTEMS are not considered to be the same as requests with a scope of SYSTEM.

The mutual insulation of the ENQ/DEQ scopes means that all of the *Model 204* instances executing on an operating system instance must use the same specification for scope. In practice, this can be very hard to ensure, and errors can compromise file integrity.

Fortunately, a GRSRNLxx parmlib member can be used to override the scope specification for ENQ/DEQ requests. This makes it possible for all of the *Model 204* file control ENQ/DEQ requests on a system to be promoted from a scope of SYSTEM to a

scope of SYSTEMS. This is accomplished by the following entries in a GRSRNLxx member:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(IFAMQA)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(IFAMQB)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(IFAMQC)
```

If the entries above are active for a particular instance of MVS, then all of the ENQ/DEQ requests used by *Model 204* to serialize database file access will be promoted to the SYSTEMS scope. This guarantees that the current *Model 204* logic for shared DASD will continue to work, but in and of itself, the promotion does not extend the functionality.

For more information on creating and maintaining GRS rings and promoting enqueues, refer to the following IBM publications:

- *MVS Authorized Assembler Services Guide*
- *MVS Planning: Global Resource Serialization*
- *MVS Initialization and Tuning Reference*

4.2 SIRENQ

The *SIRENQ* utility is designed to be run as a started task or batch job under all of the MVS instances that can host *Model 204* jobs. Its purpose is to let each *Model 204* instance determine if it is a member of a Sysplex or GRS ring, and if so, to determine the other members of the ring that are promoting IFAMQx enqueues. *Model 204* can then treat all systems so identified as a single system image, performing automatic cleanup of obsolete shared DASD enqueue list entries.

SIRENQ's unified view of *Model 204* instances and shared DASD cleanup provides significant benefit to Sysplex and GRS ring members, while it still maintains the usefulness of the current shared DASD approach for other systems (for example, for a monoplex used for testing).

SIRENQ installation and operation is summarized in “SIRENQ Utility Reference” on page 37.

SIRENQ establishes three special enqueues, each with a scope of SYSTEM:

```
IFAMQA   MODEL204.SIRENQ.smfid
IFAMQB   MODEL204.SIRENQ.smfid
IFAMQC   MODEL204.SIRENQ.smfid
```

Where *smfid* is the SMF system ID for the current system. *SIRENQ* then uses the GQSCAN system service to build a list of all MVS instances that have enqueues on these resources with a scope of SYSTEMS. By examining the last portion of the resource name, *SIRENQ* can determine if the IFAMQx enqueues from other systems will be “visible.”

SIRENQ wakes up periodically (default sleep interval is five minutes) to recheck its view of the GRS environment and update its list of other MVS instances running *SIRENQ*. *SIRENQ* also produces informational messages to the operator's console when other MVS instances either become visible or cease to be visible.

4.3 Modified Shared DASD List Maintenance

All systems that are "*SirENQ visible*" to the current system are considered to be the same system for purposes of deleting obsolete shared DASD enqueue list entries. For example, if a *Model 204* instance is running on system "V210" with a copy of *SIRENQ*, and *SIRENQ* is also running on another GRS ring member with an SMF system ID of "Z140," then the first instance will automatically detect and delete any obsolete enqueue entries for Z140.

This is possible because the *Model 204* instances on V210 know that their ENQ/DEQ calls have been promoted to the SYSTEMS scope, and they also know that the ENQ/DEQ calls on Z140 have been promoted. Further, *Model 204* instances on the two systems know that the GRS ring between the two instances is functional.

Note that if a third system is either not in the GRS ring or does not have a copy of *SIRENQ* active, then any *Model 204* instance running on that system will not be able to detect obsolete entries from V210 or Z140, and those systems in turn will not be able to detect obsolete entries from the third system. However, no loss in file integrity will result from sharing files among the three systems.

4.4 Controlling Shared DASD Enqueueing

The `AUTHCTL` command has been enhanced to better support the monitoring and management of shared DASD enqueueing within *Model 204*. The `AUTHCTL VIEW` command now indicates whether enhanced shared DASD enqueueing is active, and if so, it lists the systems that are visible and those that have become invisible:

```
***  MSIR.0688: SirSafe enhanced shared DASD active on V210 (SIRIJES2)
***  MSIR.0689: SirSafe Shared DASD visible system:   V210   (SIRIJES2)
***  MSIR.0689: SirSafe Shared DASD visible system: + Z140   (SIRIUSZO)
***  MSIR.0689: SirSafe Shared DASD visible system: - TEST   (SIRITEST)
```

The above example shows that enhanced shared DASD enqueueing is active for the current system, that the system Z140 has become visible since the last AUTHCTL VIEW or AUTHCTL REFRESH, and the the system TEST was visible but is currently not visible. Note that the eight-character "system name" is shown as well as the older four-character SMF ID for each system.

Suppose that a *Model 204* online being tested under system TEST had several files open when the testing LPAR was reset. Rather than track down all the obsolete shared

DASD enqueue list entries and delete them with an ENQCTL command, a second copy of *SIRENQ* can be run on the V210 system, with a parameter of “SMFID=TEST”. This will cause any *Model 204* instance on V210 to automatically clear the obsolete entries from TEST.

```
***  MSIR.0688: SirSafe enhanced shared DASD active on V210 (SIRIJES2)
***  MSIR.0689: SirSafe Shared DASD visible system:   Z140   (SIRIUSZO)
***  MSIR.0689: SirSafe Shared DASD visible system:   TEST   (SIRIJES2)
```

The presence of the system name makes it easy to spot duplicate copies of *SIRENQ* that were used to facilitate recovery. Note that if the system SIRITEST was restarted and rejoined the Sysplex or GRS ring, then when it's copy of *SIRENQ* was started, the operator of SIRITEST would receive an error notification indicating that SIRIJES2 was running a spoofing copy of *SIRENQ*.

Each *Model 204* instance maintains its view of visible systems in an efficient manner. You can use the AUTHCTL REFRESH command to cause *Model 204* to manually rebuild its view of visible systems:

```
AUTHCTL REFRESH [CLEAR]
```

The CLEAR option directs *Model 204* to remove from its list any systems that were once visible, but are now invisible (that is, those entries preceded by a minus sign). If a *Model 204* instance is started without a copy of *SIRENQ* running, then an AUTHCTL REFRESH command is required to enable enhanced shared DASD enqueueing.

CHAPTER 5 $SIR_CHECK_ACCESS function

The $SIR_CHECK_ACCESS function allows installations to code sophisticated interfaces between *Model 204* applications and a System Authorization Facility such as RACF or ACF2. $SIR_CHECK_ACCESS lets a system security administrator maintain controls over “dataset names” that identify intended application actions. By calling the function at strategic points, a User Language program can provide security with arbitrary granularity.

The format of the $SIR_CHECK_ACCESS call is

```
%RC = $SIR_CHECK_ACCESS(dsn, prefix, access, log)
```

Where

dsn An uppercase “dataset name,” following the usual rules.

Note: If a prefix is provided, the prefix is concatenated to the beginning of *dsn* with a separating period (`.`), and the resulting string must be less than 44 characters.

prefix A string indicating whether the provided dataset name is to be prefixed. Valid values are:

NONE No prefix is to be provided (the default).

AUTH The same HLQ used by *SirSafe*, determined by the external authorizer in use:

RACF The RACF control group name in effect for the run (default `M204RACF`).

ACF2 The character "R" with the ACF2 resource type appended (default `R204`).

TOPSECRET The Top Secret ACID in effect for the run (default `M204TOPS`).

JOB The current job name.

JOB.STEP The current job name and step name separated by a period.

AUTH.JOB The SirSafe HLQ and job name separated by a period.

CCASYS The dataset name for CCASYS.

access
Flag indicating desired access, either `R` for read or `W` for write. The default is `R`.

log Flag indicating whether failed access checks should be logged, either `Y` or `N`. The default is `N`, which suppresses logging.

$SIR_CHECK_ACCESS returns an integer value, as follows:

2 No authorizer running, call ignored
1 Access not allowed
0 Access allowed
-1 Resulting dsname (prefix.dsname) invalid
-2 Prefix argument invalid
-3 Read/write flag invalid
-4 Log/nolog flag invalid

CHAPTER 6 Command Reference

SirSafe is controlled by enhancements to the AUTHCTL command. *SirSafe* provides its services through extensions to other *Model 204* commands, including LOGCTL, LOGFILE, LOGGRP, and SECURE.

6.1 AUTHCTL

The AUTHCTL command is used to activate *SirSafe* for a *Model 204* password table and to control various aspects of its function. The basic AUTHCTL command is described in the *Rocket Model 204 Security Interfaces Manual*. This manual only describes additions and extensions specific to *SirSafe*.

6.1.1 AUTHCTL A SIRSAFE

The AUTHCTL A SIRSAFE command is used both to activate *SirSafe* for a *Model 204* password table and to specify the security environments that are allowed to access the password table. This is accomplished by adding a *SirSafe* security entry that is processed during *Model 204* initialization.

```
AUTHCTL A SIRSAFE { OPTIONAL | REQUIRED } -
       [ MVSRW | MVSRO ] -
       interface=mask [ interface=mask ] ...
```

AUTHCTL A SIRSAFE syntax

Where

OPTIONAL Indicates that the current *Model 204* password table will be accessible to any *Model 204* online, including those without *SirSafe* support. However, no visible file or group entries may be added to a password table when *SirSafe* support is optional.

Note: If a *SirSafe*-enabled *Model 204* instance opens a password table that has been set as optional, *SirSafe* will activate only if the current security environment satisfies the conditions described by the *SirSafe* security entry.

REQUIRED Indicates that the current *Model 204* password table will only be accessible to a *Model 204* online that has:

- *SirSafe* support
- A current security environment that matches one of those specified in the special *SirSafe* entry

This condition enables support for visible file and group entries.

MVSRW Indicates that *Model 204* will try to open all files in read/write mode and update their FPL pages, even if the file is being opened for read/only access. If a file cannot be opened in read/write mode, it will not be opened. This is the standard *Model 204* behavior, and it is the default.

MVSRO Indicates that if *Model 204* is not allowed to open a file in read/write mode, it will try to open it in read/only mode and so not update the FPL page of the file. Whether or not *Model 204* is allowed to open a file in read/write mode depends, of course, on the security profile of the user under which the *Model 204* job is running.

interface The name of a *Model 204* external security interface: `RACF`, `ACF2`, or `TOPSECRET`.

mask A character string that identifies an acceptable environment for the corresponding security interface, as described further, below. You can use a trailing asterisk (*) to identify a range of possible environments.

Multiple *interface=mask* pairs may be specified. During initialization, a *SirSafe*-enabled *Model 204* scans the password table for a *SirSafe* security entry. If one is found, its list of *interface=mask* pairs is checked. If an entry matching the current security environment is found, *Model 204* enters the *SirSafe*-active state. Otherwise an M204.0340 or M204.0341 error message is produced.

The interpretation of *interface* and *mask* combinations depends upon the particular interface:

- For RACF, the *mask* value applies to the field identified as the “RACF CONTROL GROUP NAME” on an `AUTHCTL VIEW` command (see the example in “AUTHCTL VIEW” on page 30).
- For ACF2, the *mask* value applies to the field identified as the “ACF2 RESOURCE TYPE” on an `AUTHCTL VIEW` command, prefixed with the letter `"R"`.
- For TOPSECRET, the *mask* value applies to the field identified as the “ACID” on an `AUTHCTL VIEW` command.

Thus, the following command

```
AUTHCTL A SIRSAFE REQUIRED RACF=M204* -
ACF2=R204 TOPSECRET=*
```

allows visible file and group passwords to be stored and requires that *Model 204* be *SirSafe*-enabled when it tries to use the password table. If RACF was being used, the RACF control group could be any name starting with "M204", which would include the default "M204RACF". If ACF2 was being used, the ACF2 resource type would need to be "204", which is the default. Any TOPSECRET environment would be allowed.

6.1.2 AUTHCTL C SIRSAFE

The `AUTHCTL C SIRSAFE` command is used to change the *SirSafe* security entry for a *Model 204* password table. It accepts the same parameters as the `AUTHCTL A SIRSAFE` command. It is especially useful for updating the security masks for a password table that contains visible file or group entries.

The `AUTHCTL C SIRSAFE` command may not be used to switch a password table to the *SirSafe*-optional state if the table contains any visible file or group entries. An MSIR.0542 error message is produced, and the command is ignored.

6.1.3 AUTHCTL D SIRSAFE

The `AUTHCTL D SIRSAFE` command is used to delete the *SirSafe* security entry from a *Model 204* password table, deactivating *SirSafe*.

Any visible file and group entries must be deleted before *SirSafe* can be deactivated. If the current *SirSafe* security entry includes the REQUIRED attribute, and visible file or group password entries have been added to the password table, an `AUTHCTL D` request will be rejected with an MSIR.0542 error message.

6.1.4 AUTHCTL LIST SIRSAFE

The `AUTHCTL LIST SIRSAFE` command is used to list the *SirSafe* security entry, if any in the current *Model 204* password table. The output from this command is the `AUTHCTL A SIRSAFE` command that could be used to recreate the *SirSafe* security entry. For example:

```
AUTHCTL LIST SIRSAFE
AUTHCTL A SIRSAFE REQUIRED MVSRO RACF=M2Ø4*
```

6.1.5 AUTHCTL REFRESH

The AUTHCTL REFRESH command causes *Model 204* to manually rebuild its list of systems for which enhanced shared DASD enqueueing is active, including whether the systems are visible or have become invisible. This list is displayed by the AUTHCTL VIEW command, as described in "Controlling Shared DASD Enqueueing" on page 21.

```
AUTHCTL REFRESH [CLEAR]
```

AUTHCTL REFRESH syntax

The CLEAR option directs *Model 204* to remove from its list any systems that were once visible, but are now invisible.

If a *Model 204* instance is started without a copy of *SIRENQ* running, an AUTHCTL REFRESH command is required to enable enhanced shared DASD enqueueing.

6.1.6 AUTHCTL TEST

The AUTHCTL TEST command is used to activate or deactivate the logging of *SirSafe* debugging and diagnostic information. When the AUTHCTL TEST facility is active, informative messages are produced that track the various calls *SirSafe* makes to the current security interface. The *SirScan* product may be used to examine the messages, or they may be retrieved with the *Model 204* VIEW ERRORS command. When using VIEW ERRORS, please remember that the most recent messages are listed first.

```
AUTHCTL TEST { ON | OFF }
```

AUTHCTL TEST syntax

AUTHCTL TEST ON produces six informational messages to the journal and *Model 204* operator. If the MVSRO feature is active, two messages are used to track the tests performed to determine access to *Model 204* database file datasets, as shown in "Monitoring and Debugging" on page 18. The remaining four messages are for debugging access to file and group passwords.

If no security environment is active, *SirSafe* produces an MSIR.0552 message, and access to file and group passwords is disallowed. Otherwise, an MSIR.0553 message is produced for each file or group password table entry that matches a password provided by a user, indicating that *SirSafe* is testing the current end user's access. If the external security interface disallows the requested access, an MSIR.0554 message is produced. If the requested access is allowed by the external security interface, an MSIR.0557 message is produced.

The MSIR.0553 message identifies the user attempting the access, the "dataset" name associated with the access, and the type of access required (read or write). The user ID

is identified in two ways: the *Model 204* "internal ID," that is, the ID that appears in messages on the journal, and the external security interface view — including the user ID and group ID. The MSIR.0553 message is the only source of the external authorizer information, which is important for several reasons:

- The access checks are performed using the external ID and group.
- The access rights are frequently conferred from group, as opposed to the ID.
- When *Model 204* processes a login with an ID from CCASTAT, the external user ID and group are defined by the *Model 204* external security interface.

In the following example, user GARY logs in with a CCASTAT ID, then opens the semi-public file PROCFILE with the password WRITE. The user does not get the expected access privileges, in fact the user gets the default privileges for the file:

```
O PROCFILE
***  M204.0347: PASSWORD
***  M204.0620: FILE PROCFILE OPENED -- NO UPDATES ALLOWED
VIEW ERRORS
MSIR.0554: SirSafe disallowed password access
MSIR.0553: GARY (M204USR,M204GRP) read to
             M204RACF.FILE.PROCFILE.INDEXA tried by
MSIR.0598: SirSafe: R/W access allowed
MSIR.0597: SirSafe: (GARY,SYS1) checking R/W to M204.GARY.PROCFILE
             on MVS204
V CURPRIV,PRIVDEF
CURPRIV   X'0761'     PRIVS FOR CURRENT FILE/GROUP
PRIVDEF   X'0761'     DEFAULT FILE PRIVILEGES
```

SirSafe first performed MVS read-only access checking, using the profile for the *Model 204* ***job*** (user ID GARY, group SYS1), and read/write access was allowed, so the file was not opened in read-only mode. Then the password was matched to the CCASTAT entry for PROCFILE with index character A, and access to that entry was denied. Since this was the only password matched (only one MSIR.0553 message), the user was given the default privileges for the file.

Note: Although the *Model 204* internal user ID was GARY, the external profile being used was user M204USR, group M204GRP. This occurred because CCASTAT contained an entry for the ID GARY, a common source of confusion.

6.1.7 AUTHCTL VIEW

The AUTHCTL VIEW command displays the status of the current *Model 204* external security interface, if any. If an interface is active, the command displays its type and current execution parameters. If *SirSafe* is installed, the AUTHCTL VIEW command also displays the status of *SirSafe*, as the following example shows:

```
AUTHCTL VIEW

RACF INTERFACE OPTIONS
GROUP       M204RACF  RACF CONTROL GROUP NAME
COMGROUP    M204RAC1  RACF COMMON CONTROL GROUP NAME
PRIORITY    STANDARD  PRIORITY DEFAULT
            M204GRP   DEFAULT USER GROUP
            M204USR   DEFAULT USERID
DLMCHECK              USE $JOB DLM CHECKING OPTION

***  MSIR.0551: SirSafe is active and required for current CCASTAT
***  MSIR.0599: SirSafe read-only file checking is active
***  MSIR.0687: SirSafe enhanced shared DASD not active, run
                  SirEnq on ZOS4 (P390)
```

For information about using AUTHCTL VIEW for monitoring enhanced shared DASD enqueueing, see “Controlling Shared DASD Enqueueing” on page 21.

6.2 LOGCTL Enhancements for Visible File/Group Entries

SirSafe provides support for *visible* file and group passwords when it is operated in the **REQUIRED** mode. Visible file and group entries are easier to administer because all of the data is visible while it is being entered, and because the password value is displayed by a LOGFILE or LOGGRP command.

LOGCTL recognizes standard file entries because the file name is prefixed by a colon (:), and group entries because the name is prefixed by a comma (,). *SirSafe* extends the LOGCTL A, LOGCTL C and LOGCTL D commands to support visible file and group entries. Visible file entries are indicated by prefixing the file name with a greater-than symbol (>). Visible group entries are indicated by prefixing the group name with a plus sign (+).

Whenever a LOGCTL A or LOGCTL C command is issued for a visible file or group entry, the subsequent line of input is not masked, remaining visible on the System

Manager's screen. The following example illustrates how to add a visible file password entry:

```
logctl a >procfile a
M204.0374: ENTER FILE/GROUP
PASSWORD,PRIVILEGES,CLASS,SELECT,READ,UPDATE,ADD
theman,X'bfff'
M204.0379: ENTER TERMINAL LIST, ALL, NONE, ADD, DEL, OR RETURN

>PROCFILE A THEMAN    X'BFFF'   0,   0,   0,   0,   0,    ALL
M204.0376: PARAMETERS ACCEPTED
M204.0345: CCASTAT UPDATED
```

Note that the password and privileges are visible while typing and are echoed to the screen. Also, when the completed entry is listed, the password is displayed. Contrast this to the case for adding a conventional file password:

```
logctl a :procfile b
M204.0374: ENTER FILE/GROUP
PASSWORD,PRIVILEGES,CLASS,SELECT,READ,UPDATE,ADD
M204.0379: ENTER TERMINAL LIST, ALL, NONE, ADD, DEL, OR RETURN

:PROCFILE B ********  X'00FF'   0,   0,   0,   0,   0,    ALL
M204.0376: PARAMETERS ACCEPTED
M204.0345: CCASTAT UPDATED
```

In the conventional case, the password and privilege line does not echo while typing, and the password is not displayed when the entry is listed.

6.3 LOGCTL R

The `LOGCTL R` command is a *SirSafe* implemented subcommand for the `LOGCTL` command. `LOGCTL R` is used to ***R**eplicate* (copy) a file or group entry in the password table.

```
LOGCTL R {:filename | >filename | ,grpname | +grpname} [indx]
```

LOGCTL R syntax

Where

:filename Indicates a classic, invisible password entry for the named file.

>filename Indicates a visible password entry for the named file.

,grpname Indicates a classic, invisible password entry for the named group.

+grpname Indicates a visible password entry for the named group.

indx A single index character used to distinguish between multiple entries for a file or group; it defaults to a single space.

If the identified entry is found in the *Model 204* password table, the user is prompted for an index character that will identify a new copy of the entry.

Note: The sequence of a `LOGCTL R` followed by a `LOGCTL D` may be used to move a file or group entry in the *Model 204* password table.

For a simple example of LOGCTL R, see "Moving file/group CCASTAT Entries" on page 14.

6.4 LOGFILE and LOGGRP Enhancements

SirSafe enhances the `LOGFILE` and `LOGGRP` commands to facilitate the management of file and group password table entries. These changes are active whenever *SirSafe* is enabled, whether the *SirSafe* mode is *required* or *optional*.

6.4.1 Viewing visible entries in sorted display

Even though visible file and group entries begin with a different character (>) than their standard counterparts, the file and group sections of the CCASTAT file are sorted in order of file or group name and index character.

```
logfile procfile
>PROCFILE A THEMAN    X'BFFF'   Ø,   Ø,   Ø,   Ø,   Ø,    ALL
:PROCFILE B ********  X'Ø761'   Ø,   Ø,   Ø,   Ø,   Ø,    ALL
>PROCFILE 4 THEMAN    X'Ø221'   Ø,   Ø,   Ø,   Ø,   Ø,    ALL
```

Note that for visible entries, the value of the password is displayed in clear text, while standard entries are flagged with a field of eight asterisks.

6.4.2 Selecting entries by password

The syntax for LOGFILE and LOGGRP is extended to include a `PWDLOCATE` option. If the keyword PWDLOCATE is present, the end user is prompted for a password value. That value is used to filter the list of entries that would have been produced without the PWDLOCATE option, and the entries with matching password values is displayed.

```
LOGFILE [PWDLOCATE] [[filename1] [filename2]]
```

LOGFILE PWDLOCATE syntax

If just one filename is provided, the command lists just the entries for that file. Otherwise, the command lists all the entries that are between the two values, inclusive.

```
LOGGRP [PWDLOCATE] [[groupname1] [groupname2]]
```

LOGGRP PWDLOCATE syntax

If just one groupname is provided, the command lists just the entries for that group. Otherwise, the command lists all the entries that are between the two values, inclusive.

The following example assumes that the user entered `theman` in response to the password prompt:

```
logfile pwdlocate
***  M204.0347: PASSWORD
>PROCFILE A THEMAN    X'BFFF'   0,   0,   0,   0,   0,    ALL
>PROCFILE 4 THEMAN    X'BFFF'   0,   0,   0,   0,   0,    ALL
```

For an additional example using PWDLOCATE, see “Identifying file/group CCASTAT Entries” on page 13

6.5 SECURE command enhancements

The variant of the `SECURE` command used to secure a file is extended to accept the qualifier `FILE` and the optional keyword `SIRSAFE`:

```
SECURE [[FILE] SIRSAFE]
```

SECURE FILE syntax

If the keyword `SIRSAFE` is present, the current file is set in a mode such that it can only be open when *SirSafe* is active.

For additional comments about this feature, see “Enhanced SECURE Command” on page 15.

CHAPTER 7 Parameter Reference

Some *SirSafe* provided facilities can be enabled or controlled via *Model 204* parameters. These parameters are described here.

7.1 APSYSEC

The APSYSEC parameter allows any system manager to START, STOP, DEBUG, or TEST any susbsystem, without having to add the system manager to the SCLASS authorized to do these things. The APSYSEC parameter is a User 0 (CCAIN) system parameter.

APSYSEC is a bitmask parameter, where the bits mean:

X'01' System managers are allowed to START, STOP, DEBUG, or TEST any subsystem. This reduces the headache of having to add a system manager to a privileged SCLASS in every subsystem in an Online to enable the system manager at least to start and stop the subsystems — a common thing for system managers to need to do.

In fact, if no users other than system managers need to start or stop subsystems, this can eliminate the need even to have SCLASSes in a subsystem to allow starting or stopping of the subsystem. In some cases, eliminating this requirement can reduce the subsystem definition to a single default SCLASS, which has performance benefits — no SCLASS lookup is required when a user enters a subsystem, and no SCLASS-specific compilations are done for the procedures in the subsystem.

Furthermore, because of the overhead associated with multiple SCLASSes in a subsystem (not huge, but possibly measurable) some sites risk simply adding START and STOP privileges (and perhaps TEST and DEBUG) to the one and only SCLASS for a subsystem. This, of course, means that any user can start and stop the subsystem, which might not be ideal from a control or security perspective. The APSYSEC parameter allows such a site to have it both ways: START and STOP privileges can be removed from the default/only SCLASS for a subsystem, and only system managers (or users running subsystems that give them system manager privileges) can start or stop subsystems.

Of course, if a site wants to continue using *Model 204*'s traditional fine-grained control of START, STOP, TEST, and DEBUG privileges, the APSYSEC X'01' bit should not be set.

APPENDIX A

SIRENQ Utility Reference

The *SIRENQ* utility is designed to be run as a started task or batch job under all the MVS instances that can host *Model 204* jobs. *SIRENQ* performs three functions that significantly enhance the operation of *Model 204* in Sysplex and multiple LPAR installations:

- Verifies that *Model 204* database file enqueues are automatically promoted from the `SYSTEM` scope to the `SYSTEMS` scope on the current MVS instance.
- Establishes an enqueue that can be “seen” by other members of a Global Resource Serialization (GRS) ring.
- Monitors and reports to the operator the other visible instances of *SIRENQ*.

For additional details about *SIRENQ*, see “SIRENQ” on page 20.

A.1 Installation

SIRENQ is distributed as a single object deck, which may be downloaded as directed by Technical Support or included on a product distribution tape.

SIRENQ must be linkage edited with `AMODE(31)` and `RMODE(ANY)` is suggested. *SIRENQ* does not require APF authorization.

The following job could be used to linkage edit *SIRENQ*:

```
//LINKENQ  JOB  CLASS=A,MSGCLASS=A
//*
//* LINK SIRENQ INTO SIRENQ.V1Ø1.LOAD, WHICH DOES
//* NOT NEED TO BE APF-AUTHORIZED
//*
//LINK     EXEC PGM=HEWLKED,REGION=ØM,
//          PARM='LIST,LET,MAP,NCAL,SIZE=(2Ø48K,512K),AC=Ø'
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   UNIT=VIO,SPACE=(32ØØ,(3ØØ,15Ø))
//SYSLMOD  DD   DSN=SIRENQ.V1Ø1.LOAD,DISP=SHR
//SYSLIN   DD   *
 <insert SIRENQ object deck>
 ENTRY   SIRENQ
 NAME   SIRENQ(R)
/*
//
```

Before *SIRENQ* can be used, a GRSRNLxx parmlib member must be modified and activated to override the scope specification for the enqueue requests used by *Model 204*. This is accomplished by adding the following entries in a GRSRNLxx member:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(IFAMQA)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(IFAMQB)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(IFAMQC)
```

A.2 Executing SIRENQ

SIRENQ can be invoked as a started task or a batch job, depending upon local preference. If the current configuration does not correctly promote the enqueues used by *Model 204*, *SIRENQ* issues an error message and terminates.

SIRENQ consumes very few system resources, although it does maintain three exclusive enqueues that are issued with a scope of SYSTEM and promoted to scope of SYSTEMS:

```
IFAMQA   MODEL204.SIRENQ.smfid
IFAMQB   MODEL204.SIRENQ.smfid
IFAMQC   MODEL204.SIRENQ.smfid
```

Where *smfid* is the SMF system ID for the current system, or the value of the `SMFID` parameter (described below).

The following JCL could be used to invoke *SIRENQ*, assuming it had previously been linkage-edited into the dataset identified by STEPLIB:

```
//SIRENQ  JOB ,'SIRSAFE SHARED DASD',CLASS=A,MSGCLASS=A
//*
//* FIRE UP SIRENQ
//*
//TEST    EXEC PGM=SIRENQ,REGION=1024K
//STEPLIB  DD DSN=SIRENQ.V101.LOAD,DISP=SHR
//
```

SIRENQ accepts the following parameters from the PARM field of its EXEC statement:

INTERVAL=nnn

Number of minutes to wait before re-scanning for visible systems. The default is 5 minutes, sufficient for avoiding 522 ABENDs, while still noticing systems entering and leaving the GRS ring. A value of `0` eliminates the timer, and *SIRENQ* just waits for a MODIFY or STOP command from the operator.

SMFID=cccc

If specified, this parameter provides the SMF system ID to be used by this copy of *SIRENQ*. If not specified, the SMF system ID of the current system is used. This is useful in certain hot recovery environments to avoid the need for ENQCTL commands to clear obsolete shared DASD enqueues.

A.3 Operation

SIRENQ listens for operating system `STOP` or `MODIFY` commands. If a STOP command is received, *SIRENQ* drops its enqueues, and it exits, causing the current OS/390 instance to become invisible.

SIRENQ processes the following MODIFY commands:

REFRESH

Immediately rebuild the list of *SIRENQ* copies seen on other systems, without waiting for the completion of the current time interval, if any.

EOD or EXIT

Processed as a STOP command: immediately drop the enqueues and exit.

Index

$

$SIR_CHECK_ACCESS function ... 23

A

Access privileges, *see* Privileges
ACF2 interface ... 10, 26
APSYSEC system parameter ... 35
AUTHCTL A SIRSAFE command ... 13, 25
AUTHCTL C SIRSAFE command ... 13, 27
AUTHCTL command ... 25
AUTHCTL D SIRSAFE command ... 13, 27
AUTHCTL LIST command ... 17
AUTHCTL LIST SIRSAFE command ... 13, 27
AUTHCTL REFRESH command ... 22, 28
AUTHCTL TEST command ... 18, 28
AUTHCTL VIEW command ... 10, 21, 30

C

CCASTAT file ... 1-3, 9
  entries mapped to dataset names ... 10
  entries selected by password ... 32
  rearranging entries ... 15, 32
  SirSafe mode ... 11, 25
  sorted visible entries ... 32
  visible passwords, *see* Visible passwords

D

Dataset access, *Model 204* ... 5, 28
Debugging information, *SirSafe* ... 18, 28

E

ENQ/DEQ facility ... 5-6, 19
ENQCTL command, *Model 204* ... 7
Enqueue list, shared DASD ... 5, 21
Enqueue promotion ... 19, 37
Enqueueing, *Model 204* database file ... 5
EOD command, operator ... 39
EXIT command, operator ... 39

G

GRS (Global Resource Serialization) ring ... 19, 37
GRSRNLxx parmlib member ... 19, 37

I

IBM Security Server interface ... 10
IFAMQA queue name ... 5
IFAMQB queue name ... 5
IFAMQC queue name ... 5
Index character, CCASTAT entry ... 3, 11, 13-14
INOUT option ... 5
INTERVAL parameter, *SIRENQ* ... 38

L

LOGCTL A command ... 30
LOGCTL C command ... 30
LOGCTL command ... 3, 30
LOGCTL D command ... 30
LOGCTL R command ... 15, 31
LOGFILE command ... 12-13, 32
LOGGROUP command ... 12-13, 32

M

Mode, *SirSafe* ... 11, 25
MODIFY command, operator ... 38-39
MVSRO mode, *SirSafe* ... 17-18, 26, 28
MVSRW mode, *SirSafe* ... 17, 26

O

OPTIONAL mode, *SirSafe* ... 11, 25

P

Password processing, file or group ... 9
Password table, *Model 204*
  See CCASTAT file
Passwords, visible
  See Visible passwords
Private file ... 2
Privileges, file or group ... 2

Public file ... 2
PWDLOCATE option,
LOGFILE/LOGGROUP ... 13, 32

Q

Queue names ... 5

R

RACF interface ... 10, 26
Read-only files ... 1, 17, 26
REFRESH command, operator ... 39
REQUIRED mode, *SirSafe* ... 11, 25

S

SECPLIST parameter, user zero ... 10
SECURE command, *Model 204* ... 15, 33
Security Environment, *Model 204* ... 10-11, 25
Security manager ... 1, 3, 9-10
Security, *Model 204* database ... 2
Semipublic file ... 2
Shared DASD enqueueing ... 5, 17, 19-21, 28
SIRENQ utility ... 20, 37
SirSafe
activation ... 13, 25
deactivation ... 13, 27
status ... 21, 30
SirSafe mode, CCASTAT file ... 11
SMF parameter, *SIRENQ* ... 38
STOP command, operator ... 38-39
Subsystem privileges ... 35
Sysplex environment support ... 19-20
System Authorization Facility (SAF) ... 1
SYSTEM scope, ENQ/DEQ ... 19-20, 37
SYSTEMS scope, ENQ/DEQ ... 19-20, 37

T

TOPSECRET interface ... 10, 26

V

Visible enqueues ... 20-22, 28
Visible passwords, CCASTAT ... 12-13, 25, 30, 32
Visible systems, *SIRENQ* ... 38