
Sirius Mods Installation Guide



Sirius Software, Inc.
875 Massachusetts Avenue, Suite 21
Cambridge, MA 02139

Telephone: (617) 876-6677

FAX: (617) 234-1200

E-mail: support@sirius-software.com

World Wide Web: <http://sirius-software.com>

November 26, 2006

© 2006 Sirius Software, Inc.

Proprietary Notices

Sirius Mods is a package of object modules that are linkage edited with *Model 204* to provide support for one or more proprietary products of Sirius Software, Inc.:

Sirius Software, Inc.
875 Massachusetts Avenue, Suite 21
Cambridge, Massachusetts 02139
USA

The following products are not included with the *Sirius Mods*, but are proprietary products of Sirius Software, Inc.:

- *Fast/CRAM*
- *Fast/Unload*
- *SirAud*
- *SirPIPE*
- *SirPUT*
- *SirTune*
- *SirZap*

Model 204™ is a proprietary product of Computer Corporation of America:

Computer Corporation of America
500 Old Connecticut Path
Framingham, Massachusetts 01701
USA

Model 204™ is a registered trademark of Computer Corporation of America.

Contents

Proprietary Notices	ii
Contents	iii
Summary of Changes	v
Sirius Mods Version 6.9	v
Sirius Mods Version 6.8	v
Sirius Mods Version 6.7	v
Sirius Mods Version 6.6	v
Sirius Mods Version 6.4	vi
Sirius Mods Version 6.3	vi
Sirius Mods Version 6.2	vi
Sirius Mods Version 6.1	vi
Sirius Mods Version 6.0	vi
Sirius Mods Version 5.6	vi
Sirius Mods Version 5.5	vii
Sirius Mods Version 5.4	vii
Sirius Mods Version 5.2	vii
Sirius Mods Version 5.0	vii
Chapter 1: Overview	1
Chapter 2: System Requirements	5
Chapter 3: Web-Based Installation	7
Chapter 4: MVS Installation from Cartridge	9
Chapter 5: CMS Installation from Cartridge	11
Chapter 6: Defining Sdaemon Threads	13
Chapter 7: Fast/Unload User Language Interface	15
Chapter 8: The Restart Utility	17
Commands supported by the RESTART utility	18
BUMP <i>user_num</i>	19
MONITOR	19
RESTART	19
STATUS	21

Chapter 9: Customization Zaps 23

 Disable or restrict APPDATE command 23

 Disable SIRIUS command generated during startup 24

 Suppress product expiration warnings from operator console 25

Summary of Changes

This section describes significant changes to the documentation. In most cases these changes correspond to changes in the installation process which occur with the release of some version of the *Sirius Mods*.

Note that this manual applies to all generally supported versions of the *Sirius Mods*, whether or not there is mention of a change for that version in this section.

Sirius Mods Version 6.9

- *SirTune* data collector functionality is now integrated into the *Sirius Mods* product. *SirTune* is no longer installed as a separate product. The *SirTune* Report Generator (SIRTUNER), however, is still installed separately. See the ***SirTune Reference Manual*** for installation details.

Sirius Mods Version 6.8

- *Sirius Mods* version 6.8 supports *Model 204 V5R1* and *V6R1* only.

Sirius Mods Version 6.7

- Support is added for *Model 204 V6R1*, with some qualifications for Large Object fields (LOBs) (as described in the ***Notes for Sirius Mods Release 6.7***).
- Support is dropped for *Model 204 V4R2.0*.

Sirius Mods Version 6.6

- Support is dropped for *Model 204 V4R2.1* and *V5R2*. The only supported versions are 4.2.0, 5.1, and 5.3.

Sirius Mods Version 6.4

- Support is dropped for *Model 204 V4R1.1*. Support is added for *Model 204* version 5.3.

Sirius Mods Version 6.3

- Support is dropped for *Model 204 V4R1.0*.

Sirius Mods Version 6.2

- Support is added for *Model 204* version 5.1.

Sirius Mods Version 6.1

- *Limited Janus Web Server* is included within *Sirius Mods*.

Sirius Mods Version 6.0

- Customization zap added to suppress expiration warning (this feature is also available after applying a maintenance zap to Version 5.4).

Sirius Mods Version 5.6

- Documentation is added for the new *Sirius Performance Enhancements V2* parameter, MAXINCL.
- Documentation for *Sirius Performance Enhancements* and *Sirius Performance Enhancements V2* is moved to ***Sirius Performance Enhancements Reference Manual***.

Sirius Mods Version 5.5

- Support is dropped for *Model 204 V3R2*.
- Mention of non-Y2K compliant and no longer supported versions of *Model 204* (V3R2 and earlier) is nearly eliminated.

Sirius Mods Version 5.4

- Support is dropped for *Model 204 V2R2* and *V3R1*.
- Customization zaps added to disable or restrict APPDATE.
- Customization zap added to disable automatic SIRIUS command (MSIR.0447) during startup.
- Documentation is added for the *Sirius Performance Enhancements*.

Sirius Mods Version 5.2

- Support is added for *Model 204 V4R2*.

Sirius Mods Version 5.0

The manual is converted for PDF bookmarks and other formatting changes.

Overview

This manual describes the process for installing the *Sirius Mods*. *Sirius Mods* is a package of object modules that are linkage edited with your *Model 204 ONLINE*, **BATCH204**, **IFAM1**, and **IFAM4**, load modules to provide support for one or more proprietary products of Sirius Software, Inc. The *Sirius Mods* comprise support for several Sirius products which have been integrated into a single package to simplify installation and maintenance. You need only perform the installation once, no matter how many *Sirius Mods* products you install.

The following products are separately purchased components of the *Sirius Mods*:

Product	Where documented
<i>Fast/Backup</i>	<i>Fast/Backup User's Guide</i>
<i>Fast/Reload</i>	<i>Fast/Reload Reference Manual</i>
<i>Fast/Reorg</i> (<i>Fast/Reload</i> component only)	<i>Fast/Reload Reference Manual</i>
<i>Fast/Unload User Language Interface</i> (functions)	<i>Fast/Unload Reference Manual</i> and <i>Sirius Functions Reference Manual</i>
<i>Janus Debugger</i>	<i>Janus/Sirius Debugger User's Guide</i>
<i>Janus Network Security</i>	<i>Janus Network Security Reference Manual</i>
<i>Janus Open Client</i>	<i>Janus Open Client Reference Manual</i>
<i>Janus Open Server</i>	<i>Janus Open Server Reference Manual</i>
<i>Janus Specialty Data Store</i>	<i>Janus Specialty Data Store Reference Manual</i>
<i>Janus SOAP</i>	<i>Janus SOAP Reference Manual</i>
<i>Janus Sockets</i>	<i>Janus Sockets Reference Manual</i>
<i>Janus TCP/IP Base</i>	<i>Janus TCP/IP Base Reference Manual</i>
<i>Janus Web Server</i>	<i>Janus Web Server Reference Manual</i>
<i>SirDBA</i>	<i>SirDBA User's Guide</i>
<i>SirFact</i>	<i>SirFact Reference Manual</i>
<i>SirFile</i>	<i>SirFile User's Guide</i>
<i>Sirius Debugger</i>	<i>Janus/Sirius Debugger User's Guide</i>
<i>Sirius Functions</i>	<i>Sirius Functions Reference Manual</i>
<i>Sirius Performance Enhancements</i>	<i>Sirius Performance Enhancements Reference Manual</i>
<i>Sirius Performance Enhancements V2</i>	<i>Sirius Performance Enhancements Reference Manual</i>
<i>Sirius Performance Enhancements V3</i>	<i>Sirius Performance Enhancements Reference Manual</i>
<i>SirLib</i>	<i>SirLib User's Guide</i>
<i>SirMon</i>	<i>SirMon User's Guide</i>
<i>SirPro</i>	<i>SirPro User's Guide</i>

<i>SirSafe</i>	<i>SirSafe Reference Manual</i>
<i>SirScan</i>	<i>SirScan User's Guide</i>
<i>SirTune</i>	<i>SirTune Reference Manual</i>
<i>SirXref</i>	<i>SirXref User's Guide</i>
<i>Sir2000 Field Migration Facility</i>	<i>Sir2000 Field Migration Facility Reference Manual</i>
<i>Sir2000 User Language Tools</i>	<i>Sir2000 User Language Tools Reference Manual</i>
<i>Fast/CRAM Trusted Login for Model 204</i>	<i>Fast/CRAM Trusted Login for Model 204 Reference Manual</i>

In addition, several features are available to a user licensed for any of the above products: for example, “[The Restart Utility](#)” on page 17, and *Limited Janus Web Server*, which is described in the ***Janus Web Server Reference Manual***.

Several products (for example, *SirMon*) comprise a *Sirius Mods* component that extends *Model 204* and User Language code that must be installed as a *Model 204* APSY subsystem. Refer to the ***UL/SPF Installation and Maintenance Guide*** for instructions on how to perform the User Language installation steps for these products.

All parameters and commands associated with *Sirius Mods* products are documented in the ***Sirius Mods Command and Parameter Reference Manual***. Those parameters and commands restricted to sites authorized for particular products are so indicated.

Installation method options

You install the *Sirius Mods* using either of the following:

- An object file downloaded from the Sirius Software web site. The web-based method for distribution is described in “[Web-Based Installation](#)” on page 7.
- A product distribution cartridge. Installation from cartridge is described in “[MVS Installation from Cartridge](#)” on page 9 and “[CMS Installation from Cartridge](#)” on page 11.

Note: Cartridge distribution is not available for VSE customers.

The *Sirius Mods* is shipped with *all* maintenance including a site-specific authorization zap pre-applied: up to the time the cartridge was cut for cartridge-based installations, and up to the time the object file was downloaded for web-based installations.

Maintenance method options

You maintain the *Sirius Mods* in one of two ways:

- Reinstalling the appropriate release from a new cartridge or from an object file downloaded from the web (picking up all available maintenance).
- Applying maintenance ZAPs to an existing installation.

You can have maintenance ZAPs sent on cartridge or via e-mail from Sirius Software support personnel, or you can download them from the Sirius Software web site. The latter option is the most convenient, as it can be done any time and without extra human intervention which might produce turnaround delays.

You can find instructions for downloading maintenance on the “Customer Service” page of the Sirius web site:

<http://sirius-software.com/support.html>

You will need a userid and password to access either of the two “Download fixes” links:

<https://sirius-software.com/maint/fixlist>

<https://sirius-software.com/maint/altfixlist>

The fix files are cumulative, that is, they contain *all* fixes for a particular *Sirius Mods* and *Model 204* version combination, whether or not a particular site has any of these fixes already installed. The *SirZap* utility provides a particularly convenient way to apply maintenance, since it detects if a ZAP has already been applied. If you do not use *SirZap*, you must manually remove any ZAPs that you have already applied from the fix files before attempting to apply the fixes.

SirZap is available without cost to all Sirius customers. *SirZap* is shipped on all product distribution cartridges, and it can also be downloaded from the Sirius web site “Download object files” page:

<https://sirius-software.com/maint/objlist>

Even if you choose re-linking from fresh object files as your approach to maintenance, the *Sirius Mods* fix files are useful for tracking the various bug fixes and enhancements that are distributed. All fixes in these ZAP files are preceded by comments describing the problem(s) being fixed. You can read these sequentially to determine if any new fixes might be applicable or critical to a site, or you can search them for keywords describing a symptom of a problem encountered at a site.

System Requirements

This chapter lists the various system components required by the *Sirius Mods*.

As of version 6.9 of the *Sirius Mods*, the support of *Model 204* versions prior to V5R1.0 is discontinued. Therefore, this manual restricts the discussion of versions of *Model 204* to V5R1.0 and later, even though, at the time of release of many versions of the *Sirius Mods*, an earlier version of *Model 204* was supported.

The *Sirius Mods* require the following components:

- Mainframe operating systems:
 - MVS/SP Version 1.3 or later (or MVS/XA, MVS/ESA, OS/390, or z/OS), or
 - VSE Version 2 or later, or
 - CMS (releases currently supported by IBM) running under VM/ESA, or z/VM
- *Model 204* Version 5 Release 1.0 or later.
- If you are also installing Janus products, one of the following mainframe TCP/IP implementations:
 - IBM TCP/IP for VM or MVS - Version 2 or later
 - IBM TCP/IP for OS/390 1.3 or later
 - InterLink TCP/IP for MVS - Version 1.1 or later
 - TCP/IP for VSE (Connectivity Systems, Inc., Columbus, OH) - Version 1 Release 4.0 or later (starting with version 6.1 of the *Sirius Mods*)

You can download the *Sirius Mods* object files from the Sirius Software web site (<http://sirius-software.com>). The download process requires a userid and password.

1. Click the **Support** navigation link to go to the Customer Service page (<http://sirius-software.com/support.html>), a public page that explains how to get the required userid and password and that also contains links to the various download pages.
2. Click the **Download object files** link to go to a protected page (<https://sirius-software.com/maint/objlist>) that contains a dynamically-generated list of the various Sirius products that you may download.

The page also contains a **Click here** link that causes the page to be re-displayed with detailed download and installation instructions. These instructions supercede any information in this manual.

3. On the **Download object files** page, select the link to the currently available *Sirius Mods* object file for the version you require. This file is used for all the versions of *Model 204* that are currently supported by the *Sirius Mods*.

MVS Installation from Cartridge

The *Sirius Mods* are distributed on a single installation cartridge. In order to install the *Sirius Mods* under MVS, you must first Load the three standard labeled files from the product cartridge. The following job will perform this function when you modify it for your local requirements:

```

//*
//*      Load Sirius products from distribution cartridge
//*
//LOAD   EXEC PGM=IEBCOPY,REGION=4M
//T1     DD UNIT=3480,LABEL=(1,SL, or z,DSN=SIRIUS.LIB,
//        VOL=SER=SIRIUS,DISP=(OLD,PASS)
//T2     DD UNIT=3480,LABEL=(2,SL),DSN=SIRIUS.LOAD,
//        VOL=SER=SIRIUS,DISP=(OLD,PASS)
//T3     DD UNIT=3480,LABEL=(3,SL),DSN=SIRIUS.ULSPF,
//        VOL=SER=SIRIUS,DISP=(OLD,PASS)
//D1     DD DISP=(NEW,CATLG),SPACE=(CYL,(10,2,5),
//        DSN=SIRIUS.LIB,UNIT=SYSDA,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//D2     DD DISP=(NEW,CATLG),SPACE=(CYL,(5,2,2),
//        DSN=SIRIUS.LOAD,UNIT=SYSDA,
//        DCB=(RECFM=U,LRECL=0,BLKSIZE=6144)
//D3     DD DISP=(NEW,CATLG),SPACE=(CYL,(10,10,2),
//        DSN=SIRIUS.ULSPF,UNIT=SYSDA,
//        DCB=(RECFM=F,LRECL=6184,BLKSIZE=6184)
//*
//SYSPRINT DD SYSOUT=*
//SYSIN   DD   *
COPY INDD=T1,OUTDD=D1
COPY INDD=T2,OUTDD=D2
COPY INDD=T3,OUTDD=D3
/*

```

Of course you can change the name for the datasets identified in the job by **D1**, **D2**, and **D3**, but these installation instructions will refer to them by the name shown in the sample JCL. The dataset **SIRIUS.LIB** will contain members like the following:

<i>Member</i>	<i>Function</i>
SIROBJ53	Object file for linking with <i>Model 204</i> Version 5 Release 3.
SIROBJ51	Object file for linking with <i>Model 204</i> Version 5 Release 1.
SIROBJ42	Object file for linking with <i>Model 204</i> Version 4 Release 2.0.
LNKSIR53	Linkedit JCL for <i>Sirius Mods</i> with <i>Model 204</i> Version 5 Release 3.
LNKSIR51	Linkedit JCL for <i>Sirius Mods</i> with <i>Model 204</i> Version 5 Release 1.
LNKSIR42	Linkedit JCL for <i>Sirius Mods</i> with <i>Model 204</i> Version 4 Release 2.0.

Note: If a version of *Model 204* listed above is not supported on the version of the *Sirius Mods* you are installing, the relevant members for it will not be present on the cartridge. “[Summary of Changes](#)” on page v tracks the changes to the supported *Model 204* versions for the various versions of the *Sirius Mods*. Also note, as described in “[System Requirements](#)” on page 5, that various older versions of *Model 204* are no longer supported by any versions of the *Sirius Mods*.

To build the *Sirius Mods* load modules, perform the following steps:

1. Edit the linkedit JCL “LNKSIxxx” and change the JOB card as necessary for your environment. Change the M204LOAD DD card to point to your *Model 204* load library. This library should contain your production load module, with Early Warnings applied.

If you are using KNET TCP/IP, you'll need to link the KNET K-routines with the online module. To do this, add a DD statement to the LNKSIxxx JCL in SIRIUS.LIB:

```
//KNETOBJ DD DISP=SHR,DSN=KNET.Vxx.SOCKOBJ
```

Then add the following statement to the link-edit control statements:

```
INCLUDE KNETOBJ(KRTNS)
```

2. Change the SYSLMOD DD statement to point to the output load library. **It is highly recommended that you do not replace your unmodified *Model 204* ONLINE, BATCH204, IFAM1, and IFAM4 load modules with the new load modules.** This is so that you can apply CCA early warnings to both the unmodified and the custom load modules. By doing this you will have an unmodified load module with all early warnings applied to use in rebuilding your custom load module whenever you receive updated *Sirius Mods* object files.
3. Apply any of the customization zaps described in “[Customization Zaps](#)” on page 23. Use the ZAP command, or use *SirZap*, which is located in the second dataset on the installation cartridge (**SIRIUS LOAD**).
4. Test the newly created **ONLINE** load module. This involves bringing up an **ONLINE**, using the customized load module, and then logging in. Once you have logged in, issue the **SIRIUS** command to verify that the *Sirius Mods* components for all of the products you have ordered are correctly installed.

You are now ready to move the load modules created in the installation process to an area where they can be accessed by the appropriate *Model 204* job streams. This can be done by either copying the load module from one load module PDS to another, or by simply concatenating the library containing the custom *Model 204* **ONLINE** to the STEPLIB for the appropriate job(s).

CMS Installation from Cartridge

The *Sirius Mods* are distributed on a single installation cartridge, in VMFPLC2 DUMP format. In order to install the *Sirius Mods* under CMS, you must perform the following steps:

1. Define and format a minidisk to receive the UL/SPF distribution. This minidisk should have space for 10,800 4K-byte CMS blocks. A 60 cylinder minidisk on a 3390 is sufficient for this purpose. You should access this disk as your A disk. Access your *Model 204* disk as read/only, so none of the modifications will affect your current configuration.
2. Attach a cartridge drive at virtual address 181 to the owner of the installation minidisk defined in step 1. Mount the product distribution cartridge on this drive. The product distribution cartridge consists of 1 physical file in **VMFPLC2 DUMP** format:

File 1 Files required to install the *Sirius Mods*.

Load the files from File 1 with the 'VMFPLC2 LOAD' command.

3. Build a new load list for the *Sirius Mods* version of *Model 204*. You must add a line for object file SIROBxxx, where “xxx” is as described in the list of members of *SIRIUS.LIB* as shown in [“MVS Installation from Cartridge” on page 9](#); for example, **SIROBJ53** for *Model 204* Version 5 Release 3.

Sirius Software provides an EXEC to convert your existing load list. Invoke it as follows:

```
LOADCONV loadlist
```

where *loadlist* is the name of your *Model 204* ONLINE load list. LOADCONV will produce a new EXEC loadlist named “**loadlist EXEC SIR A**”.

Inspect this new load list for accuracy, place it on either your primary Sirius functions installation minidisk or on a local work minidisk, and change the filetype to “EXEC” with the CMS **RENAME** command.

4. Modify the “M204CCA PARMS” so the **ONLINE** module will be generated correctly. First, edit the file and find the label **ONLN:**. Following this label, but before the label **END.ONLN**, find the statement `start.mod = MLNK$`. Change the parameter from MLNK\$ to SLNK\$, then save the file on your A disk.

5. Create the custom **ONLINE** load module(s) appropriate to your site. To do this, you should have your standard *Model 204* installation disks accessed.

It is recommended that you build the custom *Sirius Mods* **ONLINE**(s) on the same disk on which you built the customized load list in the previous step. This is accomplished by accessing the appropriate minidisk as your A disk. This prevents you from overlaying your standard *Model 204* **ONLINE** load module. This also ensures that the custom load list that you built in the previous step will be used when you generate your new **ONLINE** load module.

Either issue the following command to build an **ONLINE** without saved segments:

```
M204GEN ONLINE
```

Or issue the following command to build an **ONLINE** with saved segments:

```
M204GEN ONLINE (SEGMENT NOSAVE
```

Note: The module you build in this step has **no** early warnings applied. Because of this the **NOSAVE** option is recommended when generating the **ONLINE** load module.

6. Apply early warnings to the load module(s) generated in the previous step. Use the **ZAP** command, or use *SirZap*, which is located on the installation cartridge.

You can continue to maintain the custom load module(s) much as you would the standard *Model 204* **ONLINE** load module: by simply applying early warnings to the custom *Sirius Mods* load module(s) as required. Similarly, any **ZAP** maintenance distributed by Sirius Software for the *Sirius Mods* can be applied to the same load module.

Note: Keep all CCA Early Warnings that have been applied in an easily retrievable file. You will need to reapply all CCA early warnings whenever you relink, as for example when you receive new *Sirius Mods* object files.

7. Apply any of the customization zaps described in “[Customization Zaps](#)” on page 23. Use the **ZAP** command, or use *SirZap*.
8. Install the saved segments if required. Use the **M204SAVE EXEC**, created by the *Model 204* installation and described in the appropriate CCA documentation.
9. Test the newly created **ONLINE** load module. Bring up an **ONLINE**, using the new **M204ONLN MODULE** or saved segments, and issue a **LOGON** and **LOGOFF**. This should verify that the installation has not introduced any new problems.

You are now ready to move the load modules created in the installation process to an area where they can be accessed by the appropriate *Model 204* service machines.

Defining Sdaemon Threads

Many of the Sirius products make use of special background threads to offload processing from the online user's thread. These *Model 204* threads that operate without terminals are called **sdaemon** threads, and they are analogous to IODEV=3 threads.

For proper operation of several Sirius products, you must have sufficient numbers of sdaemon threads defined in your **ONLINE** configuration. The Janus products *Janus Web Server*, *Janus Sockets* (for server ports), *Janus Specialty Data Store*, and *Janus Open Server* all use sdaemon threads to process client connections. Many *UL/SPF* products require sdaemon threads for brief units of work.

It is recommended that a site have at least 4 sdaemon threads if using any *UL/SPF* products, and have an sdaemon thread for each concurrent connection that will be maintained by a Janus product.

To define the sdaemon threads for an **ONLINE**:

1. Specify the IODEV number to be used for sdaemon threads.

Use the User 0 parameter SDAEMDEV. The SDAEMDEV parameter can be any odd number from 1 to 53.

Note: The IODEV number selected for SDAEMDEV will be unavailable for its normal function. Thus if SDAEMDEV is set to 7, IODEV 7's will not be available for their normal function (VTAM full screen support).

The recommended setting for SDAEMDEV is 15, *unless* a site is using BTAM TTY terminals. Adding a line to the User 0 parm card like the following will indicate that any IODEV 15 thread is to run as an sdaemon thread:

```
SDAEMDEV=15
```

2. Increase NUSERS by the number of sdaemon threads to be defined.

Note: Increasing NUSERS might require adding space to the *Model 204* server datasets.

3. Add IODEV statements to CCAIN for each sdaemon thread.

It is recommended that the LECHO parameter be set to 0 for sdaemon threads.

The following statements define six sdaemon threads (assuming SDAEMDEV was set to 15):

```
IODEV=15 , POLLNO=1 , NOTERM=6 , LECHO=0  
IODEV=15 , POLLNO=2  
IODEV=15 , POLLNO=3  
IODEV=15 , POLLNO=4  
IODEV=15 , POLLNO=5  
IODEV=15 , POLLNO=6
```

 CHAPTER 7 *Fast/Unload User Language Interface*

The *Fast/Unload User Language Interface* (\$FUNLOAD) has some special installation requirements:

1. To make the *Fast/Unload User Language Interface* available, the User 0 parameter FUNTSKN must be set to a non-zero value.

FUNTSKN is the maximum number of concurrent *Fast/Unload* tasks that will be allowed in an Online address space. If FUNTSKN is 0, the User Language interface to *Fast/Unload* will not be available. When setting FUNTSKN, increase NSUBTKS by the value specified for FUNTSKN.

2. The *Fast/Unload* load module (usually called FUNLOAD) must be made accessible to the Online.

- For MVS systems, this means that the *Fast/Unload* load module (usually called **FUNLOAD**) must be placed in a library that is part of the STEPLIB concatenation for the **ONLINE**. This means either adding the *Fast/Unload* load library to the STEPLIB concatenation for the **ONLINE** job step, or moving the *Fast/Unload* load module into one of the libraries currently in the STEPLIB concatenation.

Note: Concatenating the *Fast/Unload* load library to the standard libraries will lose APF authorization for the **ONLINE** unless the *Fast/Unload* load library is added to the APF authorized library list.

- For VM systems, this means either accessing the *Fast/Unload* disk from the **ONLINE**, or moving the *Fast/Unload* load module onto a disk already accessed by the **ONLINE** (the Y disk might make sense at some sites).

3. If the *Fast/Unload* load module is not called FUNLOAD, the User0 parameter called FUNPGM must be set to the name of the *Fast/Unload* load module.

4. *Fast/Unload* log information can go either to the standard *Model 204* audit trail (CCAAUDIT) or to a *Fast/Unload*-specific audit trail (FUNAUDIT).

The action required to direct output to FUNAUDIT depends upon the operating system type, as follows:

- For MVS systems, add a DD card for FUNAUDIT to the JCL for the **ONLINE** job step. The following is an example of a valid DD card for FUNAUDIT:

```
//FUNAUDIT DD SYSOUT=*
```

- For VM systems, add a FILEDEF for FUNAUDIT to M204FDEF EXEC (or to whatever exec does FILEDEFs for the Online). The following is an example of a valid FILEDEF for FUNAUDIT:

```
FILEDEF FUNAUDIT DISK FUNAUDIT LISTING A (LRECL 133 BLOCK 4096 RECFM VB
```

For more information on the *Fast/Unload User Language Interface*, refer to the ***Fast/Unload Reference Manual***.

The RESTART utility allows an operator to force a user restart in an ONLINE address space from an operator console. It is intended to provide an alternative to the CANCEL system command. It is not intended as a means of solving performance problems or “bumping” users. Sirius Software recommends that this utility only be used when the ONLINE would ordinarily be cancelled. In this case, the worst that could happen is that the online would come down instantly. The next worst case is that the ONLINE would come down after going through standard *Model 204* termination thus eliminating the need to run recovery. If the restart is successful the ONLINE will continue operation after a probable SNAP dump which will help isolate the cause of the original problem. One should, of course, indicate how the SNAP was generated when reporting the problem to CCA product support.

For current users of the RESTART utility, it should be noted that the means of invoking the RESTART utility and the RESTART utility commands have been changed in version 3.0 of the *Sirius Mods*. For users of SIRTUNE, SIRTUNE provides identical functionality to the RESTART utility. As such, if the RESTART utility is started in an **ONLINE** (or **BATCH204**) job that is running under SIRTUNE, the RESTART utility will always defer to SIRTUNE and allow SIRTUNE to provide the required functionality. To simplify this coexistence, the RESTART utility commands are identical to the SIRTUNE SMSG/MODIFY commands, except for commands that have no applicability when not running SIRTUNE (CLOSE, SAMPLE and STOP).

The RESTART utility is integrated with the *Sirius Mods* and requires no additional installation. To enable the RESTART utility, the RESTARTU parameter must be set to 1. The RESTARTU parameter must be passed on the EXEC card (under MVS) as a PARM or as a parameter to the command that invokes the M204CMS load module (under CMS). Setting the RESTARTU parameter in CCAIN will have no effect.

The RESTART utility allows users outside the *Model 204* address space to issue certain commands while the **ONLINE** or **BATCH204** job is running. Users can be authorized to do this with the RESTAUTH command. Since under MVS it is impossible to determine the originator of a MODIFY command, any user that is capable of issuing a MODIFY command is considered to be authorized by the RESTART utility.

Under CMS, a user must be authorized to issue RESTART utility commands. This authorization is accomplished via the RESTAUTH command. RESTAUTH is a *Model 204* command that can only be issued by a system manager or system administrator. The RESTAUTH command should be followed by a list of userids that are authorized to issue RESTART utility commands. For example:

```
RESTAUTH SKIPPER GILLIGAN
```

authorizes CMS users SKIPPER and GILLIGAN to issue RESTART utility commands (via SMSG). Wildcards can also be used in the userids. An asterisk matches any group of characters and a question mark matches any single character. For example

```
RESTAUTH PROF* *ANN GING?? *HOWE*
```

authorizes any CMS userid that begins with the letters 'PROF', ends with the letters 'ANN', begins with the letters 'GING' and is exactly six characters long, or has the letters 'HOWE' anywhere in it to issue RESTART utility commands (via SMSG).

Under MVS, RESTART utility commands can be issued via the MODIFY operator command. This command can be issued at an operator console or under SDSF or equivalent virtual console system. For example, to issue the STATUS command to the RESTART utility running under job PRODONLN from SDSF's LOG screen, one can simply type:

```
/MODIFY PRODONLN,STATUS
```

or

```
/F PRODONLN,STATUS
```

Responses to MODIFY commands go to the system log and should be viewable under SDSF.

Under CMS, RESTART utility commands can be issued via the (CP) SMSG command. For example, to issue the STATUS command to the RESTART utility running on a virtual machine named PRODONLN, one can simply enter:

```
SMSG PRODONLN STATUS
```

or

```
CP SMSG PRODONLN STATUS
```

Responses to SMSG commands are sent via MSGNOH if the *Model 204* service machine is authorized to use MSGNOH and via MSG otherwise.

8.1 Commands supported by the RESTART utility

Available MODIFY/SMSG commands are listed here. Optional statement parameters are enclosed in brackets ({}). Alternatives are separated by a vertical line (|).

8.1.1 BUMP *user_num*

The BUMP command requests that the RESTART utility issue the equivalent of the *Model 204* BUMP command against the indicated user number. This command is useful if a high priority user is looping in an Online, making it impossible for anyone else to do anything. To “bump” user number 18, for example. issue the command:

```
BUMP 18
```

Unlike the *Model 204* BUMP command, the RESTART utility bump command only accepts a single user number and does not accept userids or file names. To determine the user number of the running user, issue the RESTART utility's MONITOR command. It is recommended that the MONITOR command be issued several times to ensure that a single user is indeed looping and that the performance problem is not simply the result of excess demand.

If a loop situation is caused by a *Model 204* bug, it is possible that a BUMP command will fail to force the looping user off the system. In this case, it might be necessary to issue the RESTART command to break the loop.

8.1.2 MONITOR

The MONITOR command requests that the RESTART utility return information about what is currently happening in *Model 204*. This command is especially useful if *Model 204* appears to be hung or looping and, making it impossible to log on to *Model 204* in order to determine the cause of the problem.

Information is returned for the main *Model 204* task and any subtasks (if the MP/204 feature is being used). The information returned indicates whether each task is running or waiting (corresponding to looping and hung situations respectively) and where in the *Model 204* load module the task is running or waiting. This latter piece of information is most useful to *Model 204* internals experts.

If any *Model 204* task is indicated as running, information is also provided on the user number and userid of the running user, the current activity (evaluating, compiling, etc.) and the name of the procedure that is currently running. This information can be used to determine whether a BUMP or RESTART command should be issued.

8.1.3 RESTART

The RESTART command requests that SIRTUNE issue the equivalent of a user abend in *Model 204* for the indicated user or task. This command should be used only as a last resort when a *Model 204* Online is looping or hung and all efforts to clear up the situation have failed (see the BUMP command). In this situation, the only options left are to cancel (FORCE under CMS) the run or to issue the RESTART utility's RESTART command. The RESTART command is preferable because:

- The Online might intercept the abend, possibly take a snap dump, and then continue running.
- Even if the Online does not continue, it might at least intercept the abend and terminate “cleanly”, preventing files from being broken and eliminating the need to run recovery. Cancelling or forcing the Online ensures that any file with active updating transactions will be left broken and that recovery will have to be run before the Online files can be used again.
- The RESTART command will generally result in a CCASNAP being taken rather than (or in addition to) a SYSMDUMP, SYSUDUMP or VMDUMP. CCASNAPs are generally easier to deal with than other types of dumps.

The format of the RESTART command is:

```
RESTART abend_code USER user_num | TASK task_num
```

The *abend_code* must be a 1- to 3-digit hexadecimal code that indicates the abend code to be used for the artificially generated abend. Under CMS, the abend code must be between 0C1 and 0CF. Any dumps will indicate the specified abend code. It is important to inform support personnel that are examining the dump that the abend code was artificially generated by the RESTART utility and that the situation was actually a hung or looping Online.

To prevent accidentally terminating a user or an Online, a user number can be specified on the RESTART command. When a user number is specified on the RESTART command, no simulated abend will occur if the indicated user is not running in the Online. For example:

```
RESTART 0C4 USER 23
```

simulates a 0C4 abend if user 23 is running. If user 23 is not running, no abend will be simulated.

If no user is running (a hung Online) it might be necessary to issue the RESTART command with a task number instead. Unless running MP/204 the task number must be specified as 0. If MP/204 is running, the task number must be 0 (for the maintask) or the subtask number (from 1 to NMPSUBS) to be restarted. For example

```
RESTART 0A9 TASK 0
```

simulates a 0A9 abend on the *Model 204* maintask. Note that it is almost always preferable to specify a user number on the RESTART command rather than a task number.

8.1.4 STATUS

The STATUS command requests the current status of the RESTART utility. The response to the STATUS command indicates the number of restarts that have been performed (hopefully 0) since the Online started.

Customization Zaps

In some cases, you may wish to change the default behavior of the *Model 204* load module. The zaps described in this chapter allow you to do this. You can apply these zaps using the standard zap approach for your installation.

The zaps are provided as one or more *examples*, showing the bit or bits that must be set to achieve the desired result. If you are using multiple customization features within the same byte, you must use the combined bits of those features in the VER and REP data of the zap.

Note: These zaps cannot be converted and invoked with the *Model 204* *ZAP command; they must be applied to the *Model 204* load module (before it is executed, of course).

9.1 Disable or restrict APPDATE command

The *Sir2000 User Language Tools Reference Manual* describes how various APPDATE commands in the user 0 input can restrict or disable subsequent use of APPDATE. You can also apply a customization zap to achieve this result for all executions of a load module. If, for example, you wanted to disallow the use of the APPDATE command in a production load module, or if you wanted to only allow one system-wide APPDATE setting for the entire *Model 204* run, you can use a zap based on one of the examples below.

```
* -----
* Example: disable any execution of APPDATE
*
* If you apply the following zap, any APPDATE
* command will cause an error message to be
* issued and the command otherwise is ignored.
*
* Note: this example assumes you have no other
* customization zaps; if you do, include those
* bits in the VER and REP data.
*
NAME ONLINE SIRK$
VER 070E 00
REP 070E 80
* ----- End of zap -----
```

```
* -----
* Example: restrict APPDATE to pre-user 0 params
*
* If you apply the following zap, the APPDATE
* command is only allowed prior to the user zero
* parameter line, and any other use of APPDATE
* will cause an error message to be issued and
* the command otherwise is ignored.
*
* Note: this example assumes you have no other
* customization zaps; if you do, include those
* bits in the VER and REP data.
*
NAME ONLINE SIRK$
VER 070E 00
REP 070E 40
* ----- End of zap -----
```

These zaps can only be used with version 5.4 and later of the *Sirius Mods*.

9.2 Disable SIRIUS command generated during startup

```
* -----
* Example: disable SIRIUS command during startup
*
* If you apply the following zap, the SIRIUS
* command is not issued during startup; normally
* this command displays the mods version, etc.,
* on the audit trail and CCAPRINT.
*
* Note: this example assumes you have no other
* customization zaps; if you do, include those
* bits in the VER and REP data.
*
NAME ONLINE SIRK$
VER 070F 00
REP 070F 80
* ----- End of zap -----
```

This zap can only be used with version 5.4 and later of the *Sirius Mods*.

Notes:

1. You can also place either or both the following commands before the user zero parameter line in CCAIN:

```
MSGCTL MSIR.0447 NOTERM
MSGCTL MSIR.0447 NOAUDIT
```

The first of these eliminates the messages from the user zero normal output (CCAPRINT), and the second eliminates them from the audit trail.

2. See “[Suppress product expiration warnings from operator console](#)” for another zap to suppress the MSIR.0427 operator messages warning about product expiration.

9.3 Suppress product expiration warnings from operator console

If you receive operator console MSIR.0427 messages warning about the expiration of one or more Sirius products, you can disable these messages for a period of one month (after which they will disappear, because the time period for issuing the warnings will have elapsed).

```
* -----  
* Example: stop operator expiration warnings  
*  
* A zap similar to the following will suppress  
* operator MSIR.0427 expiration warnings until  
* they are no longer produced. The date, in  
* MMDDYYYY format, is the date when you want  
* to start suppressing them; usually just use  
* the current date when you apply the zap.  
*  
NAME ONLINE SIRK$  
VER 070A 0000,0000  
REP 070A 0713,2000 Suppress starting 13 Jul 2000  
* ----- End of zap -----
```

Do **not** literally copy the indicated ZAP — it won't work. The date in the ZAP must be 30 days or less before the current date for the message suppression to work.

