



CSI

INTERNATIONAL

Original Printing .....01/20/2003  
Last Revised .....01/20/2003

# TABLE OF CONTENTS

---

---

<b>GETTING STARTED</b> .....	<b>1</b>
Introduction .....	3
Benefits of FSI-JOURNAL.....	4
What is Recovery?.....	4
Which Part is provided by FSI-JOURNAL? .....	5
Which Part is provided by FSI-RECOV? .....	5
Features of FSI-JOURNAL.....	5
Migration Requirements .....	6
Installation.....	8
<b>USING FSI-JOURNAL</b> .....	<b>11</b>
Considerations.....	13
Purpose of journaling .....	13
What to journal.....	13
What the journal looks like.....	13
When is the Journal Opened?.....	14
When is the Journal Closed?.....	14
Journal dataset JCL .....	14
Journal dataset management.....	14
Primary / alternate extent journals .....	15
Alternate indexes .....	16
VSAM Share Option 4 .....	16
Restrictions.....	17
Compatibility .....	18
Activating FSI-JOURNAL .....	19
Deactivating FSI-JOURNAL .....	21
Option Table for FSI-JOURNAL .....	23
Execution Time Overrides for FSI-JOURNAL .....	29
Commands and Operands.....	31
Syntax .....	31
Commands.....	32
DATASET Command.....	33
JOURNAL Command .....	36
IDENTIFY Command.....	43
CONTROL Command.....	45
START Command.....	49
STOP Command .....	50
GENERATE Command.....	51
EXEC Command .....	53
OPTION Command.....	55
SELECT Operand.....	56
<b>MESSAGES</b> .....	<b>59</b>
FSIJRNL-nnnn.....	61
<b>APPENDIX A</b> .....	<b>83</b>
How to Code Commands (Syntax).....	85

# TABLE OF CONTENTS

---

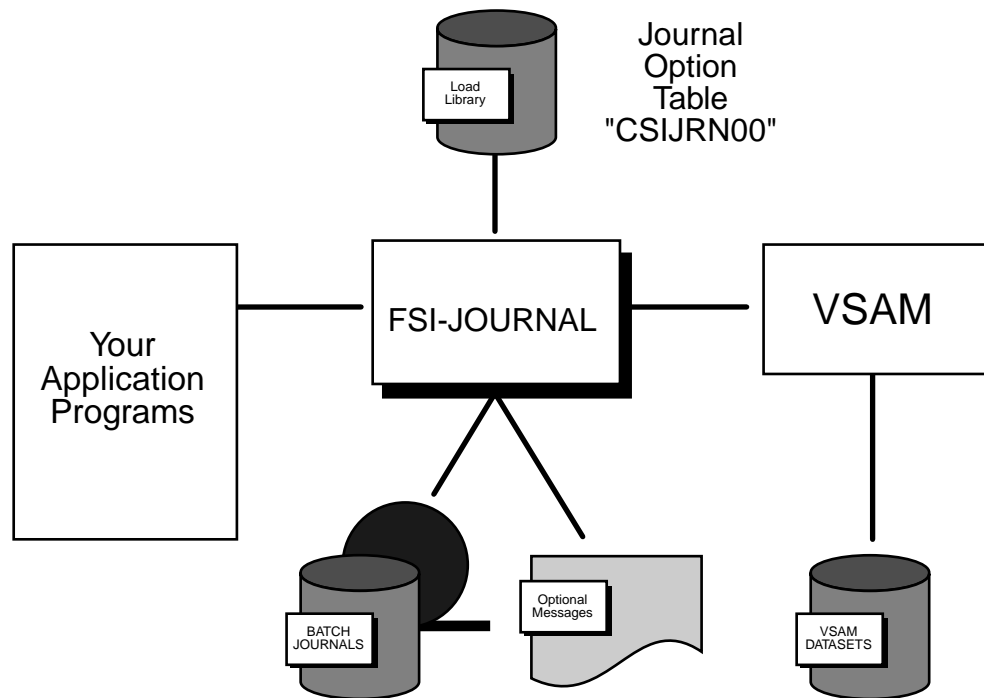
**APPENDIX B** ..... 89  
    Journal Record Formats ..... 91

## **GETTING STARTED**



## Introduction

FSI-JOURNAL is a system designed to provide your batch programs with the type of journaling support for VSAM datasets that is provided by CICS for your on-line programs. This is a companion system to FSI's CICS VSAM recovery system FSI-RECOV. The journals created by FSI-JOURNAL can be input to FSI-RECOV along with CICS created journals to provide total recovery for your on-line and batch systems.



**Overview of FSI-JOURNAL**

## Benefits of FSI-JOURNAL

- You do not have to modify or re-compile your application programs to get Batch Journaling support when you use FSI-JOURNAL.
- You will not have to re-run batch jobs because an on-line VSAM dataset has to be recovered. Batch updates can be merged with on-line updates for dataset recovery, archival or reporting purposes.
- Any number of VSAM datasets in the same jobstep can be journaled concurrently to one or more batch journals, which simplifies JCL and reduces storage requirements.
- Each journal can be defined as a single dataset which allows for the use of secondary allocation, or as a primary and secondary extent with automatic switching, similar to CICS.
- Journal datasets can be dynamically allocated by FSI-JOURNAL, eliminating the need to change jobstreams.
- Journaling messages can be generated to provide visual assurance that the journaling process is working properly. You won't get to the end of a job before finding out that no journal has been created.
- Batch Journaling is controlled by a single Option Table. You have complete control over which VSAM datasets are journaled, and when and where they are journaled.

In summary, FSI-JOURNAL, when used with FSI-RECOV, will save your company time and money when a VSAM dataset is lost or damaged.

## What is Recovery?

Forward Recovery is a process that restores a damaged dataset to the state it was in prior to the point of damage.

Damage may be caused by hardware or software failure, or by human error, and the extent of the dataset damage can vary from a single record in one dataset to many datasets in several systems. Accordingly, the recovery process could be different each time a failure occurs.

**Which Part is provided by FSI-JOURNAL?**

FSI-JOURNAL provides the facility to record VSAM dataset changes made by batch programs to separate datasets called journals. These journals can be used by a recovery package such as FSI-RECOV to perform the actual dataset recovery.

**Which Part is provided by FSI-RECOV?**

FSI-JOURNAL will not use these journal records to re-apply changes when a dataset has been destroyed. This is where FSI-RECOV comes in. After you have restored the damaged VSAM dataset (or datasets) from the latest backup tape, FSI-RECOV will re-apply changes made since the latest backup using the journaled records created by FSI-JOURNAL and/or CICS.

**Features of FSI-JOURNAL**

- All types of VSAM datasets can be processed, KSDS, RRDS, ESDS.
- Complete selection control over which VSAM datasets will be journaled from which programs.
- No modifications required to any application programs.

## Migration Requirements

The following sections cover the changes in the installation of FSI-JOURNAL that may be required if you are migrating from an earlier release.

### Changes for Release 3.0

- The load module names have been changed to a CSI prefix.
- You must regenerate your Journal Option Table (now CSIJRN00) due to an internal format change.

### Changes for Release 2.2

- The internal format of the Journal Option Table has changed. You must regenerate your CSIJRN00 table for this release of FSI-JOURNAL.
- A new parameter has been added to the GENERATE command. TERMEXIT, an optional parameter, has been added to control the closure of open journal files. With the TERMEXIT parameter specified, all journal files will be closed at Job-Step termination time rather than when the associated VSAM file(s) are closed. Closure of journals at Job-Step termination time eliminates some problems associated with system ENQUEing that occurs during a VSAM file closure.
- A new module, CSIJRNIX, has been added. This module is an end-of-step exit that is invoked when TERMEXIT is specified. CSIJRNIX gains control at Job-Step termination time to manage closure of all journal files. CSIJRNIX will also close all orphaned VSAM files being journaled that are found open at Step termination.
- The name of the Open/Close processor for FSI-JOURNAL has been changed from CSIJRN09 to CSIJRNIP.
- Module CSIJRN16 is no longer used.

### Changes for Release 2.1

- If you have jobsteps that open more than one journal dataset, you will need to review the ddnames assigned to these journals. FSI-JOURNAL now requires that journal ddnames be unique within the first 5 characters, if more than one will be required within the same jobstep.
- FSI-JOURNAL now supports journals defined with a primary and alternate extent. This provides an automatic switching facility for full journal files, similar to CICS. Refer to “Primary / alternate extent journals” on page 15 for more information.
- If a failure occurs during the dynamic allocation of a journal, a message is generated on the System Console, allowing the operator to correct the problem, if possible, and allow the application to continue.
- FSI-JOURNAL no longer uses the VSAM OPEN/CLOSE Router from Softworks, Inc. Refer to the installation section for instructions on the installation of the new router used by FSI-JOURNAL and other FSI software products.
- The new VSAM OPEN/CLOSE Router used by FSI-JOURNAL does not require that module CSIJRN09 be loaded from your LPA.
- The DD statement for **LPALIBM** used during the activation of FSI-JOURNAL, has been changed to **BIMLPA**.

### Changes for Release 2.0

- FSI-JOURNAL supports a user exit point that can be used to alter the normal journaling activity. Refer to source member CSIJRNXP for additional information.

## Installation

The FSI-JOURNAL programs described in this manual run under MVS, and are distributed on a tape or cartridge containing two datasets created by IEBCOPY. The first dataset contains pre-generated load modules to be transferred to your Load Library. The second dataset contains source modules and sample JCL to be transferred to your Source Library.

### Load Modules:

<b>CSIJRNL</b>	Batch Journaling control module
<b>CSIJRN01</b>	Override anchor module
<b>CSIJRN07</b>	VSAM interface module
<b>CSIJRNIP</b>	VSAM interface module
<b>CSIJRN11</b>	Journaling module
<b>CSIJRNIX</b>	End-of-step exit module

### Source Modules:

<b>CSIJRN00</b>	Journal Option Table Sample
<b>CSIJRNXP</b>	User Exit Parameter Definitions
<b>CSIJRNXS</b>	Copybook for (short) format journal records
<b>CSIJRNXL</b>	Copybook for (long) format journal records

### Installation Step 1.

Create the following JCL which will load the distribution tape to your Source and Load Libraries.

```
//JOBID      JOB (accounting),'LOAD FSI-JOURNAL',  
//              CLASS=A,MSGCLASS=*,MSGLEVEL=(1,1)  
//*  
//*  LOAD FSI-JOURNAL DISTRIBUTION TAPE  
//*  
//IEBCOPY EXEC PGM=IEBCOPY  
//SYSPRINT DD SYSOUT=*  
//TAPEF1   DD DSN='CSIJRNL.LOADLIB',UNIT=TAPE,DISP=OLD,  
//              LABEL=(1,SL),VOL=(,RETAIN,SER=FSISYS)  
//TAPEF2   DD DSN='CSIJRNL.SOURCLIB',UNIT=TAPE,DISP=OLD,  
//              LABEL=(2,SL),VOL=REF=*.TAPEF1  
//LOADLIB  DD DSN='authorized.linklist.library',DISP=SHR  
//SRCELIB  DD DSN='your.source.library',DISP=SHR  
//SYSIN    DD *  
          COPY I=((TAPEF1,R)),O=LOADLIB  
          COPY I=((TAPEF2,R)),O=SRCELIB  
//*  
//
```

### Installation Step 2

The programs for FSI-JOURNAL must be executed from an **APF Authorized Load Library**. To prevent the need for changes to all of your Batch JCL, the FSI-JOURNAL Load Library should also be placed in the Linklist. This means that the library must be listed in both members, **IEAAPFxx** and **LNKLSTxx**, in **SYS1.PARMLIB**.

### Installation Step 3

FSI-JOURNAL uses an SVC Routing Facility to intercept VSAM OPEN and CLOSE processing. This facility allows several FSI products to intercept OPEN and/or CLOSE processing in a prescribed order. The Routing Facility is implemented transparently in FSI-JOURNAL by means of the START and STOP commands of program CSIJRNL.

The SVC Routing Facility is shipped separately on a tape labeled: **BIMVSR/MVS**. A cover letter should have been shipped with the tape that will explain how to install it.

**Do NOT proceed to Installation Step 4 until you confirm that the SVC Routing Facility has been installed.**

**Installation Step 4**

FSI-JOURNAL has now been installed. You will need to generate a Journal Option Table to control which VSAM datasets are to be journaled and from which Batch Jobs. Review the section "Using FSI-JOURNAL" for an overview of Batch Journaling and for assistance in generating your Journal Option Table. FSI-JOURNAL must be activated before it can be used. Refer to the section "Activating FSI-JOURNAL" on page 19.

## **USING FSI-JOURNAL**



## Considerations

Before the description of how to set up and use FSI-JOURNAL, we have listed a number of things to consider. Some of these issues may influence your approach to Batch Journaling, and we recommend that you read the remainder of this section before setting up FSI-JOURNAL on your system. Only Batch Journaling is discussed here, as CICS journaling and dataset recovery is fully covered in the documentation for FSI-RECOV (VSAM Recovery system).

## Purpose of journaling

Journaling itself does not have any benefit. FSI-JOURNAL is designed to be used with FSI-RECOV to provide VSAM dataset recovery. However, the journal records that it creates can also be used by programs you may write to suit other needs. FSI-JOURNAL is totally independent of the application programs that actually update the VSAM datasets. This makes it ideal to use for implementing “add-on” applications that are driven by the journaled updates of existing systems.

## What to journal

FSI-JOURNAL can selectively journal any type of VSAM I/O: read only, read for update, update, add, delete. The type of I/O that you want journaled is described to FSI-JOURNAL in a control table, on a dataset-by-dataset basis if necessary. Forward Recovery requires only updates, adds and deletes to be journaled, but if you use the journal for other purposes, you may need to journal the read requests also. If you journal read for update requests, FSI-JOURNAL will normally journal only the records which are actually updated or deleted. This prevents batch programs, which usually update a small percentage of the records they read, from filling the journal with unneeded records.

## What the journal looks like

The journals produced by FSI-JOURNAL share the same format as FSI-RECOV's OUTPUT BACKUP, and are written as sequential tape or disk datasets. The journal record format is described in Appendix B.

## When is the Journal Opened?

FSI-JOURNAL opens the journal dataset when the first VSAM request (RO, RU, WN, WD) is made by the application program. If the application program does not perform any I/O to a journaled VSAM dataset, the journal will never be opened.

## When is the Journal Closed?

FSI-JOURNAL closes all open journals for a jobstep when the last open VSAM dataset is closed by the application program. If the logic flow of the application causes a single VSAM dataset to be opened and closed repetitively, or several VSAM datasets to be processed one at a time, the journal will be opened and closed several times during the jobstep. This process can be prevented by use of the JCLOSE operand of the EXEC command. Refer to the section "Execution Time Overrides for FSI-JOURNAL" on page 29 for more information.

## Journal dataset JCL

When FSI-JOURNAL opens a journal dataset, a DD statement for the journal must be provided in the JCL, unless the ALLOCATE operand has been specified for the journal in the Journal Option Table.

If a DD statement is provided, it must include a DCB=BLKSIZE=nnnnn. Single extent journals are created in a VBS format, dual extent journals are created in a VB format.

The ALLOCATE option causes the journal to be dynamically allocated at execution time. Refer to the ALLOCATE operand of the JOURNAL command on page 38 for more information.

## Journal dataset management

Since you probably will have multiple jobs creating journal datasets that you will want grouped together, you need to implement a method for combining these journal datasets. FSI-RECOV can be used to merge FSI-JOURNAL (and CICS) journal datasets into a single "journal backup" dataset. A standard sort/merge utility could also be used to combine multiple FSI-JOURNAL datasets.

## Primary / alternate extent journals

FSI-JOURNAL incorporates the ability to manage multiple journal extents for a single journal entity much as CICS manages its journals. Command operands allow users to define a journal with two extents eliminating the possibility that the application job will be terminated if a journal fills. With dual extents defined, the user has the ability to offload the filled extent while the application job is running. By archiving the data on the filled extent, the user frees the journal to be overwritten when the alternate or second extent has been filled and FSI-JOURNAL requests authority to switch back to the other extent.

Console messages to the operator requiring responses, allow the operator to control the use of the journals defined for the application's job. There is a danger that the operator may fail to start an archive job to offload the filled journal extent, and merely allow FSI-JOURNAL to overlay the previous extent's journal records. This danger is not minimized by any built-in controls in FSI-JOURNAL. The intent of this feature is to eliminate the loss of time and resources when jobs failed because available journal space was insufficient.

There is a caveat in the dual extent journal scheme. The BSAM logic used to write to the journal has no support for secondary extents allocated to the file space defined. This means that the dual extents of the journal can only handle the primary space allocated to the file. Therefore, sufficient primary space must be available on the DASD device to allow a sufficiently large journal extent to be defined for both the first and second extents of the dual extent journal.

Dual extent journals cannot be used by more than one step in a multi-step job, unless the journals are archived between application steps. FSI-JOURNAL does not have adequate controls to carry journal extent status information across step boundaries. Therefore, for multiple step jobs, a separate set of dual journal extents must be setup to handle each step of the job.

FSI-JOURNAL creates single extent journals in a VBS format (variable spanned blocked). This allows single records to span multiple blocks. Dual extent journals cannot allow this, because a single record could begin in one extent, and end in another. Therefore, dual extent journals are created in a VB format (variable blocked). This means that you must choose a BLKSIZE for your dual extent journals that will be large enough to contain the largest VSAM record to be journaled, plus the journal record prefix generated by FSI-JOURNAL. Refer to "APPENDIX B" starting on page 89 for the formats and lengths of the journal record prefix.

This feature is activated via the MULTIPLE, PSUFFIX and ASUFFIX operands of the JOURNAL command.

### Alternate indexes

FSI-JOURNAL processes an alternate index as a Key Sequence Data Set (KSDS). The name of the PATH that the application program opens is the dataset name used by FSI-JOURNAL. The alternate key, used by the application program to access the data records, is the key that is recorded in the journal.

### VSAM Share Option 4

Concurrent updates between batch programs and CICS transactions cannot be recovered properly, even with VSAM share option 4, because the time-stamp on CICS journal records is accurate only to one tenth of a second. This is not accurate enough to guarantee the correct sequencing of updates during recovery when the same VSAM record is updated by CICS and a batch program during the same tenth of a second.

## Restrictions

FSI-JOURNAL is designed to journal the VSAM activity of batch application programs. Nearly all types of VSAM access are supported without modification of user programs. There are, however, a few situations in which journaling is not supported.

- **CICS.** Since CICS has its own journaling facility, FSI-JOURNAL specifically excludes all VSAM requests made by CICS. This also applies to any dataset that is opened by a program executing in the same partition as CICS.
- **Control Interval Access.** Any VSAM dataset requests that use control interval access will not be journaled. Control interval access does not deal with logical records, and therefore cannot be journaled. This will normally affect only system utility programs.
- **Chained RPLs.** This method of issuing VSAM requests is available only to assembler language programs, and is normally used only by system utility programs. FSI-JOURNAL will journal only the request specified by the first or only RPL.
- **VSAM share option 4** cannot be relied on to provide recovery for concurrent updates between batch programs and CICS transactions because the time-stamp on CICS journal records is accurate only to one tenth of a second. This is not accurate enough to guarantee the correct sequencing of updates during recovery when the same VSAM record is updated by CICS and a batch program during the same tenth of a second.

## Compatibility

FSI-JOURNAL generally works with most other VSAM file support products. However, care must be exercised when initiating the products to ensure that the order of support functions is logical. Generally, VSAM compressors and other similar products should be initiated before FSI-JOURNAL.

Softwork's VSAM compression package, "Capacity Plus for VSAM", uses a VSAM "registry" control file that should be excluded from journaling by FSI-JOURNAL. Add a "DATASET" command for the registry control file to your Journal Options Table specifying "JREQ(NONE)". This will prevent problems during VSAM open requests when FSI-JOURNAL is active. Refer to the section "Option Table for FSI-JOURNAL" on page 23.

LEGENT's file compression product, "Extend-DASD", will only work with FSI-JOURNAL for VSAM files that are both journaled and compressed. That is, both EXTEND-DASD and FSI-JOURNAL are "managing" the file. If EXTEND-DASD does not manage a VSAM file, FSI-JOURNAL will lose control of it during file open. This is because EXTEND-DASD "unhooks" FSI-JOURNAL for VSAM files that are not compressed.

## Activating FSI-JOURNAL

FSI-JOURNAL is activated as an independent extension of the operating system. From this vantage point, it monitors the opening and closing of each VSAM dataset, and interrogates an option table to determine if the dataset is to be journaled. FSI-JOURNAL is activated system-wide, that is, it is in effect for all regions of your MVS system. FSI-JOURNAL will automatically exclude journaling for all VSAM datasets accessed in a region that is running CICS.

Before FSI-JOURNAL can be activated, the modules on the installation tape must be cataloged into an **APF Authorized Linklist Load Library**. Once this has been done, execute program CSIJRNL followed by a START command (see "START command" for details). The following example will accomplish this:

```
//CSIJRNL JOB          activate FSI-JOURNAL
//STEP1 EXEC PGM=CSIJRNL
//FSILIB DD DSN=authorized.linklist.library,DISP=SHR
//BIMLPA DD DSN=SYS1.LPALIB(IFG0192A),DISP=SHR
//CSIJRNSO DD SYSOUT=*
//CSIJRNSI DD *
        START STARTCODE('authcode')
/*
//
```

The DD statement "FSILIB" must point to the authorized library containing the FSI-JOURNAL and Routing Facility programs.

Program CSIJRNL and the Journal Option Table 'CSIJRN00' must be in an APF Authorized Load Library, or the START will abend with a S047.

The JOB doing the START must be authorized for read access to 'SYS1.LPALIB' to verify the DFP and VSAM release levels.

The activation job must be run after **every IPL** of each MVS system that is to have journaling active.

If you want to activate FSI-JOURNAL from within a JCL procedure, specify the START command in the PARM= operand of the EXEC statement as follows:

```
//CSIJRNL JOB          activate FSI-JOURNAL
//STEP1 EXEC PGM=CSIJRNL,
//          PARM='START STARTCODE(authcode)'
//FSILIB DD DSN=authorized.linklist.library,DISP=SHR
//BIMLPA DD DSN=SYS1.LPALIB(IFG0192A),DISP=SHR
//CSIJRNSO DD SYSOUT=*
//
```

Note: The DD statement for CSIJRNSI must not be present.

The activation is accomplished by the following procedure:

- The routing programs of BIMVSR/MVS are loaded into CSA storage and queued onto the CDE chain.
- Module IDA0192A is read from SYS1.LPALIB. The CESD and RLD records are scanned for the VCONS for IDA019A and IDA0200T. The addresses of the routers are then inserted into these VCONS.
- The addresses of IDA0192A and IDA0200T are stored in a control section of BIMVSRTB.

**Since this process changes internal control blocks within MVS, it is important that you test this in your environment during a non-production time period.**

If you experience problems with your system after activating FSI-JOURNAL, first attempt to deactivate it using the procedure found in the next section of this manual. If you still experience problems, then remove the VSAM OPEN/CLOSE Router using the procedure found in section "Removing the VSAM OPEN/CLOSE Router".

After FSI-JOURNAL has been activated, the option table and any JCL-supplied overrides will determine which VSAM datasets are to be journaled.

## Deactivating FSI-JOURNAL

FSI-JOURNAL will not have to be deactivated under normal circumstances. The option table and JCL overrides should provide all the flexibility needed for controlling the operation of FSI-JOURNAL.

For system maintenance or debugging severe VSAM problems, however, it may be necessary to totally remove FSI-JOURNAL from your system. The following example will accomplish this:

```
//CSIJRNL JOB          deactivate FSI-JOURNAL
//STEP1  EXEC PGM=CSIJRNL
//CSIJRNSO DD SYSOUT=*
//CSIJRNSI DD *
           STOP STOPCODE('authcode')
/*
//
```

See “STOP command” for details. Program CSIJRNL and the Journal Option Table 'CSIJRNO0' must be in an APF-Authorized Load Library, or the STOP will abend with a S047.

If you want to deactivate FSI-JOURNAL from within a JCL procedure, you can specify the STOP command in the PARM= operand of the EXEC statement as follows;

```
//CSIJRNL JOB          deactivate FSI-JOURNAL
//STEP1  EXEC PGM=CSIJRNL,
//          PARM='STOP STOPCODE(authcode)'
//CSIJRNSO DD SYSOUT=*
//
```

**Note** The DD statement for CSIJRNSI must not be present.

**If FSI-JOURNAL is stopped while Batch Journaling is in progress for one or more jobs, the journal records will continue to be written for as long as the ACBs are left open, but the journals for those jobs will not be closed, because the VSAM close will not have been intercepted. Due to buffering, this could lead to the loss of the last block of journal records.**

To re-activate FSI-JOURNAL, simply execute CSIJRNL with the START command.

## Option Table for FSI-JOURNAL

FSI-JOURNAL uses an option table to control the journaling of VSAM activity. When a VSAM dataset is opened, FSI-JOURNAL searches the option table for entries with selection criteria that match the dataset name and the environment (job name, program name, etc.). You code the option table and generate it using program CSIJRNL, then catalog it to an APF Authorized Load Library as phase CSIJRN00 using the standard MVS linkage editor.

The option table consists of four types of entries: DATASET, JOURNAL, IDENTIFY, and CONTROL. There may be any number of each type of entry, and entries of different types may be intermixed in any convenient order. However, for two of the types, DATASET and IDENTIFY, the sequence of the entries of the same type may be important.

The DATASET table entries are searched in the same order as they were coded, and if more than one table entry matches the dataset being opened, the first one found is the one that is used. The DATASET table entry refers to a JOURNAL table entry, which defines the attributes of the journal dataset to be used.

The JOURNAL table entries may be specified in any order, and are selected by the Journal ID (JID) from the previously selected DATASET table entry. Each JOURNAL table entry contains the ddname and other attributes of the physical journal dataset.

The IDENTIFY table entries are searched only if the dataset is to be journaled. The IDENTIFY table entries are searched in the same order as they were coded, and if more than one table entry matches the selection criteria, the first one found is the one that is used. The IDENTIFY table entry provides information to be used to identify the source of the journal records (equivalent of CICS TRANID, TERMID, etc.).

The CONTROL entries are used to specify optional messages you want displayed for certain VSAM and/or journaling events. The CONTROL table entries are treated somewhat differently than the DATASET and IDENTIFY entries. To allow you to trace VSAM activity without journaling, the CONTROL table entries are searched even if the dataset being opened is not going to be journaled. You probably will not specify trace messages in the option table for normal situations, but the facility may be useful for debugging systems without changing programs or JCL.

The other unique feature of CONTROL entries is that instead of using the options from the FIRST entry that matches the SELECT criteria, the options from ALL entries that match are combined. This could be used, for example, to allow one set of criteria (such as dataset name) to determine which messages to display, and another set of criteria (such as job name) to determine the destinations to which the messages should be routed.

The following example shows how to generate and catalog the FSI-JOURNAL Option Table, CSIJRN00. Comments are shown in lower case for clarity.

```
//CSIJRN00 JOB          catalog journal option table
//STEP1 EXEC PGM=CSIJRNL
//SYSUT1 DD UNIT=VIO,SPACE=(8192,(10,10))
//SYSLIN DD UNIT=VIO,SPACE=(3200,(10,10)),
          DSN=&&SYSLIN,DISP=(NEW,PASS)
//CSIJRN00 DD SYSOUT=*
//CSIJRN01 DD *
          GENERATE          /* create option table */ -
                          STARTCODE('authcode') -
                          STOPCODE('authcode')
          DATASET NAME(...) -
                          JID (...) -
                          JREQ(...)
          JOURNAL JID (...) -
                          DDNAME(...)
          IDENTIFY ...
          CONTROL ...
/*
//STEP2 EXEC PGM=IEWL,COND=(3,LT),
//          PARM='XREF,LIST,LET,REUS'
//SYSUT1 DD UNIT=VIO,SPACE=(1024,(100,50))
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&SYSLIN,DISP=OLD
//SYSLMOD DD DSN=your.loadlib(CSIJRN00),DISP=SHR
//
```

The SYSLMOD must point to the Authorized Linklist Load Library containing the FSI-JOURNAL programs.

**Note that any regions that have open VSAM datasets will not be affected by the new version. The new version will take effect at the first VSAM open of the next job step in each region.**

If FSI-JOURNAL detects any errors in the table, it will not produce an option table module, and the link edit will not catalog a new version of CSIJRN00.

### Example Journal Option Table

Shown below is an example of a complete Journal Option Table. Following the example is a description of each of the entries specified:

```

.....+....1.....+....2.....+....3.....+....4.....+....5.....+....6.....+....7..
1      GENERATE -
2          STARTCODE('STARTIT') -
3          STOPCODE('STOPTT')
4      DATASET NAME(*) -
5          JREQ(NONE) -
6          SELECT(INCLUDE(PROGNAME(CSIRECOV IDCAMS)))
7      DATASET NAME(FSI.KSDS.DATA/RLNKSDS -
8          FSI.ESDS.DATA/RLNESDS) -
9          JID(01) -
10         JREQ(ALL)
11     DATASET NAME(FSI.KS.* FSI.ES*) -
12         JID(01) -
13         JREQ(ALL) -
14         SELECT(INCLUDE(PROGNAME(CSIPROG1)))
15     DATASET NAME(FSI.KS* FSI.ES*) -
16         JID(02) -
17         JREQ(RU WU WN WD) -
18         SELECT(EXCLUDE(PROGNAME(CSIPROG2)))
19     JOURNAL JID(01) -
20         DDNAME(CSLJRN1)
21     JOURNAL JID(02) -
22         DDNAME(CSLJRN1)
23     IDENTIFY -
24         SELECT(INCLUDE(DATASET(FSI.KS* FSI.ES*) -
25             PROGNAME(CSIPROG*))) -
26         APPLID('SPECIAL')
27     IDENTIFY -
28         SELECT(INCLUDE(DATASET(*))) -
29         APPLID(JOBNAME)
30     CONTROL -
31         SELECT(INCLUDE(DATASET(*))) -
32         STATS(DEST(CONSOLE) JOOPEN JCLOSE)

```

### Notes on example Journal Option Table

#### Lines 1 through 3:

The GENERATE command instructs FSI-JOURNAL to build a Journal Option Table. This command also provides passwords to be used for activating and de-activating FSI-JOURNAL. The STARTCODE and STOPCODE specified here must also be specified on the START and STOP commands..

#### Lines 4 through 18:

This example contains four DATASET commands. The sequence of the individual commands can be very important. FSI-JOURNAL will use the first DATASET command that matches the NAME operand, and any SELECT requirements.

You normally want to place the most restrictive DATASET commands first, and the least restrictive commands last.

DATASET commands that use specific VSAM cluster names should usually appear before commands that use generic cluster names, unless very specific SELECT operands are specified, such as the first example shown in **Lines 4 through 6**.

The contents of the SELECT operand should also be used when you are determining the order of your DATASET commands. SELECT operands that INCLUDE specific jobs or programs should usually be placed before DATASET commands that EXCLUDE specific jobs or programs.

**Lines 4 through 6:**

The first DATASET command uses a generic dataset name (\*) to cause all datasets to match the entry. The SELECT operand limits the matches to only include programs CSIRECOV and IDCAMS.

The JREQ(NONE) specifies that no journaling should occur for any VSAM datasets processed by programs CSIRECOV and IDCAMS.

This DATASET command is specified first to prevent any other DATASET commands from possibly matching, and causing journaling to occur for these specific programs.

**Lines 7 through 10:**

The second DATASET command specifies two specific VSAM dataset names.

Since no SELECT operand has been specified, any job or program referencing either of these two VSAM cluster will have journaling performed.

The JREQ(ALL) specifies that all I/O requests against these datasets should be journaled to the journal dataset defined by JID(01) which is defined by the JOURNAL command in **Lines 19 through 20**.

The cluster names are followed by a ddname RLNKSDS and RLNESDS. This causes FSI-JOURNAL to use these names as the ddnames in the journal records. If these operands are omitted, FSI-JOURNAL will use the ddnames defined in the application program. This option should be used if your batch programs are accessing datasets used by CICS, but using different ddnames than CICS.

**Lines 11 through 14:**

The third DATASET command specifies two generic VSAM dataset names. These will match any clusters with names starting with 'FSI.KS' or 'FSI.ES'.

The SELECT operand specifies that this DATASET command should only be used for dataset activity from program CSIPROG1.

The JREQ(ALL) specifies that all I/O requests against these datasets from program CSIPROG1 should be journaled to the journal dataset defined by JID(01) which is defined by the JOURNAL command in **Lines 19 through 20**.

**Lines 15 through 18:**

The fourth DATASET command specifies the same two generic VSAM dataset names as the third DATASET command. These will also match any clusters with names starting with 'FSI.KS' or 'FSI.ES'.

The SELECT operand specifies that this DATASET command should match the dataset activity from any program except CSIPROG2.

The JREQ operand specifies that read-for-updates, updates, adds, and deletes against these datasets should be journaled to the journal dataset defined by JID(02) which is defined by the JOURNAL command in **Lines 21 through 22**.

**Lines 19 through 22:**

This example contains two JOURNAL commands. The first for JID(01) and the second for JID(02).

Both of these commands reference the same journal ddname (CSIJRNL). The updates for each of these “logical journals” will be written to the same “physical journal”. Each journal record will be identified with the JID of the “logical journal”.

You can journal all of your updates to the same journal, as shown in this example, or you can specify separate journals for specific combinations of VSAM datasets. Each “physical journal” must have a unique ddname.

**Lines 23 through 29:**

IDENTIFY commands provide information for FSI-JOURNAL to use when creating journal records. Batch programs do not have the same characteristics as CICS transactions. They do not have APPLIDs, Tran IDs, or Term IDs, as in CICS. The IDENTIFY command allows you to specify to FSI-JOURNAL what information you want to appear in each of these fields.

Just as with DATASET commands, the sequence that you specify your IDENTIFY commands is important. The same general rules for placement apply here. Refer to the notes for **Lines 4 through 18**.

**Lines 23 through 26:**

The first IDENTIFY command specifies that the character string 'SPECIAL' is to be used for the APPLID of all journal records for datasets whose names begin with 'FSI.KS' or 'FSI.ES' that are generated from programs whose names begin with 'CSIPROG'.

**Lines 27 through 29:**

The second IDENTIFY command specifies that each jobs job name is to be used for the APPLID of all journal records other than those matching the first IDENTIFY command. Since this the last IDENTIFY command, and it applies to all datasets, the SELECT operand could have been omitted.

**Lines 30 through 32:**

CONTROL commands determine which trace and/or statistical messages FSI-JOURNAL will generate, if any.

This example command specifies that a message should be generated on the System Console each time a journal dataset is opened or closed. Since this the only CONTROL command, and it applies to all datasets, the SELECT operand could have been omitted.

## Execution Time Overrides for FSI-JOURNAL

The option table (CSIJRN00) should be designed to handle all your production jobs. However, there may be special cases, such as testing or debugging, when you want to override the option table for the duration of a program. This is easily accomplished by executing CSIJRNL, giving it override table entries and a command to execute your application program. The following page shows an example of how to do this.

A second reason for using this method of executing your application, is if you are not using the TERMEXIT option of the GENERATE command, and you have applications that cause a single VSAM dataset to be opened and closed repetitively, or several VSAM datasets to be processed one at a time. If the TERMEXIT option is not being used, FSI-JOURNAL will close all open journals for a jobstep when the last open VSAM dataset is closed by the application program. These applications would cause the journal to be opened and closed several times during the jobstep. This process can be prevented by use of the JCLOSE operand of the EXEC command. Refer to the section "GENERATE Command" on page 51 for information on the TERMEXIT option.

The EXEC command of FSI-JOURNAL causes the DATASET, JOURNAL, IDENTIFY and CONTROL commands to be processed as they would with the GENERATE command, but the table is left in memory instead of being written to the SYSLIN dataset.

When a program that is executed by CSIJRNL opens a VSAM dataset, the override table entries for DATASET, JOURNAL and IDENTIFY are searched first, followed by the entries in the option table CSIJRN00 (if it exists).

CONTROL table entries are treated differently. If any CONTROL overrides exist, the option table CSIJRN00 will not be searched. This avoids having any CONTROL table entries in the option table CSIJRN00 combined with those in the overrides. If they were combined, there would be no way to "turn off" the effects of a CONTROL table entry in CSIJRN00.

If the EXEC command is used when FSI-JOURNAL is not active in your system, message FSIJRNL-30 will be generated, and the EXEC function will be terminated. If you want the EXEC to continue even if journaling is not active, include an OPTION TOLERATE(8) command in the EXEC jobstream.

### Original Test Job

```
//TESTJOB JOB          normal test job
//STEP1 EXEC PGM=TESTPROG
//VSAMDS DD   DSN=vsam.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN      DD *
TEST RECORD 1
TEST RECORD 2
/*           end of data for TESTPROG
//
```

### Same Job, with Journal Overrides Added

```
//TESTJOB JOB          job with journal override
//STEP1 EXEC PGM=CSIJRNL
//CSIJRNSO DD SYSOUT=*  messages from FSI-JRNL
//CSIJRNSI DD *        input for FSI-JRNL
    EXEC TESTPROG /* from original exec card */
    DATASET NAME(...) -
                JID (...) -
                JREQ(...)
    JOURNAL   JID (...) -
                DDNAME(...)
    IDENTIFY ...
    CONTROL  ...
/*
//VSAMDS DD   DSN=vsam.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN      DD *
TEST RECORD 1
TEST RECORD 2
/*
/ &
```

You must make a copy of module CSIJRNL in a **non-APF Authorized** Library, and execute it from there. An abend will occur if you attempt to use the APF Authorized version of CSIJRNL for Execution Override processing.

## Commands and Operands

Parameter cards are supplied to program CSIJRNL to create the journaling option table 'CSIJRN00'. This table specifies the names of the VSAM datasets to be journaled, along with the conditions that must be met to cause the journaling to occur. CSIJRNL can also be used to specify overrides to the Journal Option Table for a single execution of a particular program. The following sections define each of the supported commands.

### Syntax

The syntax of FSI-JOURNAL's commands is the same as the one used by the IBM utility IDCAMS:

- A **command followed by keyword operands**, with the value to be assigned to a keyword specified in parentheses.
- **Blanks** or **commas** are used to separate operands from the command and from other operands.
- **Continuation** is specified by coding a dash as the last non-blank on the line to be continued.
- **Comments** may appear before, between, and/or after operands.
- A **comment** starts with the characters “/\*”, is ended by the characters “\*/”, and may be continued to multiple lines as needed.

In the following instructions, the FSI-JOURNAL commands and operands should be interpreted as follows:

- UPPER CASE = the exact characters to be entered
- lower case = values to be supplied by you
- **boldface/underlined** keywords or values = default values
- Vertical bar “|” separates mutually exclusive operands
- [values enclosed in brackets] are optional

Refer to “APPENDIX A” starting on page 83 for a more complete description of how to code commands and operands.

## Commands

The commands that are used in FSI-JOURNAL are the following:

1. **DATASET**            Define datasets to journal
2. **JOURNAL**            Define journal dataset attributes
3. **IDENTIFY**           Define source of record identification
4. **CONTROL**           Define journaling message requirements
5. **START**              Activate FSI-JOURNAL
6. **STOP**                Deactivate FSI-JOURNAL
7. **GENERATE**          Generate FSI-JOURNAL Option Table
8. **EXEC**                Execute an application program with override parameters
9. **OPTION**             Set processing options for CSIJRNL

Please see the following pages for detailed descriptions of all commands/operands.

## DATASET Command

The DATASET command is used to specify the cluster names of the VSAM datasets to be journaled by FSI-JOURNAL.

Any number of DATASET commands can be specified. Each DATASET command provides a list of one or more VSAM cluster names, optional selection criteria used to qualify the conditions required to cause journaling of the specified cluster(s), a list of the types of activity against the cluster(s) that should be journaled, and the logical Journal ID that the updates should be written to. You can make the VSAM cluster names generic by specifying an "\*" to mean any number of characters, or "+" to mean any "single" character. The logical Journal ID is defined by the JOURNAL command.

### How to enter

DATASET	-
NAME(cluster.name/ddname, ...)	-
JREQ(RO,RU(ALL),WU,WD,WN,	-
ALL,UPDATE,BACKOUT,NONE)	-
[JID(nn)]	-
[SELECT(...)]	-

### Explanation

NAME(cluster.name/ddname, ...)

Required. Specifies a list of one or more VSAM cluster names to be considered for Batch Journaling.

Optional SELECT criteria can be supplied to limit the use of these VSAM cluster names.

Each cluster name can be followed by an optional DDNAME to be used as an override for whatever DDNAME is specified by the application program. This may be used to force the journaled records to have the same DDNAME as used in your CICS FCT.

You can make the VSAM cluster names generic by specifying an "\*" to mean any number of characters, or "+" to mean any "single" character. If generic cluster names are used with the optional DDNAME, then the same DDNAME will be used for all VSAM clusters that satisfy the generic name.

JREQ (RO , RU ( ALL ) , WU , WD , WN , ALL , UPDATE , BACKOUT , NONE )

Required. Specifies a list of one or more functions that are to be journaled for the VSAM clusters specified by the NAME operand.

RO	Read only. (Not required for Recovery.)
RU	Read for Update. (Only required for Backout Recovery.) If ALL is <u>not</u> specified, only the read for updates that are followed by some type of update will be journaled. If RU(ALL) is specified then <u>all</u> read for updates will be journaled even if no update is performed by the application program
WU	Write Update. (Replace). If WU is specified, WD is also assumed (same as the CICS FCT). If you want WU but not WD, specify "WU,NOWD".
WD	Write Delete.
WN	Write New. (Add.)
ALL	All requests will be journaled.
UPDATE	Same as JREQ(WU,WD,WN). These are the update requests needed for Forward Recovery.
BACKOUT	Same as JREQ(RU,WU,WD,WN). These are the requests needed for backout recovery.
NONE	No journaling will take place, and no other DATASET table entries will be searched. This may be used in a "dummy" DATASET table entry which is used only to exclude one or more datasets from journaling.

Journaling options can be specified in any combination desired. To negate selected functions from UPDATE, ALL or BACKOUT, specify the negated function preceded with "NO". The following specifications will have the same result:

JREQ(UPDATE)

JREQ(ALL,NORO,NORU)

JID(nn)

Required unless JREQ(NONE) is specified. Specifies the logical Journal ID to which the journaled activity will be written. A JOURNAL command must be specified for each JID value specified in a DATASET command. Any value from 0 to 255 can be specified.

SELECT(...)

Allows the specification of selection criteria that will be used to qualify when the VSAM clusters specified via the NAME operand will be journaled. Refer to the section "SELECT Operand" on page 56 for complete format and rules.

## JOURNAL Command

The JOURNAL command is used to specify the physical characteristics of the journal for each logical Journal ID (JID). One JOURNAL command is required for each unique JID value specified on a DATASET command.

### How to enter

```
JOURNAL -
      JID(nn) -
      DDNAME(ddname) -
      [FORMAT(SHORT | LONG) -]
      [ALLOCATE(dataset.name) -]
      [      BLKSIZE(8192 | nnnnn) -]
      [      DISP(status,normal,abend) -]
      [      EXPDT(yyddd) -]
      [      FREE | NOFREE -]
      [      LABEL(seq,type) -]
      [      MODDCB(model.dsname) -]
      [      RETPD(nnnn) -]
      [      SPACE(type(pri,sec),RLSE | NORLSE) -]
      [      UNIT(uuuuuuuu) -]
      [      VOLSER(vvvvvv) -]
      [      ) -]
      [ALLOCATE-A(... ) -]
      [MULTIPLE | NOMULTIPLE -]
      [PSUFFIX(xx) -]
      [ASUFFIX(xx) -]
      [USEREXIT(xxxxxxxx, | IGNORE | , | FORCEOPEN | )]
```

### Explanation

JID(nn)

Required. Specifies the logical Journal ID being defined. Can be any value from 0 through 255. One JOURNAL command must be specified with a JID for each JID value specified in a DATASET command.

DDNAME(ddname)

Specifies the DDNAME to be used to access this logical JID.

If the MULTIPLE operand is specified, the ddname must be exactly 5 characters long, and is used in conjunction with the PSUFFIX and ASUFFIX operands to form the actual ddnames of the primary and secondary extents of the journal.

If more than one JOURNAL command specifies a ddname with the same first 5 characters, the first JOURNAL command referenced at execution time will cause the journal to be opened, and all other JOURNAL commands with the same first 5 characters will be written to that journal dataset.

A DD statement is needed in the jobstream of the application, unless the ALLOCATE operand is also provided. The DD statement must include a DCB=BLKSIZE=nnnnn.

If the ALLOCATE operand is used, you can omit the DDNAME operand. This will result in EVERY VSAM dataset being journaled to a separate journal dataset with a system defined DDNAME (SYSnnnnn). You must also specify a dataset name for the ALLOCATE operand that will be unique for each VSAM dataset. This can be accomplished by using the FSI-JOURNAL system variables.

FORMAT ( **SHORT** | LONG )

Specifies the journal record format to be generated. The default is SHORT, which will generate the smallest records. If you intend to read the journal files by COBOL application programs, you may want to specify FORMAT(LONG). Refer to the section "APPENDIX B" on page 89 for the formats generated by each option.

MULTIPLE | **NOMULTIPLE**

Specifies that the physical journal is to have a primary and alternate extent.

Journaling will begin on the primary extent, and will automatically switch to the alternate extent if the primary extent fills. Switching back and forth will continue until the batch jobstep completes.

Messages will be generated on the console at each switch, and a message requiring operator response will be displayed before FSI-JOURNAL will overwrite an extent.

The PSUFFIX and ASUFFIX operands are required if MULTIPLE is specified.

PSUFFIX(cc)

This operand is required if the MULTIPLE operand has been specified. It provides the characters to be appended to the 5 character DDNAME to form the complete DD name of the primary journal extent.

ASUFFIX(cc)

This operand is required if the MULTIPLE operand has been specified. It provides the characters to be appended to the 5 character DDNAME to form the complete DD name of the alternate journal extent.

ALLOCATE(...)  
ALLOCATE-A(...)

Specifying these along with the required sub-operands, indicates that the journal dataset should be dynamically allocated at execution time. The sub-operands represent a subset of the standard DD statement parameters. Although the syntax used by FSI-JOURNAL is not always exactly the same as a DD statement, the rules for what can be specified are the same.

The ALLOCATE-A operand is only used when the MULTIPLE option is specified. It is used to allocate the alternate extent journal. It contains the same sub-operands as the ALLOCATE operand.

If a DD statement for the DDNAME of the journal is specified in the jobstream, the dynamic allocation will not be attempted. You can use this feature to override the dynamic allocation for selected jobs.

&&dsname  
dataset.name  
dataset.name(+nnn)

Required. Specifies the dsname of the allocated journal dataset. The dataset name can be any valid dsname supported by a DD statement, including temporary names (&&dsname).

FSI-JOURNAL provides a series of system variables that can be used to create dataset names that are execution-time dependent. System variables can be specified at any location in the dataset name, and in any combination. It is your responsibility to ensure that the resulting generated name follows the rules for a valid dsname. The following table lists the system variables and what they generate:

<b>%APPLID</b>	APPLID from IDENTIFY command	1-8 char.
<b>%CPUID</b>	Last 6-digits of current CPU-ID	6 char.
<b>%DDNAME</b>	VSAM DDNAME from ACB	1-8 char.
<b>%JID</b>	JID value	1-3 char.
<b>%JOBNAME</b>	Name from JOB	1-8 char.
<b>%JOBSTEP</b>	Stepname from EXEC PGM=	1-8 char.
<b>%JULDATE</b>	Current date (yyddd)	5 char.
<b>%PROCSTEP</b>	Stepname from EXEC within a procedure	1-8 char.
<b>%PROGNAME</b>	PGM from EXEC	1-8 char.
<b>%TIME</b>	Current time (hhmmss)	1-7 char.

**NOTE:** **%PROCSTEP** will be the same as **%JOBSTEP** if the jobstep is NOT contained within a procedure.

For the following examples, the values for the specified system variables are assumed to be:

<b>%JOBNAME</b>	<b>SAMPLE</b>
<b>%JULDATE</b>	<b>88001</b>
<b>%JID</b>	<b>1</b>
<b>%DDNAME</b>	<b>RLNKSDS</b>

entered - **%DDNAME.JOURNAL.J%JID**  
 generated - **RLNKSDS.JOURNAL.J1**

entered - **JRNL.%JOBNAME.D%JULDATE**  
 generated - **JRNL.SAMPLE.D88001**



MODDCB(model.dsname)

Specifies the dataset name to be used as the model when allocating a GDG.

RETPD(nnnn)

Specifies the dataset retention period. This operand follows the same rules as the RETPD operand of the DD statement and is mutually exclusive with the EXPDT operand.

SPACE(type(pri[,sec])[ ,RLSE|**NORLSE**])

Specifies the dataset space allocation for disk journals. This operand follows most of the same rules as the SPACE operand of the DD statement.

“type” is the allocation unit type. It can be TRK, CYL or nnnn (nnnn is the block size for block allocation).

“pri” is the primary allocation value

“sec” is the secondary allocation value

“RLSE” is the same as RLSE on the DD statement. If RLSE is omitted, NORLSE is assumed.

UNIT(uuuuuuuu)

Specifies the dataset unit allocation type. This operand follows the same rules as the UNIT operand of the DD statement.

VOLSER(vvvvvv)

Specifies the volume serial number of the device to contain the journal. This operand follows the same rules as the VOLSER operand of the DD statement.

USEREXIT(xxxxxxxx, | **IGNORE** | , | **FORCEOPEN** | )  
 ABEND | DEFEROPEN | )

Specifies the name and processing options for a user exit to be called while processing this JOURNAL.

Refer to source member CSIJRNXF for complete information on writing and using a user exit.

xxxxxxxx The phase name of the user exit. This module will be loaded (CDLOAD) by FSI-JOURNAL.

IGNORE	If the phase 'xxxxxxx' cannot be loaded, issue warning message FSIJRN-251 , but continue. This is the default.
ABEND	If the phase 'xxxxxxx' cannot be loaded, issue error message FSIJRN-250 and terminate.
FORCEOPEN	Open the journal at the first request of any type by the application program. This is the default.
DEFEROPEN	Defer the open of the journal until it is required. If the user exit handles all journal requests, it will never be opened.

## IDENTIFY Command

The IDENTIFY command is used to specify identifying information to be included in the journal records. This is information that is provided in CICS journals but does not have equivalent information in a batch environment.

Any number of IDENTIFY commands can be specified. The SELECT operand is used to specify the conditions to be met before a particular IDENTIFY command is used. The IDENTIFY commands are searched in the sequence specified; the first IDENTIFY command with matching SELECT criteria will be used.

## How to enter

```
IDENTIFY -
  [APPLID(JOBNAME | PROGNAME | JOBSTEP | PROCSTEP | 'cccccccc' ) ] -
  [TRANTERM(PROGNAME | JOBNAME | JOBSTEP | PROCSTEP | 'cccccccc' ) ] -
  [TASK(Q | nnnnn) ] -
  [SELECT( . . . ) ]
```

## Explanation

APPLID(JOBNAME | PROGNAME | JOBSTEP | PROCSTEP | 'cccccccc' )

Specifies the value to be inserted in the APPLID field of the journal record.

JOBNAME            Current MVS job name.  
(From // JOB)

PROGNAME           Program name.  
(From // EXEC PGM=)

JOBSTEP            Step name.  
(From // step EXEC PGM=)

PROCSTEP           Step name within procedure.  
(From // EXEC PGM=)  
Will equal JOBSTEP if no procedure is  
being executed.

'cccccccc'        Specific character string.  
(8 characters max.)

TRANTERM( **PROGNAME** | JOBNAME | JOBSTEP | PROCSTEP | 'cccccccc' )

Specifies the value to be inserted in the TRAN and TERM fields of the journal record. The first four bytes go into the TRAN field, the second four bytes go into the TERM field.

JOBNAME	Current MVS job name. (From // JOB)
PROGNAME	Program name. (From // EXEC PGM=)
JOBSTEP	Step name. (From //step EXEC PGM=)
PROCSTEP	Step name within procedure. (From // EXEC PGM=) Will equal JOBSTEP if no procedure is being executed.
'cccccccc'	Specific character string. (8 characters max.)

TASK( 0 | nnnnn )

Specifies a number from 0 - 99999 to be placed in the journal Task ID field.

SELECT( . . . )

Allows the specification of selection criteria that will be used to qualify when the IDENTIFY command should be used. Refer to the section "SELECT Operand" on page 56 for complete format and rules.

## CONTROL Command

The CONTROL command is used to specify that messages are to be generated for statistics and/or trace information.

Any number of CONTROL commands can be specified. The SELECT operand is used to specify the conditions to be met before a particular CONTROL command is used. All message options from matching CONTROL commands are combined for each opened VSAM dataset. If conflicting destinations are specified, the destination from the first matching CONTROL command is used. The CONTROL commands are searched in the sequence specified.

Different CONTROL options can be specified for each VSAM cluster, if desired, by specifying the appropriate SELECT operands.

## How to enter

```

CONTROL -
  NOSTATS | STATS(                                     -
    [DEST(SYSPRINT | CONSOLE | ddname [ , ddname ] ) ] -
    [ NOOPEN | OPEN ]                                 -
    [ NOCLOSE | CLOSE ]                               -
    [ NOJOPEN | JOOPEN ]                               -
    [ NOJCLOSE | JCLOSE ]                             -
  )                                                     -
  NOTRACE | TRACE(                                     -
    [DEST(SYSPRINT | ddname) ]                         -
    [ NOENTRY | ENTRY ]                               -
    [ NOACB | ACB | ACB ( SNAP ) ]                   -
    [ NORPL | RPL | RPL ( SNAP ) ]                   -
    [ NOGET | GET ]                                   -
    [ NOPUT | PUT ]                                   -
    [ NOERASE | ERASE ]                               -
    [ NOARG | ARG ]                                   -
    [ NODATA | DATA ]                               -
    [ NOJWRITE | JWRITE | JWRITE ( SNAP ) ]         -
  )                                                     -
  [ SELECT ( . . . ) ]

```

**Explanation****NOSTATS** | STATS

Specifies that selected statistics messages are to be generated. These are messages with numbers 200 through 209. Additional operands on this keyword supply the destination and options for the messages that are desired.

## DEST (SYSPRINT | CONSOLE | ddname [ , ddname ] )

Specifies the destination(s) for the STATS messages being generated. Specify the ddname of the printer to which the selected messages should be sent. Two destinations can be provided. One of the destinations for STATS can be 'CONSOLE' which will cause the output to be sent to the System Console.

## SELECT ( . . . )

Allows the specification of selection criteria that will be used to qualify when the CONTROL command should be used. Refer to the section "SELECT Operand" on page 56 for complete format and rules.

**NOOPEN** | OPEN

STATS option to specify that message 201 is desired. This message is generated at the first request for a VSAM ACB for which journaling has been specified. This message provides the name of the journal being used, and lists the type of VSAM dataset activity that will be journaled.

**NOCLOSE** | CLOSE

STATS option to specify that message 202 is desired. This message is generated when the VSAM ACB is closed. This message provides the number of records journaled and individual counts by type of activity.

**NOJOPEN** | JOPEN

STATS option to specify that message 200 is desired. This message is generated at the time a journal dataset is opened. This message provides the dataset name of the journal being used.

**NOJCLOSE** | JCLOSE

STATS option to specify that message 203 is desired. This message is generated when a journal dataset is closed. This message provides the number of records journaled and individual counts by type of activity.

**NOTRACE** | TRACE

Specifies that selected trace messages are to be generated. These are messages with numbers 290 through 299. Additional operands on this keyword supply the destination and options for which trace messages are desired.

## DEST (SYSPRINT | ddname)

Specifies the destination(s) for the TRACE messages being generated. Specify the ddname of the printer to which the selected messages should be sent.

**NOENTRY** | ENTRY

TRACE option to specify that message 290 is desired. This message is generated each time the journaling module is entered with a VSAM request from an application program. This message identifies the type of VSAM request, and the addresses of the VSAM ACB and RPL in the application.

**NOACB** | ACB | ACB (SNAP)

TRACE option to specify that message 291 is desired. This message is generated the first time each VSAM ACB is accessed. This message displays various information about the VSAM dataset. If (SNAP) is specified, a snap dump of the actual ACB is also generated.

**NORPL** | RPL | RPL (SNAP)

TRACE option to specify that message 292 is desired. This message is generated after every request has been processed by VSAM. This message displays various information about the VSAM request. If (SNAP) is specified, a snap dump of the actual RPL is also generated.

**NOGET** | GET  
**NOPUT** | PUT  
**NOERASE** | ERASE

TRACE options to specify that message 293 is desired for the specified type(s) of VSAM requests. This message is generated after every request has been processed by VSAM. This message displays various information about the VSAM request. If the ARG and/or DATA operands are specified, snap dumps of the key and/or data areas will also be generated.

**NOARG** | ARG

TRACE option to specify that the VSAM KEY/RBA/RRN field is to be displayed in a snap dump format each time message 293 is generated. Refer to operands GET, PUT and ERASE defined above.

**NODATA** | DATA

TRACE option to specify that the VSAM data record is to be displayed in a snap dump format each time message 293 is generated. Refer to operands GET, PUT and ERASE defined above.

**NOJWRITE** | JWRITE | JWRITE (SNAP)

TRACE option to specify that message 294 is desired. This message is generated each time a record is written to the journal. This message displays various information about the record being written to the journal. If (SNAP) is specified, a snap dump of the actual journal record is also generated.

## START Command

The START command is used to activate FSI-JOURNAL for system wide processing.

This will cause every VSAM open in your entire MVS system to become eligible for Batch Journaling. FSI-JOURNAL will automatically exclude any VSAM open in any regions running CICS, but all other VSAM opens will be matched against the Journal Option Table (CSIJRN00) to determine whether or not to activate journaling for a particular VSAM ACB.

The START command must be the only command specified for the CSIJRNL execution. Refer to the section “Activating FSI-JOURNAL” on page 19 for additional information.

## How to enter

```
START -  
  [STARTCODE('authorization code')           -]  
  [LOADLIB(FSILIB|xxxxxxxx)                   ]
```

## Explanation

STARTCODE('authorization code')

Specifies the 1 to 8 character authorization code required to use the START command. The authorization code specified here must be the same as the code defined for the STARTCODE operand of the GENERATE command. If the wrong value is specified here, the START function will be rejected.

LOADLIB(**FSILIB**|xxxxxxxx)

Specifies the name of the DD statement that points to the Authorized Linklist Library containing the FSI-JOURNAL and Routing Facility programs.

### STOP Command

The STOP command is used to de-activate FSI-JOURNAL from system wide processing.

The STOP command must be the only command specified for the CSIJRNL execution. Refer to the section "Deactivating FSI-JOURNAL" on page 21 for additional information.

### How to enter

```
STOP    [STOPCODE('authorization code')]
```

### Explanation

STOPCODE('authorization code')

Specifies the 1 to 8 character authorization code required to use the STOP command. The authorization code specified here must be the same as the code defined for the STOPCODE operand of the GENERATE command. If the wrong value is specified here, the STOP function will be rejected.

**GENERATE Command**

The GENERATE command is used to create a new Journal Option Table (CSIJRN00). This function will create an object module containing all options specified via DATASET, JOURNAL, IDENTIFY and CONTROL commands. The object module will be written to SYSLIN. The object module can then be link edited to phase CSIJRN00. This will completely replace the previous Journal Option Table. Refer to the section "Option Table for FSI-JOURNAL" on page 23 for a sample jobstream for the GENERATE command.

**How to enter**

GENERATE	-
[ <b>NOTERMEXIT</b>   TERMEXIT	-]
[STARTCODE('authorization code')	-]
[STOPCODE('authorization code')	-]

**Explanation**

**NOTERMEXIT** | TERMEXIT

Controls journal closure time. The default is NOTERMEXIT which cause journals to be controlled by the closure of associated VSAM file(s). This was the normal method used in all prior releases of FSI-JOURNAL. TERMEXIT causes all journals regardless of VSAM file association to be closed at Job-Step termination time. With TERMEXIT specified, CSIJRNIX will be loaded into the JOB Pack Area and given control after the application returns control back to the system. CSIJRNIX will close any orphaned VSAM files and all open journals when it receives control.

STARTCODE('authorization code')

Specifies the 1 to 8 character authorization code required to use the START command. IF an authorization code is specified here, then the STARTCODE operand of the START command must be specified with the same value, or the START function will be rejected.

STOPCODE('authorization code')

Specifies the 1 to 8 character authorization code required to use the STOP command. IF an authorization code is specified here, then the STOPCODE operand of the STOP command must be specified with the same value, or the STOP function will be rejected.

## EXEC Command

The EXEC command is used to provide temporary journaling options for the execution of a specific application program. It allows the specification of override journaling options for DATASET, JOURNAL and IDENTIFY commands, and replacement options for CONTROL commands. The overrides are applied to the Journal Option Table (CSIJRN00) by logically placing the overrides BEFORE the entries in CSIJRN00. Any CONTROL commands specified will completely replace the CONTROL commands contained in CSIJRN00.

If JCLOSE(LASTVSAM) is specified, or allowed to default, FSI-JOURNAL will transfer control to the application via an MVS XCTL macro.

If JCLOSE(RETURN) is specified, FSI-JOURNAL will transfer control to the application via an MVS LINK Macro.

If OPTION SUSPEND is specified, FSI-JOURNAL will ignore the Journal Option Table, and no journaling will be performed. Refer to the section "OPTION Command" on page 55.

You must make a copy of module CSIJRNL in a **non-APF Authorized** Library, and execute it from there. An abend will occur if you attempt to use the APF Authorized version of CSIJRNL for Execution Override processing.

Refer to the section "Execution Time Overrides for FSI-JOURNAL" on page 29 for a sample jobstream for the EXEC command.

## How to enter

```
EXEC    [PGM=]progname          -  
        [JCLOSE(LASTVSAM|RETURN) ]
```

## Explanation

[PGM=]progname

Required. Specifies the name of the program to be executed.

JCLOSE (LASTVSAM | RETURN)

Specifies when FSI-JOURNAL will close the journal datasets. This operand is ignored if the TERMEXIT option is specified on the GENERATE command. Refer to the section “GENERATE Command” on page 51 for information on the TERMEXIT option.

**LASTVSAM** This parameter indicates that FSI-JOURNAL is to close all open journals whenever all VSAM datasets have been closed by the application program. This is the method normally used by FSI-JOURNAL.

**RETURN** This parameter indicates that FSI-JOURNAL is not to close the journal datasets until the application program has finished processing and returns to program CSIJRNL. (CSIJRNL will have transferred to the application via an MVS LINK Macro.)

## OPTION Command

The OPTION command is used to specify special processing options for CSIJRNL.

### How to enter

```
OPTION -  
    [ NOSUSPEND | SUSPEND - ]  
    [ NOTOLERATE | TOLERATE ( 12 | nn ) ]
```

### Explanation

**NOSUSPEND** | SUSPEND

SUSPEND specifies that no journaling is required for the current application program. This option is only valid when used with the EXEC command. It causes the Journal Option Table to be ignored for the current execution of the application program.

**NOTOLERATE** | TOLERATE ( **12** | nn )

NOTOLERATE causes CSIJRNL to terminate processing if any errors are detected in the input commands.

TOLERATE allows processing to continue in CSIJRNL even if errors are detected in the input commands. If a tolerate value is specified, it represents the maximum condition code that will allow processing to continue. Each error message has an associated condition code. Refer to the section "MESSAGES" on page 59.

## SELECT Operand

The SELECT operand is used to specify the criteria to be met to have DATASET, IDENTIFY and CONTROL commands selected for processing.

### How to enter

```
Command ... SELECT( selection-criteria-operands )
```

### How to enter selection-criteria-operands

```
INCLUDE|EXCLUDE( field1(list1) [field2(list2)...] )...
```

### Explanation

The "field(list)" operands represent the names of the fields to test, and the data to be compared.

### Field name Keywords

<b>CPUID(string ...)</b>	Last 6 digits of your CPU serial number
<b>DATASET(string ...)</b>	44-character VSAM cluster name
<b>DDNAME(string ...)</b>	DDNAME from current VSAM ACB
<b>JOBNAME(string ...)</b>	Job name from // JOB
<b>JOBSTEP(string ...)</b>	Step name from //step EXEC
<b>JREQ(string ...)</b>	5-character Y/N flag. Each position represents whether or not the corresponding journaling function (RO, RU, WN, WU, WD) has been specified. It is valid for IDENTIFY and CONTROL commands.
<b>JSEL(string ...)</b>	Y/N flag if any journaling specified (valid for IDENTIFY and CONTROL commands)
<b>PROCSTEP(string ...)</b>	Step name from executed procedure
<b>PROGNAME(string ...)</b>	Program name from // EXEC PGM=

Field name keywords can be prefixed by a 'relational operator' to change the meaning of a 'match' with the strings in the specified list.

**Field name Relational Operators**

<b>EQ-field(...)</b>	(default) match if field is equal to any one string
<b>NOT-field(...)</b>	match if the field is equal to NONE of the strings
<b>NE-field(...)</b>	match if the field is not equal to the string
<b>GT-field(...)</b>	match if the field is greater than the string
<b>GE-field(...)</b>	match if the field is greater than or equal to the string
<b>LT-field(...)</b>	match if the field is less than the string
<b>LE-field(...)</b>	match if the field is less than or equal to the string

**Selection Rules**

- Any number of 'field(list)' operands can be specified.
- If more than one 'field(list)' operand is specified, ALL fields must be 'matched' for the SELECT or EXCLUDE to be matched.
- Any number of INCLUDE and/or EXCLUDE operands can be used, in any sequence. They are processed in the sequence specified. As soon as an INCLUDE or EXCLUDE is matched, selection processing stops, and the INCLUDE or EXCLUDE takes effect.
- If only EXCLUDE operands are specified, then SELECTs are considered matched unless specifically excluded. If one or more INCLUDE operands are specified, then the SELECT is not considered matched unless at least one INCLUDE operand is matched.
- Selection 'lists' can contain any number of data values to be tested against the 'field'. If more than one value is specified, 'matching' any ONE value 'matches' the list.

**Rules for strings**

- 'strings' can be any character string. If it contains any blanks, commas, quotes or parenthesis, enclose the string in single quotes. If the string contains any quotes, specify them as (2) single quotes.
- 'strings' can contain special generic operators if they are being tested for an EQ-, NE- or NOT- condition.

'\*' represents ANY number of characters.

'+' represents ANY single character.

If you need to specify a string with an '\*' or '+' and you want to test for those characters, specify the string as - C'string'

- 'strings' can be specified as Hexidecimal values - X'hhhhhhhh'.

### SELECT Operand Example

In this example, the SELECT operand is used to define when to journal any dataset with a name that begins with "PROD.VSAM". Journaling is to take place only if all of the following criteria are met:

- The program name is not "PROD001" or "PROD002"
- The program name starts with "PROD"

The following DATASET command will produce this result:

```
DATASET -
NAME ( PROD . VSAM . * ) -
JID ( 01 ) -
JREQ ( WU WN WD ) -
SELECT ( -
    EXCLUDE ( -
        PROGRAMNAME ( PROD001 PROD002 ) -
    ) -
    INCLUDE ( -
        PROGRAMNAME ( PROD * ) -
    ) -
) -
```

## **MESSAGES**



**FSIJRNL-nnnn**

- FSIJRNL-1**            **PROCESSING COMPLETED,  
HIGHEST CONDITION CODE WAS 0**  
This message is displayed on SYSPRINT at end of job. It indicates the Operating System Return/Condition code that the Jobstep ended with.
- FSIJRNL-2**            **A START, STOP, GENERATE OR EXEC COMMAND HAS ALREADY  
BEEN SPECIFIED - THIS COMMAND IS IGNORED  
CONDITION CODE = 12**  
This message is displayed on SYSPRINT if more than one of the listed commands was entered in the same execution of CSIJRNL. The START, STOP, GENERATE and EXEC commands are mutually exclusive in function. The condition code for the command is set to 12.
- FSIJRNL-3**            **DUPLICATE JOURNAL ID - DEFINITION IGNORED  
CONDITION CODE = 12**  
This message is displayed on SYSPRINT if more than one JOURNAL command is entered with the same JID value. The condition code for the command is set to 12.
- FSIJRNL-4**            **JOURNAL ID nn DEFINED BUT NOT REFERENCED  
CONDITION CODE = 4**  
This message is displayed on SYSPRINT if any JID values specified in JOURNAL commands have not been specified in DATASET commands. The condition code for the command is set to 4, unless it has already been set to a higher value.
- FSIJRNL-5**            **JOURNAL ID nn REFERENCED BUT NOT DEFINED CONDITION  
CODE = 4**  
This message is displayed on SYSPRINT if any JID values specified in DATASET commands do not have corresponding JOURNAL commands with the same JID value. The condition code for the command is set to 4, unless it has already been set to a higher value.
- FSIJRNL-7**            **DATASET, JOURNAL, IDENTIFY, AND CONTROL COMMANDS ARE  
NOT ALLOWED WITH START OR STOP  
CONDITION CODE = 12**  
This message is displayed on SYSPRINT when a START or STOP command is entered along with one or more of the listed commands. The listed

commands can only be included with GENERATE or EXEC commands. The condition code for the command is set to 12.

- FSIJRNL-8**            **'nnn' IS NOT A VALID JOURNAL ID NUMBER**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT when a JID value outside the valid range of 0 through 255 is entered. The condition code for the command is set to 12.
- FSIJRNL-9**            **OUTPUT FILE IS NOT AVAILABLE FOR GENERATE COMMAND**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT when a GENERATE command is entered, but no DD statement for SYSPUNCH has been provided. Refer to the section "Option Table for FSI-JOURNAL" on page 23 for sample JCL to use with the GENERATE command. The condition code for the command is set to 12.
- FSIJRNL-10**          **SPECIFIED PARAMETERS REQUIRE nnnnn BYTES, MAXIMUM IS**  
**65,535**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT when an individual command would require more than 65,535 bytes to be stored in the Journal Option Table. Separate the command into more than one command to reduce the storage required. The condition code for the command is set to 12.
- FSIJRNL-11**          **THE xxxxxxxx PARAMETER CANNOT CONSIST OF BOTH DATA AND**  
**A KEYWORD**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT if more than one value is entered for the APPLID or TRANTERM keywords of the IDENTIFY command. The condition code for the command is set to 12.
- FSIJRNL-12**          **NO START, STOP, GENERATE, OR EXEC COMMAND GIVEN - NO**  
**ACTION TAKEN**  
**CONDITION CODE = 4**  
This message is displayed on SYSPRINT if none of the listed "action" commands have been supplied. Add the command for the action you had intended. The condition code for the command is set to 4, unless it has already been set to a higher value.
- FSIJRNL-13**          **PROGRAM NAME 'xxxxxxx' EXCEEDS 8 CHARACTERS**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT if a program name is supplied on the EXEC command that exceeds the maximum size for a valid name. The condition code for the command is set to 12.

- FSIJRNL-14**      **JID(...) PARAMETER IS MISSING - JID(0) ASSUMED**  
**CONDITION CODE = 4**  
This message is displayed on SYSPRINT if the JID keyword is missing from a DATASET or JOURNAL command. This is a required keyword unless JREQ(NONE) is specified. The condition code for the command is set to 4, unless it has already been set to a higher value.
- FSIJRNL-15**      **JREQ(...) PARAMETER IS MISSING - JREQ(NONE) ASSUMED**  
**CONDITION CODE = 8**  
This message is displayed on SYSPRINT if the JREQ keyword is omitted from a DATASET command. DATASET commands are only required for datasets intended for journaling. DATASET commands are not required if only tracing is required. Trace options are specified via the CONTROL command. The condition code for the command is set to 8, unless it has already been set to a higher value.
- FSIJRNL-16**      **'SUSPEND' OPTION IGNORED - ONLY VALID WITH EXEC**  
**COMMAND**  
**CONDITION CODE = 8**  
This message is displayed on SYSPRINT if OPTION SUSPEND is specified with a command other than EXEC. The condition code for the command is set to 8, unless it has already been set to a higher value.
- FSIJRNL-17**      **FSI-JOURNAL IS ALREADY ACTIVE IN THE SYSTEM**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT and the System Console if a START command is executed when FSI-JOURNAL is already active due to a previous START. A START is only required after an IPL of your system or a preceding STOP command. The condition code for the command is set to 12.
- FSIJRNL-18**      **FSI-JOURNAL IS NOT ACTIVE IN THE SYSTEM**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT and the System Console if a STOP command is executed when FSI-JOURNAL is not active. The condition code for the command is set to 12.
- FSIJRNL-20**      **FSI-JOURNAL SUCCESSFULLY ACTIVATED**  
This message is displayed on SYSPRINT and the System Console after a START command has been successfully executed.
- FSIJRNL-21**      **FSI-JOURNAL SUCCESSFULLY DEACTIVATED**  
This message is displayed on SYSPRINT and the System Console after a STOP command has been successfully executed.
- FSIJRNL-22**      **NAME(...) PARAMETER MISSING - DEFINITION IGNORED**  
**CONDITION CODE = 12**  
This message is displayed on SYSPRINT if the NAME keyword is omitted from a DATASET command. This is a required keyword. If no dataset names

are specified, the DATASET command will have no effect. The condition code for the command is set to 12.

- FSIJRNL-23**      **'BLK' SPACE ALLOCATION REQUIRE BLOCK SIZE, 8192 ASSUMED  
CONDITION CODE = 8**  
This message is displayed on SYSPRINT if the SPACE operand is specified with a zero block length. The condition code for the command is set to 12.
- FSIJRNL-24**      **RETPD AND EXPDT OPERANDS ARE MUTUALLY EXCLUSIVE,  
EXPDT IGNORED  
CONDITION CODE = 8**  
This message is displayed on SYSPRINT if the RETPD and EXPDT operands are specified together. The condition code for the command is set to 12.
- FSIJRNL-25**      **BLKSIZE(0) INVALID, 8192 ASSUMED  
CONDITION CODE = 8**  
This message is displayed on SYSPRINT if the BLKSIZE operand is specified with a zero block length. The condition code for the command is set to 12.
- FSIJRNL-26**      **UNKNOWN VARIABLE IN DATASET NAME STARTING AT  
POSITION nn, VARIABLES ARE IGNORED  
CONDITION CODE = 12**  
This message is displayed on SYSPRINT if the dataset name for an ALLOCATE operand contains a "%" that is not followed by a valid FSI-JOURNAL system variable name. The dataset name will be used as coded, which will probably result in a dynamic allocation error. The condition code for the command is set to 12.
- FSIJRNL-27**      **DATASET NAME IS REQUIRED FOR ALLOCATE  
CONDITION CODE = 12**  
This message is displayed on SYSPRINT if the ALLOCATE operand is specified without a dataset name. The condition code for the command is set to 12.
- FSIJRNL-28**      **INVALID OR MISSING AUTHORIZATION CODE,  
START/STOP FUNCTION REJECTED**  
This message is displayed on SYSPRINT and the System Console if a STARTCODE and/or STOPCODE has been specified on the GENERATE command and the matching code has not been provided for the START or STOP command. The START or STOP function is rejected, and the jobstep is abended.
- FSIJRNL-29**      **JOURNAL OPTION TABLE (CSIJRN00) NOT FOUND,  
START/STOP FUNCTION REJECTED**  
This message is displayed on SYSPRINT and the System Console if a START or STOP command has been specified and no Journal Option Table can be located. The START or STOP function is rejected, and the jobstep is abended.

- FSIJRNL-30**      **FSI-JOURNAL IS NOT ACTIVE IN THE SYSTEM - EXECUTED PROGRAM WILL NOT BE JOURNALED**  
**CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console if an EXEC command is used when FSI-JOURNAL is not active (has not been started). If you want the EXEC function to continue, you must include an OPTION TOLERATE(8) command.
- FSIJRNL-31**      **JOURNAL OPTION TABLE (CSIJRN00) RELEASE LEVEL IS INCOMPATIBLE, START FUNCTION REJECTED**  
**CSIJRNL=x.xx, CSIJRN00=x.xx**  
This message is displayed on SYSPRINT and the System Console if a START command has been specified and the Journal Option Table is determined to be from an incompatible release of FSI-JOURNAL. The START function is rejected, and the jobstep is abended. Regenerate your Journal Option Table with the current release of program CSIJRNL.
- FSIJRNL-32**      **JOURNAL OPTION TABLE (CSIJRN00) RELEASE LEVEL IS INCOMPATIBLE, JOURNALING WILL NOT BE ACTIVATED**  
**CSIJRNL=x.xx, CSIJRN00=x.xx**  
This message is displayed on SYSPRINT and the System Console if an EXEC command has been specified and the Journal Option Table is determined to be from an incompatible release of FSI-JOURNAL. The EXEC function is processed, but no journaling will be performed. Regenerate your Journal Option Table with the current release of program CSIJRNL.
- FSIJRNL-34**      **SUFFIX NOT SPECIFIED FOR PRIMARY EXTENT OF JOURNAL SET. MULTIPLE EXTENT WAS SPECIFIED. CHECK PARAMETERS AND RESTART. CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console when the primary suffix information is missing when multiple extents has been requested. Suffix information for the primary extent must be provided either as a separate entry in PSUFFIX or as part of the base DDNAME entry
- FSIJRNL-35**      **SUFFIX NOT SPECIFIED FOR ALTERNATE EXTENT OF JOURNAL SET. MULTIPLE EXTENT WAS SPECIFIED. CHECK PARAMETERS AND RESTART. CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console when the alternate suffix information is missing when multiple extents has been requested. Suffix information for the alternate extent must be provided with an ASUFFIX operand and value.
- FSIJRNL-36**      **SECONDARY ALLOCATION FOR ALTERNATE EXTENT OF JOURNAL SET NOT SPECIFIED**  
**CHECK PARAMETERS AND RESTART. CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console when the alternate allocation information has not been provided for the alternate extent of a dual extent journal set with dynamic allocation requested. An ALLOCATE-A operand with sub-parameters is required.

- FSIJRNL-37**      **PRIMARY AND ALTERNATE SUFFIX ARE THE SAME VALUE, UNIQUE SUFFIXES ARE REQUIRED. CHECK PARAMETERS AND RESTART. CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console when the alternate suffix and the primary suffix values provided are the same characters. Unique suffix values are required to create unique ddnames for each dataset of a dual extent journal set.
- FSIJRNL-38**      **DDNAME DOES NOT HAVE 5 CHARACTERS IN IT. MULTIPLE EXTENTS REQUIRES A 5 CHARACTER DDNAME BASE. CHECK PARAMETERS AND RESTART. CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console when the base DDNAME value does not contain the necessary 5 characters that will fill the concatenation variable completely in order to ensure that blanks do not fall within the ddname when the suffix data is concatenated to the base.
- FSIJRNL-39**      **MULTIPLE EXTENTS ARE IMPLIED BY PRESENCE OF SUFFIX DEFINITIONS, YET 'MULTIPLE' PARAMETER IS NOT PROVIDED. CHECK PARAMETERS AND RESTART. CONDITION CODE = 8**  
This message is displayed on SYSPRINT and the System Console when FSI-JOURNAL encounters a parameter error in the journal definition during generation of the Journal Option Table. If multiple extents are desired, the MULTIPLE or MULT keyword must be specified.
- FSIJRNL-100**      **SHOWCB-xxxxxxx ERROR,  
RC=X'nn'(nn), EC=X'nn'(nn),  
ACB=A(aaaaaaaaa),  
PROCESSING WILL BE TERMINATED**  
This message is displayed on the System Console if a VSAM SHOWCB error occurs during Journal selection processing. xxxxxx describes the type of SHOWCB being attempted, RC and EC are the VSAM return and error codes in hex and decimal. Processing is terminated with a Jobstep abend code of 100.
- FSIJRNL-101**      **JOURNAL ID nnn CANNOT BE LOCATED IN OPTION TABLE,  
DATASET NAME 'cluster.name',  
PROCESSING WILL BE TERMINATED**  
This message is displayed on the System Console if no JOURNAL entry can be located in module CSIJRN00 (or execution-time overrides) for the JID specified in the DATASET entry matching the VSAM cluster specified in the message. This error indicates a logic error in program CSIJRNL. This situation should be detected in the GENERATE or EXEC functions before the application program is executed. Processing is terminated with a Jobstep abend code of 101.

- FSIJRNL-102**      **TESTCB-xxxxxxx ERROR,  
RC=X'nn'(nn), EC=X'nn'(nn),  
ACB=A(aaaaaaa),  
PROCESSING WILL BE TERMINATED**  
This message is displayed on the System Console if a VSAM TESTCB error occurs during Journal selection processing. xxxxxx describes the type of TESTCB being attempted, RC and EC are the VSAM return and error codes in hex and decimal. Processing is terminated with a Jobstep abend code of 102.
- FSIJRNL-103**      **xxxxxxx IS NOT AT THE SAME PRODUCT RELEASE LEVEL AS  
yyyyyyy  
JOURNALING WILL NOT BE ACTIVATED**  
This message is displayed on the System Console if all of the FSI-JOURNAL programs and tables are not part of the same product release level. xxxxxx is the program or table that was found to be in error. yyyyyy is the program that made the check. Journaling will not be activated.
- FSIJRNL-104**      **JOURNAL OPTION TABLE (CSIJRN00) RELEASE LEVEL IS  
INCOMPATIBLE, JOURNALING WILL NOT BE ACTIVATED  
CSIJRNL=x.xx, CSIJRN00=x.xx**  
This message is displayed on the System Console if the Journal Option Table is determined to be from an incompatible release of FSI-JOURNAL. No journaling will be performed. Regenerate your Journal Option Table with the current release of program CSIJRNL.
- FSIJRNL-200**      **JOURNAL FILE 'xxxxxxx', SUCCESSFULLY OPENED,  
DATASET NAME 'data.set.name'**  
This message is displayed on up to two destinations defined by the DEST operand of the STATS keyword on the CONTROL command. It is generated if the JOPEN operand of the STATS keyword is specified on a selected CONTROL command. It indicates that a journal dataset has been opened. 'xxxxxxx' is the DDNAME, and 'data.set.name' is from the DD statement.
- FSIJRNL-201**      **JOURNALING ESTABLISHED FOR 'xxxxxxx' AS 'yyyyyyy' DATASET  
NAME 'cluster.name'  
JOURNAL FILE 'jjjjjjj'  
FUNCTIONS JOURNALED (RO RU WU WN WD)**  
This message is displayed on up to two destinations defined by the DEST operand of the STATS keyword on the CONTROL command. It is generated if the OPEN operand of the STATS keyword is specified on a selected CONTROL command. It indicates that a VSAM dataset has been selected for journaling. 'xxxxxxx' is the VSAM DDNAME. 'yyyyyyy' is the 'forced' ddname from the DATASET command. 'cluster.name' is the VSAM cluster name. 'jjjjjjj' is the journal DDNAME. 'RO RU WU WN WD' is based on the JREQ parameter of the DATASET command.

- FSIJRNL-202**      **DATASET 'xxxxxxx' CLOSED,  
nnnnn REQUESTS JOURNALED TO 'jjjjjjj',  
FUNCTION TOTALS (RO=nnn RU=nnn WU=nnn WN=nnn WD=nnn)**  
This message is displayed on up to two destinations defined by the DEST operand of the STATS keyword on the CONTROL command. It is generated if the CLOSE operand of the STATS keyword is specified on a selected CONTROL command. It indicates that a VSAM dataset has been closed and shows the total number of records journaled to journal ddname 'jjjjjjj'. It also provides totals by each type of VSAM function.
- FSIJRNL-203**      **JOURNAL FILE 'jjjjjjj' SUCCESSFULLY CLOSED,  
nnnnn REQUESTS JOURNALED,  
FUNCTION TOTALS (RO=nnn RU=nnn WU=nnn WN=nnn WD=nnn)**  
This message is displayed on up to two destinations defined by the DEST operand of the STATS keyword on the CONTROL command. It is generated if the JCLOSE operand of the STATS keyword is specified on a selected CONTROL command. It indicates that a journal dataset has been closed and shows the total number of records journaled to journal ddname 'jjjjjjj'. It also provides totals by each type of VSAM function.
- FSIJRNL-204**      **JOURNAL FILE <ddname>, PREMATURELY CLOSED, <nnnn>  
REQUESTS JOURNALED BEFORE CLOSURE.  
FUNCTION TOTALS (RU=nn WU=nn WD=nn ADD=nn)**  
This message is displayed on the designated output media when journal closures have been requested. It is the same as FSIJRNL-203 except it indicates an unsatisfactory closure. The function totals indicate only those records captured before the forced close. See FSIJRNL-209 for an indication of the number of records lost in the unwritten I/O buffer.
- FSIJRNL-209**      **JOURNAL FILE <ddname>, FOUND CLOSED, PROBABLY BY THE  
SYSTEM DURING TASK TERMINATION. AN I/O BUFFER WITH  
<nnnn> RECORDS WAS LOST.  
FUNCTION TOTALS (RU=nn WU=nn WD=nn ADD=nn)**  
This message is displayed on the SYSPRINT and SYSTEM CONSOLE when FSI-JOURNAL determines that the journal file has been force closed by the system. This can occur when applications leave VSAM files open at application termination or ATTACHED subtasks open VSAM files which causes a first open for associated journals. An unwritten I/O buffer was lost containing the number of records shown.
- FSIJRNL-210**      **'DD' STATEMENT MISSING FOR JOURNAL 'jjjjjjj',  
PROCESSING WILL BE TERMINATED**  
This message is displayed on the System Console if the required JCL for journal 'jjjjjjj' cannot be located. Processing is terminated with a Jobstep abend code of 210.

## FSIJRNL-211

**JOURNAL OPEN ERROR,**

```

... ERROR CODE -      xx
... FEEDBACK   -      xxxxxxxx
... FUNCTION   -      xx
... MODNAME    -      ppppppppp
... DDNAME     -      dddddddd
... DATASET    -      cluster.name

```

**PROCESSING WILL BE TERMINATED**

This message is displayed on the System Console if an error occurs while the program is attempting to open a journal dataset. Processing is terminated with a Jobstep abend code of 211.

ERROR CODE	FEEDBACK	FUNCTION
01 GETMAIN/GETVIS failed	Amount of GETMAIN(hex)	00 Open input
02 FREEMAIN/FREEVIS failed		04 Open output
03 FREEMAIN/FREEVIS wrong area		08 Close
04 Bad block length field	Blocklength and "LL" field	0C Sequential input
05 Block too large	Blocklength and "LL" field	10 Sequential output (buffered)
06 Spanned record to large	Blocklength and "LL" field	14 Sequential output (force write)
07 VARBLK record to large	Blocklength and "LL" field	18 Extent open output
08 FIXBLK record to large	Blocklength and "LL" field	1C Extent close
09 FIXBLK invalid blocksize	Blocklength and "LL" field	20 Build output block
10 End of file		24 Write output block
11 Misc write error		28 Test close
12 OPEN/CLOSE error		2C Point
13 EOX entered with rc04		
14 EOX entered with rc08		
20 File already closed		

## FSIJRNL-212

**JOURNAL WRITE ERROR,**

```

... ERROR CODE -      xx
... FEEDBACK   -      xxxxxxxx
... FUNCTION   -      xx
... MODNAME    -      ppppppppp
... DDNAME     -      dddddddd
... DATASET    -      cluster.name

```

**PROCESSING WILL BE TERMINATED**

This message is displayed on the System Console if an error occurs while the program is attempting to write to a journal dataset. Refer to error number 211 for a description of the returned information. Processing is terminated with a Jobstep abend code of 212.

## FSIJRNL-213

**JOURNAL CLOSE ERROR,**

```

...ERROR CODE      -      xx
...FEEDBACK        -      xxxxxxxx
...FUNCTION        -      xx
...MODNAME         -      ppppppppp
...DDNAME          -      dddddddd
...DATASET         -      cluster.name

```

**PROCESSING WILL BE TERMINATED**

This message is displayed on the System Console if an error occurs while the program is attempting to close a journal dataset. Refer to error number 211

for a description of the returned information. Processing is terminated with a Jobstep abend code of 213.

- FSIJRNL-215**      **AN ERROR OCCURED DURING THE DYNAMIC ALLOCATION FOR JOURNAL 'jjjjjjj', AS JID(nn), WITH A DATASET NAME OF dsname, THE FOLLOWING MESSAGE GIVES THE REASON FOR THE DYNAMIC ALLOCATION ERROR**
- This message is displayed on the System Console if the Dynamic Allocation for journal 'jjjjjjj' is rejected by MVS. Message FSIJRNL-2001 will also be generated providing the reason for the rejection. Processing is then terminated with a Jobstep abend code of 216.
- FSIJRNL-217**      **IF THIS IS A DEVICE 'NOT AVAILABLE' OR 'NOT READY' AND IT IS POSSIBLE TO MAKE THE DEVICE AVAILABLE, READY DEVICE AND REPLY 'RETRY' OTHERWISE REPLY 'ABORT'**
- This message is displayed on the System Console when a dynamically allocated device fails allocation. it follows the message that gives the reason for the allocation failure. If the device can be placed in service and made available to the application, do so and reply 'RETRY' to the message. Otherwise you must reply 'ABORT' to clear the application from the system.
- FSIJRNL-220**      **SHOWCB-xxxxxxx ERROR,  
RC=X'nn'(nn), EC=X'nn'(nn),  
RPL=A(aaaaaaaaa), ACB=A(aaaaaaaaa),  
PROCESSING WILL BE TERMINATED**
- This message is displayed on the System Console if a VSAM SHOWCB error occurs during Journal output processing. "xxxxxxx" describes the type of SHOWCB being attempted, RC and EC are the VSAM return and error codes in hex and decimal. Processing is terminated with a Jobstep abend code of 220.
- FSIJRNL-221**      **TESTCB-xxxxxxx ERROR,  
RC=X'nn'(nn), EC=X'nn'(nn),  
RPL=A(aaaaaaaaa), ACB=A(aaaaaaaaa),  
PROCESSING WILL BE TERMINATED**
- This message is displayed on the System Console if a VSAM TESTCB error occurs during Journal output processing. "xxxxxxx" describes the type of TESTCB being attempted, RC and EC are the VSAM return and error codes in hex and decimal. Processing is terminated with a Jobstep abend code of 221.
- FSIJRNL-232**      **('jobname') END OF VOLUME FOUND FOR 'current' JOURNAL EXTENT ON DSN 'dddddd.dddd' SWITCHING TO 'next' JOURNAL EXTENT.**
- This message is displayed on the System Console when the journal manager determines that it has reached the end of the current journal dataset and has switched to the next extent of the defined journal set. The value of 'current' may be PRIMARY or SECONDARY. The value of 'jobname' will be the application Jobname. The value of 'next' will be either SECONDARY or PRIMARY.

- FSIJRNL-233** (**'jobname'**) **OPEN FOR 'current' JOURNAL EXTENT WAS SUCCESSFUL, NOW JOURNALLING ON 'current' JOURNAL EXTENT 'ddname'**  
This message is displayed on the System Console when the journal manager has successfully opened the 'current' extent of the journal set. The value of 'current' will be PRIMARY or SECONDARY depending upon how many switches have been executed.
- FSIJRNL-235** (**'jobname'**) **CLOSE FOR 'current' JOURNAL EXTENT WAS SUCCESSFUL, JOURNAL DATA ON JOURNAL DATASET ' 'dataset name' MAY BE ARCHIVED.**  
This message is displayed on the System Console when the journal manager has successfully closed the 'current' extent of the journal set. The value of 'current' will be PRIMARY or SECONDARY depending upon how many switches have been executed. The dataset named may be archived to another dataset name on DASD or tape to free the journal extent for the next possible switch to it.
- FSIJRNL-236** (**'jobname'**) **JOURNAL DATA ON 'dataset.name' IS STILL PROTECTED, A SWITCH TO THE 'next', JOURNAL EXTENT IS HELD. IF DATA WAS ARCHIVED, RESPOND ACCORDINGLY ELSE ARCHIVE DATA AND RESPOND OR REQUEST OVERLAY OF DATA OR ABORT THE JOB, ENTER ARCHIVED, OVERLAY, OR ABORT'**  
This message is displayed on the System Console when the journal manager attempts to switch to a previously written extent of the journal set. Since there is no interface between any archive utility and the journal manager, the protection switch is never altered except when the reply is "ARCHIVED" or "OVERLAY" A response to this message is required to allow the application job to continue processing or to be terminated abnormally.
- If the journal extent has not been archived yet, the journal manager will hold the job until the operator can accomplish the archival of the data. When the archival job completes satisfactorily, the operator may respond with "ARCHIVED"
- If the operator responds "OVERLAY" without archiving the data contained on the journal extent protected, all the data will be overlaid, destroying any chance of using that data to support a recovery attempt.
- FSIJRNL-237** (**'jobname'**) **CLOSE FOR 'current' JOURNAL EXTENT FAILED, PROGRAM IS TERMINATED CONDITION CODE = 8**  
This message is displayed on the System Console when the journal manager has attempted to close the 'current' extent of the journal set and has failed for some reason. The data on the journal may be good up to the last record block, but the data set may not be properly closed and may not be useable. The job is terminated abnormally to prevent loss of recover capability.

- FSIJRNL-238**      **(jobname) OPEN FOR 'next' JOURNAL EXTENT FAILED, PROGRAM IS TERMINATED CONDITION CODE = 8**  
 This message is displayed on the System Console when the journal manager has attempted to open the 'next' extent of the journal set and has failed for some reason. The journal manager will not allow the application job to run without journal support. The job is terminated abnormally.
- FSIJRNL-239**      **REPLY FROM OPERATOR 'reply'**  
 This message is displayed on the System Console when the operator replies to message FSIJRNL-236 as a means of tracking the response that was entered to the request.
- FSIJRNL-240**      **ABEND INTERCEPT PROCESSING STARTED**  
 This message is displayed on the System Console if an abend causes the FSI-JOURNAL abend exit to be entered. The abend exit will attempt to close all open journals. Message FSIJRNL-203 will be generated on the System Console for each journal closed. Message FSIJRNL-242 will be generated on the System Console for any journal that could not be closed.
- FSIJRNL-241**      **ABEND INTERCEPT PROCESSING COMPLETED**  
 This message is displayed on the System Console when the FSI-JOURNAL abend exit completes its processing. Refer to message FSIJRNL-240 for more information.
- FSIJRNL-242**      **JOURNAL CLOSE ERROR,**  
     ... ERROR CODE            - xx  
     ... FEEDBACK             - xxxxxxxx  
     ... FUNCTION             - xx  
     ... MODNAME             - pppppppp  
     ... DDNAME              - dddddddd  
     ... DATASET             - cluster.name  
 This message is displayed on the System Console if an error occurs while the FSI-JOURNAL abend exit is attempting to close a journal dataset. Refer to message FSIJRNL-211 for a description of the returned information.
- FSIJRNL-250**      **USEREXIT 'pppppppp' COULD NOT BE LOADED FOR JOURNAL 'jjjjjjj' PROCESSING WILL BE TERMINATED**  
 This message is displayed on the System Console if the USEREXIT "pppppppp" specified in a JOURNAL command cannot be loaded from any library in the current JOBLIB/STEPLIB search chain. Processing is terminated with a Jobstep abend code of 250.
- FSIJRNL-251**      **USEREXIT 'pppppppp' COULD NOT BE LOADED FOR JOURNAL 'jjjjjjj', IGNORE OPTION HAS BEEN SPECIFIED**  
 This message is displayed on the System Console if the USEREXIT "pppppppp" specified in a JOURNAL command cannot be loaded from any library in the current JOBLIB/STEPLIB search chain. The IGNORE option

has been specified, therefore the USEREXIT is ignored, and processing continues.

- FSIJRNL-290**      **ENTRY, REQ=X'xx'(dddddd), RPL=A(rrrrrrrr), ACB=A(aaaaaaaaa), DDN=nnnnnnnn, WS=wwwwwwww**
- This message is displayed on the destination defined by the DEST operand of the TRACE keyword on the CONTROL command. It is generated if the ENTRY operand of the TRACE keyword is specified on a selected CONTROL command. It indicates that the journaling module has been entered for the specified request (in hex and a descriptive name) with the VSAM RPL and ACB addresses displayed. The DDNAME is shown after the first request for any given ACB. The WS is an internal work area associated with the ACB used by the journaling module.
- FSIJRNL-291**      **ACB EXTRACT, ACB=A(aaaaaaaaa), WS=wwwwwwww, DDN=nnnnnnnn, DSN=cluster.name, TYPE=xxDS, KEYLEN=000, RKP=000, LRECL=00000, CINV=00000, CIMODE=N, JFLAGS=nnnnnn**
- This message is displayed on the destination defined by the DEST operand of the TRACE keyword on the CONTROL command. It is generated if the ACB operand of the TRACE keyword is specified on a selected CONTROL command. It is generated each time a new ACB is detected by the journaling module. It displays various required information about the cluster. CIMODE is a Y/N flag for whether the ACB is in Control interval mode. FSI-JOURNAL cannot journal processing performed in Control interval mode. JFLAGS is a series of six Y/N flags. The first will be 'N' if NO journaling has been specified. The remaining five indicate via the Y/N if each of the functions are to be journaled (RO, RU, WU, WD, WN). The WS is an internal work area associated with the ACB used by the journaling module.
- FSIJRNL-292**      **RPL EXTRACT, RPL=A(aaaaaaaaa), DDN=nnnnnnnn, RECL=00000, RBA=X'00000000', AREA=X'aaaaaaaa', ARG=X'aaaaaaaa', UPD=n, LOC=n, CIMODE=n**
- This message is displayed on the destination defined by the DEST operand of the TRACE keyword on the CONTROL command. It is generated if the RPL operand of the TRACE keyword is specified on a selected CONTROL command. It is generated each time a VSAM request is completed. It displays various required information from the RPL. RPL address, DDNAME, current record length, current RBA in hex, address of current record (AREA), address of current key (ARG). UPD is a Y/N flag for whether the RPL is in update mode (add versus replace). CIMODE is a Y/N flag for whether the RPL is in Control interval mode. FSI-JOURNAL cannot journal processing performed in Control interval mode.
- FSIJRNL-293**      **rrrrrrrr PROCESSING, FC=cc, RC=x'rc', JFLAG=y, DDN=nnnnnnnn, ACB=A(aaaaaaaaa), RPL=A(aaaaaaaaa), RECL=00000, RBA=X'00000000', AREA=X'aaaaaaaa', ARG=X'aaaaaaaa', UPD=n, LOC=n, CIMODE=n**
- This message is displayed on the destination defined by the DEST operand of the TRACE keyword on the CONTROL command. It is generated if the GET, PUT or ERASE operands of the TRACE keyword are specified on a

selected CONTROL command. It is generated each time one of the specified requests is completed.

It displays various required information from the RPL and Journal Option Table. FC is the function code for the journal (RO,RU,WU,WN,WD), RC is the VSAM return code for the request, JFLAG is a Y/N indicating whether the request is to be journaled, DDN is the ddname, RECL is the current record length, RBA is the current RBA in hex, AREA is the address of current record, ARG is the address of current key, UPD is a Y/N flag for whether the RPL is in update mode (add versus replace), LOC is a Y/N flag for whether the current request is in locate mode (versus move mode), CIMODE is a Y/N flag for whether the RPL is in Control interval mode. FSI-JOURNAL cannot journal processing performed in Control internal mode.

**FSIJRNL-294**      **JOURNAL WRITE TO 'jjjjjjj', FUNC=(fc), DATASET='dataset.name',  
RPL=Aaaaaaaaa, ACB=Aaaaaaaaa, JBXRPL=Aaaaaaaaa,  
JBXBAR=Aaaaaaaaa)**

This message is displayed on the destination defined by the DEST operand of the TRACE keyword on the CONTROL command. It is generated if the JWRITE operand of the TRACE keyword is specified on a selected CONTROL command. It is generated each time a record is written to a journal. It displays the journal ddname, FUNC is the function code for the journal (RO,RU,WU,WN,WD). JBXRPL and JBXBAR are used internally by FSI-JOURNAL to allow RU records to be journaled only if an update occurs. If JBXBAR is not zero, the current record was held for the RPL at JBXRPL.

**FSIJRNL-297**      **CSIJRNIX EXIT HAS FAILED TO FIND CSIJRN01 CLOSE FOR  
JOURNALS ABORTED.**

This message is displayed on the System Console if the End Of Job (EOJ) exit, CSIJRNIX, fails to find CSIJRN01, the base options table containing addresses necessary to control close processing and return to system. Since the system return address is also lost, CSIJRNIX will force an abend.

**FSIJRNL-298**      **CSIJRNIX COULD NOT FIND DUMMY BJCA CLOSE FOR JOURNALS  
ABORTED**

This message is displayed on the System Console if the End Of Job (EOJ) exit, CSIJRNIX, fails to find the pointer to the control block needed to control close processing. An abend will be forced to invoke system close functions.

**FSIJRNL-299**      **CSIJRNIX FOUND OPEN VSAM FILE CONTROL BLOCKS IN  
CONTROL CHAIN**

This message is displayed on the System Console if the End Of Job (EOJ) exit, CSIJRNIX, finds more than one batch journal control block on the chain. This indicates that some VSAM files may still be open. The EOJ exit will attempt closure of all open files.

**FSIJRNL-300**      **ERROR ON CSIJRNIX LINK TO CSIJRN20**

This message is displayed on the System Console if the End Of Job (EOJ) exit, CSIJRNIX, finds that the link to the journal module was not successful.

This may indicate that the module address was corrupted and the link could not be executed. Check SYSLOG for other messages that might indicated cause.

- FSIJRNL-301      ERROR ON CSIJRNIX LINK TO CSIJRN07**  
This message is displayed on the System Console if the End Of Job (EOJ) exit, CSIJRNIX, finds that the link to the journal module was not successful. This may indicate that the module was not found on a load and thus the link could not be executed. Check SYSLOG for other messages that might indicated cause.
- FSIJRNL-302      CLOSE FOR VSAM FILE <ddname> BY CSIJRNIX HAS FAILED WITH RC=nn, EC=nn**  
This message is displayed on the System Console if the End Of Job (EOJ) exit, CSIJRNIX, detects that the VSAM close request has failed. The RC and EC values will be displayed in HEX. Consult appropriate VSAM manuals to determine cause of failure. No further journal action will be attempted with the VSAM file. Journal closure will proceed despite the failure.
- FSIJRNL-303      CSIJRNIX SUCCESSFULLY CLOSED FILE DDN=<ddname>**  
This message is displayed on the System Console when the End Of Job (EOJ) exit, CSIJRNIX, has successfully closed an open VSAM file found on the control block chain. This message is displayed in conjunction with FSIJRNL-299.
- FSIJRNL-304      NO ACB ADDRESS FOR DDN=<ddname> CLOSE ABORTED**  
This message is displayed on the System Console when the End Of Job (EOJ) exit, CSIJRNIX, determines that the journal control block does not contain a valid VSAM ACB address that can be used for file closure. This message is displayed in conjunction with FSIJRNL-299.
- FSIJRNL-305      ACB FOR DDN=<ddname> IS ALREADY CLOSED OR NEVER OPENED**  
This message is displayed on the System Console when the End Of Job (EOJ) exit, CSIJRNIX, determines that the journal control block points at a VSAM ACB that is or has not been opened. This message is displayed in conjunction with FSIJRNL-299.
- FSIJRNL-1000     'x' IS AN INVALID COMMAND**  
The first word in the line, "x", is not a valid command. If it is a parameter intended to be part of a previous command, check for valid continuation on the end of the previous line. If it is supposed to be part of a comment, check for the proper use of "/" (start of comment), ";" (end of command), and continuation characters.
- FSIJRNL-1001     'x' IS AN INVALID KEYWORD FOR 'y'  
SUB-PARAMETERS FOLLOWING IT ARE IGNORED**  
The character string "x" was specified as a sub-parameter of "y", and is not a valid keyword for "y". Any sub-parameters coded for "x" are ignored. If "x"

was not intended to be a sub-parameter of "y", check your use of parentheses. If "x" is a command, a previous command may have been continued improperly.

- FSIJRNL-1002**    **'x' EXCEEDS THE NUMBER OF POSITIONAL PARAMETERS ALLOWED FOR 'y'**  
**SUB-PARAMETERS FOLLOWING IT ARE IGNORED**  
Positional parameters are allowed for "y", but the maximum number allowed had already been specified before "x" was found. Any sub-parameters coded for "x" are ignored. If "x" was not intended to be a sub-parameter of "y", check your use of parentheses. If "x" is a command, a previous command may have been continued improperly.
- FSIJRNL-1003**    **'x' IS INVALID - NO PARAMETERS ALLOWED FOR 'y'**  
**SUB-PARAMETERS FOLLOWING IT ARE IGNORED**  
No sub-parameters are allowed for "y", but "x" was coded as a sub-parameter. Any sub-parameters coded for "x" are ignored. If "x" was not intended to be a sub-parameter of "y", check your use of parentheses.
- FSIJRNL-1004**    **'x' CAN BE SPECIFIED ONLY n TIMES FOR 'y'**  
**SUB-PARAMETERS FOLLOWING IT ARE IGNORED**  
A list of repeated sub-parameters (maximum of "n") are allowed for "y", but "x" exceeds this maximum. Any sub-parameters coded for "x" are ignored. If "x" was not intended to be a sub-parameter of "y", check your use of parentheses.
- FSIJRNL-1006**    **'x' IS NOT A VALID NUMERIC VALUE**  
The character string "x" was coded where a numeric value is expected, but it is not a valid number. If "x" was not intended to be a number, it may be positioned incorrectly in relation to other sub-parameters at the same level, or incorrect use of parentheses may have caused it to be associated with the wrong parameter.
- FSIJRNL-1007**    **'x' EXCEEDS THE ALLOWED NUMBER OF n CHARACTERS/DIGITS**  
For a non-numeric positional parameter, "x" contains too many characters. For a numeric parameter, "x" is too large a number for the positional parameter. If "x" is correct as coded, it may be positioned incorrectly in relation to other sub-parameters at the same level, or the incorrect use of parentheses may have caused it to be associated with the wrong parameter.
- FSIJRNL-1009**    **DELIMITER 'x' IS INVALID FOLLOWING 'y', AND IS IGNORED**  
The delimiter "x" was coded where it is not allowed. A "/" delimiter is allowed only after certain parameters, and a "(" cannot immediately follow another delimiter.
- FSIJRNL-1010**    **EXTRA RIGHT PARENTHESES FOLLOWING "x" IGNORED**  
More right parentheses ")" have been found than are needed. Either a left parenthesis "(" is missing, or too many right parentheses were coded.

- FSIJRNL-1011 MISSING n RIGHT PARENTHESES AT END OF COMMAND**  
The end of command has been reached, but “n” left parentheses “(” have not been paired with right parentheses “)”. Either a continuation character is missing (the next line will start a new command), or too few right parentheses “)” were coded.
- FSIJRNL-1012 COMMENT IS NOT TERMINATED WITH '\*/', OR CONTINUATION IS MISSING ON THIS LINE**  
A comment was started with “/\*” but was not terminated properly. Comments follow the same continuation rules as parameters, and must be terminated by “\*/”.
- FSIJRNL-1013 STRING 'x' IS NOT TERMINATED WITH A QUOTE, OR CONTINUATION IS MISSING ON THIS LINE**  
The character string “x” was started as a quoted string, but an ending quote (') was not found.
- FSIJRNL-1014 STRING 'x' IS NOT FOLLOWED BY A SEPARATOR CHARACTER, OR ONE QUOTE WAS CODED WHERE TWO ARE REQUIRED**  
The character string “x” was started as a quoted string, and another quote (') was found which was not followed by a separator character (to end the string).
- FSIJRNL-1015 END OF INPUT REACHED - LAST INPUT LINE CANNOT BE CONTINUED**  
The last line in the SYSIN dataset ended with a continuation character. Since there are no more lines to process, continuation is invalid. If you did not intend the SYSIN dataset to end at this point, check your input for “/\*” appearing in the first position of a SYSIN line. This is treated as end-of-file, not the start of a comment.
- FSIJRNL-1016 COMMAND NOT PROCESSED DUE TO SYNTAX ERRORS - CONDITION CODE n**  
The condition code set by previous errors for this command exceeds the maximum allowed. The highest condition code for all errors for this command is “n”.
- FSIJRNL-1096 PATCH HEX STRING CONTAINS ODD NUMBER OF DIGITS**  
The internal debugging facility has been invoked to patch a module. A hexadecimal verify or replace string was coded which does not represent a whole number of bytes. For example, X'47F0' is valid; x'47F' is not valid.
- FSIJRNL-1097 PATCH VERIFY ERROR**  
The internal debugging facility has been invoked to patch a module, but the data being verified does not match what is in the module.

- FSIJRNL-1098**      **INTERNAL ERROR - CANNOT FIND TYPE=FINAL**  
An internal logic error has occurred. Request assistance from a support representative.
- FSIJRNL-1099**      **INTERNAL ERROR - CANNOT FIND TYPE=SEG FOR xxxxxxxx**  
An internal logic error has occurred. Request assistance from a support representative.
- FSIJRNL-1100**      **(module+offset) INTERNAL ERROR - INVALID MESSAGE NUMBER**  
**X'xxxxxxx'(n)**  
An internal program error has occurred at the location indicated by the module and offset information. The module has attempted to issue an error number which is not in any message tables. When requesting technical assistance for this error, please state the full error message.
- FSIJRNL-1101**      **(module+offset) function/macro INTERNAL ERROR - RETURN CODE**  
**X'xxxxxxx'(n)**  
An internal program function has issued an operating system macro which failed. The function was called at the location indicated by the module and offset information. When requesting technical assistance for this error, please state the full error message.
- FSIJRNL-1102**      **(module+offset) function/macro INTERNAL ERROR - RETURN CODE**  
**X'xxxxxxx'(n) WHILE ISSUING MESSAGE mmmm**  
While attempting to issue message number "mmmm", an internal program function issued an operating system macro which failed. The function was called at the location indicated by the module and offset information. Also refer to message "FSIJRNL-mmmm" which may help you determine the cause of the problem. When requesting technical assistance for this error, please state the full error message.
- FSIJRNL-1103**      **GETMAIN(GM) / FREEMAIN(FM) STATISTICS**  
**GMCOUNT=n      GMTOT=n**  
**FMCOUNT=n      FMTOT=n**  
**CURRENT=n      MAXIMUM=n**  
**LARGEST SINGLE GETMAIN=n**  
An internal debugging function has requested a display of the memory statistics.
- FSIJRNL-1104**      **(module+offset) function/macro PROGRAM 'xxxxxxx' NOT FOUND**  
FSI-JOURNAL attempted to load program "xxxxxxx", but it was not found in the Load Libraries available in the job step. The load function was called at the location indicated by the module and offset information. This message is issued only if the program cannot be found. Other load errors will produce message FSIJRNL-1105 or FSIJRNL-1108

- FSIJRNL-1105**      **function/macro REQUEST FOR n BYTES OF MAIN STORAGE FAILED. USE A LARGER REGION PARAMETER**  
 The indicated internal function issued an operating system macro which required more main storage than was available. If the number of bytes needed is not known, the message will say "REQUEST FOR MAIN STORAGE FAILED". Execute CSIJRNL with a larger region parameter. If the problem persists, determine the cause of excessive main storage use.
- FSIJRNL-1106**      **(module+offset) function/macro INVALID PRINT FILE 'filename'**  
 FSI-JOURNAL was unable to open "filename" as a printer dataset. Check your JCL and parameters. If you request technical assistance for this error, please state the full error message.
- FSIJRNL-1108**      **(module+offset) function/macro INTERNAL ERROR - ABEND CODE X'xxxx', REASON CODE X'xxxx'**  
 An internal program function has issued an operating system macro which failed. The function was called at the location indicated by the module and offset information. The system abend code and reason code are displayed. When requesting technical assistance for this error, please state the full error message.
- FSIJRNL-1150**      **(module+offset) xxxxxxxx FUNCTION HAS BEEN ENTERED**  
 The operating system interface module has been called from (module+offset) to perform a "xxxxxxx" function. This message is issued only if the internal trace has been turned on.
- FSIJRNL-2001**      **ALLOCATE REQUEST FAILED, RC=xx, EC=xxxx, IRC=xxxx**  
 This error occurs if a dynamic allocation request is rejected by MVS.

RC is the "Return Code", it indicates the type of error:

- 04 - An error resulted from the current environment, the unavailability of a system resource, or a system routine failure; there will also be an error reason code.
- 08 - The installation validation routine denied this request.
- 0C - The error is due to an invalid parameter list; there will also be an error reason code.

EC is the "Error Reason Code", it explains the reason for the return code. The following table contains a partial list of error reason codes, a complete list can be found in the MVS or MVS/XA System Macros and Facilities manual 'GC28-1150'.

CLASS 2 CODES (UNAVAILABLE SYSTEM RESOURCE)	
0204	Real storage unavailable.
020C	Request for exclusive use of a shared data set cannot be honored.
0210	Requested data set unavailable.
0214	Unit(s) not available.
0218	Specified volume or an acceptable volume is not mounted.
021C	Unit name specified is undefined.
0220	Requested volume not available.
0224	Eligible device types do not contain enough units.

0228	Specified volume or unit in use by system.
022C	Volume mounted on ineligible permanently resident or reserved unit.
0230	Permanently resident or reserved volume on required unit.
0240	Requested device is a console.
0260	Unit does not meet specified status requirements.
0264	Invalid request due to current unit status.
0268	Tape device is broken.
CLASS 3 CODES (INVALID PARAMETER LIST)	
0340	Dsname not specified and disposition not "new".
035C	Invalid PARM specified in text unit. <sup>1</sup>
0364	JOBLIB/STEPLIB/JOB CAT/STEP CAT specified as ddname.
0368	Authorized function requested by unauthorized user.
0380	Mutually exclusive KEY specified. <sup>1</sup>
0384	Mutually inclusive KEY specified. <sup>1</sup>
0388	Required KEY not specified.
0390	GDG specified with relative generation number exceeds 35 chars.
0394	Status and relative generation number are incompatible.
039C	Device type and volume are incompatible.
CLASS 4 CODES (ENVIRONMENTAL ERROR)	
0410	Specified ddname unavailable.
0420	Specified ddname or dsname associated with an open data set.
0440	Specified dsname not found.
0448	Request for a new data set failed; data set already exists.
044C	Request was made for a data set that has a disposition of delete.
0458	Dsname in DCB reference is a GDG group name.
0484	Request denied by operator.
0488	GDG Pattern DSCB not mounted.
048C	GDG Pattern DSCB not found.
<sup>1</sup> The Information Reason Code contains the value of the key that caused the error.	

IRC is the "Information Reason Code", it further explains the reason for the return code. The contents of this field will vary depending on the Error Return Code. The primary contents of this field will be the value of the "KEY" that caused the error. The following table contains a list of the KEY fields used:

0001	Ddname
0002	Data Set Dsname
0003	GDG / Member name
0004	Data Set Status
0005	Data Set Normal Disposition
0006	Data Set Conditional/Abnormal Disposition
0007	Track Space Type
0008	Cylinder Space Type
0009	Block length (Space allocation)
000A	Primary Space Quantity
000B	Secondary Space Quantity
000D	Unused Space Release (RLSE)
0010	Volume Serial number (VOLSER)
0015	Unit Description (UNIT)
001C	Unallocation at Close (FREE)
001E	Label Type
001F	Label Sequence Number
0022	Expiration Date (EXPDT)
0023	Retention Period (RETPD)
002C	DCB=dsname
0030	DCB=BLKSIZE
0042	DCB=LRECL
0049	DCB=RECFM
0055	Returned DDNAME
0056	Returned DSNAME
0057	Returned DSORG

**FSIJRNL-2002**

**DE-ALLOC REQUEST FAILED, RC=xx, EC=xxxx, IRC=xxxx**

This error occurs if a dynamic de-allocation request is rejected by MVS. Refer to message 2001 for a description of RC, EC and IRC.

**FSIJRNL-9998      WARNING - PRODUCT AUTHORIZATION EXPIRES IN 30 DAYS OR LESS**

This message is displayed on the system console if the copy of FSI-JOURNAL you are using is about to expire. Contact your FSI product sales representative for a new Expiration Password.

**FSIJRNL-9999      PRODUCT AUTHORIZATION HAS EXPIRED**

FSI-JOURNAL is protected with an Expiration Password. This message is displayed on the system console if that Password has expired. Contact your FSI product sales representative for a new Expiration Password.



## **APPENDIX A**



## How to Code Commands (Syntax)

This section describes the rules for coding FSI-JOURNAL commands and their operands. The meaning and use of the commands themselves are not described here. The examples given here are intended only to clarify the coding syntax, not to show how to use a particular command or operand.

Commands are coded as one or more lines in a SYSIN type dataset or as one or more logical lines in the PARM= operand of the EXEC JCL statement. Each command may be continued using any number of lines.

FSI-JOURNAL reads its SYSIN type commands from a DD statement named CSIJRNSI. If this DD is found in the JCL, the PARM= operand on the EXEC statement will be ignored.

A command may contain comments, which can be coded anywhere within the command. Comments may also be coded separately, rather than as part of a command.

### Terminology

The special terms defined below are used to describe the command syntax.

- **COMMAND**: A command is a FSI-JOURNAL-defined command name followed by operands and comments. Each command is defined and processed by FSI-JOURNAL as a single entity, and may or may not be related to other commands. When coding commands on the PARM= operand of the EXEC statement, each command is terminated by coding a semi-colon.
- **KEYWORD operand**: Keyword operands are represented in command descriptions with upper-case letters, and indicate FSI-JOURNAL-defined words which either identify sub-operands that follow the keyword in parentheses, or modify the action taken during the processing of a command.
- **POSITIONAL operand**: Positional operands are represented in command descriptions with lower-case letters, and indicate user-provided data needed by the command.

- **SEPARATOR CHARACTER:** A separator character is used to end an operand, and to indicate the relationship of that operand to other operands. The separator characters are: space, comma, left and right parentheses, slash, and semi-colon.
- **COMMENTS:** FSI-JOURNAL allows you to code comments in the SYSIN dataset. Comments are for your own documentation and are printed, but not processed, with the commands.

### Coding FSI-JOURNAL Operands

FSI-JOURNAL commands all have this general structure:

COMMAND-NAME operands ... ;

The semi-colon (;) is optional, and signals the end of the command. If you are using SYSIN commands, anything coded after a semi-colon is ignored. If you are using PARM= commands, the semi-colon marks the end of the current command and the beginning of the next command. The command-name must be the first word in a command, terminated by a space. It is normally followed by operands which further define the action to be taken by the command. The operands which are allowed after the command-name are separated using one or more spaces. If an operand requires one or more sub-operands, the sub-operands are coded in parentheses in this manner

```
COMMAND  PARM-1  PARM-2(SUB-A(... ) SUB-B) PARM-3...
```

where PARM-1, PARM-2, SUB-A, SUB-B, and PARM-3 are valid keyword or positional operands defined for the command named "COMMAND". Only operands defined as valid for PARM-2 may be coded as sub-operands following PARM-2. The operands PARM-1, PARM-2, and PARM-3 are considered to be at the same logical level, regardless of any sub-operands specified for each of them. operands and sub-operands of all logical levels follow the same coding rules.

The command-name, operands, and separator characters may be preceded or followed by any number of spaces. This can be used to make commands more readable, and easier to maintain.

**Keyword** operands can be coded in any order, unless the description of a command states otherwise. Some keywords may be defined as having one or more prefixes which alter the meaning of the keyword.

**Positional** operands may be coded in a variety of ways, to allow you to specify any type of data. If the data needed for an operand contains separator characters or non-printable characters, code the operand as a **quoted string** or as a **hexidecimal string**. If an operand is defined as a number, you may code it either as a decimal number, or use hexadecimal notation. The relative position of each positional operand within the command determines the sequence in which they are acted upon.

### Coding Continuation

You may specify any FSI-JOURNAL command on more than one input line by coding either a "+" or "-" as the **last non-blank character** of all but the last line of the command. If a "-" is used as the continuation character, the command is continued starting at the **left margin** of the following line. If a "+" is used, the command is continued starting with the **first non-blank character** of the following line. All data on the line up to, but not including, the "+" or "-" is used. The following examples will all have the same result:

```

COMMAND PARM-1 PARM-2 (SUB-A(DATA) SUB-B) PARM-3

COMMAND PARM-1 PARM-2 (SUB-A(DA+
      TA) SUB-B) PARM-3

COMMAND PARM-1 PARM-2 (SUB-A(DA-
      TA) SUB-B) PARM-3

COMMAND PARM-1
      PARM-2 (
                SUB-A(DATA)
                SUB-B
            )
      PARM-3

```

Continuation works in the same way regardless of whether it takes place in the middle of a word, within a quoted string or comment, between operands, or after a semi-colon (;). It takes place as the lines are read, before any other characters are interpreted.

### Coding Quoted Strings

Quoted strings are valid for any positional operand, to allow you to:

- Code data that contains separator characters - 'NOT FOUND'
- Code data that would otherwise be interpreted as a keyword.
- Code data in hexadecimal notation - x'C8C5E7'

- Code lower case data - 'This will remain lower case'

To specify the quote character within a quoted string, code two consecutive quotes:

```
'THIS STRING CONTAINS "QUOTES" IN IT'
```

### Coding Comments

It is often useful to put comments with the operands. Since a semi-colon ";" ends the operand processing of a command, anything coded after a semi-colon is treated as a comment. If continuation is specified after a semi-colon, the command is continued to the next line.

If your are specifying commands with the PARM= operand of the EXEC statement, the semi-colon cannot be used for comments. The semi-colon is used on the PARM= to indicate the end of one command and the beginning of the next.

A more flexible method of coding comments is provided. A comment may appear at any point that a separator character is valid, by coding "/\* comment \*/". Any characters may appear between the "/\*" and "\*/", and the comment may be continued to any number of lines by using the normal continuation method. If you do not end the comment, or code a continuation character on the line, an error message is issued. In the following examples, the lower case characters represent data that are treated as comments:

```
COMMAND PARM-1;parm-2(sub-a(data) sub-b) parm-3

COMMAND PARM-1      /* comment for parm-1 */           -
      PARM-2(      /* comment for parm-2 */           -
              /* more comments */                     -
      SUB-A(DATA)   -
      )             -
      /* a single comment can be-
      continued */
```

## **APPENDIX B**



## Journal Record Formats

The following record layouts describe the formats of FSI-JOURNAL Journal records. Journal datasets are created using variable-length spanned blocked records. The format is determined by the FORMAT operand of the JOURNAL command.

FORMAT(SHORT)			
OFFSET	LENGTH		DESCRIPTION
000	(000)	2	"LL" field - Length of record
002	(002)	2	"BB" field - Set to x'0000'
004	(004)	2	Prefix length (includes LLBB fields)
006	(006)	2	Length of Key (following prefix)
008	(008)	2	Length of VSAM Record (following prefix)
010	(00A)	2	(reserved)
012	(00C)	8	Dataset ID (from ACB or DATASET command)
020	(014)	4	Date (packed 00YYDDD)
024	(018)	4	Time (packed HHMMSS)
028	(01C)	8	Journal Block Number (hex xxxxxxxx)
036	(024)	2	Relative Record in Block (packed nnn)
038	(026)	1	Journal ID (binary JID value)
039	(027)	2	Source of this record: FC =VSAM File Control BJ = Batch Journal (FSI-JRNL)
041	(029)	8	APPLID (from SMF or APPLID parm)
049	(031)	4	Transaction ID
053	(035)	4	Terminal ID
057	(039)	3	Task number (packed nnnnn)
060	(03C)	2	Function Code: RO = Read Only RU = Read for Update WU = Write Update WN = Write New WD = (Write) Delete
062	(03E)	1	(reserved)
063	(03F)	1	(reserved)
----- End of Prefix -----			
064	(040)	variable	Key field (RIDFLD)
064	(040)	variable	Record Data (follows Key field)



FORMAT(LONG)			
OFFSET		LENGTH	DESCRIPTION
000	(000)	2	"LL" field - Length of record
002	(002)	2	"BB" field - Set to x'0000'
004	(004)	2	Prefix length (includes LLBB fields)
006	(006)	8	Dataset ID (from FCT)
014	(00E)	1	(reserved)
015	(00F)	1	(reserved)
016	(010)	2	Key length
018	(012)	256	Key field (RIDFLD), Left justified, x'00' filled
274	(112)	4	Date (packed 00YYDDD)
278	(116)	4	Time (packed HHMMSS)
282	(11A)	8	Journal Block Number (hex xxxxxxxx)
290	(122)	2	Relative Record in Block (packed nnn)
292	(124)	1	(reserved)
293	(125)	1	Journal ID (binary JID value)
294	(126)	2	Source of this record: FC = VSAM File Control BJ = Batch Journal (FSI-JRNL)
296	(128)	8	APPLID (from SMF or APPLID parm)
304	(130)	4	Transaction ID
308	(134)	4	Terminal ID
312	(138)	3	Task number (packed nnnnn)
315	(13B)	2	Function Code: RO = Read Only RU = Read for Update WU = Write Update WN = Write New WD = (Write) Delete
317	(13D)	1	(reserved)
318	(13E)	2	(work area for FSIRECOV)
320	(140)	2	length of data record
322	(142)	44	(reserved)
----- End of Prefix -----			
366	(16E)	variable	Record Data



# Index

---

---

- (continuation) .....	87
+	
+ (continuation) .....	87
<b>A</b>	
ABEND	
JOURNAL Command .....	41
ACB	
CONTROL Command .....	47
Activating FSI-JOURNAL .....	19
ALLOCATE	
JOURNAL Command .....	38
ALLOCATE-A	
JOURNAL Command .....	38
Alternate Extent .....	15
Alternate indexes .....	16
APPLID	
IDENTIFY Command .....	43
ARG	
CONTROL Command .....	48
ASUFFIX	
JOURNAL Command .....	38
<b>B</b>	
BLKSIZE	
JOURNAL Command .....	40
<b>C</b>	
Chained RPLs .....	17
CICS .....	13, 17
Journals .....	13
CLOSE	
CONTROL Command .....	46
Coding Operands .....	86
Commands .....	31
Coding Syntax .....	86
Continuation Syntax .....	87
CONTROL Command .....	45
DATASET Command .....	33
EXEC Command .....	53
GENERATE Command .....	51
IDENTIFY Command .....	43
JOURNAL Command .....	36
OPTION Command .....	55
SELECT Operand .....	56
START Command .....	49

STOP Command .....	50
Syntax .....	31
Comments .....	86
Comments (syntax) .....	88
Compatibility .....	18
Continuation Syntax .....	87
CONTROL command	
OPEN .....	46
CONTROL Command .....	45
ACB .....	47
ARG .....	48
CLOSE .....	46
DATA .....	48
DEST .....	46, 47
ENTRY .....	47
ERASE .....	48
GET .....	48
how used .....	23
JCLOSE .....	47
JOPEN .....	46
JWRITE .....	48
PUT .....	48
RPL .....	47
SELECT .....	46
STATS .....	46
TRACE .....	47
Control Interval Access .....	17
CSIJRN00 .....	23
example .....	24
<b>D</b>	
DATA	
CONTROL Command .....	48
DATASET Command .....	33
how used .....	23
JID .....	35
JREQ .....	34
NAME .....	33
SELECT .....	35
DDNAME	
JOURNAL Command .....	36
Deactivating FSI-JOURNAL .....	21
DEFEROPEN	
JOURNAL Command .....	41
DEST	
CONTROL Command .....	46, 47
DISP	
JOURNAL Command .....	40

# Index

---

---

<b>E</b>	
ENTRY	
CONTROL Command .....	47
ERASE	
CONTROL Command .....	48
Example Journal Option Table .....	25
EXCLUDE.Operand .....	57
EXEC Command .....	53
JCLOSE.....	53
PGM= .....	53
Execution Time Overrides.....	29
EXPDT	
JOURNAL Command .....	40
<b>F</b>	
FORCEOPEN	
JOURNAL Command .....	41
FORMAT	
JOURNAL Command .....	37
Format, journal records.....	91
Forward recovery .....	13
Forward Recovery	
Introduction.....	4
FREE	
JOURNAL Command .....	40
FSI-RECOV .....	13
<b>G</b>	
GENERATE Command .....	51
STARTCODE.....	51
STOPCODE .....	52
TERMEXIT .....	51
GET	
CONTROL Command .....	48
<b>H</b>	
Hexidecimal Notation.....	87
<b>I</b>	
IDENTIFY Command .....	43
APPLID .....	43
how used.....	23
SELECT .....	44
TASK .....	44
TRANTERM.....	44
IEAAPFxx .....	9
IGNORE	
JOURNAL Command .....	41
INCLUDE Operand.....	57
Installation .....	8
Load Modules .....	8
Source Modules.....	8
Introduction.....	3
Features.....	5
What is Recovery .....	4
<b>J</b>	
JCLOSE	
CONTROL Command .....	47
EXEC Command .....	53
JID	
DATASET Command.....	35
JOURNAL Command .....	36
JOPEN	
CONTROL Command .....	46
Journal	
dataset type.....	13
format .....	13
record formats .....	91
JOURNAL Command .....	36
ABEND.....	41
ALLOCATE .....	38
ALLOCATE-A.....	38
ASUFFIX .....	38
BLKSIZE.....	40
DDNAME .....	36
DEFEROPEN .....	41
DISP .....	40
EXPDT .....	40
FORCEOPEN .....	41
FORMAT .....	37
FREE .....	40
how used.....	23
IGNORE .....	41
JID .....	36
LABEL .....	40
MODDCB.....	40
MULTIPLE.....	37
PSUFFIX.....	38
RETPD .....	41
SPACE .....	41
UNIT .....	41
USEREXIT .....	41
VOLSER .....	41
Journal Option Table	
Example.....	25
JREQ	

# Index

---

---

DATASET Command .....	34	Positional Operand.....	85
JWRITE		Primary Extent .....	15
CONTROL Command.....	48	PSUFFIX	
<b>K</b>		JOURNAL Command.....	38
Keyword Operand .....	85	PUT	
<b>L</b>		CONTROL Command .....	48
LABEL		<b>Q</b>	
JOURNAL Command.....	40	Quoted Strings .....	87
LNKLSTxx .....	9	<b>R</b>	
LOADLIB		Recovery .....	13
START Command .....	49	Restrictions .....	17
Lower Case Data.....	87	Chained RPLs.....	17
<b>M</b>		CICS.....	17
Messages		Control Interval Access .....	17
FSIJRNL-10nn .....	75	Share option 4 .....	17
FSIJRNL-11nn .....	77	RETPD	
FSIJRNL-20nn .....	79	JOURNAL Command.....	41
FSIJRNL-nnn.....	61	RPL	
Migration Requirements .....	6	CONTROL Command .....	47
MODDCB		RPLs.....	17
JOURNAL Command.....	40	<b>S</b>	
MULTIPLE		SELECT	
JOURNAL Command.....	37	CONTROL Command .....	46
Multiple Extent Journals.....	15	DATASET Command .....	35
<b>N</b>		IDENTIFY Command .....	44
NAME		SELECT Operand .....	56
DATASET Command .....	33	Example .....	58
<b>O</b>		How to enter .....	56
OPEN		Rules for strings.....	57
CONTROL command.....	46	Selection Rules .....	57
Operands		Separator character.....	85
Coding Syntax .....	86	Share option 4 .....	16, 17
Comments .....	88	SPACE	
Continuation Syntax .....	87	JOURNAL Command.....	41
Quoted Strings.....	87	START Command .....	49
OPTION Command .....	55	LOADLIB.....	49
SUSPEND.....	55	STARTCODE.....	49
TOLERATE .....	55	STARTCODE	
Option Table.....	23	GENERATE Command .....	51
example.....	24	START Command .....	49
<b>P</b>		STATS	
PGM=		CONTROL Command .....	46
EXEC Command .....	53	STOP Command .....	50
		STOPCODE .....	50
		STOPCODE	
		GENERATE Command .....	52

# Index

---

STOP Command .....	50
SUSPEND	
OPTION Command .....	55
Syntax	
Coding Operands .....	86
Comments.....	88
Continuation .....	87
Quoted Strings .....	87
Terminology .....	85
SYS1.PARMLIB.....	9
<b>T</b>	
TASK	
IDENTIFY Command .....	44
TERMEXIT	
GENERATE Command .....	51
TOLERATE	
OPTION Command .....	55
TRACE	
CONTROL Command .....	47
TRANTERM	
IDENTIFY Command .....	44
<b>U</b>	
UNIT	
JOURNAL Command .....	41
USEREXIT	
JOURNAL Command .....	41
Using FSI-JOURNAL .....	13
Activating .....	19
Deactivating .....	21
Execution Time Overrides.....	29
Option Table.....	23
<b>V</b>	
VOLSER	
JOURNAL Command .....	41