

z/OS
3.2

DFSMSdfp Checkpoint/Restart



Note

Before using this information and the product it supports, read the information in [“Notices” on page 87.](#)

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-09-30

© **Copyright International Business Machines Corporation 1976, 2025.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures.....	vii
Tables.....	ix
About This Book.....	xi
Required product knowledge.....	xi
z/OS information.....	xi
How to provide feedback to IBM.....	xiii
Summary of changes.....	xv
Summary of changes for z/OS 3.2.....	xv
Summary of changes for z/OS 3.1.....	xv
Chapter 1. Introduction.....	1
Components.....	1
CHKPT Macro Instruction.....	1
End-of-Volume Exit Routine.....	1
Checkpoint at End-of-Volume Facility.....	1
Restart Definition (RD) Parameter.....	1
RESTART Parameter.....	2
SYSCHK DD Statement.....	2
SYSCKEOV DD Statement.....	2
Dependencies.....	2
Controlled Use of Checkpoints.....	2
Specification of Codes Eligible for Restart.....	2
Job Journal Requirements.....	2
User Data Set Security.....	3
Checkpoint Data Set Security.....	3
Restrictions.....	3
Chapter 2. Coding a Checkpoint.....	5
When to Code a Checkpoint.....	5
Checkpoint at End-of-Volume.....	5
Checkpoint in Exit Routines.....	7
Coding the DCB for a Checkpoint Data Set.....	7
Required DCB Parameters.....	7
DCB Options.....	7
Notes on the DCB for a Checkpoint Data Set.....	8
Coding the DD Statement for a Checkpoint Data Set.....	8
Notes on the DD Statement.....	8
Using Checkpoint Data Sets.....	9
Storage Estimates for Checkpoint Data Sets.....	9
How Checkpoint Entries Are Written.....	11
How to Ensure Restart with Sequential Checkpoint Data Sets.....	12
How Checkpoint Entries Are Identified.....	13
Coding the CHKPT Macro Instruction.....	14
CHKPT Return Codes.....	16
Use of CHKPT with Other Macro Instructions.....	16

Canceling a Checkpoint.....	18
Serially Reusable Resources.....	19
Requesting Serially Reusable Resources.....	19
Special Operating System Features.....	20
VIO Data Sets.....	20
Dynamic Allocation and Dynamically Concatenated Data Sets.....	20
Shared Resources.....	21
Cross-memory support restrictions.....	21
Data Spaces.....	22
Name/Token Services.....	22
Chapter 3. Requesting Restart.....	23
Types of Restart.....	23
Automatic Restart.....	23
Operator Message Sequence.....	23
Operator Considerations.....	24
How to Request Automatic Checkpoint/Restart.....	25
Automatic Step Restart.....	25
How Data Sets Being Extended Are Handled during Automatic Step Restart.....	26
JCL Requirements and Restrictions on Automatic Restart.....	27
Resource Variations Allowed in Automatic Restart.....	27
How the System Works at Automatic Restart.....	27
Deferred Restart.....	28
Message Sequence.....	28
Operator Considerations.....	29
Deferred Step Restart.....	29
Deferred Checkpoint/Restart.....	31
How the System Works during a Deferred Checkpoint/Restart.....	34
Coding the RD and RESTART Parameters.....	35
RD Parameter.....	35
RESTART Parameter.....	35
SYSCHK DD Statement.....	36
Chapter 4. Checkpoint/Restart Processing.....	39
Notes for User Data Sets.....	39
BDAM Data Sets.....	39
Extended Format Data Sets.....	40
Generation Data Sets.....	40
Graphics Data Sets.....	41
IEFDDSRV macro.....	41
Large Format Data Sets.....	41
Multivolume Data Sets.....	41
Partitioned Data Sets (PDSS).....	41
Partitioned Data Sets Extended (PDSEs).....	42
SMS-Managed Sequential Data Sets.....	43
Card Readers.....	43
SYSABEND or SYSUDUMP Data Sets.....	43
SYSIN Data Sets.....	43
SYSOUT Data Sets.....	44
Temporary Data Sets.....	44
VSAM Data Sets.....	44
Work Data Sets.....	47
Notes for Tape Labels and Files.....	47
VSE Tape Files.....	47
ISO/ANSI Tape Labels.....	47
Nonstandard Labels.....	47
Bypass Label Processing (BLP).....	47

Automated Tape Library.....	47
Repositioning and Preserving the Contents of Data Sets.....	48
Recording Data Set Positioning.....	48
Buffering of Data Set Records.....	49
Preserving Data Set Contents.....	49
Allocating Devices during Deferred Checkpoint/Restart.....	50
Handling Checkpoint/Restart Errors.....	50
I/O errors.....	50
Internal Errors.....	51
User Errors.....	51
Using Checkpoint/Restart with Other Programs.....	51
Job and Job Step Accounting.....	51
JES3.....	52
COBOL.....	52
DFSORT™.....	53
PL/I.....	53
Virtual Fetch.....	53
Appendix A. Checkpoint/Restart Codes.....	55
Return Codes.....	55
Reason codes.....	56
Modules and Return Codes for IHJ Messages.....	81
Appendix B. End-of-Volume Checkpoint Routine.....	83
Appendix C. Accessibility.....	85
Notices.....	87
Terms and conditions for product documentation.....	88
IBM Online Privacy Statement.....	89
Policy for unsupported hardware.....	89
Minimum supported hardware.....	89
Trademarks.....	91
Glossary.....	93
Index.....	105

Figures

- 1. Using One Sequential Checkpoint Data Set to Ensure Restart..... 13
- 2. Using Two Sequential Checkpoint Data Sets to Ensure Restart..... 13
- 3. Recording a Checkpoint Identification Assigned by the Control Program..... 14
- 4. Obtaining Updated TCB Information after Restart.....17
- 5. Canceling a Request for Automatic Restart.....19
- 6. Requesting a Resource after Restart..... 19
- 7. Requesting Automatic Checkpoint/Restart..... 25
- 8. Requesting Automatic Step Restart.....26
- 9. Requesting a Deferred Step Restart..... 30
- 10. Requesting a Deferred Checkpoint/Restart..... 32

Tables

1. Determining the Size of a Checkpoint Data Set.....	9
2. Cross-memory support restrictions.....	21

About This Book

This book is intended to help you understand the checkpoint/restart facility provided by DFSMS. The checkpoint/restart facility lets you:

- Record information about a job at designated checkpoints.
- Restart a job at the beginning of a step or at a checkpoint in a step.

Required product knowledge

To use this book effectively, you should be familiar with job control language (JCL) and data management.

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see [How to send feedback to IBM](#).

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under [IBM z/OS Product Documentation Update Policy](http://www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy) (www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy).

Summary of changes for z/OS 3.2

The following content is new, changed, or no longer included in z/OS 3.2.

New

The following content is new.

September 2025 release

- None.

Changed

The following content is changed.

September 2025 release

- None.

Deleted

The following content is deleted.

September 2025 release

- None.

Summary of changes for z/OS 3.1

The following content is new, changed, or no longer included in z/OS 3.1.

New

The following content is new.

May 2024 refresh

- The topic, [“IEFDDSRV macro” on page 41](#), is added.

September 2023 release

- None.

Changed

The following content is changed.

September 2023 release

- Return code 08 of reason code 117 is updated. For more information, see [“Reason codes” on page 56](#).

Chapter 1. Introduction

This chapter contains the following sections:

- [“Components” on page 1](#)
- [“Dependencies” on page 2](#)

Checkpoint/restart is a method of recording information about a job at programmer-designated checkpoints so that the job can be restarted at one of these checkpoints or at the beginning of a job step.

A checkpoint is taken when your program or the system end-of-volume function issues the CHKPT macro instruction. This macro causes the contents of the program's virtual storage area and certain system control information to be written as a series of records in a data set. The restart function retrieves records from the data set if the job terminates abnormally or produces erroneous output, and the job can be restarted. Restart can take place immediately (allowed by the operator at the console) or be deferred until the job is resubmitted. In either case, you can avoid the time-consuming process of rerunning the entire job from the beginning.

Components

Checkpoint/restart consists of the following components.

CHKPT Macro Instruction

Your program or end-of-volume optionally issues the CHKPT macro.

When a CHKPT macro is executed, the contents of the program's virtual storage data area and certain system control information are written in a data set as a series of records. The series of records is called a checkpoint entry, and the data set in which they are written is called a checkpoint data set. The checkpoint entry, which has a unique programmer-specified or system-generated identification called a *checkid*, is retrieved from the data set when restart occurs.

For a detailed explanation about establishing a checkpoint see [Chapter 2, “Coding a Checkpoint,” on page 5](#).

End-of-Volume Exit Routine

You can code an end-of-volume exit routine that can issue a CHKPT macro instruction each time the processing of a multivolume physical sequential user data set is continued on another volume. [Appendix B, “End-of-Volume Checkpoint Routine,” on page 83](#) contains a brief description of this routine. For more detailed information about coding the EOVS exit for physical sequential data sets see [z/OS DFSMS Using Data Sets](#).

Checkpoint at End-of-Volume Facility

A system-supplied routine takes checkpoints at end-of-volume occurrences for multivolume data sets opened with QSAM, BSAM, or EXCP and can be invoked with a JCL parameter. “Checkpoint at End-of-Volume” on [page 5](#) contains a brief description of this routine. See [z/OS DFSMS Using Data Sets](#).

Restart Definition (RD) Parameter

When the RD parameter is used, it is coded in the JOB or EXEC statements and is used to request automatic step restart if job failure occurs, or to suppress, partially or totally, the action of the CHKPT macro instruction. For more information about this parameter see [Chapter 3, “Requesting Restart,” on page 23](#).

RESTART Parameter

The RESTART parameter, coded in the JOB statement, is required when a job is resubmitted for deferred restart. It specifies the step (for deferred step restart) or the step and the checkpoint within that step (for deferred checkpoint/restart) where restart should begin. For more information about this parameter see [Chapter 3, “Requesting Restart,” on page 23](#).

SYSCHK DD Statement

The SYSCHK DD statement is required when a job is resubmitted for deferred checkpoint/restart. It defines the checkpoint data set for the job being restarted. For more information about this statement, see [“SYSCHK DD Statement” on page 36](#).

SYSCKEOV DD Statement

The SYSCKEOV DD statement defines the checkpoint data set that is to contain the checkpoint records generated when a checkpoint is taken at end-of-volume. For more information about this statement see [“SYSCKEOV DD Statement” on page 7](#).

Dependencies

When using checkpoint/restart, consider the following dependencies.

Controlled Use of Checkpoints

Your installation may have special procedures in place to control the use of checkpoints on DASD. Contact your system administrator to determine whether this is the case for your system. For more information about protecting checkpoint data sets on shared DASD, see [Control of Checkpoint Data Sets on Shared DASD Volumes in z/OS DFSMS Using Data Sets](#).

Specification of Codes Eligible for Restart

The CKPTREST macro is no longer used to define codes eligible for restart. Instead, use the following technique.

During system installation, your system programmer uses a standard IBM® supplied list of system completion codes (codes produced when the system executes abend) to create the table of codes eligible for automatic restart. To change this table, the SYS1.PARMLIB member SCHEDxx RESTART/NORESTART statement type is specified. The RESTART statement type is used to add user-defined completion codes to the table of codes eligible for automatic restart. The NORESTART statement type is used to delete both IBM-supplied and user-defined codes from the table.

Job Journal Requirements

The job journal is a sequential data set that resides on the spool volume of the job entry subsystem. It contains a set of selected job-related control blocks that are critical to automatic restart processing.

The job journal is necessary because the scheduler control blocks are maintained in the scheduler work area (SWA) in pageable storage, rather than in a job queue on external storage. When a job or the system itself fails, the space that contains the SWA and its job control blocks is lost. Because it preserves up-to-date copies of certain critical control blocks, the job journal makes it possible to reconstruct the SWA. SWA control blocks are restored to the state they were in just prior to the failing step for automatic step restart. For automatic checkpoint/restart, SWA control blocks are reconstructed as they appeared at the time of the most recently issued CHKPT.

Job journaling is required before a job is eligible for an automatic restart.

For JES2: Job journaling is provided to a job in JES2 in one of the following ways:

- JOURNAL is specified on the JOB CLASS CHARACTERISTICS statement in the SYS1.PARMLIB member used for initializing JES2 (see *z/OS JES2 Initialization and Tuning Guide*).
- The JCL for the job specifies RD parameter on the JOB statement or the EXEC statement (see *z/OS MVS JCL Reference*).

For JES3: Job journaling is provided to a job in JES3 in one of three ways:

- JOURNAL=YES is specified on the CLASS initialization statement.
- JOURNAL=YES is specified on the MAIN JCL statement that overrides the CLASS initialization statement. For more information about the MAIN statement see *z/OS MVS JCL Reference*.
- The JCL has specified either the RESTART or the RD parameter on the JOB statement or the RD parameter on the EXEC statement.

The system creates a job journal to hold restart information for any job specifying job journaling. After a system failure and the system restart of the failing main processor, the jobs in execution that requested job journaling are restarted (warm started) by the system. If a job is eligible for automatic restart, the operator is sent the message IEF225D asking if the job should be restarted. If the job is not eligible for restart or if the operator indicates that the restart should not be attempted, any scratch or VIO data sets the job allocated are deleted, and the job terminates. Jobs that frequently use scratch or VIO data sets should request job journaling.

User Data Set Security

In order to restart a job step that processes RACF[®] protected data sets, you must be authorized to access all RACF[®] protected data sets that were open when the checkpoint was taken. In order to restart, all password-protected VSAM data sets that were open at checkpoint time must have the pointer to their password in the access method control block (ACB) when the checkpoint is taken, or the password must be known to the operator. For RACF[®] protected data sets, the userid specified on the RESTART job statement must be authorized to access the data set.

Checkpoint Data Set Security

An unauthorized program is defined as one that is not authorized by the authorized program facility (APF), not in supervisor state and does not have a system key (keys 0 through 7). An unauthorized program cannot communicate directly with the checkpoint data set. The unauthorized program can take checkpoints and do restarts, but after the operation has begun, a security interface is used by the system to prevent unauthorized alteration of the checkpoint data set.

If an unauthorized program tries to access a checkpoint data set directly, an error message appears, and the job terminates. In addition, the unauthorized program cannot take a checkpoint on a new checkpoint data set if another data control block (DCB) is already open to that data set.

Offline security of the checkpoint data set is ensured by the operator.

Restrictions

When using checkpoint/restart, consider the following restrictions that might cause the checkpoint, restart, or both operations to fail. This list is in addition to other restrictions contained in this documentation.

- Having open sequential extended format data sets, including compressed and encrypted data sets.
- Having open sequential encrypted basic or large format data sets.
- Having open partitioned data sets extended (PDSEs).
- Use of any application or z/OS function that obtains 64-bit virtual storage (storage above the bar).
- Use of the EXECUTABLE=NO option on the STORAGE OBTAIN macro (Instruction Execution Protection storage).
- Use of the Language Environment option "CEENXSTG ON", which obtains EXECUTABLE=NO heap storage.

- Use of AMP='RMODE31=CB' or AMP='RMODE31=ALL' on VSAM data sets, which results in 64-bit virtual storage being used by VSAM. If the application uses either of these parameters, AMP='RMODE31=BUFF' can be used on the JCL DD statement for the data set to override what it specified in the application or SMS data class.
- Environments that do not disable the use of non-executable memory for the JCL PARM area with the CBATTR EXECUTABLE(JCLPARM) option in the DIAGxx member of Parmlib. By default, the system obtains storage for the parameter area passed to the program using EXECUTABLE=NO memory.
- Users of CICS Transaction Server for z/OS 6.1 and above who specify the feature toggle **com.ibm.cics.sm.iep=true** to require DSA instruction execution protection.

Chapter 2. Coding a Checkpoint

The CHKPT macro instruction is coded in your program. When the CHKPT macro executes, job step information (about your program, virtual storage data areas, data set position, and supervisor control) is written as a checkpoint entry in a checkpoint data set. The point at which this information is saved becomes a checkpoint from which a restart may be performed. After the checkpoint entry is written, control returns to your program at the instruction following the CHKPT macro.

This chapter defines a checkpoint and explains how to establish checkpoints to restart job steps. This chapter contains the following sections:

- [“When to Code a Checkpoint” on page 5](#)
- [“Coding the DCB for a Checkpoint Data Set” on page 7](#)
- [“Coding the DD Statement for a Checkpoint Data Set” on page 8](#)
- [“Using Checkpoint Data Sets” on page 9](#)
- [“Coding the CHKPT Macro Instruction” on page 14](#)
- [“Canceling a Checkpoint” on page 18](#)
- [“Serially Reusable Resources” on page 19](#)
- [“Special Operating System Features” on page 20](#)

When to Code a Checkpoint

Using the CHKPT macro instruction, you can code a checkpoint within your program or at end-of-volume. The checkpoint routine records information about certain system information, user storage (including user programs in user storage), and the data sets used by the step executing the CHKPT macro. Recorded information about data sets includes the following:

- For all data sets, the checkpoint routine records the information which can be coded on a DD statement, such as the device type and volume serial numbers. The contents of the step's job file control blocks (JFCBs) and job file control block extensions (JFCBEs) are also recorded, as well as the contents of the generation data group name tables (GDGNT).
- For data sets which are (1) open at the checkpoint, (2) are processed on either magnetic tape or direct access devices, (3) and use the BSAM, QSAM, BPAM, VSAM, or EXCP access methods, the checkpoint routine records the information needed to reposition the data set if restart occurs at a checkpoint.

If a checkpoint is taken and an output data set is extended onto a second direct access volume because end-of-volume occurred on the first volume and there was no more space available on the volume, or because the data set contained 16 extents, restart subsequently occurs at that checkpoint and the system does not delete the extension of the data set.

Note that, when a step using the universal character set (UCS) feature is restarted, the system does not determine if the UCS buffer or forms control buffer (FCB) buffer is properly loaded, nor does it alert the operator to the UCS or FCB requirements of the step.

Checkpoint at End-of-Volume

The checkpoint at end-of-volume provides an external checkpoint function with no user program modifications. It is executed immediately prior to the invocation of any checkpoint in a user EOVS exit routine. Redundancy occurs if a user exit routine is supplied and checkpoint at end-of-volume is invoked for the same data set's end-of-volume occurrence.

Checkpoint at end-of-volume is executable only if the CHKPT=EOV parameter is specified for multivolume data sets or for the second, third, etc., data sets of a sequential concatenation. If the first data set of a concatenation is a multivolume data set, this parameter is also valid on that DD statement.

Checkpoint at end-of-volume issues a CHKPT macro. If, at EOVS, a return code of 08 is presented, the CHKPT macro is executed again. Return code 08 indicates that end-of-extent or end-of-volume occurred for the SYSCKEOV DD before completion of the checkpoint entry. The retry of return code 08 uses secondary space, if provided.

For the other unsuccessful return codes (0C, 10, 14, or 18) and for unsuccessful retry of return code 08, the following message is generated:

IEC067I CHKPT=EOV FACILITY EXECUTED UNSUCCESSFULLY

This message is preceded by a checkpoint/restart error message (prefixed "IHJ") that describes the nature of the problem. For more details about these error messages see *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, *z/OS MVS System Messages, Vol 2 (ARC-ASA)*, *z/OS MVS System Messages, Vol 3 (ASB-BPX)*, *z/OS MVS System Messages, Vol 4 (CBD-DMO)*, *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*, *z/OS MVS System Messages, Vol 6 (GOS-IEA)*, *z/OS MVS System Messages, Vol 7 (IEB-IEE)*, *z/OS MVS System Messages, Vol 8 (IEF-IGD)*, *z/OS MVS System Messages, Vol 9 (IGF-IWM)*, and *z/OS MVS System Messages, Vol 10 (IXC-IZP)*.

In any event, processing continues, and this message is generated for each unsuccessful invocation of the checkpoint at end-of-volume until step termination occurs. Operator intervention is required to halt further processing prior to step end.

CHKPT at EOVS JCL Parameter

The CHKPT=EOVS parameter on a DD statement requests that a checkpoint be taken for this job step at EOVS for the data set whose DD has this parameter. The following restrictions apply to the use of this JCL parameter:

1. The DD must define a QSAM or BSAM sequentially processed data set.
2. The QSAM or BSAM sequentially processed data set must be a multivolume data set or the second, third, etc., set of a concatenated data sets.
3. The DD statement must not define a SYSOUT data set. It can be a DD *, SUBSYS, or DD DATA type data set in a concatenation if it is after the first DD.
4. The JCL parameters DDNAME and DYNAM cannot be specified on the same DD statement with this parameter.
5. The DD must not define a checkpoint data set.

If restriction 1, 2, or 5 is not observed, the following will result:

- No action, no checkpoints are taken
- Processing continues as if the CHKPT=EOVS parameter were not specified.

If restriction 3 or 4 is not observed, the following will result:

- JCL error messages appear
- Processing is not initiated.

An example follows:

```
//DD1 DD DSN=DSN1,DISP=OLD,VOL=SER=(TAPE01,TAPE02),
//      UNIT=TAPE,CHKPT=EOV

//DD2 DD DSN=DSN2,DISP=OLD
//      DD DSN=DSNX,DISP=OLD,CHKPT=EOV
//      DD DSN=DSNY,DISP=OLD,CHKPT=EOV

//DD3 DD DSN=DSN3,DISP=NEW,VOL=(,,5),
//      UNIT=DISK,SPACE=(CYL,(300,300)),CHKPT=EOV
```

- DD1 — Multivolume data set
- DD2 — Concatenated data sets

- DD3 — Multivolume data set on disk

SYSCKEOV DD Statement

The SYSCKEOV DD statement defines the checkpoint data set to contain the checkpoint records generated from the checkpoint at end-of-volume. The same restrictions that apply to other checkpoint DD statements (see [“Coding the DD Statement for a Checkpoint Data Set” on page 8](#)) also apply to this DD statement, with the following exceptions:

- DISP=MOD is recommended to reduce risk of loss of checkpoint data in the event of a system failure while taking a checkpoint. This is because restart normally is from the most recent checkpoint in the data set. If you do not code DISP=MOD, each checkpoint overlays the previous one. See [“How Checkpoint Entries Are Written” on page 11](#) and [“How to Ensure Restart with Sequential Checkpoint Data Sets” on page 12](#).
- This DD must define a sequential BSAM data set (BPAM is not supported)
- All the DCB parameters are provided by the checkpoint at EOVR routine and should not be coded on the DD statement.

An example follows:

```
//SYSCKEOV DD DSN=CKPTDS,UNIT=TAPE,DISP=MOD
```

Checkpoint in Exit Routines

The CHKPT macro instruction must not be used in an exit routine other than the end-of-volume exit routine. You can take a checkpoint when a BSAM or QSAM data set reaches end-of-volume. For more detailed information about coding the EOVR exit for physical sequential data sets, see [Appendix B, “End-of-Volume Checkpoint Routine,” on page 83](#).

Coding the DCB for a Checkpoint Data Set

You must provide either a data control block (DCB) for the checkpoint data set or the name of the DD statement for the checkpoint data set.

Required DCB Parameters

If you provide a DCB, you must code the following parameters:

- DSORG=PS or PO (BSAM or BPAM data set organization)
- MACRF=W (WRITE macro instruction)
- DDNAME= name of DD statement for checkpoint data set.

To control space allocation you can provide DSORG also on the DD statement or in the data class definition. Optional DCB macro and DD statement parameters are in [“DCB Options” on page 7](#)

DCB Options

You may, optionally, provide the following DCB parameters:

- OPTCD=W (write validity checking)
- NCP=2 (number of channel programs)
- NCP=2 and OPTCD=C (chained scheduling)
- RECFM=U or UT (undefined record format)
- DEVD=DA or TA (direct access or tape device)
- TRTCH=C (data conversion with odd parity; parameter required).

[z/OS DFSMS Macro Instructions for Data Sets](#) contains detailed information about coding DCBs.

Notes on the DCB for a Checkpoint Data Set

The following restrictions apply to the DCB for a checkpoint data set:

- The checkpoint routine writes all checkpoint records in both 400- and 4096-byte blocks.
- Requests for two channel programs or chained scheduling apply to the writing of all checkpoint records. Such requests do not apply to the reading of checkpoint records for a restart.
- OPTCD=Q cannot be specified in the DCB, because checkpoints are not character data.
- Following errors that leave the DCB invalid (for example, a D37 abend), the DCB must be closed and opened again, if additional checkpoints will be taken using this DCB.
- Do not code the DCBE parameter on the DCB macro.

Coding the DD Statement for a Checkpoint Data Set

The DD statement for the checkpoint data set must define the data set in the usual way. The publication *z/OS MVS JCL Reference* contains detailed information about coding the DD statement.

[“Coding the DCB for a Checkpoint Data Set” on page 7](#) describes certain DCB parameters that you can code on the DD statement.

The only restrictions on the DD statement are:

- A checkpoint data set cannot be concatenated, because it is an output data set.
- The UNIT parameter must specify a tape or direct access device supported by BSAM or BPAM. The device can be specified by referring to a specific device, a device type, or a group of devices.

If direct access is specified, the device cannot be shared with another processor, unless your installation has special procedures in place to control the use of checkpoints on shared DASD. See *z/OS DFSMS Using Data Sets* for specific details. To avoid allocating to a shared DASD, a special generic device name should be generated at system definition time to include a nonshared DASD of a single device type.

- DEFER should not be coded in the DD statement.
- DISP=SHR cannot be specified.
- Secondary space allocation may be requested by the increment subparameter (see [“Notes on the DD Statement” on page 8](#)). The increment subparameter must specify at least enough space to contain two complete checkpoint entries. For information about calculating the length of the checkpoint entry see [“Storage Estimates for Checkpoint Data Sets” on page 9](#).
- The LABEL parameter of the DD statement describes the labels of a data set on magnetic tape. For a checkpoint data set, you must specify only IBM standard labels (SL or SUL). Nonstandard labels (NSL), no labels (NL), or International Organization for Standardization/American National Standards Institute (ISO/ANSI) labels (AL or AUL) cannot be specified for a checkpoint data set. If the label type is not specified, the operating system assumes that the data set has IBM standard labels (SL).
- OPTCD=Q cannot be specified as a DCB subparameter.
- CHKPT=EOV is ignored (checkpoint resets the value in the JFCB).
- The RLSE subparameter of the SPACE parameter cannot be used on the DD statement for a checkpoint data set.
- A checkpoint data set cannot be an extended format data set, a PDSE file, or an HFS file.

Notes on the DD Statement

- The initial disposition of the data set (as specified in the DISP operand of the DD statement) is used to position the checkpoint data set each time it is opened. For a discussion of a specific circumstance in which the only checkpoint entry that exists on a data set will be the last checkpoint taken (whether that entry is valid or not), see [“How Checkpoint Entries Are Written” on page 11](#).

- The final and conditional dispositions of the data set have their normal meanings. However, if termination is occurring and an automatic restart at a checkpoint is to occur, the system automatically keeps all data sets that are in use by the job, including the checkpoint data set.
- If you code UNIT or VOL=SER on the checkpoint DD statement, it has no effect unless changed by your installation's ACS routines. If the checkpoint data sets are to be restricted to specific volumes, a storage group should be defined for this purpose.
- Filling the current DASD extent or tape volume is called an end-of-volume (EOV) condition. If EOV is encountered while writing a checkpoint on a direct access or tape volume, a second attempt is made to create the checkpoint entry. If EOV occurs before the second attempt at writing the entry is completed, message IHJ0001 (invalid checkpoint) is issued. Control is returned to you with a X'08' return code in register 15 and a reason code of 027 in register 0.

The following example shows DD statements for the checkpoint data set:

```
//ddname DD DSN=dsname,UNIT=TAPE,DISP=(MOD,KEEP)
//ddname DD DSN=dsname,UNIT=SYSDA,
// DISP=(NEW,DELETE,KEEP),SPACE=(CYL,(15,17)),
// VOL=SER=CKPTDS
```

Using Checkpoint Data Sets

The checkpoint data set must be empty at the beginning of a job step. For more information, see [“How Checkpoint Entries Are Written”](#) on page 11.

Storage Estimates for Checkpoint Data Sets

The following information can be used as a guideline in determining the size of this data set.

Table 1 on page 9 contains the size and number of records written when a checkpoint is taken. The number of tracks or the amount of tape occupied by the checkpoint data set can be determined by applying the number of records and their sizes against either the track capacities of the direct access device or the recording density and type for the magnetic tape device.

Table 1. Determining the Size of a Checkpoint Data Set

Description of Record	Size (in bytes)	Number Required
Checkpoint header record (CHR)	400	1
DD name table (DDNT)	400	$((D+20)/21)+1)/2$
Data set descriptor record (DSDR)	400	$(A+1)/2$
Problem program image record (PPIR)	4096	B
Supervisor record (SUR)	4096	C
Subsystem checkpoint record (SSCR)		
subsystem data sets	4096	Number of SYSIN or SYSOUT plus number of DDs with SUBSYS=
SPIE	20/4096	E/20
VSAM	4096	1

Table 1. Determining the Size of a Checkpoint Data Set (continued)

Description of Record	Size (in bytes)	Number Required
Records Code:		
A = the number of data sets defined in the job step		
B = (Storage allocated - freespace + (20 * number of gaps in allocated storage))/4076.		
Example of calculation of B:		
Storage allocated = 9000 to 4FFFF and 1A000000 to 2FFFFFFF		
Freespace=		
Address	Size	Number of Gaps
A200 to A3FF	200	1
32900 to 35FFF	3700	1
40000 to 47FFF	8000	1
50000 to 19FFFFFF	0	1 (not allocated; counts as a gap)
1F900000 to 210F8FFF	17F9000	1
28700000 to 29FFFFFF	1900000	1
=	3104900	6

Therefore:

B = ((47000 + 16000000) - 3104900 + (X'14' * 6)) / X'FEC'

B = 130C0

Note: All values in the preceding example are in hexadecimal.

C =

A variable number of records for system and protected data management control blocks. This will be approximately three records plus one record for every three DASD data sets and one record for every ten non-DASD data sets.

D =

Number of JCL-specified data sets dynamically unallocated at checkpoint.

E =

One for every 90 ESPIEs established before checkpoint.

Table 1. Determining the Size of a Checkpoint Data Set (continued)

Description of Record	Size (in bytes)	Number Required
Checkpoint/restart's dynamic storage requirement varies with the resources used by the caller. The largest impact on the dynamic storage requirement results from the amount of storage allocated to the job and the amount of storage fragmentation. Checkpoint/restart's dynamic storage is allocated out of subpools 229 and 230.		

Fragmented Storage

User fragmented storage may result in:

- Reduced performance of a checkpoint
- Insufficient storage for a checkpoint to contain a description of the user's storage (checkpoint is not completed).

How Checkpoint Entries Are Written

Unopened DCB Passed to Checkpoint

- Sequential data sets
 - When a sequential data set's disposition is NEW or OLD, the checkpoint entry is written at the beginning of the data set, even if a checkpoint entry was written on a previous call.
 - When a sequential data set's disposition is MOD, the checkpoint entry is written after the last entry existing in the data set.
- PDSs
 - When the data set is partitioned, each checkpoint entry is a member, and its checkid is its member name. After it writes a checkpoint entry, the checkpoint routine executes the STOW macro to add the checkid of the entry to the directory of the data set.
 - If the data set is partitioned, regardless of its disposition, the checkpoint entry is written after the last entry existing in the data set.
 - If an identical checkid already exists in the directory, the related address of a member is changed and becomes the address of the new checkpoint entry. The initial disposition specified for the checkpoint data set has no effect on the STOW operation.

Opened DCB Passed to Checkpoint

If your program has a DCB and opens the checkpoint data set for output, the checkpoint routine writes a checkpoint entry at the data set's current position and does not close the data set. A user-opened checkpoint data set does not have to be closed after taking the last checkpoint for the job step. All the checkpoint entries are saved in this case, without specifying DISP=MOD. This provides the ability to request a deferred restart from any of the checkpoints.

- Sequential data sets

For sequential data sets, DISP=OLD, NEW, or MOD has no effect on how the checkpoint entries are written on the checkpoint data set. The first time each checkpoint data set is used, the checkpoint entry is written at the beginning of the file. Each following checkpoint entry is written after the previous entry.
- PDSs

PDSs are treated in the same manner as unopened DCBs passed to checkpoint.

The DCB is left open when checkpoint returns to the user. The open checkpoint data set is closed by step termination.

DDNAME Passed to Checkpoint

If your program provides a DDNAME instead of a DCB, the checkpoint routine opens the checkpoint data set the first time it receives that DDNAME.

- Sequential data sets

For a sequential checkpoint data set, the data set's disposition controls where the first checkpoint is written. If the data set's disposition is NEW or OLD, the first checkpoint entry is written at the beginning of the data set, even if a previous checkpoint call has written an entry. If the data set's disposition is MOD, the first checkpoint entry is written after the last entry existing in the data set. DISP=OLD, NEW, or MOD has no effect on where checkpoint entries other than the first are written. Each following entry is written after the previous entry.

- PDSs

PDSs are treated in the same manner as nonopened DCBs passed to checkpoint.

Note: If your program uses more than one DDNAME on executions of the CHKPT macro, the data set for the previous DDNAME is closed and the data set for the new DDNAME is opened.

General Considerations

If end-of-volume is encountered during writing of a checkpoint entry, a second attempt is made to create the checkpoint entry on another volume if another volume has been allocated. If the checkpoint data set extends during writing of a checkpoint entry, a second attempt is made to create the checkpoint entry by writing over the first attempt and using the new extents. If EOVS occurs or the data set extends before the second attempt at writing the entry is completed, message IHJ0001 (invalid checkpoint) is issued. Control is returned to you with a X'08' return code in register 15 and a reason code of 27 in register 0.

The status (opened or closed) and position of a checkpoint data set remain the same at restart as they were after execution of the CHKPT macro instruction that established the checkpoint.

The first time a checkpoint data set is opened, checkpoint routines cause the OPEN macro to request security information for the data set from the operator.

Note: A checkpoint data set must contain only checkpoint entries. If a program attempts to write its own data in a checkpoint data set, message IEC954I will be issued, followed by an abend code of 23F.

How to Ensure Restart with Sequential Checkpoint Data Sets

Abnormal termination or system failure may occur while the new entry is written. To ensure that restart at the most recent valid checkpoint is possible, a new checkpoint entry must not be written over a preceding one. The following figures suggest methods by which you can ensure that restart is possible. All of the methods use sequential checkpoint data sets.

Note: Because of the characteristics of PDSs, checkpoint entries with different member names are not written over each other.

Figure 1 on page 13 shows the use of one sequential checkpoint data set, one data control block, and one DD statement (CHECKDD) specifying MOD disposition. The user allows the checkpoint routine to open and close the data set each time it writes a checkpoint entry. Checkpoint entries are written sequentially in the data set.

```

Program
.
.
CHKPT      CHKDCB
.
.
CHKDCB     DCB   DDNAME=CHECKDD,MACRF=W,DSORG=PS
.
.
DD Statement

//CHECKDD DD UNIT=TAPE,DISP=(MOD,KEEP),DSN=WAREHOUS.DB1.CHK

```

Figure 1. Using One Sequential Checkpoint Data Set to Ensure Restart

Alternatively, you can have your program open the data set and leave it open. The disposition of the data set may be NEW, OLD, or MOD. This method uses one sequential checkpoint data set, one DCB, and one DD statement.

Figure 2 on page 13 shows a way to alternate data sets when all checkpoints are taken by one CHKPT macro instruction. The data sets are opened by the control program and identified by two DD statements, CHECKDD1 and CHECKDD2. The data control block (DCB) initially refers to CHECKDD1. Before the second checkpoint, the DCB is changed to refer to CHECKDD2; before the third checkpoint, the DCB is again changed to refer to CHECKDD1; and so forth. One data control block can be used for two data sets that are not open at the same time.

An alternative method of using two sequential data sets is to use two DCBs and two DD statements specifying NEW or OLD dispositions, and to execute alternately two CHKPT macro instructions. Each CHKPT macro instruction then refers to a different data set.

The method illustrated in Figure 1 on page 13 saves all checkpoint entries for possible use in deferred restart. The method illustrated in Figure 2 on page 13 conserves auxiliary storage. None of the methods requires a particular device type.

```

Program

*      LA      2,CHECKDCB      Establish CHECKDCB as
                                base address for
                                IHADCB
      USING  IHADCB,2
      XC     DCBDDNAM(8),DDHOLD Exchange ddname in
                                CHECKDCB for ddname
      XC     DDHOLD(8),DCBDDNAM CHECKDCB for ddname
      XC     DCBDDNAM(8),DDHOLD in DDHOLD
      CHKPT  CHECKDCB          Open, checkpoint, close
      .
      .
DDHOLD     DC   C'CHECKDD1'
CHECKDCB   DCB   DSORG=PS,MACRF=(W),DDNAME=CHECKDD2
           DCBD  DSORG=PS      Define IHADCB (dummy
*                                section that defines
*                                DCBDDNAM)

DD Statements

//CHECKDD1 DD UNIT=SYSDA,DISP=NEW ...
//CHECKDD2 DD UNIT=SYSDA,DISP=NEW ...

```

Figure 2. Using Two Sequential Checkpoint Data Sets to Ensure Restart

How Checkpoint Entries Are Identified

The number of checkpoint entries, as well as the number of checkpoint data sets used concurrently, is unlimited. In a sequential checkpoint data set, checkkids of valid or invalid checkpoint entries in one data set should be unique. In a PDS, checkkids of valid entries should be unique.

If you specify checkkids instead of having the system generate them, incorrect duplicates might be specified. The system cannot recognize the error of duplicate checkkids. When deferred restart at a

checkpoint occurs and the checkpoint data set is sequential, the system searches the data set from its beginning for the specified checkpoint entry. The system uses the first entry that it finds which has the specified checkid.

If the data set is partitioned, the system searches the directory to find the location of the specified checkpoint entry. If two or more entries having the same checkid were written in the data set, the most recent of those entries is the one pointed to by the directory, and restart occurs from the most recent entry.

Checkpoint entries have two identifications: primary and secondary.

- The primary identification is the programmer-generated or system-generated checkid specified or requested by the CHKPT macro instruction. The primary identification is used when a search is made for a checkpoint entry.
- The secondary identification is identical to the system-generated checkid that might have been requested by CHKPT. The secondary identification is then used as a base to compute the system-generated checkids of entries written after restart has occurred.

The control program identifies each checkpoint in a message to the operator and user. On request, the control program also makes the identification available to your program. In [Figure 3 on page 14](#), the CHKPT macro instruction requests the control program to supply an identification ('S' parameter) and place it in the 8-byte field named ID. When the checkpoint is successfully taken, the program prints the identification as part of a message to the programmer.

```

      .
      .
      .
      CHKPT  CHKDCB,ID,'S'      Take checkpoint
      LTR    15,15              Was checkpoint taken
      BNZ    PHASE2             No, branch to PHASE2
      PUT    STEPLLOG,MESSAGE    Yes, print
                                checkpoint ID
*
PHASE2
      .
      .
      .
MESSAGE DS    0F
        DC    H'44'             Record length
        DC    H'0'              Flags
        DC    C'SUCCESSFUL CHKPT AT PHASE2...ID='
ID       DS    CL8
STEPLLOG DCB    DSORG=PS,MACRF=(PM),RECFM=V,LRECL=124,DDNAME=LOGDD
CHKDCB   DCB    DSORG=PS,MACRF=(W),RECFM=U,DDNAME=CHKDD

```

Figure 3. Recording a Checkpoint Identification Assigned by the Control Program

Coding the CHKPT Macro Instruction

The CHKPT macro instruction is coded in your program. When the CHKPT macro executes, job step information (about your program, virtual storage data areas, data set position, and supervisor control) is written as a checkpoint entry in a checkpoint data set. The point at which this information is saved becomes a checkpoint from which a restart may be performed. After the checkpoint entry is written, control returns to your program at the instruction following the CHKPT macro.

The CHKPT macro instruction optionally refers to the data control block (DCB) for the checkpoint data set. If the data set is not open, the checkpoint routine opens it and then closes it after writing the checkpoint entry. If the data set is open, the checkpoint routine writes the checkpoint entry, but does not close the data set.

The checkpoint data set must be on one or more magnetic tape volumes or direct access volumes. A checkpoint data set that resides on a magnetic tape must have IBM standard labels. Direct access volumes cannot be shared. For exceptions to this restriction on shared DASD see [“Coding the DD Statement for a Checkpoint Data Set” on page 8.](#))

The standard form of the CHKPT macro instruction is:

[<i>symbol</i>]	CHKPT	{ <i>dcbaddr</i> DDNADDR= <i>ddnaddr</i> CANCEL} [, { <i>checkid addr</i> IDADDR= <i>checkid addr</i> } [, { <i>checkid length</i> IDLNG= <i>checkid length</i> ‘S’ IDLNG=‘S’}]] [, MF={S (S, <i>addr</i>)} {L (L, <i>name</i>)} {(E, <i>addr</i>) (E, <i>addr1</i> , <i>addr2</i>)}]
-------------------	-------	---

The options on the CHKPT macro are discussed in alphabetic order as follows.

CANCEL

Cancels the request for automatic checkpoint/restart. CANCEL can be used only with the standard form (MF= omitted or MF=S or MF=(S,NAME)). An automatic checkpoint/restart can be suppressed by issuing a CHKPT with CANCEL. See [“Canceling a Checkpoint” on page 18](#).

checkid addr

Specifies the address of a programmer-provided field that is to contain a unique, printable identification of the checkpoint entry. This identification is called a checkid. The checkpoint routine writes the checkid as part of the entry and prints it in a message on the operator's console when it finishes writing the entry. The programmer must use the checkid and code it in the JOB statement RESTART parameter to use the corresponding entry to perform a deferred restart at a specific checkpoint. If the *checkid addr* operand is omitted, the *checkid length* or ‘S’ operand is invalid. Valid characters for a checkid are:

A - Z

0 - 9

\$ #

! *) ; - / , % _ > ? : ' = " and blanks are valid except as first characters or for PDSs.

With the execute form, *checkid addr* is an RX type address or register (2-12). With the standard or list form, it is an A type address.

checkid length or ‘S’

checkid length is the length in bytes of the field that contains the checkid. The maximum length of this field is 16 bytes when the checkpoint data set is physical sequential, 8 bytes when it is partitioned. By coding this operand or by omitting it entirely (in which case a length of 8 bytes is implied), you specify that your program will form an identification and store it in the checkid field before CHKPT is executed. If the *checkid addr* operand is omitted, the *checkid length* operand is invalid.

By coding this operand as ‘S’, you specify that the checkpoint routine is to generate an identification 8 bytes in length and store it in the checkid field. If the *checkid addr* operand is omitted, this operand is invalid.

If either *checkid length* or ‘S’ and *checkid addr* and the keyword equivalents are omitted, the checkpoint routine generates an identification. The checkpoint routine then writes this identification in the checkpoint entry and on the operator's console but does not return it to your program.

If you provide the checkpoint identification and the checkpoint data set is sequential, the identification can be any combination of up to 16 alphanumerics, special characters, and blanks. For a PDS, the checkpoint identification must be a valid member name of up to eight alphanumerics. The identification for each checkpoint should be unique. If two identifications differ only by having a different number of trailing blanks, the control program considers them to be the same. For more information see [“How Checkpoint Entries Are Identified” on page 13](#). When a deferred step restart takes place, the checkpoint identification is reset to 0.

One way to use the *checkid addr* operand, or its keyword equivalent, is to allow your program to select fields in the records of an input data set and use them as checkids. Alternatively, your program can use the *checkid addr* and the ‘S’ operands and include a system-generated checkid in the current record of an output data set.

dcbaddr

is the address of the DCB for the checkpoint data set. With the execute form, *dcbaddr* is an RX type address or register (2-12). With the standard or list form, it is an A type address.

DDNADDR=

Points to an 8-character, fixed-length field that has the name of the DD statement for the checkpoint data set. This keyword may be used instead of the first positional operand DCB. This keyword and the positional operand DCB are mutually exclusive.

With the execute form, *ddnaddr* is an RX type address or register (2-12). With the standard or list form, it is an A type address.

IDADDR=

Points to a variable-length field that contains, or receives, the checkid. (This keyword may be used instead of the second positional operand, *checkid addr*.) This keyword is mutually exclusive with and follows the same rules as the second positional operand.

With the execute form, *checkid addr* is an RX type address or register (2-12). With the standard or list form, it is an A type address.

IDLNG=

Specifies the length of the field that contains, or receives, the checkid. This keyword may be used instead of the third positional operand, *checkid length*. This keyword is mutually exclusive with and follows the same rules as the third positional operand.

Keyword MF= Standard Form (default)**MF=S**

Generates an inline parameter list and executable code.

MF=(S,name)

Generates executable code and an inline parameter list with the label name at the beginning of the list. *name* is a symbol.

Keyword MF= List Form**MF=L**

Generates a parameter list. *Symbol* is required with this form.

MF=(L,name)

Generates a parameter list with the label name at the beginning of the list. *name* is a symbol.

Keyword MF= Execute Form

The execute form is useful for programs that must be reentrant.

MF=(E,addr)

Generates executable code referencing to the parameter list whose name is *addr*. *addr* can also be expressed as a register, for example, MF=(E,(R5)), if the register is preloaded to point to a parameter list.

MF=(E,addr1,addr2)

Generates executable code that moves a model parameter list from *addr2* to *addr1* and updates the list in *addr1* with the positional or keyword operands specified. *addr1* and/or *addr2* are RX type addresses or may be expressed in register form, for example, MF=(E,(R5),(R6)) if the register or registers are preloaded to point to the respective parameter list.

For more information about the MF= keyword see [z/OS DFSMS Macro Instructions for Data Sets](#).

CHKPT Return Codes

The CHKPT macro returns a code in register 15 to indicate whether the CHKPT macro instruction executed successfully. If the value in register 15 does not equal 0, register 0 gives a more detailed code. Appendix A, "Checkpoint/Restart Codes," on page 55, contains a list of these codes and their meanings.

Use of CHKPT with Other Macro Instructions

ATTACH

If ATTACH is issued in the program using CHKPT, all subtasks created must terminate before a CHKPT is issued; the job-step task must be the only task of the step.

EXTRACT

The EXTRACT macro obtains information from the task control block (TCB). TCB information may change when the task terminates and the job step restarts. If the information is needed after restart, the EXTRACT macro instruction should be reissued after the checkpoint is taken, as shown in [Figure 4 on page 17](#).

	.		
	.		
	.		
*	EXTRACT	ANSADDR,FIELDS=(ALL)	Obtain TCB information
	.		
	.		
	CHKPT	CHKPTDCB	Establish checkpoint
*	CH	15,=H'4'	Is restart in progress
*	BNE	NRESTART	No, branch to NRESTART
*	EXTRACT	ANSADDR,FIELDS=(ALL)	Yes, obtain new information
*	NRESTART		
	.		
	.		
	.		

Figure 4. Obtaining Updated TCB Information after Restart

FEOV

A user cannot issue an FEOV macro for the checkpoint data set.

IMGLIB

A checkpoint cannot be created while a DCB created by the IMGLIB macro exists in the address space. You can delete the DCB by issuing IMGLIB CLOSE.

PCLINK

If a PCLINK STACK is issued, a PCLINK UNSTACK must also be issued against all existing stacks before a CHKPT can be issued. There must not be any PCLINK STACKs in existence when a CHKPT macro is issued.

SETDEV

If an IBM 3890 Document Processor unit record device is used, the following SETDEV macro instruction must be issued after a successful restart:

```
SETDEV dcbaddr,DEVT=3890,IREC=irecaddr [,PROGRAM=programe]
```

where:

dcbaddr

is the address of the associated DCB.

irecaddr

is the address of an initialization record.

programe

is the name of the stacker control instruction program loaded in SYS1.IMAGELIB. This parameter is optional.

SETPRT

The SETPRT macro in data management selects the universal character set (UCS) buffer for an IBM 3203-5, 3211, 4245, 4248, or 1403 Printer with the Universal Character Set feature and the forms control buffer (FCB) for the 3203-5 or 3211 Printers, 3262 Model 5, or the IBM 3800 Printing Subsystem.

For some printers, such as the 3800, the buffer contents are not saved when a checkpoint is taken. See [z/OS DFSMS Introduction](#) for a more complete list. To ensure that all lines are printed before taking a checkpoint, you should take one of the following actions:

- Close the printer with the CLOSE macro instruction
- Issue the “Clear Printer” command, perhaps with the SETPRT macro
- Reprint the data after restart.

For the IBM 3800 Printing Subsystem, (Model 1, Model 3, or Model 8 in compatibility mode), the SETPRT macro instruction initially sets or dynamically changes the control information for the printer. For more information about the SETPRT macro see *IBM 3800 Printing Subsystem Programmer's Guide* and [z/OS DFSMS Macro Instructions for Data Sets](#).

For the IBM 3800 Printing Subsystem (Model 1, Model 3, or Model 8 in compatibility mode) at checkpoint/restart, the job has the job file control block extension (JFCBE) parameters as modified during the last JFCBE exit. If no exit was requested or if the JFCBE was not flagged as being modified during the exit, the JFCBE reflects the values coded in the restart JCL. For more information about the JFCBE exit see [z/OS DFSMS Using Data Sets](#).

STIMER

A time interval established by the STIMER macro must be completed before a CHKPT is issued.

WTOR

The reply to a WTOR macro instruction must be received before a CHKPT is issued.

Canceling a Checkpoint

Under the following conditions, you may not wish to have a job automatically restarted in case of failure.

- If you have not yet taken a checkpoint and the job has already been restarted.
- If you are updating critical data that must be complete and accurate before an automatic restart can be successful.

In these cases, you can take a checkpoint, but avoid automatic restart, by using the CANCEL option on the CHKPT macro. If CHKPT CANCEL is used after a checkpoint is taken, the job is not automatically restarted after failure. For the format of the CHKPT macro see [“Coding the CHKPT Macro Instruction” on page 14](#).

After being restarted, the job step may again terminate abnormally. If the job terminates abnormally, it may be restarted from the same checkpoint, subject to operator authorization. To avoid restarting the job step twice from the same checkpoint, code the sequence shown in [Figure 5 on page 19](#).

NRESTART

Figure 5. Canceling a Request for Automatic Restart

After the successful completion of a restart from a checkpoint, the system places a return code of X'04' in register 15 and returns control to your program at the instruction following the CHKPT macro instruction. At this time, a request for another automatic restart at the same checkpoint is normally in effect. In [Figure 5 on page 19](#), the instruction that follows the CHKPT macro instruction tests the return code to determine whether control has been returned as the result of a restart. If the return code is X'04', a restart has just occurred, and a second CHKPT macro instruction is executed. This macro instruction has a CANCEL operand, which cancels the existing request for an automatic restart. If the job step again terminates abnormally after a restart from the checkpoint, automatic restart can occur only at the beginning of the step. It will not occur at the checkpoint preceding the canceled checkpoint.

Serially Reusable Resources

When a job step terminates, it loses control of serially reusable resources. If the step is restarted, it must request all the resources needed to continue processing.

Requesting Serially Reusable Resources

Explicit use of a serially reusable resource is requested when the user's program issues the ENQ or RESERVE macro instruction. If the program issues the ENQ or RESERVE macro and takes a checkpoint, it must issue the ENQ or RESERVE macro again whenever restart occurs at the checkpoint.

Figure 6 on page 19 shows a program that requests a serially reusable resource by issuing an ENQ before establishing a checkpoint. After the checkpoint, the program tests for a restart. If a restart has occurred, the program requests the same resource again. The program requests the resource again because the job step terminated, lost control of the resource, and then restarted from the checkpoint.

NRESTRT

Figure 6. Requesting a Resource after Restart

Some serially reusable resources are requested implicitly by issuing data management macro instructions. These resources may be records that you are processing or tracks on a direct access device. To ensure correct processing, you must *not* establish checkpoints during control of these resources:

- If the basic direct access method (BDAM) is used, your program must, before executing the CHKPT macro, execute either the WRITE or the RELEX macro to release a record that is read with exclusive control.
- If BDAM is used to add a record to a data set with variable-length or undefined records, BDAM issues an ENQ macro instruction for the capacity record (R0). Your program must execute the WAIT or CHECK macro to check completion of the write operation before it executes CHKPT. After execution of the WAIT or CHECK macro, the resources are removed from the queue.
- If a VSAM cluster is implicitly specified, the restart program obtains the names of the data set and the index from the catalog and reissues an ENQ macro against each of them. Therefore, no special considerations are required.

Tape Volume DEQ at Demount

Tape volume DEQ at demount is not affected by restarts. If restart is from a checkpoint, all volumes that are dequeued during the processing of the job are enqueued when the restart job is initiated. This includes all volumes referenced in the JFCB at the time of checkpoint. Subsequent restart processing tests each tape data set open at the time of checkpoint to determine if:

- the data set was open for OUTPUT, OUTIN, EXTEND, OUTINX, or INOUT when the checkpoint was taken.
- JFCDQDSP was set to 1 in the JFCB at the time of checkpoint.
- the program restarting is APF authorized.

If these conditions are satisfied, the DEQ at demount is reestablished so that the data set's current volume is dequeued when it is demounted. All volume serials in the JFCB and JFCB extensions prior to the current volume are dequeued during the restart process for that data set.

Because all volumes, including those dequeued through the DEQ at demount, are enqueued at restart, if only for a short time, automatic checkpoint/restart may result in a “waiting for volumes” condition if any of the dequeued volumes are allocated to another job. A deferred checkpoint/restart can be timed to avoid such a delay.

Special Operating System Features

This following sections contain special operating system features.

VIO Data Sets

If a VIO data set is open when the checkpoint is taken, it is supported for automatic restart only.

If restart is deferred, VIO data sets for BSAM or QSAM must be redefined as dummy data sets. If they are not redefined or if an access method other than BSAM or QSAM accesses the VIO data set, the job fails.

Dynamic Allocation and Dynamically Concatenated Data Sets

Checkpoint/restart supports the use of dynamic allocation by problem programs. There are, however, several restrictions with a checkpointed job involving the use of dynamically concatenated data sets. See [*z/OS MVS Programming: Authorized Assembler Services Guide*](#) for information about the specification of dynamically concatenated data sets.

If you code VOL=REF to refer to a nonspecific tape DD request, then after a restart the two DDs may have different volumes from each other. This occurs when the program writes the first data set before the checkpoint, unallocates that data set and does not open the second data set until after the restart.

If a VIO data set is dynamically allocated prior to a checkpoint and unallocated after checkpoint, the step is ineligible for restart until the next checkpoint is taken.

When the checkpointed job requests dynamic concatenation, the following restrictions apply:

- On a deferred or automatic checkpoint/restart, JCL-defined DD statements that were dynamically concatenated (by SVC 99) with the permanently concatenated attribute will not be properly resolved. This will result in one or more missing DD statements. If the restart attempt fails, message IEF007I, with reason code 34, will be issued.
- On an automatic checkpoint/restart of a job that uses dynamically allocated data sets that had the permanently concatenated attribute, the restart may produce incorrect results. For example, the DDs in the concatenated group may not be matched properly, causing concatenation of the wrong DDs. Restart processing may not detect this type of error. This type of job or step should specify RD=NR in its JOB or EXEC statement. If the job or step fails, it can be resubmitted for a deferred checkpoint/restart.
- On either a deferred or automatic checkpoint/restart of a job which uses dynamically concatenated data sets, unit affinity requests for the concatenated DDs may or may not be accepted on the restart. This is true with or without the permanently concatenated attribute.
- You cannot use the nocapture option of dynamic allocation.

Shared Resources

Checkpoint/restart is supported for the local shared resources (LSR) feature, including use with Hiperspace™ buffers, but repositioning is not allowed. Also, checkpoint is not allowed and cannot be taken when the global shared resources feature is used.

For automatic checkpoint/restart, deleting a data set open at the time the checkpoint was taken or deallocating a SYSOUT data set to the job entry subsystem makes the step ineligible for restart until the next checkpoint is taken.

For automatic step restart, any of the following makes the step ineligible for restart:

- KEEP, CATLG, or UNCATLG a NEW JCL-specified data set
- DELETE JCL-specified data set whose initial status upon entry to the step was OLD
- DELETE an OLD dynamically allocated data set that had volumes specified when allocated
- UNCATLG JCL-specified data set whose initial status upon entry to the step was OLD and did not have volumes specified.

Cross-memory support restrictions

A checkpoint is not allowed when storage access across address space boundaries has been established. When using cross-memory support, the restrictions shown in [Table 2 on page 21](#) may apply.

Table 2. Cross-memory support restrictions

Macros	Restriction Code	Instructions	Restriction Code
LXRES	2	EPAR	2
ETCRE	2	ESAR	2
ETCON	2	IAC	2
AXRES	2	IVSK	2
AXERT	2	MVCP	3
AXSET	2	MVCS	3
ATSET	2	MVCK	3
PCLINK	1	PC*	4
		PT*	3 and 4
		SAC	1

Code definitions:

1. You must release this resource before checkpoint and reestablish it after checkpoint or restart.
2. You must reestablish this resource after a restart.
3. You must be sure that the keys and ASIDs are correct after a restart.
4. Any attempt to restart from a checkpoint taken in a program call (PC) routine is unpredictable. If a PC routine issues a PCLINK, and then a CHKPT, the error is detected and a checkpoint request is refused. It is impossible to detect a PC routine when the PC routine does not issue a PCLINK and the PC routine is entered without a space switch. If a PC routine issues a checkpoint, that PC routine must ensure that all the requisite values needed for a PT instruction are current before issuing the PT instruction. These values may change after a restart (for example, the ASID might be different).

Data Spaces

A program can use checkpoint/restart while it has one or more entries for a data space on its access list (DU-AL or PASN-AL). In such a case, the CHKPT macro only processes successfully if the program has specified on the ALESERV macro that entries made to the access list for the data space should be ignored for checkpoint/restart processing (CHKPT=IGNORE). A program that specifies the CHKPT=IGNORE option assumes full responsibility for managing the data space storage after a restart operation.

See [z/OS MVS Programming: Assembler Services Reference ABE-HSP](#) and [z/OS MVS Programming: Assembler Services Guide](#) for more information about the ALESERV macro.

Name/Token Services

You cannot take a checkpoint if the program has a name/token pair that is not a task-level name/token that was created with the IEANT_CHECKPOINTOK value in the persist parameter. See the IEANTASM macro. You can undefine a name and redefine it after taking a checkpoint or you can add the IEANT_CHECKPOINTOK option. Ensure that the name and the token contain information that will be valid after the restart. For example virtual storage addresses in user subpools will be unchanged but the ASID (address space identifier) probably will be different. The name/token service is described in [z/OS MVS Programming: Assembler Services Reference ABE-HSP](#).

Chapter 3. Requesting Restart

This chapter contains the following sections:

- [“Types of Restart” on page 23](#)
- [“Automatic Restart” on page 23](#)
- [“Deferred Restart” on page 28](#)
- [“Coding the RD and RESTART Parameters” on page 35](#)

Types of Restart

The system allows two types of step restart:

- Automatic step restart
- Deferred step restart

The system allows two types of checkpoint/restart:

- Automatic checkpoint/restart
- Deferred checkpoint/restart

Automatic restarts occur in different ways:

- Automatic step restarts at the beginning of a job step that has abended
- Automatic checkpoint/restart restarts at the last checkpoint taken before the job failed

Deferred restart occurs when a job is resubmitted with the RESTART parameter specified on the JOB statement:

- Deferred step restart occurs at the beginning of the job step specified in the RESTART parameter of the JOB statement.
- Deferred checkpoint/restart occurs at the checkpoint specified in the RESTART parameter of the JOB statement.

Automatic Restart

Because an automatic step restart and an automatic checkpoint/restart are similar, they are discussed together in the following sections.

Operator Message Sequence

During processing related to an automatic checkpoint/restart, the system issues the following sequence of messages to the operator:

1. A message is issued each time a checkpoint entry is written. Each message contains the checkpoint identification.
2. If the job step terminates because of an abend condition, an abend message is issued for the job step.
3. If the abend code makes the job step eligible for restart and a job journal is present, an authorization for restart message (IEF225D) is issued that requires a reply.
4. A message is issued, indicating the virtual storage requirements (beginning address and ending address) of the job step to be restarted.
5. Normal mount and restart messages are issued, except when using the Tape Library Dataserver.
6. If password-protected data sets are to be repositioned at restart time, a password message is issued.
7. If additional checkpoint data sets (other than the data set used for restart) are encountered at restart time, checkpoint data set security messages are issued.

8. A successful restart message is issued.

During processing related to an automatic step restart after a job step has terminated abnormally, the system issues the following sequence of messages to the operator.

1. An abend message is issued for the job step.
2. If the abend code makes the job step eligible for restart, and a job journal is present, an authorization message is issued that requires a reply.
3. Normal mount messages are issued.

Note: The abend message, which is issued as:

```
IEF4501 jobname.stepname.procstepname. ABEND code
```

is always displayed if a job step terminates abnormally. In addition, if the job step is being executed and the system fails, this message will be displayed during the next IPL if a system-supported restart is performed. The *code* part of the message has the form Shhh (S followed by a 3-character hexadecimal number) if the system executed the abend macro instruction, or Udddd (U followed by a 4-digit decimal number) if your program executed the abend. It is S2F3 if a system failure occurred. For more information about messages see [z/OS MVS System Messages, Vol 1 \(ABA-AOM\)](#), [z/OS MVS System Messages, Vol 2 \(ARC-ASA\)](#), [z/OS MVS System Messages, Vol 3 \(ASB-BPX\)](#), [z/OS MVS System Messages, Vol 4 \(CBD-DMO\)](#), [z/OS MVS System Messages, Vol 5 \(EDG-GLZ\)](#), [z/OS MVS System Messages, Vol 6 \(GOS-IEA\)](#), [z/OS MVS System Messages, Vol 7 \(IEB-IEE\)](#), [z/OS MVS System Messages, Vol 8 \(IEF-IGD\)](#), [z/OS MVS System Messages, Vol 9 \(IGF-IWM\)](#), and [z/OS MVS System Messages, Vol 10 \(IXC-IZP\)](#).

Operator Considerations

If a step requests automatic restart and is eligible for restart, the system displays the following message to request authorization for the restart:

```
xxIEF225D SHOULD jobname.stepname.procstepname  
[checkid.] RESTART
```

Checkid appears in the message only if restart at a checkpoint is requested. It contains from 1 to 16 characters and identifies the checkpoint entry to be used to perform the restart. The operator must reply to the request for authorization as follows:

```
REPLY xx,{ 'YES' | 'NO' | 'HOLD' }
```

YES authorizes the restart, HOLD postpones it, and NO prohibits it.

If it is not clear that the restart should proceed, the operator should reply HOLD to the authorization message. Later, if restart should be allowed, initiate the restart by using the RELEASE command, which will be the same as an initial YES reply. If the decision is to deny the restart authorization, the operator can cancel the job in the HOLD queue. However, that HOLD, as well as YES, causes special disposition processing to occur during the abnormal termination. This processing keeps all OLD (or MOD) data sets and deletes all NEW data sets if a step restart is requested and keeps all data sets if a checkpoint/restart is requested. If you decide to disallow the restart but want to allow normal disposition (as requested on the job's DD statements) of data sets that were kept, release the job, wait until restart has begun, and then cancel the job.

If a job is canceled in the HOLD queue, any paging space allocated to a VIO data set is not released until the system is reinitialized. To avoid this situation, the operator should release the job, wait until the restart begins, and then cancel the job.

Using the VARY and UNLOAD commands, before the operator replies YES, and after the authorization request, may cause the system's volume and device configuration during a restart execution of the job to be different from what it is during the original execution of the job. The operator may eliminate use of defective volumes and devices.

The ability to use a different volume at restart usually exists only in the case of a NEW data set on a nonspecific volume that is not open at the checkpoint. If a checkpoint/restart is performed, the data set

must not be open at the checkpoint. The ability to use a different device does not apply to the device or devices containing the system residence (SYSRES), SPOOL, or PAGE packs. If a checkpoint/restart is to be performed and a data set is open at the checkpoint, the same type of device must be allocated to the data set during both the original and restart executions.

Note: When an initiator selects a job for automatic step restart and the job is reinterpreted, no message is issued to the operator regarding virtual storage requirements, because its execution is not location dependent.

If ADDRSPC=REAL is specified on the JOB or EXEC statement, the restart may be delayed by the system waiting for the allocation of storage. If another job is using the required storage, you do not receive a message; you receive only a delay. Enter a DISPLAY A command to see if a system task or another job is using the storage required by the job with a nonpageable region. You can stop or cancel the conflicting task or job.

How to Request Automatic Checkpoint/Restart

An automatic checkpoint/restart occurs if:

- The job journal option is specified, or
- The step requests restart (RD=R), and
- A successful checkpoint has been taken, and
- The step is eligible for restart because it was terminated by an ABEND macro that returned an eligible completion code (specified by the SYS1.PARMLIB member SCHEDxx RESTART/NORESTART statement type at system initialization), or because system failure occurred, and
- The operator authorizes the restart. This authority enables the operator to control the number of restarts of the same step or from the same checkpoint.

If a step fails and an automatic checkpoint/restart is requested, restart occurs automatically at the last checkpoint taken.

Execution of the CHKPT macro requests this type of restart and establishes the checkpoint. You must provide an ordinary DD statement for the checkpoint data set. [Figure 7 on page 25](#) illustrates a job requesting automatic restart at a checkpoint. For more information see [z/OS MVS Initialization and Tuning Guide](#).

//MYJOB	JOB	MSGLEVEL=1 ¹ , RD=R ²	
//STEP1	EXEC		
	.		
	.		
//STEP2	EXEC	PGM=MYPROG	MYPROG issues the CHKPT macro
//NAME1	DD	DSN=NAME2	Describes the data set into
//*			which checkpoint entries
//*			are to be written

Figure 7. Requesting Automatic Checkpoint/Restart

Figure 7 on page 25 Notes:

1

MSGLEVEL=1 is optional.

2

RD=R need not be specified if automatic checkpoint/restart is used through job journaling. See [“Job Journal Requirements” on page 2](#).

Automatic Step Restart

An automatic step restart occurs if all of the following are satisfied:

- The step requests restart (RD=R or RD=RNC); and
- The step is eligible for restart because it was terminated by an ABEND macro that returned an eligible completion code (specified by the SYS1.PARMLIB member SCHEDxx RESTART/NORESTART statement type at system initialization), or because system failure occurred; and
- The operator authorizes the restart. This authority enables the operator to control the number of restarts of the same step or from the same checkpoint.

If a step fails and automatic step restart is requested, restart occurs automatically at the beginning of the step that failed.

Automatic step restart may be requested by coding the RD parameter (**RD=R** or **RNC**) on the JOB or EXEC statement in the originally submitted job. Checkpoint processing is suppressed if **RD=RNC**.

Figure 8 on page 26 illustrates the JCL of a job requesting automatic step restart.

```
//MYJOB   JOB   MSGLEVEL=11,RD=R   Requests automatic
//*                               restart at the
//*                               beginning of any step
//*                               that terminates
//*                               abnormally
//STEP1   EXEC
//STEP2   EXEC   PGM=MYPROG,RD=R2   Requests automatic
//*                               restart of STEP2 if it
//*                               terminates abnormally
//STEP3   EXEC
```

Figure 8. Requesting Automatic Step Restart

Figure 8 on page 26 Notes:

1

MSGLEVEL=1 is optional.

2

Note that if RD=R appears on the JOB statement, it is not required on the EXEC statement.

How a Step Restart Is Initiated

A step restart is initiated in the same way as it is during a normal execution. The devices allocated to the restart step can be different (but of the same device type) from the devices allocated originally. If the allocated devices differ, volumes must be moved from one device to another. If automatic volume recognition (AVR) is used and if the devices are available for allocation, devices containing the required volumes are allocated.

Unless the volumes are already mounted, normal mounting messages request the operator to mount the required volumes on the devices after devices are allocated to the restart step. The requested volumes resume processing.

How Data Sets Being Extended Are Handled during Automatic Step Restart

The following sections describe how data sets being extended are handled when automatic step restart is requested.

DISP=MOD

When automatic step restart is requested, the system saves the TTR and track balance of the end of the data set, for each data set with DISP=MOD that is on a direct access volume and used by the step. Saving occurs when each data set is first opened. If restart occurs, the saved TTRs indicate the ends of the data sets when the data sets open again. Thus, if the step writes data in such a data set during the original execution, the step writes over the data during the restart. The action described here does not occur if restart at a checkpoint occurs.

If a data set with **DISP=MOD** on tape is used in the restart step, the data set is not repositioned at the start of the restart execution. Data written into it during the restart execution follows the data written during the original execution. You may reposition the data set so the data written during the restart execution overlays the data written during the original execution.

OPEN EXTEND or OUTINX

If the data set opens with the **OUTINX** or **EXTEND** options, and its DD disposition parameter was

DISP=NEW: there are no restrictions, because the data set reallocates prior to the restart.

DISP=OLD or **DISP=SHR:** the data set repositions after the last record added before the job terminated. It is possible to add records twice to the data set.

DISP=MOD: the following restrictions apply to a multivolume data set:

- If the volume sequence does not point to the last volume, records overlay or an abend occurs.
- If the volume sequence is not supplied, loading starts on the last volume on which the data set exists. For TAPE data sets only, if the end of data occurs on a volume other than the last allocated volume, processing starts on the wrong volume.

JCL Requirements and Restrictions on Automatic Restart

To allow an automatic step restart or an automatic checkpoint/restart, you must observe the following rules when preparing the job used in the original execution:

1. If a step restart is desired, the **RD** parameter must be coded to request the restart.
2. If a checkpoint/restart is desired, a DD statement for the checkpoint data set must be included in the step that executes the **CHKPT** macro instruction.
3. The **EXEC** statements in the job must have unique names. (Upon restart, the system searches for the named step.)
4. If commands are in the original job, the commands are not reexecuted when restart occurs.
5. The DD statement **VOL=REF** parameter is ignored if restart is attempted from a checkpoint taken when the DD statements are opened out of order and the referenced DD statement requested nonspecific tape volumes.
6. When using passed data sets, the same temporary data set name should not be reused after the temporary data set has been deleted.
7. If the checkpoint data set is SMS-managed, the automatic class selection (ACS) routine must not override the unit or volume selection.

Resource Variations Allowed in Automatic Restart

The system's device and volume configuration during a restart execution of a job can differ from the status during the original execution of the job.

The ability to use a different volume at restart exists only in the case of a new data set on a nonspecific volume that is not open at the checkpoint. If a checkpoint/restart is to be performed, the data set must not be open at the checkpoint. The ability to use a different device does not apply to the device or devices containing the system residence (**SYSRES**) volume and the **LINKLIB** data set. Also, if a checkpoint/restart is to be performed, the same device type must be allocated to the data set during both the original and restart executions.

How the System Works at Automatic Restart

This section describes how the system works when an automatic restart occurs.

How Data Set Disposition Is Determined

When a step requests restart and is eligible for restart, disposition processing of the data sets used by the step or by the job does not occur until the operator replies to the request for authorization. If the operator denies restart, disposition processing occurs normally and programmer-specified final or conditional dispositions are performed. If you indicate that a step is to be executed after abnormal termination (COND=EVEN or COND=ONLY) the step is executed. The next step in a job that will be executed based on abnormal termination (COND=ONLY), will not execute if a restart is successfully performed.

- If step restart is to occur, all data sets with OLD or MOD dispositions in the restart step and all data sets passed around the restart step are kept, even if they are declared temporary. Temporary data sets normally cannot be kept.
- All data sets with NEW dispositions in the restart step are deleted.
- If a checkpoint/restart is to occur, all data used by the job (data sets that were not previously disposed of) is kept.

If the operator performs an operator-deferred restart by replying HOLD to the request for authorization, a CANCEL command for the job can be issued instead of a RELEASE command. If CANCEL is issued, no further data set disposition processing or step executions occur. The disposition of these data sets remains as it was when the HOLD was issued.

How the Job Is Reinterpreted

After disposition processing is completed, the job is reenqueued by the job entry subsystem and is eligible for selection. After the job is selected, the system begins restart processing by reinterpreting the job. The system uses an internal representation of the original job to perform this function. The job is not read in again.

When the information is merged and step restart occurs, the following changes occur:

- Except for data sets not opened during the original execution that requested nonspecific tape volumes, data sets specified as NEW in the restart step have their dispositions changed to OLD.
- In the case of data sets for which nonspecific volumes are requested in the restart step, the device type and serial numbers of the volumes assigned to the data sets during the original execution are used.
- In the case of multivolume data sets, the volume in use at the checkpoint, and not the first volumes of the data sets, are mounted (if they have not remained mounted) during the restart. For VSAM, however, volume mounting is based on the present catalog information and may result in the mounting of unneeded volumes; for example, the first volume of a sequential data set may be temporarily mounted although it is no longer required at the checkpoint being restarted and may be demounted almost immediately. This situation may be avoided through specification of the parallel mount subparameter of the DD statement UNIT parameter (UNIT=(,P)). For further information, see the [z/OS MVS JCL User's Guide](#).
- The system-generated name for any temporary data set uses the time stamp obtained when the job was originally interpreted.

In addition, any modification made to the job's environment by the use of the dynamic allocation facilities prior to the last checkpoint appears as it did at the time of the checkpoint.

Deferred Restart

The following sections describe a deferred restart.

Message Sequence

To perform a deferred checkpoint/restart, you resubmit in an input job stream the job to be restarted. Messages that contain checkpoint entry identifications are displayed on the console during the original execution of the job and thus can be used by the programmer preparing the job for resubmission. When the resubmitted job is restarted, messages appear on the console in the following sequence:

1. A message asking if the checkpoint data set that is being used for restart is a secure volume.
 - For Tape Library Dataserver tape library devices, the tape library configuration data base will be read for confirmation of a secure checkpoint volume.
2. If additional checkpoint data sets, other than the data set used for restart, are encountered at restart time, checkpoint data set security messages are issued.
3. A message indicating the virtual storage requirements of the job.
4. Normal mount messages.
5. If password-protected data sets are to be repositioned at restart time, a password message is issued.
6. A successful restart message.

To perform a deferred step restart, the job to be restarted is resubmitted. Normal mount messages are displayed.

Operator Considerations

When a job is resubmitted to perform a deferred checkpoint/restart (the RESTART parameter is coded on the JOB statement with a checkid operand), the processing is essentially the same as that during an automatic checkpoint/restart after the restart reader has reinterpreted the job. A message is issued to the operator, indicating the virtual storage requirements of the job.

The required virtual storage area can be unavailable for the following reasons:

- The REGION size parameter for the step is larger when the job is resubmitted than in the original execution, and the area required is at different addresses than the area available. The parameter REGION=OK causes this type of restart failure.
- A new IPL is performed and, because of different IPL options specified by the operator, the area required is larger than the area available.

If these conditions exist, a message is displayed, indicating that virtual storage for the job step to be restarted is unavailable.

Deferred Step Restart

You cause a deferred step restart of a job by coding the RESTART parameter without a checkid on the JOB statement and resubmitting the job. The parameter specifies a job step or a step of a cataloged procedure. It restarts the job at the beginning of the specified step. Steps preceding the restart step are interpreted, but not initiated.

The CHKPT macro instruction may or may not be coded in your program. [Figure 9 on page 30](#) illustrates a job originally submitted and the same job resubmitted for step restart. Assume that the results of STEP2 were unsatisfactory because of abnormal termination or incorrect data when the job was executed originally.

Original Job			
//MYJOB	JOB	MSGLEVEL=1 ¹	No automatic restart requested
//*			
//STEP1	EXEC		
	.		
	.		
//STEP2	EXEC	PGM=MYPROG	
	.		
	.		
//STEP3	EXEC		
Resubmitted Job			
//MYJOB	JOB	MSGLEVEL=1, ¹	Causes restart of job
//		RESTART=STEP2	at STEP2
//STEP1	EXEC		
	.		
	.		
//STEP2	EXEC	PGM=MYPROG	
	.		
	.		
//STEP3	EXEC		

Figure 9. Requesting a Deferred Step Restart

Figure 9 on page 30 Note:

¹ MSGLEVEL=1 is optional.

JCL Requirements and Restrictions on Deferred Step Restart

To perform a deferred step restart, you must provide the data set environment required by the restart job. This can be accomplished by using the conditional disposition subparameter in the appropriate DD statements during the original execution of the job. Conditional dispositions in the original job must be used to:

- Delete all NEW data sets used by the step to be restarted.
- Catalog all data sets passed from steps preceding the restart step, to the restart step, or to steps following the restart step. Abnormal termination of the restart step, when it is originally run, causes cataloging of the passed data sets. The information is available to the following steps when the job is resubmitted.
- Keep all OLD data sets used by the restart step, other than those passed to the step.

If a data set with DISP=MOD on tape is used in the restart step, the data set is not repositioned at the start of the restart execution, and data written into it during the restart execution follows the data written during the original execution. You may want to reposition the data set after restart so the data written during the restart execution overlays the data written during the original execution. Any data sets that are dynamically deallocated have the disposition specified at the time deallocation occurred. Conditional disposition processing is not done during abend.

The following rules apply to the restart job:

1. Code the RESTART parameter on the JOB statement.
2. If data sets are passed from steps preceding the restart step to the restart step or to steps following the restart step, the DD statements receiving the data sets must entirely define the data sets. They must explicitly specify volume serial number, device type, data set sequence number, and label type, unless this information can be retrieved from the catalog. It is recommended that passed data sets be conditionally cataloged during abnormal termination of the original execution. Label type cannot be retrieved from the catalog.
3. For jobs that use GDGBIAS=JOB, either explicitly specified on the JOB statement or through the job class, generation data sets created and cataloged in steps preceding the restart step must not

be referred to in the restart step or in steps following the restart step by the relative generation numbers used to create them. They must be referred to by their actual relative generation numbers. For example, a data set created as the +1 data set must be referred to as the 0 data set (assuming that the +2 data set was not also created).

For jobs that use GDGBIAS=STEP, relative generation numbers are resolved in each step, so it is normally unnecessary to modify the relative generation numbers used in the restart step or subsequent steps. However, if another job creates or deletes generation data sets in the generation data group before running the restart job, it may be necessary to modify the relative generation numbers in the restart step or subsequent steps to reflect the current state of the generation data group.

4. The EXEC statement PGM and COND parameters and the DD statement VOL=REF parameter must not be used in the restart step or in steps following the restart step if they contain values of the form *stepname* or *stepname.procstepname*, referring to a step preceding the restart step.

Resource Variations Allowed in Deferred Step Restart

A deferred step restart only allows the restarted execution of a job to begin at other than the first step of the job. Job step initiation and allocation of resources are accomplished normally. The following variations are allowed upon restart:

- Variation of device and volume configuration
- Variation in JCL and data in the resubmitted job.

Deferred Checkpoint/Restart

You can cause a deferred checkpoint/restart of a job by the following procedure:

1. Use the option of coding a special form of the RD parameter (RD=NR) in the original job. This specifies that, if the CHKPT macro instruction executes, a checkpoint entry is written but an automatic checkpoint/restart is not requested. Omission of RD=NR will not preclude a deferred checkpoint/restart.
2. Cause execution of the CHKPT macro instruction, which writes checkpoint entry.
3. Resubmit the job whether or not it terminated abnormally. For example, resubmit it because a volume of one of its input data sets was in error and had caused the corresponding part of an output data set to be in error.
4. Code the RESTART parameter (**RESTART=(*stepname*,*checkid*)**) on the JOB statement of the restart job. The parameter specifies both the step to be restarted and the *checkid* that identifies the checkpoint entry to perform the restart.
5. Place a SYSCHK DD statement immediately before the first EXEC statement in the restart job. It specifies the checkpoint data set from which the specified checkpoint entry is read and is additional to any DD statements in the job that define data sets into which checkpoint entries are written. [Figure 10 on page 32](#) illustrates a job when it is originally submitted and when it is resubmitted for a deferred checkpoint/restart. Assume in [Figure 10 on page 32](#) that STEP2, when originally executed, terminates abnormally at some time after CH04 is written. In the resubmitted job, you request that STEP2 be restarted using the checkpoint entry identified as entry CH04.
6. The DD statement VOL=REF parameter is ignored if restart is attempted from a checkpoint taken when the DD statements are opened out of order and the referenced DD statement requested nonspecific tape volumes.

Original Job			
//MYJOB	JOB	RD=NR	Requests that automatic restart not occur (optional)
//*			
//*			
//STEP1	EXEC		
		.	
		.	
//STEP2	EXEC	PGM=MYPROG	MYPROG issues CHKPT macro
//*			
//*			
//NAME1	DD	DSN=NAME2,DISP=(,CATLG,KEEP)...	Describes checkpoint data set
//*			
//*			
		.	
		.	
//STEP3	EXEC		
Resubmitted Job			
//MYJOB	JOB	RESTART=(STEP2,CH04)	Request restart at CH04 in STEP2
//*			
//SYCHK	DD	DSN=NAME2,DISP=OLD	Describes data set that contains CH04
//*			
//STEP1	EXEC		
		.	
		.	
		.	
//STEP2	EXEC	PGM=MYPROG	
//NAME1	DD	DSN=NAME2,DISP=(,CATLG,KEEP)...	Describes data set in which new checkpoint entries will be written
//*			
//*			
//*			
//*			
		.	
		.	
		.	
//STEP3	EXEC		

Figure 10. Requesting a Deferred Checkpoint/Restart

JCL Requirements and Restrictions on Deferred Checkpoint/Restart

To perform a deferred checkpoint/restart, you must provide the data set environment required by the restart job, using conditional dispositions during the original execution.

Conditional dispositions should be used to:

- Keep all data sets used by the restart step.
- Catalog all data sets passed from steps preceding the restart step to steps following the restart step. Although the step that terminates abnormally is not using the passed data sets, its termination causes the cataloging of the data sets if the conditional catalog parameter is used in the preceding steps.

Note:

1. Temporary data sets cannot be kept.
2. If the DD statement for any data set at checkpoint time had DISP=(NEW,DELETE,CATLG), you probably want to uncatalog after completion of the restarted job or job step by using access method services (IDCAMS), ISPF or IEHPROGM.

Dynamically deallocated data sets have the disposition processed as specified at the time of deallocation. Conditional disposition processing is not performed during abend.

The following rules must be adhered to when resubmitting a job for a deferred checkpoint/restart:

1. A RESTART parameter with a *checkid* subparameter must be coded on the JOB statement.
2. A SYCHK DD statement must be placed in the job immediately before the first EXEC statement.
3. The EXEC statements in the job must have unique names. (Upon restart, the system searches for the named restart step.)

4. The JCL statements and data in steps preceding or following the restart step can be different from their original forms. However, all backward references must be resolvable.
5. The restart step must have a DD statement corresponding to each DD statement present in the step in the original, and the names of the statements must be the same as they were originally. However, the restart step can contain, in any position, more DD statements than it contained originally. Except as noted in item 9, following. The total number of volumes specified at restart must equal or exceed the number specified at the checkpoint.
6. If a DD statement in the restart step in the original job defined a data set that was open at the checkpoint to be used, the corresponding statement in the restart job must refer to the same data set, and the data set must be on the same volume, and, in general, have the same extents recorded in its DSCB as it did originally. (See the exceptions in the note that follows.) This means no one can use the DEFRA function of DFSMSdss to consolidate extents. If the data set is multivolume and is processed by the sequential access method (SAM), only the part of the data set on the volume in use at the checkpoint needs to be the same as it was originally.

Note: The extents can differ as follows:

- In the DD statement, you can request that additional space be allocated to the data set when the space currently available is exhausted. If space is allocated after a checkpoint is taken, this space is indicated in the DSCB; after restart from the checkpoint, the space is released, and the DSCB contents are changed to what they were at the checkpoint.
- In the DD statement, you can code RLSE to request that unused space be released when the data set is closed. If the space is released, the DSCB may indicate a reduced extent for the data set when deferred restart at a checkpoint occurs; no space is allocated to replace released space. Space is not released when a data set is open when the step abends, and step termination is followed by automatic restart.

JCL specified data sets that were dynamically unallocated prior to the checkpoint are not allocated at restart.

When there is no need to read or modify a data set after restart, the data set can be replaced by a dummy data set if the original data set was processed by SAM and the job step is not restarted from a checkpoint within the data set's end-of-volume exit routine. A VSAM data set using the ISAM interface to VSAM cannot be replaced by a dummy data set. Any dummy data set present at the time of the checkpoint must be present as a dummy data set at restart. Allocation is done for each DD statement in the job step where the checkpoint is taken, even if the data set is closed at the time of the checkpoint.

7. The data in the restart step need not be the same as it was originally. If data following a DD * statement was present originally and is entirely omitted in the restart job, the delimiter (/*) statement following the data may also be omitted. The delimiter statement following a DD DATA statement must not be omitted.
8. The VOL parameter of a DD statement must refer to at least those volumes it refers to at checkpoint time. More volumes may be added if desired. The following DD statement parameters cannot be changed: DISP, DCB, LABEL, and UNIT.
9. Except for the requirements stated in rules 4 through 8, the JCL statements and data in the restart step can be different from their original forms. In particular, the DUMMY parameter can be used for any data that was not open at the checkpoint.
10. If data sets are passed from steps preceding the restart step to steps following it, the DD statements receiving the data sets must entirely define them. They must explicitly specify volume serial number, device type, data set sequence number, and label type, unless this information can be retrieved from the catalog. It is recommended that passed data sets be conditionally cataloged during abnormal termination of the original execution. Label type cannot be retrieved from the catalog.

11. The EXEC statement PGM and COND parameters and the DD statement VOL=REF parameter must not be used in the restart step or in steps following the restart step if they contain values of the form *stepname*, or *stepname.procstepname*, referring to a step preceding the restart step.
12. The DD statement VOL=REF parameter is ignored if restart is attempted from a checkpoint taken when the DD statements are opened out of sequence and the referenced DD statement requested nonspecific tape volumes.
13. The volume serial number must be coded on a restart DD for a generation data group if the deferred checkpoint/restart volume serial number list is to be different from the original.

Resource Variations Allowed in a Deferred Checkpoint/Restart

The system's device and volume configuration can differ from the status during the original execution of the job. The allowable differences are those described under [“Resource Variations Allowed in Automatic Restart”](#) on page 27.

System Restrictions for Deferred Checkpoint/Restart

The following restrictions apply when submitting a job for deferred checkpoint/restart:

- Jobs specifying nonpageable storage (ADDRSPC=REAL) require an area of storage identical to the storage originally requested at the time the checkpoint is taken and starting at the same address.
- The link pack area modules in use at the time the checkpoint is taken must reside in the same storage locations for the job submitted for deferred checkpoint/restart.
- Any time a change link pack area (CLPA) is required, the locations of modules in the link pack area (LPA) might change without any changes to the LPA modules themselves. This prevents the restart from any checkpoint taken prior to the IPL in which the CLPA is required.

Changes to hardware features and other system reconfigurations may require a CLPA. Review the instructions included with any such change carefully to determine if a CLPA will be required.

- The nucleus must not be changed between checkpoint and restart.

If the required storage is not available, it may be for one of the following reasons:

- The link pack area expands into the required storage
- The system queue area expands into the required storage.

If either of the preceding conditions occurs, contact your system programmer for a respecification of the system parameter to modify the area size and repeat initial program loading using the new value.

How the System Works during a Deferred Checkpoint/Restart

After the system reads and interprets the restart job, it reads the specified checkpoint entry and merges information from it into the scheduler work area entry for the job. As a result, the work queue entry differs from the entry existing during the original execution, as described earlier. See [“How the Job Is Reinterpreted”](#) on page 28.

Next, the system initiates the restart step normally. The system reads the specified checkpoint entry again and functions as in automatic restart. Restart is delayed until the required virtual storage area is available.

When a job restarts correctly, the system operator receives two messages: IHJ006I and IHJ008I. If these messages do not appear for nonpageable storage jobs (ADDRSPC=REAL), enter a DISPLAY A command to see if a system task or another job is using the required storage. The system operator can then stop or cancel the conflicting job.

If there are multiple tape or disk volumes, the system may ask the operator to mount data volumes other than those required at the beginning of the job. In addition, any card input data sets that are used by the failing job step must again be made available to the system.

Coding the RD and RESTART Parameters

The following sections describe how to code the RD and RESTART parameters.

RD Parameter

You code the RD parameter in the JOB or EXEC statement to request automatic step restart if failure occurs, and/or to suppress, partially or totally, the action of the CHKPT macro instruction. If the RD parameter requests an automatic step restart if failure occurs, or if the RD parameter is not coded, the action of CHKPT is normal: CHKPT writes a checkpoint entry and requests a checkpoint/restart to be performed if failure occurs. The RD parameter is ignored for system tasks and generalized start jobs.

The RD parameter, coded on an EXEC statement, applies to the step corresponding to the statement or to all steps of the cataloged procedure referred to by the statement. Coded on a JOB statement, the RD parameter applies to all steps of the corresponding job and overrides an RD parameter coded in any EXEC statement of the job. The syntax of the parameter is:

```
RD[,procstepname]={R|NC|NR|RNC}
```

The possible definitions are:

RD=R

(Restart) Requests an automatic step restart if failure occurs. If the CHKPT macro instruction is executed in the step, the resulting request for an automatic checkpoint/restart overrides the request for an automatic step restart. This parameter is ignored if the job does not contain a job journal. For JES2 and JES3, this parameter forces job journaling.

RD=NC

(No checkpoint) Does not request an automatic step restart. It totally suppresses the action of the CHKPT macro instruction if the macro instruction executes in the step. This allows use of a program containing CHKPT when the checkpoint function is not wanted. It can also suppress the checkpoint at end-of-volume.

RD=NR

(No automatic restart) Does not request an automatic step restart. It suppresses the request for an automatic checkpoint/restart that is made when the CHKPT macro instruction executes in the step. If CHKPT executes, it writes a checkpoint entry normally. A deferred restart can be performed from the checkpoint entry.

RD=RNC

(Restart and no checkpoint) Requests an automatic step restart if failure occurs. It totally suppresses the action of CHKPT if CHKPT executes in the step. It can suppress the checkpoint at end-of-volume. If the job does not contain a job journal, the step is ineligible for automatic restart. For JES2 and JES3, this parameter forces job journaling.

If **RD=***value* is coded on an EXEC statement that invokes a cataloged procedure, the parameter applies to all steps of the procedure and overrides all RD parameters present in the EXEC statements of the procedure. **RD**,*procstepname=value* can be coded instead of **RD=***value*; it applies to the specified procedure step and overrides the RD parameter that may be coded on the EXEC statement of the procedure step. **RD**,*procstepname=value* can be coded once for each step of the procedure.

RESTART Parameter

The RESTART parameter requests a deferred restart of a job. Code it in the JOB statement when you resubmit the job. If step restart is to occur, this parameter specifies which step to begin. If the restart is to occur at a checkpoint that was taken during a step, specify both the step and the identification of the particular checkpoint entry. The syntax of the parameter is:

```
RESTART=({stepname|stepname.procstepname|*},{checkid})
```

Both operands are used if restart at a checkpoint is to occur. If a step restart is to occur, *checkid* must be omitted; the enclosing parentheses can be omitted.

stepname

name of the step to be started.

stepname.procstepname

if a step of a cataloged procedure is to be restarted, *stepname* is the name of the step executing the procedure and *procstepname* is the name of the step within the procedure.

*

the first step of the job (possibly a step within a cataloged procedure) is to be restarted.

checkid

can contain up to 16 characters in any combination of alphanumeric characters, printable special characters, and blanks. If it contains any special characters or blanks, it must be enclosed in single apostrophes, and apostrophes within it must be represented as double apostrophes.

SYSCHK DD Statement

If you wish to perform a deferred checkpoint restart you must include a SYSCHK DD statement in the resubmitted job. This DD statement specifies the checkpoint data set that contains the checkpoint entry to be used in the restart. The SYSCHK DD statement cannot be included when a deferred step restart is to be performed.

The statement must precede the first EXEC statement in the job that performs a deferred restart at checkpoint. It must follow the JOBLIB DD statement if the JOBLIB DD is present. The desired checkpoint entry must be named by the *checkid* subparameter of the JOB statement RESTART parameter.

The following requirements and restrictions apply to the SYSCHK DD statement:

- The statement must contain or imply DISP=(OLD,KEEP).
- For SMS-managed data sets, UNIT and VOL=SER are not required.
- The statement must define the checkpoint data set. It must specify its name. If it is not cataloged, the statement must also specify the device type or device number and specify the volume serial number.
- If the volume containing the checkpoint data set is mounted on a JES3-managed device, the SYSCHK DD statement must not request deferred mounting.
- The SYSCHK data set cannot be multivolume or concatenated. If the checkpoint data set is multivolume, the SYSCHK DD statement must specify, as the first volume of the data set, the volume and data set name that contain the desired checkpoint entry. The serial number of the volume containing a particular entry appears in the console message that is written when the entry is written.
- If the checkpoint data set is partitioned, the DSNAME parameter on the SYSCHK DD statement must not contain a member name.
- If a RESTART parameter without the *checkid* subparameter is included in a job, a SYSCHK DD statement must not appear before the first EXEC statement of the job.
- If a RESTART parameter is not included in a job, a SYSCHK DD statement appearing before the first EXEC statement in the job is ignored.
- A SYSCHK DD statement appearing in a step or procedure step of a job is treated as an ordinary DD statement; that is, the name SYSCHK has no special meaning in that case.

An example of a non-SMS SYSCHK DD statement is:

```
//SYSCHK DD DSN=dsname,DISP=OLD,UNIT=name,
//          VOL=SER=volser
```

For existing SMS-managed data sets, the system ignores the UNIT keyword and any volume serial on the DD statement.

An example of an SMS SYSCHK DD statement is:

```
//SYSCHK DD DSN=dsname,DISP=OLD
```

Chapter 4. Checkpoint/Restart Processing

This chapter describes how the checkpoint/restart routine handles different types of user data sets, and gives additional guidelines on repositioning and preserving the contents of data sets. The end of this chapter includes notes about using checkpoint/restart with various programs and programming languages.

This chapter contains the following sections:

- [“Notes for User Data Sets” on page 39](#)
- [“SMS-Managed Sequential Data Sets” on page 43](#)
- [“VSAM Data Sets” on page 44](#)
- [“Notes for Tape Labels and Files” on page 47](#)
- [“Repositioning and Preserving the Contents of Data Sets” on page 48](#)
- [“Buffering of Data Set Records” on page 49](#)
- [“Preserving Data Set Contents” on page 49](#)
- [“Allocating Devices during Deferred Checkpoint/Restart” on page 50](#)
- [“Handling Checkpoint/Restart Errors” on page 50](#)
- [“Using Checkpoint/Restart with Other Programs” on page 51](#)

Notes for User Data Sets

The data set types discussed are as follows:

- BDAM (basic direct access method)
- Extended format
- Generation
- Graphics
- IEFDDSRV macro
- Large format
- Multivolume
- Partitioned data sets (PDSs)
- Partitioned data set extended (PDSEs)
- SMS-managed sequential data sets
- SYSABEND
- SYSIN
- SYSOUT
- TCAM (telecommunications access method)
- Temporary
- VSAM (virtual storage access method)
- Work

BDAM Data Sets

When the basic direct access method (BDAM) is used to process a data set, processing resumes normally upon restart. However, you must ensure that a particular block is read or written before a checkpoint is taken. Your program must complete the BDAM I/O operation by executing the CHECK or WAIT macro before it executes the CHKPT macro.

Any change made to a record between a checkpoint and a restart will remain in the data set. You are responsible for backing out any changes that would otherwise produce invalid results. For more information, see [“Preserving Data Set Contents”](#) on page 49. If the program does not complete the operation, the block may be read or written either before or after the checkpoint is taken.

For serially reusable resources, your program must also issue either the WRITE or RELEX macro to release a record that is read with exclusive control, before executing the CHKPT macro.

Extended Format Data Sets

A checkpoint data set cannot be an extended format data set.

A checkpoint failure occurs if a checkpoint is requested when an extended format sequential data set is open. You can take a checkpoint while an extended format VSAM data set is open. These extended format VSAM data sets can be striped, compressed format, with extended addressability or any combination thereof.

Generation Data Sets

No automatic cataloging of generation data sets (GDS) takes place. If certain generation data sets of a generation data group are to be cataloged, you must catalog them. The order in which they are cataloged determines the relative generation numbers of the generation data sets for reference by later jobs. The last generation data set becomes the 0 generation, the next-to-last cataloged generation data set becomes the -1 generation, and so on.

You can code the disposition parameter of a generation data set DISP=(NEW,CATLG,KEEP) or DISP=(NEW,CATLG,CATLG). Both will allow a restart without any change in the JCL, regardless of how the checkpointed job ended.

If you code DISP=(NEW,CATLG,KEEP) and the checkpointed job abnormally terminates, the data set is kept. When the restart job completes successfully, this data set is cataloged. If you code DISP=(NEW,CATLG,CATLG) and the checkpointed job abnormally terminates, the data set is cataloged. When the restart job completes, the data set is not cataloged because it already exists on the catalog. However, if the generation data set is to be dynamically allocated, you must code the disposition DISP=(NEW,CATLG,CATLG).

For jobs that use GDGBIAS=JOB (either explicitly specified on the JOB statement or through the job class), generation data sets created by one step of a job can be passed to subsequent steps in the same job and can be referenced by the relative generation numbers assigned at the time of creation, whether or not the generations were cataloged.

For jobs that use GDGBIAS=STEP, the base name (0 generation name) is resolved for a generation data group at the beginning of each step. When cataloging a new generation data set in one step of a job, the association between relative generation numbers and data set names for subsequent steps of the job are affected. For example, when a new +1 generation is cataloged, that data set becomes the 0 generation for the next step of the job. However, when creating a new generation data set without cataloging it, the association between relative generation numbers and data names is unchanged. For example, when creating a new +1 generation with DISP=(NEW,PASS) coded, the catalog is not updated so the step that receives the data set must still refer to that data set as the +1 generation. If the step that receives the data set catalogs it, then subsequent steps will refer to that generation as the 0 generation.

For jobs that use GDGBIAS=JOB, for deferred step restart, the generation data group name table (GDGNT) is recreated from the catalog. The last generation data set cataloged prior to termination of the job becomes the 0 generation and is used for the base name in the GDGNT. This may not be the same as the base name when the job was initially run; you must know which generation data sets were cataloged and in what order the data sets were cataloged, and the JCL must be modified accordingly. For a deferred checkpoint/restart, no modification of the relative generation number in the JCL is necessary. For jobs that use GDGBIAS=STEP, no modification of the relative generation number in the JCL is necessary for deferred step restart or deferred checkpoint/restart.

For jobs that use GDGBIAS=JOB, when a job first references a generation data set, the base name (0 generation name) of each generation data group is placed in the GDGNT. The base name stored in the

GDGNT is never changed. It is saved at a checkpoint and is available for both automatic and deferred checkpoint/restart. The restart takes place without any change in the JCL, whether or not generations previously created by the job were cataloged. For jobs that use GDGBIAS=STEP, when a step of a job first references any generation in a generation data group in that step, the base name (0 generation name) of each generation data group is placed in the GDGNT. Subsequent references to the generations in the generation data group within that step do not cause the base name in the GDGNT to be updated. Each job step will remove any base name entries from the GDGNT, and any subsequent steps of the job that reference the generation data group will determine the base name of the generation data group and place it in the GDGNT. The GDGNT is saved at a checkpoint and is available for both automatic and deferred checkpoint/restart. The restart takes place without any change in the JCL, whether or not generations previously created by the job were cataloged.

When using generation data groups, the disposition of the data set should be specified correctly to avoid having to change the JCL at restart time. If DISP=(NEW,CATLG,CATLG) was specified, then the generation data group (GDG) number is set to (0) at the time of restart and the checkpointed data set is now at GDG(-1), regardless of how the job completed. In order for generation data sets to maintain their respective generation numbers at checkpoint and restart time, the disposition should specify DISP=(NEW,CATLG,KEEP). This will prevent the generation level from changing if the job step fails. If, however, you wish to restart a job that completed without failing, you must change the JCL of the job to be resubmitted to get the correct generation levels.

Note: A job that contains generation data sets referred to by a relative generation number of +1 or greater and with a disposition of OLD, SHR, or MOD is failed by the JES3 interpreter service unless the UNIT parameter is included in the DD statement. With UNIT specified, the JES3 interpreter service permits this data set to be allocated on a deferred basis.

Graphics Data Sets

An open graphics access method (GAM) data control block (DCB) is not supported in Checkpoint/Restart.

IEFDDSRV macro

If your program uses IEFDDSRV MODIFY,TYPE=FEATURE,DSENQMGMT=MEMORY and it is enabled at the system level, the job is not restartable through the checkpoint/restart function. To enable this IEFDDSRV function at the system level, the system programmer sets the SYSTEM MEMDSENQMGMT keyword to ENABLE in the ALLOCxx parmlib member to allow the job or subsystem to use memory-based data set ENQ management.

If your program uses IEFDDSRV MODIFY,TYPE=FEATURE,SWBSTORAGE=ATB and it is enabled at the system level, the job is not restartable through the checkpoint/restart function. To enable this IEFDDSRV function at the system level, the system programmer sets the SYSTEM SWBSTORAGE keyword to ATB in the ALLOCxx parmlib member to allow the job or subsystem to place SWBs for dynamically allocated data sets in 64-bit storage.

Large Format Data Sets

Checkpoint/restart will work while a large format data set is open, and you can write a checkpoint to a large format data set.

Multivolume Data Sets

When multivolume output data sets have nonspecific volume allocation such as UNIT=SYSDA, the abnormal disposition must be coded as KEEP to ensure that if the job is restarted, all volume serial numbers are put into the catalog. For example, use DISP=(NEW,CATLG,KEEP), not DISP=(NEW,CATLG,CATLG).

Partitioned Data Sets (PDSs)

If a PDS is compressed after a checkpoint in which the PDS was open, restart at checkpoint should not be requested.

After you take a checkpoint and then open a PDS for adding or replacing members, another checkpoint should be taken before any records are written into the data set. If the second checkpoint is not taken and restart occurs at the first checkpoint, the OPEN routine positions to the current end of the data set instead of to the original end.

If a PDS is open for output (OPEN,OUTPUT) at the time of a checkpoint, then the restart function deletes any members added after the checkpoint. If the data set is open with the UPDAT, INOUT, or OUTIN option, then restart does not delete members unless another DCB is open for output to the data set. For example, the PDS is open for update, and the program is updating the third member of a five-member PDS, when checkpoint 1 is taken. If a member is updated as a newly rewritten member after checkpoint 1, and the program is restarted from that point, rewriting the updated member will cause problems because the member already exists.

Adding Members

When a PDS is updated, be careful to preserve the contents of the directory. The directory consists of entries that point to each member of the data set.

When a member is added to a PDS, an entry is also added to the directory. If one member is added, the STOW macro can be used to create the entry, or the member name can be specified in the DD statement. In the latter case, the control program creates the directory entry when the data set is closed or when the job step terminates. Programs create members by issuing a STOW macro after writing each member. You might choose to take a checkpoint after each STOW macro or after several STOW macros.

When one or more members are added to a PDS, a checkpoint should be taken immediately after opening the data set. After taking the checkpoint, the new member can be written and its entry added to the directory. If the step is restarted from a checkpoint, the restart function repositions the data set. This means restart deletes the member so your program can recreate it.

Updating Members

To update a member of a PDS, updated records can either be written back to their original locations, or the entire member (in updated form) may be rewritten as a new member of the data set. In the latter case, the directory entry must be updated to point to the rewritten member.

If a checkpoint is taken before rewriting an entire member, one must also be taken immediately after updating the directory, because the control program deletes the updated directory entry if it repositions the data set for restart from the original checkpoint. Because no entry then points to the original member, the postrestart processing will be unsuccessful.

Deleting Members

The checkpoint restart function deletes members that were added after the checkpoint. Because this action may delete members written by another job (another job may have been executed between the original and restart executions of the subject job), restart at a checkpoint should not be requested.

If a PDS is open for output at checkpoint time, any new members added later will be deleted during a restart from this checkpoint. **No members will be deleted when a PDS is open for update.** Be cautious when restarting from checkpoints having PDSs open for update.

Caution: When a program copies a PDS, the members in the copy generally reside in different locations within the data set. If you take a checkpoint while a PDS is open and then copy members, perhaps with IEBCOPY or DFSMSDss™, you cannot do a deferred checkpoint restart.

Partitioned Data Sets Extended (PDSEs)

A failure occurs if a checkpoint is requested when a DCB is open to a PDSE, or if a PDS was opened at checkpoint time but was changed to a PDSE by restart time. A checkpointed data set cannot be a PDSE. For more information on PDSEs see [z/OS DFSMS Using the New Functions](#) and [z/OS DFSMS Using Data Sets](#).

Note: A PDSE can be used as JOBLIB/STEPLIB.

SMS-Managed Sequential Data Sets

When an application program takes a checkpoint, the system records information about the status of that program in a checkpoint data set. This information includes the location on disk or tape where the application is currently reading or writing each open data set. If a data set that is open at the time of the checkpoint is moved to another location before the restart, you cannot restart the application from the checkpoint because the location-dependent information recorded by checkpoint/restart is no longer valid.

There are several system functions (for example, DFSMSHsm™ or DFSMSdss™) that might automatically move a data set without the owner specifically requesting it. To ensure that all checkpointed data sets remain available for restart, the checkpoint function sets the unmovable attribute for each SMS-managed sequential data set that is open during the checkpoint. An exception is the data set containing the actual recorded checkpoint information (the checkpoint data set), which does not require the unmovable attribute.

You can move checkpointed data sets when you no longer need them to perform a restart. DFSMSHsm™ and DFSMSdss™ FORCECP(days) let you use operations such as MIGRATE, COPY or DEFRAG to move an SMS-managed sequential data set based on a number of days since the last access. DFSMSHsm™ recall, and DFSMSdss™ restore and copy, are operations that turn off the unmovable attribute for the target data set.

Card Readers

When BSAM, QSAM, or EXCP is used to read a data set from a card reader, your program can reposition the data set upon restart. If you provide a repositioning routine, instruct the operator to position the data set to the beginning if a restart becomes necessary. The program might be designed to operate as follows:

- The program saves the first record read from the data set and keeps a count of the number of records read before each checkpoint.
- After a restart, the repositioning routine reads a record from the data set and compares it with the first record read before abnormal termination. If the records are identical, the data set has been positioned to the beginning. The routine then repositions it by reading (without otherwise processing) the number of records read before the checkpoint.

SYSABEND or SYSUDUMP Data Sets

Whether or not checkpoint/restart is used, abnormal termination causes the system to write a SYSABEND or SYSUDUMP data set if you provide a SYSABEND or SYSUDUMP DD statement. The system uses its own data control block to write the data set, and it opens the data set during abnormal termination processing. You can code or omit the SYSOUT parameter on the SYSABEND or SYSUDUMP DD statement.

When the SYSOUT parameter is coded and automatic restart occurs after abnormal termination, the SYSABEND or SYSUDUMP data set is not printed for step restart. Because the SYSABEND or SYSUDUMP data set is created by the job step, it is deleted during restart.

In all other cases, the SYSABEND or SYSUDUMP data set is printed, whether or not the restart is successful. If a second abnormal termination occurs, a second SYSABEND or SYSUDUMP data set is written. The same rules that apply to the first data sets also apply to the second data set and to all subsequent data sets.

SYSIN Data Sets

When restart at a checkpoint occurs, a SYSIN data set (data following a DD * or DD DATA statement) is repositioned. Unit record data sets are never repositioned. The checkpoint routine waits until all requested input/output operations are complete, then requests that the job entry subsystem save positioning information. When automatic restart occurs, the system keeps the SYSIN data of the job restarting. During the restart execution, the job can read data from the SYSIN data sets as it could during the original execution.

To perform deferred restart, you include any necessary SYSIN data in the resubmitted job.

Information about altering SYSIN data in a restart job is given under [“JCL Requirements and Restrictions on Automatic Restart”](#) on page 27.

SYSOUT Data Sets

The following describes how SYSOUT data sets (data sets having the SYSOUT parameter coded on their DD statements) are handled during various types of restart.

Automatic Restart

When your program writes SYSOUT data, the system stores it in spool data sets. During an automatic step restart, JES3 releases the output spool space that the original execution used. After automatic step restart JES allocates new spool spaces.

If a checkpoint/restart occurs, the data sets used during the original execution are kept. If a SYSOUT data set was open when the last checkpoint was taken, it is repositioned to its position at the time the checkpoint was taken.

The checkpoint routine waits until all requested input/output operations are complete, then requests that the job entry subsystem save positioning information. Data written during the restart execution overlays only the data written between the time the last checkpoint was taken and the time the job step terminated abnormally. If a SYSOUT data set is closed at the checkpoint, the data set is not repositioned. If the restart step opens the same data set again, the data written during the restart follows the data originally written. The data set has an implied disposition of MOD.

Deferred Checkpoint/Restart

When a checkpoint/restart occurs, and a SYSOUT data set is open at the checkpoint, the data set written into during the restart is a new data set different from the data set used originally.

Temporary Data Sets

A temporary data set is a data set where you do not supply a name or one where you code a name that starts with an ampersand. You might choose to allocate space for a temporary data set in a step before a program writes in it. You might do this to simplify the JCL. The system suppresses checkpoints and automatic restarts for any job step that uses a temporary data set that is allocated in a prior job step.

VSAM Data Sets

For VSAM data sets, you are responsible for handling checkpoint/restart problems that arise because of changes in the data. For example, consider a program that updates records in a data set by adding a number to a value already existing in some field within the record. If the program terminates and is restarted, you must ensure that the records processed between the checkpoint and the termination are not processed again after the restart.

During checkpoint/restart processing, no user ACB exits are taken. For checkpoint processing, the ACB exception exit is taken; for restart processing, the ACB exception exit is not taken.

VSAM functions you can use with checkpoint and restart include:

- VSAM Hiperbatch™ environment
- VSAM LSR environment using Hiperspace™ buffering.
- Extended format data sets, including compressed format data sets and those with extended addressability

A checkpoint cannot be taken if VSAM data sets are open in multiple address spaces and if your program uses the CBIC (control blocks in common) option.

When the checkpoint routine records positioning for a VSAM data set, all outstanding I/O requests for the data and index are completed before the contents of your address space are saved. If an error occurs

while these I/O requests are processed, the checkpoint procedure stops and a code of X'0C' is returned in register 15. You can handle the error condition and reissue the CHKPT macro.

You cannot take a checkpoint if a data set with the RLS (record level sharing) option is open.

Catalog Updates

The checkpoint program issues a VSAM CLOSE macro with TYPE=T to update the catalog. It records information about VSAM data sets in a checkpoint data set. If a failure occurs, the latest checkpoint is used to reconstruct the situation that prevailed when the checkpoint was taken.

When using VSAM data sets, a checkpoint is required after any change in the catalog information. A restart will fail if a difference in the checkpoint information and current catalog information is detected.

Note: To determine if the catalog information has changed, examine the contents of the RPL feedback word after each VSAM request. The SHOWCB macro is one way to do this (see [z/OS DFSMS Macro Instructions for Data Sets](#) for more information). An RPL feedback word of X'00000004', X'0000002B', or X'0000002C' (see [z/OS DFSMSdfp Diagnosis](#) for more information) means that EOVS has been called and the catalog information may have been updated. Immediately issue a checkpoint to keep the checkpoint data set current.

Use of Hiperbatch

Hiperbatch is a processing option which provides better performance for jobs accessing VSAM data sets by creating a copy of the data set in expanded storage. Checkpoint/restart processing is supported for VSAM data sets using hiperbatch. No external changes are required when coding the CHKPT macro or when specifying restart processing for VSAM data sets using hiperbatch.

The AMP Parameter

The AMP parameter has a subparameter for specifying checkpoint/restart options that handle two special situations in restarting a processing program:

- Modifications to the data set other than records added sequentially to the end of an entry-sequenced data set. The restart program cannot restore a data set to its checkpoint status if there have been internal modifications to it since the checkpoint, so the restart program does not attempt restart processing.
- Addition of records to the end of a data set by way of a job step other than the job step that issued the checkpoint. Any records added to the end of an entry-sequenced data set are erased in restoring the data set to its checkpoint status.

The AMP options for checkpoint/restart are:

- Let restart take its normal action for both situations
- Override either one or the other of the two actions
- Override both actions.

If you override the check for internal modification, the processing program is restarted, even though the data set being processed cannot be restored. If you override erasing data at the end of a data set and the catalog has been updated, the processing program is not restarted unless you also override the check for modification. For more detail on multiple ACBs open against the same data set, see [z/OS DFSMS Macro Instructions for Data Sets](#).

To prevent data from being erased or to allow restart with modified data sets, the CROPS subparameter of the AMP parameter must be coded in the DD statement for the cluster or data set.

When more than one ACB is opened for output for the same data set using DDN sharing and the same DD statement, or using DSN sharing, the first ACB to be opened will establish the CROPS parameter for the data set. If this is not done, the results are unpredictable.

Repositioning is mandatory for all VSAM data sets open for create mode processing, except for relative record data sets processed in direct mode. If AMP='CROPS=NRE' or AMP='CROPS=NRC', no checkpoint is

taken. If a checkpoint is attempted, message IHJ000I is issued, and a return code of X'08' is returned in register 15. A reason code of 41 in register 0 is also returned to you.

If a VSAM data set supported for repositioning extends to a new volume after the checkpoint, VSAM restart cannot reposition the data set. A restart from that checkpoint is not successful unless the no-reposition option is taken by specifying AMP='CROPS=NRE' or AMP='CROPS=NRC'.

VSAM Data Set Types

VSAM has key-sequenced, entry-sequenced, and relative record data sets. The main difference among the three is the order in which the data records are loaded into the data sets.

Key-Sequenced Data Sets

Records are loaded into a key-sequenced data set (KSDS) in key sequence, the order defined by the collating sequence of the contents of the key field in each of the records. Each record has a unique value in the key field, such as employee number or invoice number. VSAM uses an index and optional free space to insert a new record into the data set in key sequence.

A key-sequenced data set is prepared for restart by the restoration of any statistical information (such as number of records inserted) to its checkpoint status.

For a VSAM KSDS, restart does not erase any data except in create mode. It does, however, detect modification of the data set by either the checkpointed program or another program that used the data set between the checkpoint and the restart. If the data set is modified, the restart is terminated, unless you override the testing of the data set by using the AMP='CROPS=NCK' subparameter in the DD statement for the data set.

Entry-Sequenced Data Sets

Records are loaded into an *entry-sequenced data set* regardless of the contents of the records. The sequence is determined by the order in which the records are received to be placed in the data set. This is what determines the physical arrangement. New records are stored at the end of the data set.

A checkpoint cannot be taken if a VSAM entry-sequenced data set is open for output with an immediate-upgrade path (or alternate index) open over it, unless the no-reposition option, AMP='CROPS=NRE' or AMP='CROPS=NRC', is specified. VSAM immediate-upgrade data sets are key-sequenced data sets, and repositioning is not supported for them.

For a VSAM entry-sequenced output data set, all data added after the last checkpoint was taken is physically erased by restart unless the AMP='CROPS=NRE' or NRC subparameter is specified in the DD statement for the data set. If data is erased by restart, the catalog record for the data set is updated to reflect the current end of data, and the data set statistics are adjusted to reflect the new status.

During restart, entry-sequenced output data sets are restored by the elimination of all records added at the end since the checkpoint.

Relative Record Data Sets

Records are loaded into a *relative record data set* in relative record number sequence. The data set is a string of fixed-length slots, identified by a relative record number. When a record is inserted, you can assign the relative record number or allow VSAM to assign the record the next available number in sequence. No index is used.

A checkpoint cannot be taken for a VSAM relative record data set (RRDS) in load mode if direct processing was or is performed.

If a checkpoint is taken with a relative record data set open for load mode nondirect processing and direct processing is performed after the checkpoint (thus affecting data that was loaded before the checkpoint was taken), no attempt is made by restart to reset the data that existed at checkpoint time. After restart, the data set resets to checkpoint status but still contains the results of any direct processing performed on that part of the data set that existed at checkpoint time.

Work Data Sets

Many programs use "work" data sets, which are alternately written and read, rewritten and reread. If a work data set is used, a checkpoint should be taken each time you have finished reading the data set and before rewriting it.

For example, a program may perform the following sequence of operations to produce different versions of data set A:

1. Write and then read back A1.
2. Write and then read back A2.
3. Write and then read back A3.

A checkpoint should be taken at the beginning of operations 2 and 3 before any rewriting of data set A takes place. If, for example, the job step is abnormally terminated while operation 2 is in progress, the job step can be restarted from the checkpoint taken at the beginning of operation 2. At this checkpoint, there is no need for the data in version A1.

Notes for Tape Labels and Files

This section provides information on tape labels and files in relation to checkpoints.

VSE Tape Files

Checkpoints can be taken with VSE tape files opened with the bypass leading tapemark option LABEL=(LTM) and/or the bypass embedded VSE checkpoint records option DCB=(OPTCD=H) specified. However, a checkpoint must not be taken when an opened data set resides on a VSE 7-track tape, is written in translate mode, and contains embedded checkpoint records.

ISO/ANSI Tape Labels

A tape data set with ISO/ANSI labels or with OPTCD=Q cannot be used as a checkpoint data set.

Nonstandard Labels

You cannot restart from a checkpoint that was taken while a tape with nonstandard tape labels is open.

Bypass Label Processing (BLP)

If you are using a multivolume tape data set that is described by concatenated DD statements with the bypass label processing (BLP) option, the deferred option should be coded in the UNIT parameter of those DD statements. This prevents a misleading scheduler premount message that asks for the wrong volume to be mounted at restart time.

Automated Tape Library

During tape installation, checkpoint procedures must be modified when checkpoint tape volumes reside in a tape library automated by a Tape Library Dataserver. A checkpoint indicator will be recorded in the volume's tape configuration data base record, in addition to its file label. The checkpoint volume's tape configuration data base record indicator will be added and/or deleted whenever the volume changes status. Instead of requesting checkpoint volume confirmation from the operator, the volume's tape configuration data base record will be read. For more information see [*z/OS DFSMS Access Method Services Commands*](#).

When a cartridge is ejected, checkpoint status will be displayed to aid the tape library personnel in executing installation checkpoint volume procedures.

File Protection

The tape operator can file-protect a tape cartridge by manually setting a switch on the cartridge. If it is not in a tape library, a console message might direct the tape operator to set the switch.

The Tape Library Dataserver gripper is unable to manipulate the file protect switch on the cartridge. Automation makes it impossible for a tape operator to change the switch setting of the cartridge. Thus, the read/write capability of the tape cartridge is determined by the switch setting of the tape cartridge at the time it is placed in the library.

When needed, private cartridges can be used within a Tape Library Dataserver and permanently file-protected. This means that they are read-only and any attempt to write will result in a permanent error.

Cartridges can be used within a Tape Library Dataserver and not file-protected by the cartridge switch, but when accessed for input, the logical file protect feature is used. When the OPEN function enables it with a Perform Subsystem Function (PSF) order, this feature causes rejection of all attempts to write to the associated tape drive, for as long as the current volume is mounted.

The following messages will not be issued for tape devices. Instead, the following actions will be taken:

Number	Condition	Action
IEC068A	A tape volume is opened for INPUT, is not file protected and the user is not authorized to write on the volume.	An attempt will be made to logically protect the volume. If that fails, the job will be terminated.
IEC400A	A tape is to be mounted.	This message will be written to the job messages instead of being routed to the consoles.
IEC401A	A tape volume is file protected and opened for OUTPUT or OUTIN.	An attempt will be made to logically unprotect the volume. If that attempt fails or the tape is physically protected, the job will be terminated.
IEC402D	A tape volume is file protected and opened for INOUT.	An attempt will be made to logically unprotect the volume. If that attempt fails, or the tape is physically protected, the job will be terminated.

Note: If you are not using 3495 tape devices see *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, *z/OS MVS System Messages, Vol 2 (ARC-ASA)*, *z/OS MVS System Messages, Vol 3 (ASB-BPX)*, *z/OS MVS System Messages, Vol 4 (CBD-DMO)*, *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*, *z/OS MVS System Messages, Vol 6 (GOS-IEA)*, *z/OS MVS System Messages, Vol 7 (IEB-IEE)*, *z/OS MVS System Messages, Vol 8 (IEF-IGD)*, *z/OS MVS System Messages, Vol 9 (IGF-IWM)*, and *z/OS MVS System Messages, Vol 10 (IXC-IZP)* for actions to take.

Tape Label Verification

If a tape label conflict is encountered while verifying the tape volumes in the Tape Library Dataserver, checkpoint/restart will demount the volume and fail the restart.

For more information about tape labels see [*z/OS DFSMS Using Magnetic Tapes*](#).

Repositioning and Preserving the Contents of Data Sets

The control program repositions data sets, but does not preserve their contents. After taking a checkpoint, you must ensure that data set contents are not changed in such a manner as to make successful postrestart processing impossible.

Recording Data Set Positioning

The checkpoint routine records positioning information for user data sets as follows:

- Data sets are repositioned at restart only if they were open when the checkpoint was taken. The OPEN routine positions all data sets opened after the checkpoint was taken.
- Unit record data sets (printer, punch, or card reader) are not repositioned at restart.
- If you use EXCP to process a reel tape data set open at a checkpoint, ensure that the block count in the data set's data control block is correct. You can do that by coding REPOS=Y in the DCB macro. If the block count is incorrect, the data set may be incorrectly positioned by restart.

If input/output operations were requested, but not begun (for example, if a READ macro instruction was executed, but the related channel program was not started), the checkpoint routine then stops any processing associated with the I/O request, records the positioning information, and reestablishes I/O operations.

If I/O operations have already begun, the checkpoint routine waits until they are complete before recording positioning information.

Buffering of Data Set Records

When QSAM is used to process a data set, an indeterminate number of virtual-storage buffers may contain data when a checkpoint is taken. If restart at a checkpoint occurs, the system's action depends on whether a card reader or another type of device is used to process the data set:

- **Card reader used (QSAM only).** Upon restart, existing buffer contents are released. Restarting from a checkpoint causes buffers to be reprimed by reading records from the current data set into them.
- **Another device used.** Upon restart, the buffer contents are restored to virtual storage, and processing continues normally. It is not possible to predict the time (either before or after the checkpoint) at which a given record will be transferred between a buffer and the recording medium. If your program failed due to a bad record read just before or after taking a checkpoint, then the same failure probably will occur from that checkpoint. This might happen even if you repair the data set after the checkpoint.

If you close a QSAM output data set immediately after restarting the program at a checkpoint, be aware that the data set may not be restored to the same condition it was in when the checkpoint was originally taken.

Preserving Data Set Contents

The system does not save and restore the contents of data sets. You must ensure that input data sets and system data sets contain all necessary data when restart occurs. If a data set on a direct access volume is open at the checkpoint, the data set's label (the DSCB in the VTOC) must have the same location and reflect the same extents upon restart as it did when the checkpoint was taken.

If your program reads records from a data set, updates them, and writes them back to their original locations, it may be useless to take a checkpoint before completing this processing. If a checkpoint is taken earlier, postrestart processing is unsuccessful under these circumstances:

- Your program updates a record before abnormal termination and repeats the update after restart, and
- The updated record contents depend on the original contents.

For example, suppose that the purpose of the update is to exchange the positions of two fields in each record. If the record is updated twice, the fields are returned to their original positions, and the results are invalid. In a different application, an update might place a value in a record field, regardless of the field's original contents. You could then restart the step at a checkpoint taken before or during the update procedure, because an updated record would not be changed if updated again after restart.

When data set records are processed in an update-in-place manner (records are read, changed, and written back into their original location in the data set), bad data can be prevented if records updated after the last checkpoint are restored to their original state or if your program keeps track of the records that are updated and avoids updating them again during restart.

Allocating Devices during Deferred Checkpoint/Restart

When you do a deferred restart from a checkpoint, the type of device allocated for the data set depends on the specification in the UNIT parameter of the DD statement. In addition to assuring the same device type for a checkpoint and restart, the system also attempts to allocate a device with the same optional features present at the time the checkpoint was taken.

- If a device number was specified (for example, UNIT=190), then a device of that type is allocated. It may or may not be the device requested.
- If a device name was specified, then a device of that type is allocated.
- If a user-defined name for a single type of device was specified (for example, UNIT=DISK1), then a device of the defined type is allocated.
- If a name for a mixed group of devices was specified (for example, UNIT=SYSDA), then a device of the same type as that used when the checkpoint was taken is allocated.

However, if the mixed group includes devices with varying optional characteristics (for example with and without RPS or DASD shared and not shared between processors), a device with the same optional characteristics is not guaranteed. To insure that the device allocated by checkpoint/restart has the same optional characteristics as the device used when the checkpoint was taken, define some generics at system definition time that include only a single group with the same optional characteristics.

In jobs that can restart, you should avoid using generic unitnames or esoteric unitnames (for example, UNIT=TAPE) that contain more than one device type. Allocation failure may result during restart if too few units of a specific device type are available. For example, if UNIT=TAPE includes more than one type of an IBM 3400 Tape Drive (that is 3400-3, 3400-4, and 3400-6), allocation failure may occur if the device type available at restart time is not the same device type as allocated in the original run of the job. To avoid this allocation failure, define some esoteric unitnames at system initialization that include only a single device type.

Handling Checkpoint/Restart Errors

This sections provides information on checkpoint/restart errors.

I/O errors

The checkpoint routine issues return code 0C if it encounters a permanent I/O error when:

- Completing an outstanding VSAM I/O request
- Quiescing queued access method I/O operations
 - An exception occurs when QSAM is used and the skip or accept option is specified in the EROPT operand of the data set's data control block. In this case, code 00 is returned.
- Writing the checkpoint data set.

When an access method other than QSAM is used, your program can ensure that input/output operations are complete before it executes the CHKPT macro instruction; it can thereby avoid having read or written an erroneous record while quiescing.

If a permanent error occurs when the system reads a checkpoint data set to perform a restart, the restart step is terminated abnormally with the system completion code 13F. Further restart of the step is not attempted.

Internal Errors

On return to the caller, register 15 will contain a return code, and register 0 will contain a reason code describing the error. For more information see [Appendix A, “Checkpoint/Restart Codes,”](#) on page 55.

Depending on the type of error, you may want to consult a SYS1.LOGREC or a SYS1.DUMPxx data set.

User Errors

Checkpoint will make a number of checks to discover user errors. If errors are discovered *before* the first record of the checkpoint entry is written, the checkpoint request is refused; if errors are discovered *after* the first record of the checkpoint entry is written, the checkpoint entry is invalid and checkpoint gives a non-zero return and reason codes.

Using Checkpoint/Restart with Other Programs

This section provides information about using checkpoint/restart with other programs.

Job and Job Step Accounting

The system accumulates processor time used for each job step and job. To access these time values, an installation can provide an accounting routine to receive control at step initiation, step termination, and job termination. Accounting routines are discussed in [z/OS MVS Programming: Authorized Assembler Services Guide](#). The relationships between checkpoint/restart and the step time and job time values available to the accounting routine are as follows:

- At termination (either normal or abnormal) of an original execution, the step and job times accumulated are available to the accounting routine.
- At initiation of the restart step during an automatic restart, the step and job times accumulated for the original execution are again available to the accounting routine.
- At initiation of the restart step during a deferred restart, the step and job times are zero.
- At termination of a restart step and at all subsequent times when the accounting routine is given control during the restart execution, the step and job times reflect only the time used during the restart execution.
- Checkpoint/restart restores the TCBUSER field to its value at the time of the checkpoint.

For example, in an original execution, Step A uses 2 minutes of processor time, and Step B uses 3 minutes of processor time and abnormally terminates. At step termination, the step time is 3 minutes and the job time is 5.

- If automatic restart is performed for Step B, a step time of 3 minutes and a job time of 5 are again available to the accounting routine at the reinitiation of Step B.
- If Step B then uses 4 minutes of processor time and terminates, a step time of 4 and a job time of 4 are available to the accounting routine at step termination.

The two values available at the time the restart step is initiated are provided for information purposes only. They are not reflected in the step and job running times presented at termination time of the restarted job. You will not be charged twice for the time accumulated up to the abend.

Another point to consider in your accounting routine is the effect of a restart on the step sequence number available to the accounting routine. The following list indicates the sequence number presented to the accounting routine under the various restart conditions:

Condition	Step Sequence Value for Step n
Original execution	n
Automatic step restart	1
Automatic checkpoint/restart	n+1

Condition	Step Sequence Value for Step n
Deferred step restart	1
Deferred checkpoint/restart	2

Whenever an automatic restart is performed, the step sequence value accurately reflects the position of the step in the job.

In the case of a deferred restart, the restarting step is the first step of the restart job.

Job Step Time Limit

The EXEC statement TIME parameter specifies a limit on the processor time used by the related step. With any kind of restart, the entire value of the limit specified for the job step applies to the restart step. For a deferred restart, you may specify a limit different from the limit originally specified.

If the processor time used by a step exceeds the specified limit while a checkpoint entry is written, the entry is invalid and an abnormal termination occurs. You cannot do an automatic restart in this case but you can do a deferred checkpoint restart from an earlier checkpoint.

Completion of Step or Job Termination at System Restart

If a step or a job is terminating when system failure occurs, the termination is completed during the system restart that the operator may perform after the failure. This occurs whether or not the step or job uses checkpoint/restart, although the job journal option must be in use.

If other than the last step of a job is terminating when the failure occurs, the termination completes during the system restart, and the next step of the job initiates.

If the last step of a job terminates, or, if the job terminates, all necessary terminations are completed.

If a job requests an automatic restart, then abnormally terminates and system failure occurs before the restart termination processing is complete, the processing completes during the system restart.

JES3

If a job cannot complete executing because of system failure and the job requested job journaling, the job is restarted (warm started) by the system. If the job is eligible for automatic restart, the operator is sent message IEF225D asking if the job should be restarted. If the job is not eligible for automatic restart or if the operator indicated that restart should not be attempted, any temporary or VIO data sets the job allocated are deleted and the job is processed based upon the FAILURE option specified in the MAIN JCL statement. If FAILURE=RESTART is specified, JES3 automatically reschedules the job for execution from the beginning of the first step.

If automatic restart is requested for a job, the job's data sets are set up by JES3 for the first job step and not for the step being restarted. Canceling a job in a dependent network prevents successor jobs from executing if they are dependent upon successful completion of the canceled job. Any operator commands in the input stream of the job step being restarted are not executed. Restart of JES3 controlled jobs may be accompanied by message IAT2006 and/or IAT2575.

Note: A job that contains references to non-SMS-managed generation data sets that are referenced by a relative generation number of +1 or greater and with a disposition of OLD, SHR, or MOD is failed by the JES3 interpreter service unless the UNIT parameter is included on the DD statement. With UNIT specified, the JES3 interpreter service permits this data set to be allocated on a deferred basis.

COBOL

The COBOL RERUN clause can be used to provide the COBOL user with linkage to checkpoint/restart. Cautions and restrictions on the use of checkpoint/restart also apply to the use of the RERUN clause. For further information see the appropriate COBOL reference manual.

DFSORT™

When performing a sort with the IBM DFSORT™ program, you can, by including the CKPT parameter in sort control statements, cause checkpoint entries to be written and an automatic checkpoint/restart to be requested. The job control language can be used to request automatic or deferred step restarts or a deferred restart at a checkpoint.

PL/I

The PL/I user can use automatic and deferred step restart and can use COMMAND PQUIT to take checkpoints and get automatic and deferred checkpoint restarts. To cause a checkpoint entry to be written and request an automatic checkpoint/restart, the user codes CALL PLICKPT.

Each checkpoint entry in the checkpoint data set is identified by a system-generated checkid. A system message (including the checkid) at the console notifies the operator that a checkpoint entry is written.

The organization of the checkpoint data set is always physical sequential, and the data set may be written on magnetic tape or a direct access volume. Partitioned organization cannot be used.

A DD statement must be present in the job stream to define the checkpoint data set. The DISP parameter in this DD statement is used to specify whether single or multiple checkpoint entries are to be written. DISP=(NEW,KEEP) specifies a single checkpoint entry; DISP=(MOD,KEEP) specifies multiple checkpoint entries.

If CALL PLIREST is used in PL/I programs, the SCHEDxx PARMLIB member must specify 4092 as an eligible user completion code. For more information see [“Specification of Codes Eligible for Restart” on page 2](#).

Virtual Fetch

Modules managed by virtual fetch cannot issue a checkpoint. In addition, job steps that call virtual fetch cannot issue a checkpoint.

Appendix A. Checkpoint/Restart Codes

This appendix contains the following sections:

- “Return Codes” on page 55
- “Reason codes” on page 56
- “Modules and Return Codes for IHJ Messages” on page 81.

This appendix lists the return codes and reason codes associated with checkpoint/restart functions and describes the status conditions and error messages that may be encountered during normal operation. For more detailed information see the IHJxxxx messages in *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, *z/OS MVS System Messages, Vol 2 (ARC-ASA)*, *z/OS MVS System Messages, Vol 3 (ASB-BPX)*, *z/OS MVS System Messages, Vol 4 (CBD-DMO)*, *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*, *z/OS MVS System Messages, Vol 6 (GOS-IEA)*, *z/OS MVS System Messages, Vol 7 (IEB-IEE)*, *z/OS MVS System Messages, Vol 8 (IEF-IGD)*, *z/OS MVS System Messages, Vol 9 (IGF-IWM)*, and *z/OS MVS System Messages, Vol 10 (IXC-IZP)*.

Return Codes

The following return codes call attention to various successful or unsuccessful operating conditions.

Return Code (Hex)	Meaning
00	Successful completion. Code 00 is also returned if the RD parameter was coded as RD=NC or RD=RNC to totally suppress the function of CHKPT.
04	Automatic checkpoint restart has occurred at the checkpoint taken by the CHKPT macro instruction during the original execution of the job. Another restart from the same checkpoint can still occur. If a deferred restart was performed and RD=NC, NR, or RNC was specified in the resubmitted job, a request for another restart is not in effect.
	The following return codes have reason codes in register 0.
08	Unsuccessful completion. A checkpoint entry was not written, and a restart from this checkpoint was not requested. A request for an automatic restart from a previous checkpoint remains in effect.
0C	System-related or I/O-related problem: <ul style="list-style-type: none">• An error occurred during the handling of a system request, such as ESTAE, SETLOCK, or PURGE; or• an I/O error occurred in processing the CHKPT request; or• A VSAM error was detected while preparing a VSAM data set for the CHKPT request. This might occur for an attempted checkpoint or restart. If it is a restart, the checkpoint received return code 0, 10, or 18.

Return Code (Hex)	Meaning
10	<p>Successful completion with possible error condition. The task has control, by means of an explicit or implied use of the ENQ macro instruction, of a serially reusable resource; if the task terminates abnormally, it will not have control of the resource when the job step is restarted. Your program must, therefore, restore the enqueues.</p> <p>Additional information regarding explicit and implicit use of the ENQ macro instruction will be found under “Requesting Serially Reusable Resources” on page 19.</p> <p>Be sure that the dynamic link pack area is not altered until you no longer intend to attempt a restart from the identified checkpoint.</p>
14	<p>Unsuccessful completion. Internal error detected.</p> <p>This might occur for an attempted checkpoint or restart. If it is a restart, the checkpoint received return code 0, 10, or 18.</p>
18	<p>Error encountered restoring purged I/O. The checkpoint was successful, but because of the error that was detected, restart may not be possible. In addition, the job now in progress may fail because of I/O errors.</p>

When one of the errors indicated by code 08, 0C, 10, 14, or 18 occurs, the system prints an error message at the operator's console. The message contains a reason code that further identifies the error. The operator should report the contents of the message to the programmer.

Reason codes

Some reason codes may appear in messages that are associated with abends 13F, 23F, 33F, 43F, and 53F. Abend codes are indicated by an S in the following table. For more detailed information about the IHJxxxx reason codes see the following documentation:

- [z/OS MVS System Messages, Vol 1 \(ABA-AOM\)](#)
- [z/OS MVS System Messages, Vol 2 \(ARC-ASA\)](#)
- [z/OS MVS System Messages, Vol 3 \(ASB-BPX\)](#)
- [z/OS MVS System Messages, Vol 4 \(CBD-DMO\)](#)
- [z/OS MVS System Messages, Vol 5 \(EDG-GLZ\)](#)
- [z/OS MVS System Messages, Vol 6 \(GOS-IEA\)](#)
- [z/OS MVS System Messages, Vol 7 \(IEB-IEE\)](#)
- [z/OS MVS System Messages, Vol 8 \(IEF-IGD\)](#)
- [z/OS MVS System Messages, Vol 9 \(IGF-IWM\)](#)
- [z/OS MVS System Messages, Vol 10 \(IXC-IZP\)](#)

Also, see [z/OS MVS System Codes](#) for more information about system abend codes.

Some reason code programmer responses are "This is probably an error in checkpoint/restart record management". For each of them, the value and the fact that it represents an error in system logic is part of the programming interface. The nature of the error is not part of the interface.

Reason code	Return or abend code (hex)	Meaning
000	10	Possible enqueue error. Programmer Response: Ensure the program reestablished the enqueues upon restart.
001	08	A CHECKPOINT parameter list error was encountered. Programmer Response: Correct the parameter list.
002	08	The DD statement for the checkpoint data set is missing. Programmer Response: Ensure that the DD statement is correct and is included.
003	08, S13F	Insufficient storage for a checkpoint or restart. Programmer Response: Use a larger storage area. If the IEALIMIT routine has been modified, it may be necessary to alter it to request less storage.
004	0C	Open failed for checkpoint data set. Programmer Response: Locate the problem and correct it.
005	08	The checkpoint data set key length was not equal to zero. Programmer Response: Ensure that the checkpoint data set is not a keyed data set.
006	08	The checkpoint data set record format was not U. Programmer Response: Ensure that the record format is U.
007	08	The data control block (DCB) for the checkpoint data set was opened for other than BSAM or BPAM processing. Programmer Response: Ensure that only BSAM or BPAM is used.
008	08	The time interval that is specified in a STIMER macro had not elapsed. Programmer Response: Cancel the time interval before issuing the CHKPT macro instruction, and restore the interval after a restart.
009	08	The CHKPT macro instruction was issued in an exit routine other than the end-of-volume exit routine, or an RB of an unacceptable type was found on the RB chain. Programmer Response: Ensure that the CHKPT macro instruction is not issued in an exit routine other than the end-of-volume exit routine.
010	08	An open graphic data control block (DCB) has been detected, but it is not supported in Checkpoint/Restart. Programmer Response: Ensure that the CHKPT macro instruction is not issued while graphic DCBs are open.

Reason code	Return or abend code (hex)	Meaning
011	08	<p>The current task is a subtask or is not represented by the first job step TCB.</p> <p>Programmer Response: Ensure that the CHKPT macro instruction is not issued when multiple tasks (created by the ATTACH macro instruction) exist.</p>
012	08	<p>The current task has a subtask.</p> <p>Programmer Response: Ensure that the CHKPT macro instruction is not issued when multiple tasks (created by the ATTACH macro instruction) exist.</p>
013	08	<p>At the time the checkpoint was attempted, the job had issued the PCLINK macro instruction with the STACK option, but had not issued a corresponding PCLINK macro instruction with the UNSTACK parameter. (That is, the PCLINK stack was not empty.)</p> <p>Programmer Response: Ensure that the PCLINK stack is empty.</p>
014	08	<p>A WTOR macro instruction was issued prior to the CHKPT macro, and a reply had not been received.</p> <p>Programmer Response: Issue a WAIT macro instruction after the WTOR macro instruction, or a DOM macro instruction for any outstanding WTORS before the CHKPT macro instruction.</p>
015	08	<p>An incorrect checkpoint identification length or format was encountered.</p> <p>Programmer Response: When supplying a checkpoint identification, ensure that the length and format are correct.</p>
016	08	<p>The checkpoint data set device type was not magnetic tape or direct access.</p> <p>Programmer Response: Take checkpoints only to tape or DASD.</p>
017	10	<p>Insufficient storage to check enqueues—possible error.</p> <p>Programmer Response: Provide a larger storage area.</p>
018	10	<p>GQSCAN found an abnormal condition.</p> <p>Programmer Response: See the information provided by the GQSCAN service to locate and correct the problem.</p>
019	08	<p>Either MACRF=W was not specified for the checkpoint data control block (DCB), or the data control block was opened by the user's program, but was not opened for output.</p> <p>Programmer Response: Ensure that the DCB for the checkpoint data set has MACRF=W and, if opened, is opened for output.</p>
020	0C	<p>I/O error during open of CHKPT data set.</p> <p>Programmer Response: Locate the cause of the I/O error and correct the problem.</p>

Reason code	Return or abend code (hex)	Meaning
021	08	<p>LABEL was coded in the DD statement for the checkpoint data set with a value other than SL or SUL. Other values for the LABEL parameter must not be used in a checkpoint data set DD statement.</p> <p>Programmer Response: A tape data set to be used for creating a checkpoint data set must have standard labels. Correct the JCL and/or tape label.</p>
022	0C	<p>Error occurred reading or writing a SWA control block.</p> <p>Programmer Response: This is probably an error in the SWA manager. Contact your software support personnel.</p>
023	0C, S13F	<p>An uncorrectable input/output error occurred while writing the checkpoint data set.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it.</p>
024	S13F	<p>A data set has more than five volumes, but there is no JFCBX.</p> <p>Programmer Response: This is probably an error in the SWA manager. Contact your software support personnel.</p>
025	08	<p>A CHKPT request was made by a task that was open to a PDSE, or a PDSE was specified as a checkpoint data set.</p> <p>Programmer Response: Close PDSEs open at the time of checkpoint.</p>
026	0C	<p>Execution of a STOW macro instruction was unsuccessful; a permanent input/output error occurred.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it.</p>
027	08	<p>End-of-volume occurred while writing a checkpoint data set entry on a direct access volume, and secondary allocation was requested and allocated more than once; or end-of-volume occurred twice while writing a checkpoint data set entry on tape.</p> <p>Programmer Response: Ensure that all volumes (or DASD extents) have enough room for more than one checkpoint entry.</p>
028	0C, S33F, 10	<p>I/O error occurred during SYNCDEV to a user data set. The user's data set may not be available for a restart.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it.</p>
029	08	<p>The OPTCD=Q subparameter of the DCB parameter was coded. The value coded is not acceptable for a checkpoint data set. ASCII translation on checkpoint data set is not allowed.</p> <p>Programmer Response: Correct the indicated error and resubmit the job.</p>

Reason code	Return or abend code (hex)	Meaning
030	0C	<p>An uncorrectable I/O error occurred while quiescing pending I/O requests. The I/O error occurred with BSAM or with QSAM EROPT not equal to ACC.</p> <p>Programmer Response: Code EROPT=ACC (on your DD statement or DCB) for QSAM data sets, or locate the cause of the I/O error and correct it.</p>
031	S13F	<p>An uncorrectable I/O error occurred while reading from the checkpoint data set.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it. If all restrictions are met for a deferred restart, you can submit a deferred restart. Either restart at an earlier checkpoint or repeat the original execution using a different volume.</p>
033	08	<p>Cannot reposition to a tape data set or record because the block count in the DCB is negative. This may be caused by the label type being NL or BLP and the data set being open for RDBACK.</p>
034	S13F	<p>A DD statement was missing for the restart step.</p> <p>Programmer Response: Supply the missing DD statement.</p>
035	S13F	<p>Wrong length record detected when reading checkpoint data set.</p> <p>Programmer Response: Locate and correct the problem. Be alert for a possible security violation.</p>
036	0C	<p>Checkpoint received a nonzero return code from the MVS™ supervisor concerning the number of SSCR records it requires.</p> <p>Programmer Response: See “Modules and Return Codes for IHJ Messages” on page 81 for IHJ Messages.</p>
037	0C	<p>Checkpoint received a nonzero return code after requesting that the MVS supervisor fill in the SSCR records with checkpoint data.</p> <p>Programmer Response: See “Modules and Return Codes for IHJ Messages” on page 81 for IHJ Messages.</p>
038	S13F	<p>Restart received a nonzero return code when it passed the SSCR records containing MVS supervisors' checkpoint data.</p> <p>Programmer Response: See “Modules and Return Codes for IHJ Messages” on page 81 for IHJ Messages.</p>
039	08	<p>A PDS was converted to a PDSE between the time a checkpoint was taken and the time an attempt was made to restart at the checkpoint.</p>
040	S13F	<p>An uncorrectable I/O error occurred while reading standard volume labels.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it.</p>

Reason code	Return or abend code (hex)	Meaning
041	08	CHKPT was issued with a VSAM data set open for create mode processing, and no reposition was specified. Programmer Response: Ensure that a checkpoint is not issued when this condition exists, or correct this condition.
042	0C	An error occurred during repositioning for a VSAM data set. Programmer Response: Locate the error and correct it.
043	08	CHKPT was issued with an entry-sequenced data set open for output, an upgrade set open over it, and reposition required. NRE or NRC must be specified in the DD statement. Programmer Response: Locate the error and correct it.
044	08	CHKPT was issued for a relative-record data set open for create mode direct processing. Programmer Response: Ensure that a checkpoint is not issued when this condition exists, or correct this condition.
045	08	CHKPT was issued with a VSAM data set open with the GSR option. Programmer Response: Ensure that a checkpoint is not issued when this condition exists, or correct this condition.
046	08	A checkpoint is not allowed if any of the VSAM data sets in the address space are utilizing the control block update facility processing, or if any of the VSAM data sets in the address space are associated with an alternate index that was using the enhanced share option 4 processing. Programmer Response: Change the data set to be DISP=OLD or close the data set prior to issuing the CHKPT.
047	08	CHKPT was attempted with a VSAM data set that was using the UBF or CBIC option. Programmer Response: Ensure that a checkpoint is not issued when this condition exists, or correct this condition.
048	08	VSAM—media manager was in use. Programmer Response: Remove the CHKPT macro instruction.
049	08	VSAM RLS was in use. Programmer Response: Remove the CHKPT macro instruction.
050	S13F	A volume serial number at restart is not the same as it was when the checkpoint was taken. Programmer Response: Ensure that the volumes used at checkpoint are the same at restart and that units are available to mount them.
051	08	A SPIE or ESPIE exit was active when CHKPT request was made. Programmer Response: Do not issue a CHKPT request from a SPIE or ESPIE exit.

Reason code	Return or abend code (hex)	Meaning
052	S13F	<p>A module that was loaded in the link pack area (LPA) or the nucleus and was being used by the current job when the checkpoint was taken, is not in the LPA or nucleus at restart.</p> <p>Programmer Response: Ensure that the nucleus and LPA are the same at restart as when the checkpoint was taken.</p>
053	S13F	<p>A module has a different entry point address in the link pack area (LPA) or nucleus at restart than it did when the checkpoint was taken.</p> <p>Programmer Response: Ensure that the nucleus and LPA are the same at restart as when the checkpoint was taken.</p>
054	08	<p>A primary address space ID or secondary address space ID is not the same as the home address space ID. Cross-memory addressing was active when a checkpoint was requested.</p> <p>Programmer Response: Do not issue the CHKPT macro when in cross memory mode.</p>
055	0C	<p>Unacceptable return code received from system routine (VSMREGN) used to obtain starting address and size of private area regions.</p> <p>Programmer Response: Information about this problem should have been provided by the VSMREGN service of VSM (virtual storage manager).</p>
056	0C, 14	<p>Unacceptable return code received from system routine (VSMLIST) used to obtain ranges of virtual storage allocated.</p> <p>Programmer Response: Information about this problem should have been provided by the VSMLIST service of VSM (virtual storage manager).</p>
057	08	<p>Checkpoint DD statement is concatenated.</p> <p>Programmer Response: Checkpoint data sets must not be concatenated.</p>
058	14, S13F	<p>Internal parameter list is incorrect.</p> <p>Programmer Response: Use the diagnostic information provided by checkpoint/restart to locate and correct the problem.</p>
059	14, S13F	<p>Internal unexpected return code was received.</p> <p>Programmer Response: Use the diagnostic information provided by checkpoint/restart to locate and correct the problem.</p>
060	14	<p>Scheduler checkpoint routine discovered error in SWA control block.</p> <p>Programmer Response: This is probably an error in the SWA manager. Contact your software support personnel.</p>
061	08	<p>Caller of CHKPT was in secondary addressing mode.</p> <p>Programmer Response: Do not issue the CHKPT macro when in cross memory mode.</p>

Reason code	Return or abend code (hex)	Meaning
062	0C, 14	Checkpoint or Restart detected an internal error while processing checkpoint records. Programmer Response: This is probably an error in the SWA manager. Contact your software support personnel.
063	S13F	I/O error while repositioning a tape to a record within a data set. Programmer Response: Either restart at an earlier checkpoint or repeat the original execution, using a different volume.
064	10	DLPA module detected on load list. Programmer Response: Be sure that the dynamic link pack area is not altered until you no longer intend to attempt a restart from the identified checkpoint.
065	0C	Checkpoint received a nonzero return code from the MVS supervisor due to internal errors. Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.
066	0C	Restart received a nonzero return code from the MVS Supervisor due to internal errors. Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.
067	0C	Checkpoint received a nonzero return code from the MVS supervisor because a service routine failed. Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.
068	0C	Restart received a nonzero return code from the MVS supervisor because a service routine failed. Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.
069	08	Too much VSMLIST data—user's storage is fragmented beyond checkpoint's capabilities. Programmer Response: Reduce the region size, change the program to reduce storage fragmentation, or increase the size of extended private storage.
070	0C	Checkpoint received a nonzero return code from the MVS Contents Supervisor, because a provider routine return code was not zero. Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.

Reason code	Return or abend code (hex)	Meaning
071	0C	<p>Restart received a nonzero return code from the MVS contents supervisor, because a provider routine return code was not zero.</p> <p>Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.</p>
072	0C	<p>Checkpoint received a nonzero return code from the MVS supervisor because a provider routine return code was not zero.</p> <p>Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.</p>
073	0C	<p>Restart received a nonzero code from the MVS supervisor because a provider routine return code was not zero.</p> <p>Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.</p>
074	S13F	<p>An unacceptable I/O error occurred while reading a data set control block (DSCB). This reason code can also be issued if a data set was being scratched or deleted.</p> <p>Programmer Response: If it is during restart and the restart was deferred, the restart may be attempted again. If a preceding input/output error message for the restarted job identified the device in error, locate and correct the error. Ensure that no data set has been scratched or deleted.</p>
075	08	<p>STOW encountered a full directory on the checkpoint data set.</p> <p>Programmer Response: Increase the size of the directory for the checkpoint data set.</p>
076	S13F	<p>A direct access data set (other than a spooled data set) being processed for input did not occupy the same extent as it did originally; that is, discrepancies were found in comparing the space allocations described in the data extent block (DEB) and the data set control block (DSCB).</p> <p>Programmer Response: Ensure that no input data set (other than a spooled data set) has been expanded or rewritten in its original volume.</p>
078	S13F	<p>Restart was attempted on a different level of the system than when the checkpoint was taken.</p> <p>Programmer Response: Restart the job in the same processing environment as the one in which it was checkpointed.</p>

Reason code	Return or abend code (hex)	Meaning
079	S13F	<p>DUMMY was specified for an open data set; and either the data set was not being processed by the basic or queued sequential access methods, or the checkpoint at which restart was to occur was established in an end-of-volume exit routine for the data set.</p> <p>Programmer Response: Ensure that DUMMY is specified only for data sets being processed by the queued or basic sequential access methods. Also ensure that restart is not to occur at a checkpoint that was established in an end-of-volume exit routine for a data set that has been made DUMMY.</p>
080	S13F	<p>A subsystem data set was made DUMMY at restart. This usually refers to a spooled data set.</p> <p>Programmer Response: Do not make subsystem data sets DUMMY.</p>
081	S13F	<p>The checkpoint job or step specified ADDRSPC=REAL(V=R) and the restarting job or step did not, or vice versa.</p> <p>Programmer Response: Ensure that the V=R status of both the checkpoint and restarting job or step are the same. That is, if ADDRSPC=REAL was specified at checkpoint time, it can only be specified (and must be specified) at restart time.</p>
082	S13F	<p>The TCAM control program was not active at restart, but a TCAM DCB was open at checkpoint.</p> <p>Programmer Response: Ensure that the TCAM control program region is active and is large enough to add control blocks.</p>
083	S13F	<p>The value of the QNAME keyword parameter on the DD statement is not defined in the TCAM table.</p> <p>Programmer Response: Correct the DD statement or define the QNAME value in the TCAM table.</p>
084	S13F	<p>The process entry defined by the QNAME is already in use by another application program.</p> <p>Programmer Response: Restart the job when the specified QNAME is not being used.</p>
085	S13F	<p>A GETMAIN failed during TCAM restart.</p> <p>Programmer Response: Provide more storage.</p>
086	S13F, 08	<p>An abend occurred during a checkpoint or restart.</p> <p>Programmer Response: Use the diagnostic information provided to correct the error.</p>
087	S13F	<p>Record order on checkpoint data set incorrect and was detected by restart.</p> <p>Programmer Response: Use the diagnostic information provided to correct the error. Be alert for a possible security violation.</p>

Reason code	Return or abend code (hex)	Meaning
088	S13F	Storage not allocated as expected and was detected by restart. Programmer Response: Use the diagnostic information provided to correct the error.
090	14	Checkpoint or restart detected an error internal to checkpoint/restart. Programmer Response: Use the diagnostic information provided to correct the error.
091	S13F, 14	Error building a message. Checkpoint/restart detected an error while building a message for the operator. Programmer Response: Use the diagnostic information provided to correct the error.
092	S13F	An error occurred during basic partitioned access method (BPAM) processing. For example, an error was detected after the convert routine, or after reading the directory blocks. Programmer Response: Locate the problem and correct it.
093	14	An internal parameter list in an MVS supervisor call is incorrect. Programmer Response: This is probably an error in the checkpoint/restart MVS supervisor routine. Contact your software support personnel.
094	S13F	Nucleus routines or tables have moved or been deleted since checkpoint. LPA module entry points, which reside in the nucleus table, have been altered. This may result from a change link pack area (CLPA). Programmer Response: Resubmit the job on the same system with the same nucleus and LPA on which the checkpoint was taken.
095	14, 08	A control block chaining error occurred.
096	S13F	An error occurred during use of the STOW macro instruction to delete a member from the directory of a PDS. Programmer Response: Locate the cause of the error and correct it.
097	S13F	An unacceptable I/O error occurred in the installation's nonstandard label (NSL) routine (NSLREPOS). Programmer Response: Locate the cause of the I/O error and correct it.
098	S13F	Insufficient number of devices were available for restart. Programmer Response: Ensure that the UNIT parameter of the DD statement accurately reflects the device requirements for the data set.

Reason code	Return or abend code (hex)	Meaning
099	S13F	<p>A data set association block (DSAB) is missing at restart. This probably means a DD statement was omitted.</p> <p>Programmer Response: Probable system error. Locate and correct the problem.</p>
100	S13F	<p>MSS error occurred during mount processing. This is a probable system error.</p>
101	S13F	<p>An uncorrectable I/O error occurred while reading a volume label on a direct access device.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it.</p>
103	S13F	<p>A subsystem checkpoint record (SSCR) was written, but the corresponding DCB is not in the open data set table at restart.</p> <p>Programmer Response: Probable error in checkpoint/restart. Contact your software support personnel.</p>
104	S13F	<p>An I/O error occurred while repositioning to the correct data set on tape.</p> <p>Programmer Response: Locate the cause of the I/O error and correct it.</p>
105	S43F	<p>Library Automation services detected an error while processing a tape. See message text following message IHJ014I for detailed information about the error.</p> <p>Programmer Response: Probable system error. Contact your software support personnel.</p>
106	S53F	<p>A Tape Library Dataserver tape volume was mounted, and the internal label serial number did not match the volume serial number of the tape requested.</p> <p>Programmer Response: After verifying that the correct tape is available, resubmit the job.</p>
107	S53F	<p>The file protect status for a Tape Library Dataserver tape volume was incorrect. Restart attempted to reset it using PSF, but encountered an error.</p> <p>Programmer Response: Insure that the tape volume is correctly file protected, then resubmit the job.</p>
108	04	<p>Restart was unable to update statistics (i.e., last mount date and last write date) for a Tape Library Dataserver tape volume, because SMS was not active.</p> <p>Programmer Response: This is a warning indicating that checkpoint/restart was unable to update the tape configuration data base (TCDB) tape volume record; however, the restart has completed successfully.</p>

Reason code	Return or abend code (hex)	Meaning
109	04	<p>Restart was unable to update statistics (i.e., last mount date and last write date) for a Tape Library Dataserver tape volume. An error return code was returned from the SSI call.</p> <p>Programmer Response: This is a warning indicating that checkpoint/restart was unable to update the tape configuration data base (TCDB) tape volume record; however, the restart has completed successfully.</p>
110	04	<p>Restart was unable to update statistics (i.e., last mount date and last write date) for a Tape Library Dataserver tape volume. An error code was returned by SMS.</p> <p>Programmer Response: This is a warning indicating that checkpoint/restart was unable to update the tape configuration data base (TCDB) tape volume record; however, the restart has completed successfully.</p>
112	0C	<p>Checkpoint was unable to load the MVS CSV (Contents Supervisor) routine.</p> <p>Programmer Response: This is probably an error with the CSV routine. Contact your software support personnel.</p>
113	0C	<p>Restart was unable to load the MVS CSV (Contents Supervisor) routine.</p> <p>Programmer Response: This is probably an error with the CSV routine. Contact your software support personnel.</p>
114	08	<p>UNIX[®] system services is active in the current address space.</p> <p>Programmer Response: Checkpoint is not supported for HFS files. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
117	08	<p>The system cannot take the checkpoint because one or more of the following conditions are true:</p> <ul style="list-style-type: none"> • a data set was dynamically allocated with the XTIOU option, with the DSAB-above-the-line option or without UCB capture • Data set ENQs are being managed in memory for this address space • SWB control blocks for dynamically allocated data sets are placed in 64-bit storage. • The system is using non-executable memory for the parameter list passed in register 1 to the program on the JCL EXEC statement. • A CHKPT request was made by a started task. Started tasks do not support checkpoints. <p>Programmer Response: One of the following:</p> <ul style="list-style-type: none"> • Modify any dynamic allocated requests to remove the use of the XTIOU, NOCAPTURE, and DSAB-above-the-line options. • Remove the call to the IEFDDSRV MODIFY, TYPE=FEATURE, DSENQMGMT=MEMORY service. • Remove the call to the IEFDDSRV MODIFY, TYPE=FEATURE, SWBSTORAGE=ATB service. • Ask the system programmer to update the DIAGxx parmlib member to specify CBATTR EXECUTABLE(JCLPARM). • Remove the CHKPT macro instruction.
124	08	<p>A CHKPT request was made by a task that was open to a checkpoint data set with a DCBE.</p> <p>Programmer Response: Ensure that checkpoint is not issued while this condition exists. DCBE's are not supported by checkpoint.</p>
125	08	<p>A CHKPT request was made by a task that was open to a sequential extended format data set, or an extended format data set was specified as a checkpoint data set.</p> <p>Programmer Response: Ensure that checkpoint is not issued while this condition exists, or correct this condition.</p>
126	08	<p>A PDS was converted to an extended format data set between the time a checkpoint was taken and the time an attempt was made to restart at the checkpoint.</p> <p>Programmer Response: Ensure that the restart is not issued while this condition exists, or correct this condition.</p>
127	08	<p>A CHKPT request was made by a task that was open to an encrypted sequential basic or large format data set, or an encrypted sequential basic or large format data set was specified as a checkpoint data set.</p> <p>Programmer Response: Ensure that the checkpoint is not issued while this condition exists or correct this condition.</p>

Reason code	Return or abend code (hex)	Meaning
181	S13F	<p>An error occurred during VSAM restart preformat routine. <i>yyy</i> is the value from the RPL error code.</p> <p>Programmer Response: The error is for VSAM only. For more information see the RPL error codes in z/OS DFSMS Macro Instructions for Data Sets for VSAM data sets.</p>
182	S13F	<p>An error occurred during VSAM restart verify routine. <i>yyy</i> is the value from the RPL error code.</p> <p>Programmer Response: The error is for VSAM only. For more information see the RPL error codes in z/OS DFSMS Macro Instructions for Data Sets for VSAM data sets.</p>
183	S13F	<p>An error occurred during VSAM restart put routine. <i>www</i> is the value from the RPL error code.</p> <p>Programmer Response: The error is for VSAM only. For more information see the RPL error codes in z/OS DFSMS Macro Instructions for Data Sets for VSAM data sets.</p>
184	S13F	<p>An error occurred during VSAM restart index put routine. <i>www</i> is the value from the RPL error code.</p> <p>Programmer Response: The error is for VSAM only. For more information see the RPL error codes in z/OS DFSMS Macro Instructions for Data Sets for VSAM data sets.</p>
185	S13F	<p>VSAM open error during restart. <i>www</i> is the value from the ACB error code.</p> <p>Programmer Response: See message IEC161I or IEC252I.</p>
186	0C	<p>VSAM close error. <i>www</i> is the value from the ACB error code.</p> <p>Programmer Response: The message is for VSAM only. See message IEC161I or IEC252I.</p>
190	S13F	<p>An error occurred while attempting to obtain catalog information for the cluster identified by the ddname in message IHJ0091.</p> <p>Programmer Response: The error is for VSAM only. For more information, see messages IEC331I and IDA3009I, if issued.</p>
191	S13F	<p>An error occurred while attempting to mount volumes for the ddname identified by message IHJ0091.</p> <p>Programmer Response: Increase the number of units allocated for the ddname.</p> <p>The error is for VSAM only. For more information, see message IEC161I or IEC252I, if issued.</p>
193	S13F	<p>Checkpoint was taken during VSAM create mode, but, at restart time, the data set was no longer in create mode.</p> <p>Programmer Response: A restart is not allowed. Take the checkpoint after the data set is loaded.</p>

Reason code	Return or abend code (hex)	Meaning
194	S13F	<p>The catalog entries for a VSAM upgrade path have been altered since checkpoint.</p> <p>Programmer Response: If the upgrade path can be restored to its condition at checkpoint, the restart may be resubmitted.</p>
195	S13F, 0C	<p>Insufficient storage is available for a VSAM restart.</p> <p>Programmer Response: Increase the available storage.</p>
196	S13F	<p>The catalog for one or more VSAM data sets has been updated, indicating that another job successfully processed the data set after the checkpoint was taken. Restart cannot restore the data set to its checkpoint status.</p> <p>Programmer Response: Override the condition, and allow restart by coding AMP='CROPS=NCK' in the DD statement for the data set. Be aware, however, that changes might have been made to the data between the time the checkpoint was taken and when restart is attempted. A data set that has been extended to span volumes since the checkpoint was taken cannot be restarted.</p> <p>The error is for VSAM only. For more information, see messages IEC161I or IEC252I, if issued.</p>
197	S13F	<p>A restart is not allowed for control block update facility (CBUF) processing.</p> <p>Programmer Response: Do not issue the CHKPT macro instruction while using CBUF processing.</p>
198	S13F	<p>A VSAM data set extended to a new volume after the checkpoint was taken.</p> <p>Programmer Response: The checkpoint cannot be restarted. For a deferred restart, resubmit the job selecting a checkpoint taken after the data set extended to the new volume.</p> <p>The error is for VSAM only. For more information, see message IEC161I or IEC252I, if issued.</p>
199	S13F	<p>An error occurred during BLDVRP processing in VSAM restart. <i>www</i> is the return code from the BLDVRP request.</p> <p>Programmer Response: The error is for VSAM only. For more information see z/OS DFSMS Macro Instructions for Data Sets.</p>
200	0C	<p>A PURGE I/O request failed.</p> <p>Programmer Response: Locate and correct the error.</p>
201	S13F	<p>Page Fix (PGSER) failed. This is an uncorrectable error.</p> <p>Programmer Response: Probable system error. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
202	0C, S13F	<p>SETLOCK failed. This is an uncorrectable error.</p> <p>Programmer Response: Probable system error. Contact your software support personnel.</p>
203	S13F	<p>An invalid attempt has been made to invoke RESTART SVC routines.</p> <p>Programmer Response: Only the scheduler may invoke RESTART. Remove either the JCL or program logic that attempts to execute RESTART.</p>
204	0C	<p>VIO data sets could not be checkpointed for a subsequent automatic restart. This checkpoint is unavailable for automatic restarts; however, deferred restarts are possible if the VIO data sets are made DUMMY.</p> <p>Programmer Response: An error occurred while checkpointing VIO data sets to the job journal. Deferred restarts may be accomplished on the checkpoint entry; however, the VIO data sets must be made DUMMY. Locate and correct the error.</p>
205	0C	<p>VIO data sets could not be checkpointed for a subsequent automatic restart. Any checkpoints taken during this step are ineligible for automatic restarts; however, deferred restarts are possible if the VIO data sets are made DUMMY. An automatic restart cannot occur until a new and successful checkpoint is taken.</p> <p>Programmer Response: An error occurred while checkpointing VIO data sets to the job journal. Deferred restarts may be accomplished on the checkpoint entry; however, the VIO data sets must be made DUMMY. Locate and correct the error.</p>
206	08	<p>A VTAM[®] ACB was open when a CHKPT was attempted.</p> <p>Programmer Response: Ensure that all VTAM[®] ACBs are closed prior to issuing CHKPT.</p>
207	0C, S13F	<p>Failure in the subsystem interface when checkpointing or restarting subsystem data sets.</p> <p>Programmer Response: Probable subsystem error. Contact your software support personnel.</p>
208	08	<p>Erroneous data exists in tape for the new checkpoint data set presented by the user. The integrity of this data set is uncertain and therefore unusable.</p> <p>Programmer Response: If the new checkpoint data set is a sequential data set, the failure occurred because the data set was open for MOD and residual data was found. Either remove the MOD disposition, reinitialize the data set to an empty status, or reallocate the data set. If the data set is a PDS, the failure occurred as a result of residual members in the directory. Either scratch the members or reallocate the data set.</p>

Reason code	Return or abend code (hex)	Meaning
209	08	<p>Another user DCB is concurrently open to the checkpoint data set presented by the user.</p> <p>Programmer Response: Remove the code that violates this restriction, or change the DD statement for the checkpoint data set.</p>
210	08	<p>The checkpoint data set presented by the user was defined with DISP=SHR.</p> <p>Programmer Response: Change the JCL to reflect a disposition other than SHR for the checkpoint data set.</p>
211	08	<p>An attempt to create a new checkpoint data set has failed because it resides on an insecure volume.</p> <p>Programmer Response: The candidate volume for a checkpoint data set was rejected by the operator as not being secure. Contact installation personnel and verify the eligibility of the volume for creating checkpoint data sets.</p>
213	08	<p>The checkpoint data set offered by the user is a subsystem data set. Subsystem data sets cannot be used as checkpoint data sets.</p> <p>Programmer Response: Correct the JCL and resubmit the job.</p>
214	08	<p>The new checkpoint data set offered by the user exists on a shared direct access device. Concurrent opens cannot be controlled; therefore, the data set is unacceptable.</p> <p>Programmer Response: Contact operations personnel to ensure that this does not occur. See also “Coding the DD Statement for a Checkpoint Data Set” on page 8.</p>
215	18	<p>An error occurred while restoring the purged I/O operations. A checkpoint entry was successfully written; however, continuation of the user's program may produce errors.</p> <p>Programmer Response: Locate and correct the error.</p>
216	S13F	<p>Residual subpool allocations in the user's region prevented a successful reallocation of the user's checkpoint-time region image.</p> <p>Programmer Response: Possible system problem. Locate and correct the error. Make sure that REGION=0M or OK was not been specified. Also check that REGION was not overridden by an exit to be 0M or OK.</p>
219	S13F	<p>There is a password error for a password-protected tape data set.</p> <p>Programmer Response: Obtain the proper password for the password-protected data set.</p>
220	S13F	<p>The data set control block (DSCB) address of a data set to be repositioned has changed since the checkpoint was taken.</p> <p>Programmer Response: If possible, return the volume to the condition it was in at checkpoint.</p>

Reason code	Return or abend code (hex)	Meaning
221	S13F	<p>The wrong password was given for a password-protected data set that was to be repositioned.</p> <p>Programmer Response: Obtain the proper password for the password-protected data set.</p>
222	S13F	<p>The tape data set name, recorded in header label 1, has changed since the checkpoint was taken.</p> <p>Programmer Response: If possible, return the volume to the condition it was in at checkpoint.</p>
224	08	<p>An open data set has a type of ACB that checkpoint does not support. Checkpoint supports VSAM ACB's and an ACB that the system builds when the user opens a DCB for a subsystem data set, such as spooled data sets.</p> <p>Programmer Response: Ensure that all data sets of this type are closed prior to issuing the checkpoint.</p>
240	0C, S13F	<p>A recovery (ESTAE) environment could not be established.</p> <p>Programmer Response: Probable system error. Contact your software support personnel.</p>
241	0C, S13F	<p>An indeterminate error occurred when processing VSAM data sets.</p> <p>Programmer Response: Locate and correct the error using the diagnostic information provided.</p>
242	0C, S13F	<p>A machine check occurred while processing VSAM data sets.</p> <p>Programmer Response: If the machine check was transitory, resubmit the job. Otherwise, have the problem fixed before resubmitting the job.</p>
243	S13F	<p>An invalid SSCR record was encountered in the checkpoint data set entry when repositioning VSAM data sets. This is a probable system logic error.</p> <p>Programmer Response: Resubmit the job, selecting another checkpoint entry. Be alert for a possible security violation.</p>
250	08	<p>SYS1.IMAGELIB data set open with the IMGLIB macro when checkpoint taken.</p> <p>Programmer Response: Ensure that all IMAGELIB DCBs are closed prior to issuing CHKPT. See also “IMGLIB” on page 17.</p>
251	S23F	<p>A data set that was not a checkpoint data set at checkpoint time was found to be open as a secure checkpoint data set at restart time.</p> <p>Programmer Response: Ensure the proper data set volume was mounted at restart time. Also, verify the status of the subject volume by contacting the operations personnel. Be alert for a possible security violation.</p>

Reason code	Return or abend code (hex)	Meaning
252	S23F	<p>In a job using more than one checkpoint data set, one of the checkpoint data sets (not the one used for restart) was deemed not secure.</p> <p>Programmer Response: Ensure the proper data set volume was mounted at restart time. Also verify the status of the subject volume by contacting the operations personnel. Be alert for a possible security violation.</p>
255	S13F, S23F	<p>Security violation—user not authorized to access a RACF[®] protected data set at restart.</p> <p>Programmer Response: See IHJ101I or IHJ102I for information. Be alert for a possible security violation.</p>
513	14	<p>An internal error was detected for a bad parameter list identifier.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
514	14	<p>An internal error was detected for an incorrect parameter list level.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
515	14	<p>An internal parameter list is too short.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
516	14	<p>An internal storage list has been invalidated.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
517	14	<p>More than one internal requester flag was selected.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
518	14	<p>A reserved field in an internal parameter list is not zeros.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
519	14	<p>An invalid request code was passed in an internal parameter list for record management processing.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
520	14	<p>A write request was made with an invalid record address.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
521	14	<p>A write request was made with an invalid record length.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
522	14	<p>No record header address was provided in a write request and the record ID is for a record type that requires a separate record header.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
523	14	<p>No record header length was provided in a write request when a record header was provided.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
524	14	<p>The caller of an internal request is an input function and requested an output operation or vice versa.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
526	14	<p>A DCB address was not passed on an OPEN for output.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
527	14	<p>No length was specified on a write request for a record appendage.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
528	14	<p>Error positioning to the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
529	14	<p>An OPEN for input or output must precede any other operation requests.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
530	20	<p>An invalid record type was found in the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Be alert for a possible security violation. Contact your software support personnel.</p>
531	20	<p>An unrecognizable record code was found in the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Be alert for a possible security violation. Contact your software support personnel.</p>
533	12	<p>A checkpoint/restart GETMAIN request received a nonzero return code.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
537	12	<p>An attempt to open the checkpoint data set failed.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
538	20	<p>A buffer with data was found when none was expected.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
539	20	<p>A buffer that should have been full was found to be only partially filled.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
540	14	<p>The storage list pointer in an internal parameter list is zero.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
541	14	<p>An attempt to do a DEB validity check failed.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
542	14	<p>The record code passed for a write request is an end-of-block code. This code can not be passed as a record code.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
543	14	<p>A request to write a record included a record too big for a buffer and of a type that can not be split.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
544	08	<p>A checkpoint ID was specified that is not a member of the checkpoint PDS.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Possible user error. Contact your software support personnel.</p>
545	08	<p>An OPEN for input was attempted to a data set that is not a secure checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
546	08	<p>A requested checkpoint ID could not be found in the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Possible user error. Contact your software support personnel.</p>
547	20	<p>A buffer that should have been empty was marked full.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
548	08	<p>A volume switch occurred while writing a checkpoint entry in the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
549	14	<p>A request to write an SSCR record was passed with an incorrect header length.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
550	14	<p>A request to write an SSCR record was passed with nonzero values in reserved fields.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
551	14	<p>A request to write an SSCR record was passed with its data length set to zero.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
552	14	<p>A request to write an SSCR record was passed with the DCB address set to zero.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
553	14	<p>A request to write a record was passed with an unknown record code.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
554	14	<p>The DCB passed for OPEN for output was not a checkpoint protected DCB.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
555	08	<p>The ddname passed in an open for input request is not allocated.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
556	12	<p>On a call to SWA to read a JFCB, a nonzero return code was received.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
557	14	<p>No ddname was passed on an OPEN for input request by scheduler restart.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
558	14	<p>No checkpoint ID address was passed on an OPEN for input request by scheduler restart.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
559	14	<p>A zero length was passed for the checkpoint ID by scheduler restart.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>

Reason code	Return or abend code (hex)	Meaning
560	14	<p>No checkpoint header record address was passed on an OPEN for input request by scheduler restart.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
561	14	<p>A zero length was passed for the checkpoint header record by scheduler restart.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
562	12	<p>An error occurred while calculating block size for the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
563	12	<p>An optimum block size could not be calculated for the checkpoint data set.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
564	08	<p>An attempt was made by scheduler restart to open a checkpoint data set that was created on a release of DFP or MVS/SP that did not have complete scheduler checkpoint support.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
565	12	<p>A nonzero return code was returned from an ESTAE service request.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
566	08	<p>The extents of a DASD checkpoint data set have been extended.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>
567	14	<p>A record split was requested for a record type that cannot be split.</p> <p>Programmer Response: This is probably an error in checkpoint/restart record management. Contact your software support personnel.</p>

Modules and Return Codes for IHJ Messages

When Checkpoint/Restart issues messages IHJ001I and IHJ007I with reason codes 036, 037, or 038, variables **m** and **rc** might appear. **m** designates a module id, which corresponds with a module name; **rc** designates a return code issued by module **m**. The return codes are in decimal format.

Module ID	Return Code	Meaning
01		Module IEAVCKRS—Supervisor Control Checkpoint/Restart Router.
	08	The number of requested SSCRs for the second checkpoint invocation does not match the number requested in the first checkpoint invocation.
	12	The work unit is Subspace Active. Checkpoint is not allowed.
	16	The work unit has high virtual storage attached. Checkpoint is not allowed.
02		Module IEAVSPI—SPIE/ESPIE Checkpoint/Restart Routine.
	08	ESPIE exits were created while the task was running with a PSW key other than the assigned key.
	255	IEAVSPI requested a negative number of SSCRs.
03		Module IEAVCRVF—Vector Checkpoint/Restart Routine.
	04	The pointer to the subject task's VSSA is invalid.
	08	The vector SLIH determined that there are no eligible vector processors with an attached Vector Facility (VF) online.
	12	The vector SLIH could not obtain a VSSA for the task.
	16	The VF section size recorded during the checkpoint is not equal to the VF section size during the restart.
	20	In the time interval between checkpoint and the restart, the mapping of the VSSA changed.
	24	There are no processors with an attached VF in the complex.
	28	An unexpected error occurred in IEAVCRVF.
	32	VF status has been lost because of a machine check.
	255	IEAVCRVF requested a negative number of SSCRs.
04		Module IEAVLSCR—Linkage Stack Checkpoint/Restart Routine.
	08	There are entries on the task's linkage stack.
05		Module IEAVALCR—Access List Checkpoint/Restart Routine.
	08	The task has entries in use on either the dispatchable unit access list (DU-AL) or on the primary address space access list (PASN-AL). That is, the task has established addressability to another address/data space via an access list.
	0C	An unexpected error occurred during IEAVALCR processing.
06		Module IEANTCKR—Name/Token checkpoint/restart extension routine.
	08	Address space owns name/token pairs that were not created with the IEANT_CHECKPOINTOK value in the persist parameter. Remove the definition before taking the checkpoint or add the IEANT_CHECKPOINTOK option.

Appendix B. End-of-Volume Checkpoint Routine

In a data control block exit list, you can specify the address of a routine that receives control when end-of-volume is reached in processing a physical sequential data set (BSAM or QSAM). For information about forming an exit routine and coding the EOVS exit list for physical sequential data sets, see [*z/OS DFSMS Using Data Sets*](#). The routine is entered after a new volume is mounted and all necessary label processing is completed. If the volume is magnetic tape, the tape is positioned after the tape mark that precedes the beginning of the data. The end-of-volume exit routine may take a checkpoint by issuing the CHKPT macro instruction. The job step can be restarted from this checkpoint. When the job step is restarted, the volume is mounted and positioned as upon entry to the routine.

The end-of-volume exit routine returns control the same way as any other data control block exit routine. Restart is impossible if changes are made to the link pack area or nucleus after the checkpoint is taken. When the step is restarted, the virtual storage addresses of end-of-volume modules must be the same as when the checkpoint was taken.

Appendix C. Accessibility

Accessible publications for this product are offered through [IBM Documentation for z/OS \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see [How to Send Feedback to IBM](#) to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
Site Counsel
2455 South Road*

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual.

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See:

This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

ABSTR

Absolute track (value of SPACE).

ACB

Access method control block.

ACS

See *automatic class selection*.

ADSP

See *automatic data set protection*.

AL

American National Standard Labels.

allocation

Generically, the entire process of obtaining a volume and unit of external storage, and setting aside space on that storage for a data set. Also refers to the process of logically connecting a program and a data set. The program is running or is about to run.

AMODE

Addressing mode (24, 31, ANY).

ANSI

American National Standards Institute.

AOR

Application owning region.

APF

Authorized program facility.

application

The use to which an access method is put or the end result that it serves; contrasted to the internal operation of the access method.

ASCII

American National Standard Code for Information Interchange.

AUL

American National Standard user labels (value of LABEL).

ATL

See *automated tape library*.

automated tape library data server

A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. Contrast with *manual tape library*. See also *tape library*.

automatic class selection (ACS) routine

A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

automatic data set protection (ADSP)

In z/OS, a user attribute that causes all permanent data sets created by the user to be automatically defined to RACF with a discrete RACF profile.

AVGREC

Average record scale (JCL keyword).

basic format

The format of a data set that has a data set name type (DSNTYPE) of BASIC. A basic format data set is a sequential data set that is specified to be neither large format nor extended format. The size of a basic format data set cannot exceed 65 535 tracks on each volume.

BCDIC

Binary coded decimal interchange code.

BCS

Basic catalog structure.

BDAM

Basic direct access method.

BDW

Block descriptor word.

BFALN

Buffer alignment (parameter of DCB).

BFTEK

Buffer technique (parameter of DCB).

BISAM

Basic index sequential access method.

BLP

Bypass label processing.

BLKSIZE

Block size (parameter of DCB).

BLT

Block locator token.

BPAM

Basic partitioned access method.

BPI

Bytes per inch.

BSAM

Basic sequential access method.

BSM

Backspace past tape mark and forward space over tape mark (parameter of CNTRL).

BSP

Backspace one block (macro).

BSR	Backspace over a specified number of blocks or records (parameter of CNTRL).
BUFC	Buffer control block.
BUFCB	Buffer pool control block (parameter of DCB).
BUFL	Buffer length (parameter of DCB).
BUFNO	Buffer number (parameter of DCB).
BUFOFF	Buffer offset (length of ASCII block prefix by which the buffer is offset; parameter of DCB).
CA	Control area.
catalog	A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics. See <i>master catalog</i> .
CBIC	Control blocks in common.
CBUF	Control block update facility.
CCHHR	Cylinder/head/record address.
CCW	Channel command word.
CF	Coupling facility.
CI	Control interval.
CIDF	Control interval definition field.
CKDS	Cryptographic key data set.
CLPA	Change link pack area.
CONTIG	Contiguous space allocation (value of SPACE).
CSA	Common storage area.
CSW	Channel status word.
CYLOFL	Number of tracks for cylinder overflow records (parameter of DCB).
DA	Direct access (value of DEVD or DSORG).
DADSM	Direct access device space management.
DASD	Direct access storage device.

DATACLAS

Data class (JCL keyword).

data control block (DCB)

A control block used by access method routines (other than VSAM or VTAM) in storing and retrieving data.

Data Facility Storage Management Subsystem (DFSMS)

An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

Data Facility Storage Management Subsystem data facility product (DFSMSdftp)

A DFSMS functional component and a base element of z/OS that provides functions for storage management, data management, program management, device management, and distributed data access.

data security

Prevention of access to or use of data or programs without authorization. As used in this publication, the safety of data from unauthorized use, theft, or purposeful destruction.

DAU

Direct access unmovable data set (value of DSORG).

DBCS

Double-byte character set.

DCB

See *data control block* (control block name, macro, or parameter on DD statement).

DCBD

Data control block dummy section.

DCBE

Data control block extension.

DD

Data definition.

DDM

Distributed Data Management.

DEB

Data extent block.

DECB

Data event control block.

DEN

Magnetic tape density (parameter of DCB).

DES

Data encryption standard.

DEV

Device-dependent (parameter of DCB).

DFSMSdss

A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm

A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DISP

Disposition (JCL DD parameter).

DIV

Data-in-virtual.

DLF

Data lookaside facility.

DSCB

Data set control block.

DSORG

Data set organization (parameter of DCB and DD and in a data class definition).

EBCDIC

Extended binary coded decimal interchange code.

ECB

Event control block.

ECSA

Extended common service area.

ENABLE

Relationship that allows a system to allocate and access data sets in a VIO storage group, a pool storage group, or individual volumes within a pool storage group.

entry

A collection of information about a cataloged object in a master or user catalog.

entry-sequenced data set

A data set whose records are loaded without respect to their contents, and whose RBAs cannot change. Records are retrieved and stored by addressed access, and new records are added at the end of the data set.

EOB

End-of-block.

EOD

End-of-data.

EODAD

End-of-data set exit routine address (parameter of DCB, DCBE, and EXLST).

EOV

End-of-volume.

ESDS

See *entry-sequenced data set*.

ESETL

End-of-sequential retrieval (QISAM macro).

ESPIE

Extended specify program interruption exits.

exclusive control

Preventing multiple WRITE-add requests from updating the same dummy record or writing over the same available space on a track. When specified by the user, exclusive control requests that the system prevent the data block about to be read from being modified by other requests; it is specified in a READ macro and released in a WRITE or RELEX macro. When a WRITE-add request is about to be processed, the system automatically gets exclusive control of either the data set or the track.

EXCP

Execute channel program.

EXLST

Exit list (parameter of DCB and VSAM macros).

EXPDT

Expiration date for a data set (JCL keyword).

export

To create a backup or portable copy of a VSAM cluster, alternate index, or integrated catalog facility user catalog.

FCB

See *forms control buffer*.

FEOV

Force end-of-volume (macro).

file system

In the z/OS UNIX environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

FOR

File owning region.

forms control buffer (FCB)

A buffer for storing codes for control of certain printers such as the 3205-5, 3211, 3262 Model 5, or IBM 3800 Printing Subsystem.

FSM

Forward space past tape mark and backspace over tape mark (parameter of CNTRL).

FSR

Forward space over a specified number of blocks or records (parameter of CNTRL).

GCR

Group coded recording (tape recording).

GDG

See *generation data group*.

GDGNT

Generation data group name table.

GDS

Generation data set.

generation data group (GDG)

A collection of data sets with the same base name, such as PAYROLL, that are kept in chronological order. Each data set is called a generation data set.

GL

GET macro, locate mode (value of MACRF).

global resource serialization (GRS)

A component of z/OS used for serializing use of system resources and for converting hardware reserves on DASD volumes to data set enqueues.

GM

GET macro, move mode (value of MACRF).

GRS

See *global resource serialization*.

GSR

Global shared resources.

GTF

Generalized trace facility.

HCD

Hardware configuration dialog.

HFS

See *hierarchical file system*.

hierarchical file system (HFS) data set

A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using z/OS UNIX System Services. See also *file system*.

IBG

Interblock gap.

ICFC

Integrated catalog facility catalog.

ICI

Improved control interval access.

import

To restore a VSAM cluster, alternate index, or integrated catalog facility catalog from a portable data set created by the EXPORT command.

IPL

Initial program load.

INOUT

Input then output (parameter of OPEN).

I/O

Input/Output.

IOB

Input/output block.

IRG

Interrecord gap.

IS

Indexed sequential (value of DSORG).

ISAM

Indexed sequential access method.

ISMF

Interactive Storage Management Facility.

ISO

International Organization for Standardization.

ISU

Indexed sequential unmovable (value of DSORG).

JCL

Job control language.

JES

Job entry subsystem.

JFCB

Job file control block.

JFCBE

Job file control block extension.

KEYLEN

Key length (JCL and DCB keyword).

key-sequenced data set (KSDS)

A VSAM data set whose records are loaded in ascending key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in key sequence because of free space allocated in the data set. Relative byte addresses of records can change because of control interval or control area splits.

KSDS

See *key-sequenced data set*.

large format

The format of a data set that has a data set name type (DSNTYPE) of LARGE. A large format data set has the same characteristics as a sequential (non-extended format) data set, but its size on each volume can exceed 65 535 tracks. There is no minimum size requirement for a large format data set.

link-pack area (LPA)

An area of virtual storage that contains re-enterable routines that are loaded at IPL (initial program load) time and can be used concurrently by all tasks in the system.

LPA

See *Link-pack area*.

LRECL

Logical record length (JCL keyword).

LRI

Logical record interface.

LSR

Local shared resources.

MACRF

Macro instruction form (parameter of DCB and ACB).

master catalog

A catalog that contains extensive data set and volume information that VSAM requires to locate data sets, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set, and accumulate usage statistics for data sets.

MBBCHHR

Module number, bin number, cylinder number, head number, record number.

MCP

Message control program.

MGMTCLAS

Management class (JCL keyword).

MLA

Multilevel alias.

MOD

Modify data set (value of DISP).

MSHI

Main storage for highest-level index (parameter of DCB).

MSWA

Main storage for work area (parameter of DCB).

NCI

Normal control interval.

NCP

Number of channel programs (parameter of DCB).

NOPWREAD

No password required to read a data set (value of LABEL).

NRZI

Nonreturn-to-zero-inverted.

NSL

Nonstandard label (value of LABEL).

NSR

Nonshared resources.

NTM

Number of tracks in cylinder index for each entry in lowest level of master index (parameter of DCB).

NUB

No user buffering.

NUP

No update.

OPTCD

Optional services code (parameter of DCB).

OUTIN

Output then input (parameter of OPEN).

OUTINX

Output at end of data set (to extend) then input (parameter of OPEN).

partitioned data set (PDS)

A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE)

A data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PDAB

Parallel data access block.

PDS

See *partitioned data set*.

PDSE

See *partitioned data set extended*.

PE

Phase encoding (tape recording mode).

performance subsystem function (PSF)

A command to transfer data from the control program to the subsystem to control certain conditions or functions within a control unit.

PL

PUT macro, locate mode (value of MACRF).

PM

PUT macro, move mode (value of MACRF).

PO

Partitioned organization (value of DSORG).

POSIX

Portable operating system interface for computer environments.

POU

Partitioned organization unmovable (value of DSORG).

PRTSP

Printer line spacing (parameter of DCB).

PS

Physical sequential (value of DSORG).

PSF

Printing Subsystem Facility or Perform subsystem function

PSU

Physical sequential unmovable (value of DSORG).

PSW

Program status word.

QISAM

Queued indexed sequential access method.

QSAM

Queued sequential access method.

R0

Record zero.

RACF

Resource Access Control Facility.

RACF authorization

The facility for checking a user's level of access to a resource against the user's desired level of access.

The result of that check.

RBA

Relative byte address.

RDBACK

Read backward (parameter of OPEN).

RDF

See *record definition field*.

RDW

Record descriptor word.

RECFM

Record format (JCL keyword and DCB macro parameter).

record definition field (RDF)

A field stored as part of a stored record segment; it contains the control information required to manage stored record segments within a control interval.

REFDD

Refer to previous DD statement (JCL keyword).

relative record data set (RRDS)

A VSAM data set whose records are loaded into fixed-length slots.

RETPD

Retention period (JCL keyword).

RKP

Relative key position (parameter of DCB).

RLS

Record-level sharing

RLSE

Release unused space (DD statement).

RMODE

Residence mode.

RPL

Request parameter list.

RPS

Rotational position sensing.

RRDS

Variable-length or fixed-length relative record data set. Also see *relative record data set*.

RRN

Relative record number.

SAA

Systems Application Architecture®.

SBCS

Single-byte character set.

SDW

Segment descriptor word.

SEOF

Software end-of-file.

SER

Volume serial number (value of VOLUME).

SETL

Set lower limit of sequential retrieval (QISAM macro).

SF

Sequential forward (parameter of READ or WRITE).

shared resources

A set of functions that permit the sharing of a pool of I/O-related control blocks, channel programs, and buffers among several VSAM data sets open at the same time.

SI

Shift in.

simple name

The rightmost component of a qualified name. For example, APPLE is the simple name in TREE.FRUIT.APPLE.

SK

Skip to a printer channel (parameter of CNTRL).

SL

IBM standard labels (value of LABEL).

SMB

System-Managed Buffering

SMS

Storage Management Subsystem, and system-managed storage.

SMSI

Size of main-storage area for highest-level index (parameter of DCB).

SMSW

Size of main-storage work area (parameter of DCB).

SO

Shift out.

source catalog

An existing catalog that may be exported into a target catalog.

SP

Space lines on a printer (parameter of CNTRL).

SRB

Service request block.

SS

Select stacker on card reader (parameter of CNTRL).

SSCR

Subsystem checkpoint record.

SSI

Subsystem interface.

STORCLAS

Storage class (JCL keyword).

SUL

IBM standard and user labels (value of LABEL).

SWA

Scheduler work area.

SYNAD

Synchronous error routine address (parameter of DCB, DCBE, and EXLST).

T

Track overflow option (value of RECFM); user-totaling(value of OPTCD).

task control block (TCB)

Holds control information related to a task.

TCB

See *task control block*, and trusted computing base.

TCDB

Tape configuration data base.

TIOT

Task input/output table.

TMP

Terminal monitor program.

TRC

Table reference character.

TRTCH

Track recording technique (parameter of DCB and of DD statement).

TTR

Track record address.

UCB

See *unit control block*.

UCS

See *universal character set*.

UHL

User header label.

unit control block (UCB)

A control block in storage that describes the characteristics of a particular I/O device on the operating system.

universal character set (UCS)

A printer feature that permits the use of a variety of character arrays.

UPD

Update.

USAR

User security authorization record.

USVR

User security verification routine.

UTL

User trailer label.

VIO

Virtual input output.

VIO storage group

A type of storage group that allocates data sets to paging storage, which simulates the activity of a DASD volume. VIO storage groups do not contain any actual DASD volumes.

VSAM

Virtual storage access method.

VSAM RLS

VSAM record-level sharing.

VSI

VSAM shared information.

VTOC

Volume table of contents.

VVDS

VSAM volume data set.

XLRI

Extended logical record interface.

Index

Numerics

13F completion code [50](#)
3400 Tape Drive [50](#)
3800 Printing Subsystem [18](#)
3890 Document Processor [17](#)
4092 completion code [53](#)
7-track tape [7](#), [47](#)

A

ABEND

13F [50](#), [56](#)
23F [12](#), [56](#)
33F [56](#)
43F [56](#)
53F [56](#)
D37 [8](#)

abend codes [24](#)

ABEND macro [24](#), [25](#)

abnormal termination

ABEND macro [25](#)

automatic checkpoint/restart [25](#)

BSAM (basic sequential access method) [43](#)

CANCEL operand [14](#), [18](#)

checkpoint entry [12](#), [52](#)

CKPTREST macro [2](#)

COND=EVEN, COND=ONLY [28](#)

data set (job step) [47](#)

data set (passed) [30](#), [32](#), [33](#)

data set (SYSOUT) [44](#)

data set (update) [49](#)

DD statement [9](#)

HOLD queue [24](#)

job (recording information) [1](#)

job step [23](#)

restart step [30](#)

SYSABEND, SYSUDUMP [43](#)

ACB (access method control block)

VSAM (virtual storage access method) [40](#)

accessibility

contact IBM [85](#)

ACS (automatic class selection) [27](#)

ACS routine [8](#)

address (checkid) [14](#)

ALESERV macro [22](#)

allocation

avoiding shared DASD [8](#)

dynamic

checkpoint [28](#)

checkpoint/restart [20](#)

concatenated data sets [20](#)

control information change [18](#)

generation data set [40](#)

nocapture option [21](#), [69](#)

space [32](#)

storage requirement [11](#)

allocation (*continued*)

dynamic (*continued*)

VIO data set [20](#)

space

restart [44](#)

secondary [8](#)

AMP parameter [45](#)

APF (authorized program facility) [3](#)

assistive technologies [85](#)

ATL (automated tape library)

Tape Library Dataserver [47](#)

ATTACH macro [17](#), [58](#)

automatic checkpoint/restart

canceling [18](#)

data set disposition [8](#), [28](#)

description [23](#)

DFSORT [53](#)

ensuring restart [12](#)

identification (primary, secondary) [14](#)

JES3 [52](#)

job journaling [2](#), [3](#)

job reinterpretation [28](#)

requesting (conditions) [25](#)

requesting (example) [25](#)

system description [27](#)

temporary data set [44](#)

type [23](#)

automatic restart

ADDRSPC=REAL [25](#)

JCL (requirements, restrictions) [27](#)

operator options [24](#)

RD parameter [35](#)

resource variations [27](#)

step termination [33](#)

system [27](#)

VIO data set [20](#)

virtual storage [34](#)

volume [27](#)

automatic step restart

abnormal termination [24](#)

data set disposition [26](#)

eligible codes [2](#)

establishing checkpoints [6](#)

EXTRACT macro [17](#)

initiating, AVR (automatic volume recognition) [26](#)

job journaling [2](#), [3](#)

MOD data set [24](#), [26](#)

PL/I [53](#)

RD parameter [1](#), [26](#)

restrictions [30](#)

SETPRT macro [18](#)

type [23](#)

AVR (automatic volume recognition) [26](#)

B

BDAM (basic direct access method)

- BDAM (basic direct access method) (*continued*)
 - data sets [39](#)
 - serially reusable resources [19](#)
- BLDVRP processing
 - VSAM restart [71](#)
- block count
 - DCB (negative) [60](#)
 - nonstandard tape labels [47](#)
 - reel tape
 - EXCP [49](#)
- BLP (bypass label processing) [47](#)
- BPAM (basic partitioned access method)
 - checkpoint considerations [5](#)
 - DCB parameter [7](#)
 - SYSCKEOV DD statement [7](#)
 - UNIT parameter [8](#)
- BSAM (basic sequential access method)
 - DCB parameter [7](#)
 - DD statement [6](#)
 - EOV (end-of-volume) [1](#), [83](#)
 - repositioning data set [43](#)
 - UNIT parameter [8](#)
 - using [5](#)
 - VIO data set [20](#)
- bypass label processing (BLP) [47](#), [60](#)

C

- CALL PLICKPT [53](#)
- CANCEL command [28](#)
- CANCEL operand [14](#), [18](#)
- capacity
 - checkpoint data set [9](#)
- card readers [43](#)
- cataloged procedure
 - data sets [32](#)
 - EXEC statement [35](#)
 - step [29](#)
- cataloged procedure step [35](#)
- CBIC option [61](#)
- CBUF (control block update facility) option [71](#)
- chained scheduling
 - DCB option [7](#), [8](#)
- channel programs
 - DCB option [8](#)
- CHECK macro [20](#), [40](#)
- checkid (checkpoint entry identification)
 - message at restart [24](#)
 - not specified [36](#)
 - PDS (partitioned data set) [11](#)
- checkpoint canceling [18](#)
- checkpoint data set
 - DCB (data control block) [7](#)
 - DD statement [8](#)
 - disposition [13](#)
 - extending [12](#)
 - opening, ensuring restart [12](#)
 - sequential [13](#)
 - space allocation [8](#)
 - storage estimates [9](#)
 - SYSCHK DD statement [2](#)
 - VIO data set [20](#)
- checkpoint entry [27](#)
- checkpoint entry, writing [11](#)

- checkpoint EOVS [5](#), [6](#)
- CHECKPOINT parameter [57](#)
- checkpoint routine [5](#), [44](#)
- checkpoint, taking [42](#)
- checkpoint/restart
 - data spaces [22](#)
 - description [1](#)
 - ensuring restart [12](#)
 - EOV (end-of-volume) [6](#)
 - job accounting [51](#)
 - messages [12](#)
- checkpoint/restart, automatic
 - accounting [51](#)
- checkpoint/restart, deferred
 - job accounting [51](#)
- CHKPT macro [1](#), [5-7](#), [11](#), [14](#), [16](#), [18](#), [27](#), [29](#), [31](#), [35](#), [40](#), [45](#), [50](#), [55](#), [57](#), [58](#), [61](#), [62](#), [69](#), [71](#), [83](#)
- CHKPT=EOVS parameter [5](#)
- CKPT parameter [53](#)
- CKPTREST macro [2](#)
- CLASS initialization statement [3](#)
- CLOSE macro [18](#), [45](#)
- CLPA (change link pack area)
 - nucleus table [66](#)
- COBOL RERUN clause [52](#)
- command
 - CANCEL [28](#)
 - DISPLAY [25](#), [34](#)
 - RELEASE [24](#), [28](#)
- completion code (13F) [50](#)
- completion code (4092) [53](#)
- completion codes
 - returned in register [55](#), [81](#)
- concatenated data set [6](#)
- COND parameter [31](#)
- conditional disposition [30](#), [32](#)
- contact
 - z/OS [85](#)
- COPY operation [43](#)
- CROPS parameter [45](#)
- cross memory support
 - restrictions [21](#)
- cross-memory
 - addressing [62](#)
 - mode [62](#)
- CSV (Contents Supervisor) [68](#)

D

- DASD (direct access storage device)
 - nonshared, shared (space allocation) [8](#)
- data conversion (with odd parity) [7](#)
- data set
 - checkpoint security [3](#)
 - DD statement (checkpoint) [27](#)
 - dummy [33](#)
 - generation [30](#)
 - open (RDBACK) [60](#)
 - passed [27](#)
 - password protection [73](#), [74](#)
 - positioning [48](#)
 - preallocated, temporary [44](#)
 - repositioning, direct access [26](#)
 - tape (reposition) [30](#)

- data set (*continued*)
 - temporary [28](#)
 - VIO (virtual I/O) [24](#), [52](#)
 - VSAM (virtual storage access method) [45](#), [46](#)
- data set (entry-sequenced) [46](#)
- data set (key-sequenced) [46](#)
- data set (password protection) [3](#)
- data set (relative record) [46](#)
- data set buffers [49](#)
- data set disposition [8](#), [26–29](#), [31](#)
- data set move [43](#)
- data set types (VSAM) [46](#)
- data set update
 - abnormal termination [49](#)
- data spaces [22](#)
- DCB (data control block)
 - exit list [83](#)
 - invalid, ABEND [8](#)
 - opened, passed to checkpoint [11](#)
 - restrictions [8](#)
- DCB macro [49](#)
- DCB parameter [59](#)
- DCBE (data control block extension) [8](#), [69](#)
- DD statement
 - checkpoint data set [25](#)
 - DATA delimiter [33](#)
 - requesting automatic checkpoint/restart [25](#)
- DDNADDR parameter [16](#)
- DDNAME
 - passed to checkpoint [12](#)
- DEB (data extend block) [64](#), [77](#)
- DEFER parameter [8](#)
- deferred
 - DD statement [29](#)
 - step restart [30](#)
 - step restart (requesting example) [29](#)
 - step restart (requirements, restrictions) [30](#)
 - step restart (resource variations) [31](#)
 - step restart (SYSCCHK DD statement) [36](#)
 - step restart (type) [23](#)
- deferred checkpoint/restart
 - ADDRSPC=REAL [34](#)
 - checkpoint entries [11](#), [13](#)
 - description [23](#)
 - dynamic concatenation [20](#)
 - JCL (requirements, restrictions) [32](#)
 - job step time limit [52](#)
 - messages [28](#)
 - nucleus [34](#)
 - operator considerations [29](#)
 - PL/I [53](#)
 - requesting [2](#), [31](#)
 - requesting (example) [32](#)
 - resource variations allowed [34](#)
 - submitting jobs [32](#)
 - SYSCCHK DD statement [2](#), [31](#), [36](#)
 - SYSIN data set [43](#)
 - system operation [34](#)
 - system restrictions [34](#)
 - type [23](#)
 - VIO data set [20](#)
 - volume serial number [34](#)
- DEFRAG operation [33](#), [43](#)
- delimiter (/*) statement [33](#)

- demount (tape volume DEQ) [20](#)
- DEQ macro [19](#), [20](#)
- device
 - allocation (failure) [50](#)
 - step restart, deferred [31](#)
- device allocation
 - direct access [44](#)
 - restart [25](#), [27](#)
 - restart, virtual storage [26](#)
 - UNIT parameter [50](#)
- device change
 - restart [27](#)
- DFSMSDss [33](#), [42](#), [43](#)
- DFSMSHsm [43](#)
- DFSORT program [53](#)
- direct access
 - device processing [5](#)
 - EOV (end-of-volume) [9](#)
 - volume [14](#)
 - VSAM data set [46](#)
- directory (preserving contents) [42](#)
- DISP operand [8](#)
- DISP parameter [53](#)
- DISPLAY command [25](#), [34](#)
- disposition, conditional [30](#), [32](#)
- DOM macro [58](#)
- DSAB (data set association block)
 - DD statement (restart) [67](#)
- DSCB (data set control block)
 - duplicate [26](#)
- DSNAME parameter [36](#)
- dummy data set [20](#), [33](#)
- DUMMY parameter [33](#)
- duplicate DSCB (data set control block) [26](#)
- dynamic
 - allocation
 - checkpoint [28](#)
 - checkpoint/restart [20](#)
 - data sets [21](#)
 - generation data set [40](#)
 - nocapture option [21](#), [69](#)
 - space [32](#)
 - VIO data set [20](#)
 - concatenated data sets [20](#)
 - control information change
 - SETPRT macro [18](#)
 - link pack area [56](#), [63](#)
 - storage requirement
 - checkpoint/restart [11](#)
 - unallocation
 - checkpoint [10](#)
 - conditional disposition [32](#)
 - data sets [30](#)
 - JCL specified data sets [33](#)

E

- ENQ macro [19](#), [20](#), [56](#)
- EOV (end-of-volume)
 - checkpoint routine, macro [83](#)
 - exit routine (CHKPT) [7](#)
 - SYSCKEOV DD statement [7](#)
- EROPT (error options) [50](#), [60](#)
- error options (EROPT) [50](#)

- errors (input/output) [50](#)
- ESDS (entry-sequenced data set) [46](#)
- ESETL macro [20](#)
- ESPIE macro [61](#), [81](#)
- ESTAE
 - recovery environment [74](#)
 - service request [80](#)
 - system request [55](#)
- EXCP access method [49](#)
- EXCP macro [5](#)
- EXEC statement
 - automatic checkpoint/restart [25](#)
 - operator considerations [25](#)
 - TIME parameter [52](#)
- extended format [39](#)
- extended format data set [8](#), [40](#)
- extents, data set [32](#)
- external checkpoint function [5](#)
- EXTRACT macro [17](#)

F

- FAILURE option [52](#)
- FCB (forms control buffer) [5](#), [18](#)
- FEOV macro [17](#)
- file protection [29](#), [48](#)
- FORCECP(days)
 - SMS-managed sequential data set [43](#)
- fragmented storage [11](#)

G

- GAM (graphics access method) [41](#)
- GDG (generation data group) [41](#)
- GDGNT (generation data group name table) [5](#), [40](#)
- GDS (generation data set)
 - cataloging [40](#)
 - JES3 restriction [52](#)
- GETMAIN failure [65](#)
- graphic data control block [57](#)
- GSR (global shared resources) [21](#)

H

- HFS (hierarchical file system)
 - extended format data set [8](#)
 - files [68](#)
- hiperbatch
 - QSAM (queued sequential access method) [81](#)
 - VSAM (virtual storage access method) [44](#)
- hiperspace buffer [21](#)
- HOLD reply [28](#)
- home address space ID [62](#)

I

- IAT2006 [52](#)
- IAT2575 [52](#)
- IDA3009I [70](#)
- IDADDR parameter [16](#)
- identification
 - primary, secondary [14](#)
- IDLNG parameter [16](#)

- IEBCOPY restart restriction [42](#)
- IEC068A [48](#)
- IEC161I [70](#), [71](#)
- IEC252I [70](#), [71](#)
- IEC252I [71](#)
- IEC331I [70](#)
- IEC400A [48](#)
- IEC401A [48](#)
- IEC402D [48](#)
- IEC954I [12](#)
- IEF007I [21](#)
- IEF225D [3](#), [23](#), [52](#)
- IHJ0001 [9](#), [12](#)
- IHJ000I [45](#)
- IHJ001I [81](#)
- IHJ006I [34](#)
- IHJ007I [81](#)
- IHJ008I [34](#)
- IHJ009I [70](#)
- IHJ014I [67](#)
- IHJ101I [75](#)
- IHJ102I [75](#)
- IMGLIB macro [17](#), [74](#)
- increment subparameter [8](#)
- INOUT option [42](#)
- input work queues [34](#)
- interface to VSAM (ISAM data set) [33](#)
- IPL (initial program load) [34](#)
- IPL (initial program load) [24](#), [29](#), [34](#)
- ISAM (indexed sequential access method) [33](#)
- ISO/ANSI (International Organization for Standardization/
American National Standards Institute) [8](#)
- ISO/ANSI standard labels [47](#)

J

- JCL (job control language)
 - requirements, restrictions [27](#), [32](#)
- JES2
 - job journaling [2](#), [35](#)
- JES3
 - GDS (generation data set) [44](#)
 - GDS restriction [52](#)
 - job journaling [3](#), [35](#), [52](#)
 - output spool space [44](#)
- JFCB (job file control block) [5](#), [20](#), [79](#)
- JFCBE (job file control block extension) [5](#), [18](#)
- job accounting [51](#)
- job canceling [24](#), [25](#)
- job failure
 - after automatic restart [26](#)
 - automatic restart [35](#)
 - causes [26](#)
 - during checkpoint entry [12](#)
 - job journaling [52](#)
 - saving checkpoint data (MOD) [7](#)
- job journaling
 - system restart [52](#)
- job reinterpretation [28](#)
- JOB statement
 - automatic checkpoint/restart [25](#)
- job step
 - abnormal termination [23](#), [24](#)
- JOBLIB DD statement [36](#)

K

keyboard
 navigation [85](#)
 PF keys [85](#)
 shortcut keys [85](#)
KSDS (key-sequenced data set) [46](#)

L

label
 bypass label processing (BLP) [60](#)
 ISO/ANSI (AL or AUL) [8](#)
 ISO/ANSI tape (data set) [47](#)
 LTM (leading tapemark option) [47](#)
 no label (NL) [8](#), [60](#)
 nonstandard (NSL) [8](#)
 nonstandard label (NSL) [66](#)
 nonstandard tape [47](#)
 serial numbers [67](#)
 standard [14](#)
 standard (checkpoint data set) [8](#)
 standard (SL, SUL) [8](#)
LABEL parameter [8](#), [59](#)
large format data set [41](#)
library
 automated tape [47](#)
link library [27](#)
linkage stack [81](#)
LINKLIB (data set) [27](#)
LPA (link pack area)
 list, DCB exit [83](#)
 nucleus routines/tables [66](#)
LSR (local shared resources) [21](#)
LTM (leading tapemark option)
 LABEL=(,LTM) [47](#)

M

macro
 ABEND [24](#), [25](#)
 ALESERV [22](#)
 ATTACH [17](#), [58](#)
 CHECK [20](#), [40](#)
 CHKPT [1](#), [5](#), [11](#), [14](#), [16](#), [18](#), [27](#), [29](#), [31](#), [35](#), [40](#), [45](#), [50](#),
 [55](#), [57](#), [58](#), [61](#), [62](#), [69](#), [71](#), [83](#)
 CKPTREST [2](#)
 CLOSE [18](#), [45](#)
 DCB [49](#)
 DEQ [19](#), [20](#)
 DOM [58](#)
 ENQ [19](#), [20](#), [56](#)
 ESETL [20](#)
 ESPIE [61](#), [81](#)
 EXCP [5](#)
 EXTRACT [17](#)
 FEOV [17](#)
 IMGLIB [17](#), [74](#)
 OPEN [8](#)
 PCLINK [17](#), [58](#)
 READ [49](#)
 RELEX [19](#), [40](#)
 RESERVE [19](#)

macro (*continued*)

 SETDEV [17](#)
 SETL [20](#)
 SETPRT [18](#)
 STIMER [18](#), [57](#)
 STOW [11](#), [42](#), [59](#), [66](#)
 WAIT [20](#), [40](#), [58](#)
 WRITE [7](#), [19](#), [40](#)
 WTOR [18](#), [58](#)
magnetic tape
 checkpoint entry [12](#)
 data set organization [53](#)
 devices [50](#)
 processing (checkpoint entry) [5](#), [8](#)
 restriction [36](#)
 standard label [8](#)
 volume [14](#)
 volumes [9](#), [20](#), [31](#), [36](#)
member
 checkpoint entry [11](#)
 deleting [42](#)
message
 abend (data set security) [24](#)
 authorization for restart [24](#)
 checkpoint [8](#)
 checkpoint restart [12](#)
 checkpoint return codes
 associated with IHJ Messages [81](#)
 EOV (end-of-volume) [6](#)
 error [55](#), [81](#)
 IAT2006 [52](#)
 IAT2575 [52](#)
 IDA3009I [70](#)
 IEC068A [48](#)
 IEC161I [70](#), [71](#)
 IEC252I [70](#), [71](#)
 IEC252I [71](#)
 IEC331I [70](#)
 IEC400A [48](#)
 IEC401A [48](#)
 IEC402D [48](#)
 IEC954I [12](#)
 IEF007I [21](#)
 IEF225D [3](#), [23](#), [52](#)
 IHJ000I [9](#), [12](#)
 IHJ000I [45](#)
 IHJ001I [81](#)
 IHJ006I [34](#)
 IHJ007I [81](#)
 IHJ008I [34](#)
 IHJ009I [70](#)
 IHJ014I [67](#)
 IHJ101I [75](#)
 IHJ102I [75](#)
 mount [29](#)
 replies [24](#)
 restart sequence [23](#)
 sequence [28](#)
MIGRATE operation [43](#)
MOD data set
 automatic restart [24](#)
 automatic step restart [26](#)
 checkpoint entries [11](#)
 example [12](#)

- MOD data set (*continued*)
 - PL/I programs [53](#)
 - restart, repositioning [27](#)
 - SYSCKEOV DD statement [7](#)
- modules, return codes
 - associated with IHJ Messages [81](#)
- MSS error
 - system [67](#)
- multivolume data set
 - JCL parameter [1](#)
 - nonspecific volume allocation [41](#)
 - physical sequential [1](#)
- MVS/SP
 - scheduler checkpoint support [80](#)

N

- name/token
 - restriction
 - pair [22](#)
- navigation
 - keyboard [85](#)
- no-reposition option [46](#)
- nocapture option [21](#), [69](#)
- nonpageable region [25](#)
- nonpageable virtual storage [34](#)
- nonspecific type volume [20](#), [24](#), [28](#)
- nonstandard labels [47](#)
- NORESTART statement type [2](#)
- NRC subparameter [46](#)
- NSLREPOS
 - nonstandard label (NSL) routine [66](#)
- nucleus
 - deferred checkpoint/restart [34](#)

O

- OPEN macro [8](#)
- OPEN routine [49](#)
- operator considerations [24](#), [29](#)
- operator message
 - automatic checkpoint/restart [23](#)
- OPTCD parameter [8](#)
- OUTIN option [42](#)

P

- PAGE pack [24](#)
- parameter
 - AMP [45](#)
 - CHECKPOINT [57](#)
 - CKPT [53](#)
 - COND [31](#)
 - CROPS [45](#)
 - DCB [59](#)
 - DEFER [8](#)
 - DISP [53](#)
 - DSNAME [36](#)
 - DUMMY [33](#)
 - LABEL [8](#), [59](#)
 - PGM [30](#), [31](#)
 - QNAME [65](#)
 - RD [1](#), [3](#), [26](#), [35](#), [55](#)

- parameter (*continued*)
 - RESTART [2](#), [3](#), [15](#), [29–32](#), [35](#)
 - RLSE [8](#)
 - SPACE [8](#)
 - YSOUT [43](#), [44](#)
 - TIME [52](#)
 - UNIT [8](#), [41](#), [47](#), [50](#), [52](#), [66](#)
 - UNSTACK [58](#)
 - VOL [20](#), [31](#), [33](#)
- PASN-AL (primary address space access list) [81](#)
- password
 - protected [23](#)
 - protection [73](#), [74](#)
 - restart (message) [23](#), [29](#)
- PC routine, restriction [22](#)
- PCLINK macro [17](#), [58](#)
- PDS (partitioned data set)
 - checkid (checkpoint entry identification) [14](#)
 - checkid length [15](#)
 - considerations (checkpoint/restart) [41](#)
 - directory (members) [42](#)
 - open for update [42](#)
 - restriction (SYSCHK DD statement) [36](#)
 - writing checkid [11](#)
- PDSE (partitioned data set extended)
 - checkpoint [42](#)
 - extended format data set [8](#)
- PGM parameter [30](#)
- PGSER (Page Fix) [71](#)
- PL/I program [53](#)
- PLIREST [53](#)
- PQUIT command [53](#)
- preallocated data set
 - temporary [44](#)
- primary
 - address space ID [62](#)
 - identification [14](#)
- primary address space access list (PASN-AL) [81](#)
- printer
 - setup (universal character set) [18](#)
- printer repositioning
 - at restart [49](#)
- processor time [51](#)
- program control assigned
 - checkpoint identification [14](#)
- programmer-specified
 - checkid (checkpoint entry identification) [1](#), [15](#)
- PSF (file protect status)
 - Tape Library Dataserver [67](#)
- PSF (performance subsystem function) [48](#)
- PSW key [81](#)

Q

- QNAME parameter [65](#)
- QSAM (queued sequential access method)
 - card reader [49](#)
 - checkpoint/restart [5](#)
 - CHKPT JCL parameter [6](#)
 - data set [49](#)
 - EOV (end-of-volume) [83](#)
 - I/O errors [50](#)
 - VIO data set [20](#)
- queue, input work [28](#)

- quick reference
 - checkpoint/restart codes [55](#)
 - checkpoint/restart processing [39](#)
 - coding a checkpoint [5](#)
 - introduction [1](#)
 - requesting restart [23](#)

R

- RACF (Resource Access Control Facility) [3](#)
- RD parameter [1](#), [3](#), [26](#), [35](#), [55](#)
- RDBACK (read backward)
 - BLP (bypass label processing) [60](#)
 - data set open [60](#)
- READ macro [49](#)
- reason codes
 - checkpoint/restart [56](#)
- record catalog [46](#)
- record update
 - after checkpoint [49](#)
- recording data set positioning [48](#)
- reduced performance [11](#)
- REGION size parameter [29](#)
- region, nonpageable [25](#)
- register contents
 - EOV (end-of-volume) [83](#)
 - return/reason codes [55](#)
- RELEASE command
 - initiate restart [24](#)
- RELEX macro [19](#), [40](#)
- repositioning
 - data set (VSAM) [46](#)
 - data sets (VSAM) [44](#)
 - deferred step restart [30](#)
 - password protected [23](#)
- repositioning card reader [49](#)
- repositioning restriction [21](#)
- RERUN clause [52](#)
- RESERVE macro [19](#)
- resource variations
 - deferred step restart [31](#)
- restart
 - EOV (end-of-volume) [5](#)
 - responses [24](#)
 - unauthorized programs [3](#)
 - VSAM data sets [46](#)
- restart card reader [49](#)
- restart message [3](#)
- RESTART parameter [2](#), [3](#), [15](#), [23](#), [29–32](#), [35](#)
- restart request [1](#)
- RESTART statement
 - JES3 [52](#)
 - RACF protected data sets [3](#)
 - SCHEDxx
 - PLIREST [53](#)
 - SVC [72](#)
 - SYS1.PARMLIB
 - SCHEDxx [2](#)
 - SYS1.PARMLIB member
 - SCHEDxx [25](#), [26](#)
- restart types [23](#)
- restart, automatic
 - data set disposition [8](#)
 - description [23](#)

- restart, automatic (*continued*)
 - job step accounting [51](#)
- restart, deferred
 - messages [28](#)
- restart, ensuring [12](#)
- RESTART/NORESTART [2](#), [25](#), [26](#)
- return codes
 - add, delete [2](#)
 - checkpoint/restart [55](#), [57–81](#)
 - CHKPT [16](#)
- RLS (record level sharing) [45](#), [61](#)
- RLSE parameter [8](#), [33](#)
- routine (checkpoint) [5](#), [8](#)
- RPL feedback word [45](#), [70](#)
- RPS (rotational position sensing) [50](#)
- RRDS (relative record data set) [46](#)

S

- secondary
 - address space ID [62](#)
 - addressing mode [62](#)
 - allocation [59](#)
 - identification
 - checkid (checkpoint entry identification) [14](#)
 - space [6](#)
 - space allocation [8](#)
- security
 - checkpoint data set [3](#)
 - user data set [3](#)
- sequential data set
 - checkpoint/restart [12](#)
- serially reusable resources [19](#)
- SETDEV macro [17](#)
- SETL macro [20](#)
- SETLOCK system request [55](#), [72](#)
- SETPRT macro [18](#)
- shared resources [21](#)
- shortcut keys [85](#)
- SHOWCB macro [45](#)
- SMS environment [36](#)
- SMS-managed
 - data set [8](#), [27](#), [36](#)
 - sequential data sets [39](#), [43](#)
- SPACE parameter [8](#)
- space release
 - deferred checkpoint/restart [33](#)
- SPOOL pack [24](#)
- SSCR (subsystem checkpoint record)
 - DCB (restart) [67](#)
- step restart, automatic
 - accounting [51](#)
 - JES3 [52](#)
 - resource variations [27](#)
 - suppressed (JES3) [44](#)
- step restart, deferred
 - checkpoint entries [11](#), [13](#)
 - description [23](#)
 - DFSORT [53](#)
 - GDS (generation data set) [40](#)
 - JCL requirements [30](#)
 - job and job step accounting [51](#)
 - requesting [2](#)
 - VIO data set [20](#)

- STIMER macro [18](#), [57](#)
- storage estimating [11](#)
- STOW macro [11](#), [42](#), [59](#), [66](#)
- subparameter
 - NRC [46](#)
- subpools [11](#)
- subtasks [17](#)
- summary of changes [xv](#)
- suppressing action [1](#)
- SWA (scheduler work area)
 - control block [59](#), [62](#)
 - manager [59](#), [63](#)
 - pageable storage [2](#)
 - read a JFCB [79](#)
- SYNCDEV macro [59](#)
- SYS1.IMAGELIB data set [18](#)
- SYS1.PARMLIB member [2](#)
- SYSABEND [39](#)
- SYSABEND data set [43](#)
- SYSCHK DD statement [2](#), [31](#), [32](#), [36](#)
- SYSCKEOV DD statement [2](#), [7](#)
- SYSIN [39](#)
- SYSIN data set
 - automatic checkpoint/restart [43](#)
- SYSOUT [39](#)
- SYSOUT data set
 - checkpoint/restart [44](#)
- SYSOUT parameter [43](#), [44](#)
- SYSRES (system residence volume) [27](#)
- SYSRES volume [24](#)
- system failure
 - automatic restart [35](#)
 - during checkpoint entry [12](#)
 - job journaling [52](#)
 - saving checkpoint data (MOD) [7](#)
- system generated
 - checkid (checkpoint entry identification) [1](#)
- system generated name (data set) [28](#)
- system operation [27](#), [31](#)
- SYSUDUMP data set [43](#)

T

- tape
 - 7-track [7](#)
 - cartridge [48](#)
 - cartridge protection [48](#)
 - data set (ISO/ANSI label) [47](#)
 - DCB (data control block) parameters [7](#)
 - files (VSE) [47](#)
 - label conflict [48](#)
 - labels (data set) [8](#)
 - reel (data set) [49](#)
 - volume [14](#), [20](#), [67](#)
 - volume (DEQ) [20](#)
 - volume, EOVS (end-of-volume) [9](#)
- Tape Library Dataserver
 - normal mount [23](#)
 - operator message [28](#)
- TCAM (telecommunications access method) [39](#)
- TCB (task control block) [17](#)
- TCBUSER field [51](#)
- TCDB (tape configuration data base)
 - Tape Library Dataserver [68](#)

- termination
 - automatic restart [9](#)
 - completed at system restart [52](#)
 - records reprocessed [50](#)
- time limit
 - job step [52](#)
- TIME parameter [52](#)
- trademarks [91](#)
- TTR (track record address) [26](#)

U

- UCS (universal character set)
 - SETPRT macro [18](#)
- unallocation
 - dynamic
 - checkpoint [10](#)
 - conditional disposition [32](#)
 - data sets [30](#)
 - JCL specified data sets [33](#)
- unavailable virtual storage [34](#)
- UNIT keyword [36](#)
- UNIT parameter [8](#), [41](#), [47](#), [50](#), [52](#), [66](#)
- unit record
 - data sets [43](#)
 - devices [17](#), [49](#)
- UNIX system services [68](#)
- UNLOAD command [24](#)
- unmovable attribute [43](#)
- UNSTACK parameter [58](#)
- UPDAT option [42](#)
- update in place [49](#)
- user
 - data set security [3](#)
- user data sets [5](#)
- user errors [51](#)
- user interface
 - ISPF [85](#)
 - TSO/E [85](#)

V

- variable spanned record format
 - data set [42](#)
- VARY command [24](#)
- vector facility [81](#)
- VIO (virtual I/O)
 - data set allocation [20](#)
 - data sets
 - allocation, cancel [24](#)
- virtual fetch [53](#)
- virtual storage
 - buffers (data set records) [49](#)
 - CHKPT macro [1](#), [14](#)
 - contents written [1](#), [5](#)
 - EOVS (end-of-volume) [83](#)
 - job journaling [2](#)
 - operator considerations [29](#)
 - requirements [23](#), [25](#), [29](#)
 - writing, DASD [8](#)
- VOL parameter [20](#), [31](#), [33](#)
- volume
 - allocation [27](#)

- volume (*continued*)
 - automatic restart [27](#)
 - changing at restart [24](#)
 - configuration [34](#)
 - deferred step [31](#)
 - direct access [5](#), [14](#), [53](#)
 - mount (VSAM) [28](#)
 - serial (restart) [30](#)
- VSAM (virtual storage access method)
 - completion codes [45](#), [46](#)
 - data set [46](#)
 - data set repositioning [44](#)
 - ESDS (entry-sequenced data set) [46](#)
 - hiperbatch [45](#)
 - I/O errors [50](#)
 - KSDS (key-sequenced data set) [46](#)
 - RPL feedback word [70](#)
 - RRDS (relative record data set) [46](#)
 - serially reusable resources [19](#)
 - volume mount [28](#)
- VSE (virtual storage extended)
 - checkpoints [47](#)
- VSM (virtual storage manager) [62](#)
- VTAM
 - CHKPT (ACB) [72](#)

W

- WAIT macro [20](#), [40](#), [58](#)
- WRITE macro [7](#), [19](#), [40](#)
- write validity check [7](#)
- WTOR macro [18](#), [58](#)



Product Number: 5655-ZOS

SC23-6862-70

