

---

# Fast/CRAM 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](mailto:support@sirius-software.com)

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

August 5, 2010

© 2010 Sirius Software, Inc.



---

## *Proprietary Notices*

*Fast/CRAM* is a proprietary product of Sirius Software, Inc.:

**Sirius Software, Inc.  
875 First Street, Suite 21  
Cambridge, Massachusetts 02139  
USA**

**Model 204®** is a proprietary product of Computer Corporation of America, a wholly-owned subsidiary of Rocket Software, Inc., which owns the trademark:

**Rocket Software Corporate Office  
M204 Division  
275 Grove Street  
Suite 3-410  
Newton, Massachusetts 02466-2272  
USA**



---

## Contents

<b>Proprietary Notices</b> . . . . .	<b>iii</b>
<b>Contents</b> . . . . .	<b>v</b>
<b>Summary of Changes</b> . . . . .	<b>vii</b>
Fast/CRAM Version 2.7 . . . . .	vii
<b>Chapter 1: Abstract</b> . . . . .	<b>1</b>
Fast/CRAM advantages compared to CRAM . . . . .	1
Versions and Compatibility . . . . .	1
<b>Chapter 2: System Requirements</b> . . . . .	<b>3</b>
<b>Chapter 3: Storage Usage</b> . . . . .	<b>5</b>
SQA Storage . . . . .	5
CSA Storage . . . . .	6
Private Area . . . . .	6
31-Bit Considerations . . . . .	7
<b>Chapter 4: Coexistence Considerations</b> . . . . .	<b>9</b>
<b>Chapter 5: Contents of Distribution Tape</b> . . . . .	<b>11</b>
SIRIUS.LIB . . . . .	11
SIRIUS.LOAD . . . . .	11
<b>Chapter 6: Installation</b> . . . . .	<b>13</b>
Loading Distribution Tape . . . . .	13
Defining Subsystem Name . . . . .	14
Customizing SVC Modules . . . . .	14
Zapping IFAM2 Control Commands . . . . .	14
<b>Chapter 7: SNAPFAST Utility</b> . . . . .	<b>17</b>
<b>Chapter 8: FASTREP Utility</b> . . . . .	<b>19</b>
<b>Chapter 9: SNAPSUBS Utility</b> . . . . .	<b>21</b>

---

**Contents**

---

**Appendix A: Date Processing** . . . . . **23**

**Index** . . . . . **25**

---

## *Summary of Changes*

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

### **Fast/CRAM Version 2.7**

Base version of this manual.



---

**CHAPTER 1** *Abstract*

The *Fast/CRAM* program product is a functional replacement for the Cross Region Access Method (CRAM) SVC distributed with the *Model 204* database management system. CRAM provides communications support between *Model 204* and teleprocessing monitors, such as CICS or TSO. CRAM also provides support for host language interface (IFAM2) programs executed in batch as well as the BATCH2 utility program.

## 1.1 **Fast/CRAM advantages compared to CRAM**

- *Fast/CRAM* consumes considerably less system resources. Total CPU savings range from 20% to 70%, depending upon the application. Savings realized by a CICS region can be significantly higher.
- *Fast/CRAM* does not require any subtasks. This removes the single-threading of CRAM OPEN and CLOSE and dramatically reduces the overhead encountered by CICS when many CRAM connections with *Model 204* are active.
- *Fast/CRAM* makes extensive use of SRB routines, which take full advantage of multiple-processor complexes.
- *Fast/CRAM* is totally compatible with MVS/XA and MVS/ESA. All storage is allocated “above the line”. 31-bit application programs are supported using the new CRAM protocol introduced with *Model 204* Release 9.0.
- *Fast/CRAM* is reliable. Extensive use of MVS recovery facilities, ensures that *Fast/CRAM* has no “hanging states” or integrity exposures. *Fast/CRAM* adheres to the installation responsibilities for MVS integrity as outlined by IBM.

## 1.2 **Versions and Compatibility**

Sirius Software has a strong commitment to backward compatibility with its products. What this means, is that any configuration that runs correctly with a version of *Fast/CRAM* will run correctly on subsequent versions of *Fast/CRAM*.

This document, the ***Fast/CRAM Installation Guide***, assumes that a site is running *Fast/CRAM* version 2.7 or later. Any documentation that requires a later version of *Fast/CRAM* will be clearly marked to indicate this. If a feature is not indicated as requiring any specific version of *Fast/CRAM*, it can be assumed that it is available as documented in all versions of *Fast/CRAM*; that is, all versions since version 2.7.



*Fast/CRAM* executes within the IBM operating systems: MVS/SP Version 1 (MVS/370), MVS/SP Version 2 (MVS/XA), and MVS/SP Version 3 (MVS/ESA). One set of modules supports all of these system types.

*Fast/CRAM* comprises a single SVC load module and an interface module. The SVC load module may be installed as a Type 2, Type 3, or Type 4 SVC. The interface module may be installed in the Link Pack Area (LPA), or a private library, and need not be authorized.

*Fast/CRAM* requires a subsystem name, reserved through an entry in an IEFSSNxx member of SYS1.PARMLIB. Both *Fast/CRAM* modules are totally re-entrant and refreshable and do not need to be fixed in real storage. The SVC load module is linked with AMODE 31, RMODE ANY, and always executes in 31-bit addressing mode under MVS/XA or MVS/ESA. The interface load module is linked with AMODE ANY, RMODE 24, and executes in the addressing mode of its callers.

*Fast/CRAM* allocates three different types of storage: SQA, CSA, and private area. All storage is allocated “above the line” in MVS/XA and MVS/ESA systems. The SQA storage is allocated in subpool 245 when a *Fast/CRAM* channel is opened. The amount of SQA storage allocated for each channel is 128 bytes plus the number of users times 560 bytes. CSA storage is allocated in subpool 231 the first time that a particular *Fast/CRAM* subchannel is used. The size of each CSA allocation is determined by the blocksize for the *Fast/CRAM* channel. This is either IFAMBS for IFAM2 threads, LOUPTB for IODEV=11 threads, or 4 + MAX (OUTMRL, INMRL, 252) for IODEV=29 threads. The private area storage is allocated in subpool 252, above the line.



---

**CHAPTER 3** *Storage Usage*

*Fast/CRAM* allocates storage in three distinct areas: CSA, SQA, and the private area. All of this storage is “above the line” in MVS/XA and MVS/ESA systems.

### 3.1 SQA Storage

*Fast/CRAM* allocates SQA space only when *Model 204* is initialized. The amount of SQA storage allocated for a *Model 204* online depends upon parameters specified for the job. Each *Model 204* online may establish up to three CRAM channels. One channel is for any host language threads (IODEV=23); One channel is for any full screen remote threads (IODEV=11); And one channel is for any non-full screen remote threads (IODEV=29).

SQA storage for each CRAM channel is allocated in a contiguous block when the channel is opened, and freed when the channel is closed. Channel open generally corresponds to *Model 204* initialization, while channel close generally corresponds to *Model 204* termination. The amount of SQA storage allocated depends upon the channel type, as follows (all quantities in bytes):

**IODEV=23** If any IFAM2 threads are defined, one CRAM channel is opened and the following amount of SQA storage is allocated:

$$((\text{Number of IODEV=23's}) * 560) + 128$$

**IODEV=11** If any remote full screen connections are defined, one CRAM channel is opened and the following amount of SQA storage is allocated:

$$(\text{NOTERM} * 560) + 128$$

**IODEV=29** If any remote non-full screen connections are defined, one CRAM channel is opened and the following amount of SQA storage is allocated:

$$(\text{NOTERM} * 560) + 128$$

Essentially, the SQA usage is 560 bytes per possible user. In an installation with twenty host language (IODEV=23) threads and fifty full-screen (IODEV=11) threads, the total SQA usage would be 39,456 bytes, or less than eleven pages.

## 3.2 CSA Storage

*Fast/CRAM* allocates CSA space for data buffers. This space is fetch-protected, unlike the space allocated by CCA's CRAM. This provides increased data security by preventing unauthorized users from examining *Model 204* data streams which may contain passwords and other sensitive data.

*Fast/CRAM* data buffers are not allocated when a channel is opened. Instead, they are allocated the first time that a particular connection is used. *Fast/CRAM* data buffers are only freed when the channel is closed, typically corresponding to *Model 204* termination. Thus, the number of *Fast/CRAM* data buffers allocated is determined by the high water mark for the number of concurrent users. CCA's CRAM allocates data buffers in the same way, and they are the same size.

The amount of CSA storage allocated depends upon the channel type, as follows (all quantities in bytes):

**IODEV=23** The following amount of CSA storage is allocated for each IFAM2 thread the first time it is used:

IFAMBS

**IODEV=11** The following amount of CSA storage is allocated for each remote full screen connection the first time it is used:

LOUTPB

**IODEV=29** The following amount of CSA storage is allocated for each remote non-full screen connection the first time it is used:

4 + MAX (OUTMRL, INMRL, 252)

## 3.3 Private Area

Each *Fast/CRAM* private area work area is 136 bytes long, and is allocated from subpool 252, with key zero. One work area is allocated in the *Model 204* region for each *Fast/CRAM* channel open. This work area is freed when the *Fast/CRAM* channel is closed. No other private area storage is allocated or freed by *Fast/CRAM*. *Fast/CRAM* reduces the SPCORE requirement by approximately 200 bytes per IODEV=23, IODEV=29, or IODEV=11 thread.

One *Fast/CRAM* work area is allocated in each application region each time a cram connection is established. This area is freed when the connection is broken. Individual *Fast/CRAM* data transfer calls do not allocate or free storage.

### **3.4 31-Bit Considerations**

When *Fast/CRAM* is used with MVS/XA or MVS/ESA, all storage is allocated “above the line”. Application programs may execute in 31-bit addressing mode, provided that either all data areas and either the CRAM user control block are “below the line”, or that the Release 9.0 CRAM protocol is used. Release 9.0, and later, versions of IFIF can be freely used for 31-bit application programs.

Versions of IFIF prior to Release 9.0 cannot be used with 31-bit applications. However, if *Fast/CRAM* is used with older versions of *Model 204* or IFIF, all *Fast/CRAM* data structures will still be allocated “above the line”.



## *Coexistence Considerations*

*Fast/CRAM* may be installed and used in parallel with CCA's CRAM. However, successful coexistence requires that the correct operational procedures be carefully followed. Simplicity argues for just one version of CRAM. Using two versions of CRAM will require two separate SVC numbers and two separate subsystem names.

CRAM applications LOAD a copy of IGCLM244 to determine the correct CRAM or *Fast/CRAM* SVC number. The key to coexistence is two separate versions of IGCLM244, which in turn identify two different SVC numbers. The “standard” version of CRAM would be identified by a copy of IGCLM244 contained in the link pack area or linklib, while the “test” copy would be identified by a copy of IGCLM244 contained in a separate library.

The test version would be invoked whenever *Model 204* or CICS included the special IGCLM244 library in their STEPLIB concatenation. Care will be required to ensure that each partner has the correct sequence of STEPLIB libraries. If *Fast/CRAM* is installed in this fashion, IGCLM244 should be placed in an APF authorized library so that SNAPFAST will work correctly.

If this approach is used, then a special version of the TSO interface may be required. Suppose that *Fast/CRAM* is being installed in a test mode. A special library containing just the *Fast/CRAM* version of IGCLM244 would then be added to the STEPLIB concatenation for *Model 204* and any of its partners (CICS, BATCH2, etc).

The *Fast/CRAM* IGCLM244 should also be linked into SYS1.COMDLIB or the LINKLIB concatenation as FASTRLD. A second copy of the TSO interface should then be linked into SYS1.COMDLIB, with a special name such as FASTFS. Then use the IMASPZAP service aid to DUMPT this second copy of the TSO interface and locate all occurrences of the string “IGCLM244”. Then zap these to “FASTRLD”.

Any time FASTFS is used, it will load FASTRLD instead of IGCLM244, thus using the *Fast/CRAM* SVC instead of the SVC identified by the link pack version of IGCLM244. Care must be taken to ensure that the respective CRAM SVC's are only invoked by “their own” versions of IGCLM244.



## ***Contents of Distribution Tape***

*Fast/CRAM* is distributed on a magnetic tape which also contains all the other Sirius products you purchased.

The tape contains three files, each in IEBCOPY format, but only the first two are used for *Fast/CRAM* installation. The relevant members for *Fast/CRAM* in these two files are described in the following two sections.

### **5.1 SIRIUS.LIB**

The first file, *SIRIUS.LIB*, will contain the following members:

*Member*      *Description*

**FASTCUST**    Example job for customizing and installing *Fast/CRAM*.

**FASTOSIV**    Example job to linkedit FASTSVC and FASTRLD on Fujitsu OSIV systems.

**FASTVOS3**    Example job to linkedit FASTSVC and FASTRLD on Hitachi VOS3 systems.

**FASTIBM**     Example job to linkedit FASTSVC and FASTRLD on IBM MVS systems.

**FASTSNAP**    Example job for executing the SNAPFAST and SNAPSUBS utility.

**SVCREP**        *Fast/CRAM* installation utility, used to dynamically update SVC table.

### **5.2 SIRIUS.LOAD**

The second file, *SIRIUS.LOAD*, will contain the following members:

*Member*      *Description*

**IGCLM244**    *Fast/CRAM* interface module replacement.

**FASTSVC**     *Fast/CRAM* SVC module.

**SNAPFAST** *Fast/CRAM* snapping utility (SNAPCRAM replacement). Also contains the SNAPSUBS utility.

**FASTREP** *Fast/CRAM* installation utility, used to dynamically update SVC table.

---

 CHAPTER 6 *Installation*

This section presents the steps required to install *Fast/CRAM*. If you are a user of a previous version of *Fast/CRAM*, see the **Release Notes** accompanying the tape, highlighting the changes in the new version, and listing any compatibility issues with the previous version.

The following steps assume that you are replacing CCA's CRAM. If you would like to reserve the ability to switch back and forth between *Fast/CRAM* and CCA's CRAM, please refer to "Coexistence Considerations" for suggestions.

## 6.1 Loading Distribution Tape

The following job can be used to load the *Fast/CRAM* distribution tape. Note that this job will also load any other Sirius products you have ordered. You must, of course, modify the job card and output dataset names to conform to local standards.

```

//LOADSIR JOB  Ø,CLASS=A,MSGCLASS=X,NOTIFY=ME
//*
//*      Load Sirius products from tape
//*
//LIB      EXEC PGM=IEBCOPY,REGION=ØM
//T1      DD  UNIT=TAPE,LABEL=(1,SL),
//          DISP=(OLD,PASS),DSN=SIRIUS.LIB
//T2      DD  UNIT=TAPE,LABEL=(2,SL),
//          DISP=(OLD,PASS),DSN=SIRIUS.LOAD
//T3      DD  UNIT=TAPE,LABEL=(3,SL),
//          DISP=(OLD,PASS),DSN=SIRIUS.ULSPF
//*
//D1      DD  DISP=(,CATLG),SPACE=(CYL,(1Ø,Ø,5),
//          DSN=SIRIUS.LIB,UNIT=SYSDA
//D2      DD  DISP=(,CATLG),SPACE=(CYL,(5Ø,Ø,2),
//          DSN=SIRIUS.LOAD,UNIT=SYSDA
//D3      DD  DISP=(,CATLG),SPACE=(CYL,(25,Ø,2),
//          DSN=SIRIUS.ULSPF,UNIT=SYSDA
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD  *
COPY INDD=T1,OUTDD=D1
COPY INDD=T2,OUTDD=D2
COPY INDD=T3,OUTDD=D3
/*
  
```

## 6.2 Defining Subsystem Name

The IEFSSNxx member of SYS1.PARMLIB is used to define subsystems. *Fast/CRAM* requires a subsystem definition without any initialization routine. Refer to the “SPL: Initialization and Tuning” IBM manual for the format of IEFSSNxx entries. The utility SNAPSUBS can be used to get a listing of all defined subsystems.

## 6.3 Customizing SVC Modules

Once the SVC number and subsystem name have been determined, the SVC load module and its interface module must be customized and then linkage edited into the appropriate library. This can be accomplished with the job stream FASTCUST, found in SIRIUS.LIB. This assumes that the *Fast/CRAM* load modules have already been restored into SIRIUS.LOAD, the SVC is to be installed in SYS1.LPALIB, the SVC number is 250, and the subsystem name is “M204”.

Note: This job first zaps the subsystem name and SVC number into the SVC load module and interface module, then links these two modules into SYS1.LPALIB with their appropriate names. For more information on SVC installation techniques, refer to the “SPL: Supervisor” IBM manual.

## 6.4 Zapping IFAM2 Control Commands

Various infrequently used IFAM2 control commands, for example IFAMFORCE, IFAMDRAIN, and IFAMSTART, can cause timing-related problems. This is because these commands violate CRAM protocols and create race conditions.

With the CCA CRAM, this can cause various hanging conditions or protection exceptions. *Fast/CRAM* intercepts these errors and takes an SVC dump and returns a completion code of 99 to the offending call.

The following job applies a zap to the Release 2.2 version of the *Model 204* module IFIL. This zap corrects most of the IFAM2 control command timing errors. This zap should not be applied to CMS systems.

```
//IFIIZAP JOB , 'FAST/CRAM', CLASS=A, MSGCLASS=A
/**
/** JOB TO ZAP IFII TO CORRECT CRAM PROTOCOL
/** VIOLATIONS.
/**
/** DO NOT APPLY THESE ZAPS TO A CMS VERSION
/** OF IFII, SINCE THEY REUSE A CMS-ONLY
/** PARAMETER LIST
/**
//ZAPS EXEC PGM=SIRZAP, REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=,
//SYSIN DD DISP=SHR, DSN=SIRIUS.LIB(V22)
//
```



---

**CHAPTER 7**    ***SNAPFAST Utility***

The SNAPFAST utility is provided for listing the SQA storage allocated by *Fast/CRAM*, as well as the *Fast/CRAM* SVC load module and interface module. If this job is run with SNAPFAST linked into an APF-authorized library, the *Fast/CRAM* CSA data buffers are snapped as well. This utility can be invoked with the following job, assuming that the *Fast/CRAM* load modules have already been restored into SIRIUS.LOAD.

```
//SNAPF    JOB    Ø, 'SNAP SQA/CSA', CLASS=A,
//          MSGCLASS=X
//FAST     EXEC   PGM=SNAPFAST, REGION=64K
//STEPLIB  DD     DSN=SIRIUS.LOAD, DISP=SHR
//SYSPRINT DD     SYSOUT=*
//SYSUDUMP DD     SYSOUT=*
//
```



***FASTREP Utility***

The FASTREP utility is provided for dynamically installing a version of *Fast/CRAM* without requiring a system IPL. For this utility to work, FAST.CRAM.LOAD must be APF-authorized, and the correct SVC number must have been zapped into the FASTSVC load module.

FASTREP verifies that the identified SVC number is the correct type, and that the identified subsystem name has been defined and is not already in use. If these conditions are met, the FASTSVC load module is loaded into the (extended) SQA, and the SVC table is dynamically updated.

Note that an IPL will still be required to install a new version of IGCLM244 into the link pack area, but *Fast/CRAM* can be tested with IGCLM244 in a private library. Please refer to “Coexistence Considerations” for more information on such a test environment.

The FASTREP utility can be invoked with the following job:

```
//FASTREP JOB  Ø, 'UPDATE SVC TABLE', CLASS=A,
//              MSGCLASS=X
//*
//* UPDATE SVC TABLE WITH MEMBER FASTSVC
//* FROM SYSLIB
//*
//SVCUPDTE EXEC PGM=FASTREP, REGION=64K, PARM=FASTSVC
//STEPLIB DD DSN=SIRIUS.LOAD, DISP=SHR
//SYSLIB DD DSN=SIRIUS.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//
```



***SNAPSUBS Utility***

The SNAPSUBS utility is provided for listing all currently defined subsystems. This utility can be invoked with the following job, assuming that the *Fast/CRAM* load modules have already been restored into FAST.CRAM.LOAD. The output from this job is a dump of the storage for each subsystem CVT.

```
//SNAPS    JOB  Ø,'SNAP SSCVT''S',CLASS=A,  
//          MSGCLASS=X  
//SSVCTS   EXEC PGM=SNAPSUBS,REGION=64K  
//STEPLIB  DD   DSN=SIRIUS.LOAD,DISP=SHR  
//SYSPRINT DD   SYSOUT=A  
//SYSUDUMP DD   SYSOUT=A  
//
```



---

**APPENDIX A** *Date Processing*

This chapter notes there are no date processing issues for *Fast/CRAM*; *Fast/CRAM* does not produce any results which depend on the content of any data which may be date values. Please note, however, that since you use a *Fast/CRAM* thread to send and receive data to and from Model 204, if that data contains two digit year date values, you must ensure that any application using that data has an algorithm or rule for unambiguously determining the correct century for the values.

To correctly use *Fast/CRAM* past the year 1999, *Fast/CRAM* version 2.7 or later is required.

Above and beyond the post-1999 requirements specific to *Fast/CRAM*, you must examine all uses of date values in your applications to ensure that each of your applications produces correct results.



---

*Index*

