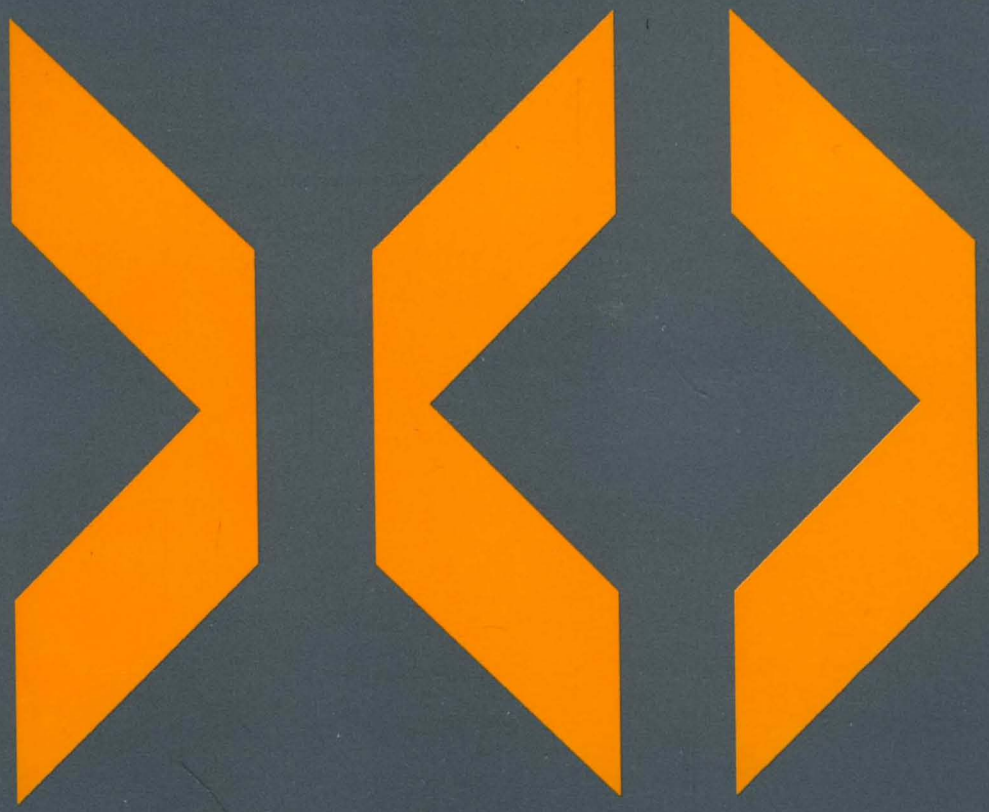
GC35-0033-13

IBM

Device Support Facilities

User's Guide and Reference

Release 10



Device Support Facilities

GC35-0033-13

User's Guide and Reference

Release 10

Twelfth Edition (July 1988)

This edition replaces and makes obsolete the previous edition, GC35-0033-12.

This edition applies to Device Support Facilities, Release 10.

The changes for this edition are summarized under "Summary of Changes" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent publication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are made periodically to this publication; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead.

Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. If you request publications from the address given below, your order will be delayed because publications are not stocked there.

A Reader's Comment Form is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publishing, P. O. Box 49023, San Jose, California, U.S.A. 95161-9023. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This publication describes the Device Support Facilities processor and commands. It also discusses hardware considerations, operating system considerations, and planning and installation. It is written for installation managers, system programmers, system operators, and other data processing personnel who manage or use direct access storage devices (DASD).

Wherever an IBM operating system is discussed, it is assumed readers understand the concepts and terms used in that operating system.

introduction to maintaining IBM 3380

For an introduction to maintaining IBM 3380 disk storage using Device Support Facilities see *Device Support Facilities: A Primer for the User of IBM 3380 Direct Access Storage*, GC26-4498. This primer includes the following:

- Considerations on handling data checks
- Description of some of the functions of the ANALYZE, INITialize, INSPECT, and INSTALL commands
- Using Device Support Facilities for various tasks such as installation, HDA replacement, etc.
- General guidelines on when and how to perform media maintenance and the use of the INITialize, INSPECT, and INSTALL commands for performing media maintenance.

if you are experiencing problems

This publication is a reference for Device Support Facilities; however, if you are experiencing device errors or media maintenance problems, you should first refer to *Storage Subsystem Library: Maintaining IBM Disk Storage Media* to determine the correct procedure.

Don't let the size of the book deter you. There is much duplication in the interest of giving you information where you need it rather than referring you to a common section. For example, "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71 and "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173 describe the ICKDSF commands. However, Part 2 *only* describes the commands used by count-key-data (CKD) devices and Part 3 *only* describes the commands used by fixed block architecture (FBA) devices. Also Part 2, Part 3, the Appendixes (which include *all* the messages), the Glossary, and Index are all reference information.

Just as a picture is worth a thousand words, an example can clearly and concisely illustrate what may take pages to explain. For this reason, numerous examples have been included in the book. Both "Examples of ICKDSF Tasks—CKD Devices" on page 37 and "Examples of ICKDSF Tasks—FBA Devices" on page 55 consist *entirely* of examples. Also, after the description of each command there are examples of that command

To assist you in finding information, see the table of contents, the figure list, and the index.

Organization of This Publication

For convenience, information on Device Support Facilities is presented under three major headings followed by the appendixes:

- "Part 1. General Information" on page 1 describes Device Support Facilities and how it is used by the system programmer and the system operator. This section includes:

  "Overview" on page 3 describes the Device Support Facilities commands and lists the devices which are supported. It includes an introduction to using Device Support Facilities in stand-alone mode or under an operating system, and installation information.

  "Using Device Support Facilities (ICKDSF)" on page 11 describes how to use Device Support Facilities in a stand-alone or an operating system environment.

  "Controlling Command Execution" on page 25 describes how to control execution of Device Support Facilities commands using IF-THEN-ELSE statements.

  "Volume and Data Set Security" on page 31 describes volume and data set security when you use Device Support Facilities.

  "Examples of ICKDSF Tasks—CKD Devices" on page 37 includes examples of Device Support Facilities tasks using count-key data devices.

  "Examples of ICKDSF Tasks—FBA Devices" on page 55 includes examples of Device Support Facilities tasks using fixed block architecture devices.

  "Syntax of the Device Support Facilities Commands" on page 67 describes the syntax of the Device Support Facilities commands.

- "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71, describes the function and syntax of the Device Support Facilities commands for CKD devices. This part includes examples and detailed reference information for each command.

- "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173, describes the function and syntax of the Device Support Facilities commands for FBA devices. This part includes examples and detailed reference information for each command.

- There are the following appendixes:

  Appendix A, "Volume Layout and Record Formats on Count-Key-Data (CKD) Devices" on page 221, contains descriptions and figures showing volume layout and record formats on CKD devices.

  Appendix B, "VM Support" on page 227, describes support for the minidisk under VM.

  Appendix C, "Device Support Facilities Messages (ICK)" on page 233, lists and explains Device Support Facilities messages.

  Appendix D, "Problem Determination Tables for MVS" on page 299, contains problem determination information.

  Appendix E, "The VTOC Index" on page 305, describes the VTOC index, and gives suggestions for calculating the size of the index data set.

Appendix F, "The Fixed Block Architecture VTOC (FBAVTOC)" on page 311, describes the fixed block architecture VTOC (FBAVTOC) and gives suggestions for calculating the FBAVTOC space requirements.

Appendix G, "Surface Checking" on page 315, describes the device surface analysis performed by Device Support Facilities.

Appendix H, "User Security Exit Module" on page 319, describes the user security exit module, which tailors the Device Support Facilities security function.

- "Glossary of Terms and Abbreviations" on page 323
- Index

Related Publications

Other books that may be helpful to you are:

- *Device Support Facilities: A Primer for the User of IBM 3380 Direct Access Storage*, GC26-4498, is an introduction to maintaining IBM 3380 disk storage using Device Support Facilities.
- *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, describes the publications that are applicable and current for MVS and VSE operating systems.
- MVS/XA Publications
  - *MVS/Extended Architecture System—Data Administration*, GC26-4149, inludes information on indexed VTOCs.
  - *MVS/XA Message Library: System Messages*, GC28-1376 and GC26-1377
  - *MVS/Extended Architecture Data Administration: Utilities*, GC26-4018, describes the utility programs available in the MVS/XA environment.
  - *Interactive Storage Management Facility (ISMF) User's Guide*, GC26-4266, describes how to use ISMF under MVS/XA.
  - *MVS/Storage Management Library: Managing Storage Pools*, SC26-4407, contains more information concerning using volumes under the Storage Management Subsystem (SMS)
- MVS/370 Publications
  - *MVS/370 System Programming Library: Data Management*, GC26-4056.
  - *MVS/370 Message Library: System Messages*, GC28-1374
  - *OS/VS2 MVS Resource Access Control Facility (RACF) General Information Manual*, GC28-0722, describes the functions and capabilities of access control.
  - *OS/VS2 System Programming Library: Supervisor Services and Macro Instructions*, GC28-0683, describes the authorized program facility and service management (SRB dispatching).
  - *OS/VS2 MVS Utilities*, GC26-3902, describes the functions of the OS/VS2 MVS utility programs and the control statements used with each.
  - *MVS/370 Utilities*, GC26-4065, describes the utility programs available in the MVS/370 environment.

- VM Publications
  - *IBM Virtual Machine Facility/370 Planning and System Generation Guide*, GC20-1801
  - *IBM Virtual Machine/System Product High Performance Option Planning and Reference*, SC19-6223
  - *CMS Macros and Function Reference*, SC24-5284
  - *CP Command Reference for General Users*, SC19-6211
  - *VM/SP System Facilities for Programming*, SC24-5288
- VSE Publications
  - *VSE/System Package, Messages and Codes*, SC33-6181
  - *VSE/Advanced Functions, System Utilities*, SC33-6100 describes the VSE/Advanced Functions System Utilities including format emulated extent utility (INITEM).
- Device Support Publications
  - *Component Summary—2835 Storage Control, 2305 Fixed Head Storage*, GA26-1589, describes the 2305 fixed head storage module.
  - *IBM System 360—Component Descriptions, 2314 Direct Access Storage Facility and 2844 Auxiliary Storage Control*, GA26-3599, describes the 2314 direct access storage device.
  - *IBM 2319 Disk Storage—A Series, Component Description*, GA26-1606, describes the 2319 direct access storage device.
  - *Component Description—2841 Storage Control, 2311 Disk Storage*, GA26-5988, describes the 2311 disk storage device.
  - *IBM 3310 Direct Access Storage Reference Manual*, GA26-1660, describes the 3310 fixed block storage device.
  - *Reference Manual for IBM 3330 Series Disk Storage*, GA26-1615, describes the 3330 direct access storage device.
  - *Reference Manual for IBM 3340 3344 Disk Storage*, GA26-1619, describes the 3340 and 3344 direct access storage devices.
  - *Reference Manual for IBM 3350 Direct Access Storage*, GA26-1638, describes the 3350 direct access storage device.
  - *IBM 3370 Direct Access Storage Description*, GA26-1657, describes the 3370 storage system and its fixed block data format.
  - *IBM 3375 Direct Access Storage Description and User's Guide*, GA26-1666, describes the 3375 direct access storage device.
  - IBM 3380 Publications
    - *IBM 3380 Direct Access Storage: Introduction*, GC26-4491
    - *IBM 3380 Direct Access Storage Reference Summary*, GX26-1678
    - *Using IBM 3380 Direct Access Storage in an MVS Environment*, GC26-4492
    - *Using IBM 3380 Direct Access Storage in a VM Environment*, GC26-4493,
    - *Using IBM 3380 Direct Access Storage in a VSE Environment*, GC26-4494,
    - *Maintaining IBM Disk Storage Media*, GC26-4495
    - *Storage Subsystem Library Master Index*, GC26-4496
  - *IBM 3850 Mass Storage System (MSS) Introduction and Preinstallation Planning*, GA32-0038, introduces the concepts of the Mass Storage System.
  - *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS*, GC35-0014, describes commands used with the Mass Storage System.

- *IBM 9332 Disk Unit: Planning*, GA21-9532, helps you to plan for the 9332 fixed block storage device.
- *IBM 9335 Direct-Access Storage Subsystem: Functional Characteristics*, SA33-3143, describes the 9335 fixed block storage device.
- *IBM 4321 and 4331 Processor Compatibility Features*, GA33-1528, describes the Direct Access Storage Compatibility Feature and emulation restrictions.
- *IBM 4361 Processor Compatibility Feature*, GA33-1569, describes the Direct Access Storage Compatibility Feature and emulation restrictions.

Summary of Changes

Release 10, July 1988

New Support

ICKDSF now has new operating system support. In Release 10 you can execute ICKDSF under the CMS operating system of VM. This new support allows you to use ICKDSF to execute functions you would perform in the stand-alone environment today, as well as to process minidisks under CMS. See Appendix B, "VM Support" on page 227 for more information on the new CMS and current minidisk support. Each ICKDSF command has a new heading in its introduction which describes the new support under CMS for that command.

A new command, INSTALL, has been added to ICKDSF. INSTALL performs the procedures neccessary for installation, head-disk assembly (HDA) replacement, and physical movement of IBM 3380 DASD.

A new command, REVALidate, has been added to ICKDSF. REVALidate combines the track validation functions of medial initialization with the problem determination and data verification functions of the ANALYZE command and also the INSPECT functions, if required.

You can now initialize a volume which is to be managed by the Storage Management Subsystem (SMS) by specifying the STORAGEGROUP parameter of the INIT command. Volumes which have been previously SMS managed, can be initialized as SMS volumes again or can be initialized as non-SMS-managed volumes.

You can now use the Interactive Storage Management Facility (ISMF) to enter the INIT command and invoke ICKDSF. See the ISMF help panels, "Using Device Support Facilities in an ISMF Environment" on page 20 in this book, and *Interactive Storage Management Facility (ISMF) User's Guide*, GC26-4266 for more information on using this facility.

The IBM 9313 is now supported. See "Command Changes" for the commands which support this new device.

A new section, "Programming Interfaces for Customers" on page 10, has been added. This section contains the programming interfaces provided by ICKDSF by which a customer-written program is to request or receive functions or services of ICKDSF.

System Support

The IBM 9313 is supported in the CMS and stand-alone environment only.

The CONTROL command with the ALLOWWRITE parameter is supported in the CMS (with ATTACHed devices) and stand-alone environment.

ICKDSF can now be executed under CMS. This allows you to process minidisks using ICKDSF in the CMS environment.

Compatibility

Current minidisk ICKDSF jobs which run in the stand-alone environment only and use the MIMIC(MINI) parameter are still valid. However, with the new minidisk support under the CMS operating system, you can now run these jobs under CMS without having to re-ipl CMS at job completion.

Existing ICKDSF jobs will run under the new support as previously. However, you should check the impact of the new IBM 9313 FBA support to your ICKDSF jobs.

- For the IBM 9313
  - Any INIT command jobs which specify the CHECK, RECLAIM, MAP, or CONTINUE parameters are not valid.
  - Any INSPECT jobs which specify CHECK, BLOCKRANGE, or MAP are not valid.
  - Any REFORMAT jobs run as coded now
  - FBA Commands not supported for the IBM 9313
    - ANALYZE
    - CONTROL
    - MAPALT

Command Changes

ICKDSF now executes under control of CMS. Under CMS you can process minidisks as follows:

- If you do not have DEVMAINT authority you can use ICKDSF commands on your own minidisk
- If you have DEVMAINT authority you can use ICKDSF commands on another user's minidisk.

Under CMS the following commands are valid if you do not have DEVMAINT authority:

- ANALYZE
- INIT
- REFORMAT

Under CMS the following commands are valid if you have DEVMAINT authority:

- ANALYZE
- INIT
- INSPECT
- REFORMAT

See Appendix B, "VM Support" on page 227 for more information on the new CMS and current minidisk support. Also, each ICKDSF command has a new heading in its introduction which describes the CMS support for that command.

A new command, INSTALL, has been added to ICKDSF. This command allows you to perform a combination of equivalent ICKDSF functions which includes a drive test, validation of the home address/record zero records, data verification of the factory functional verification data patterns (FFVDP), and surface checking. The INSTALL command is valid for the IBM 3380 *only*.

The default VTOC size for CKD devices has been changed. If you do not specify the VTOC size, (and the device is not a minidisk) the default VTOC size is now the number of tracks per cylinder minus one.

The UNITADDRESS parameter for all commands has been extended to accept a 3 or 4 digit unit address in all system environments

For the IBM 9313 the following commands are supported:

- INIT
- INSPECT
- REFORMAT
  - INIT 9313 support

    Minimal INIT only is supported. This consists of formatting the VOL labels and the VTOC.

    VM minidisk support is minimal INIT and consists of formatting the VOL labels and the VTOC.

    The INIT parameters, CHECK, RECLAIM, BLOCKRANGE, MAP, and CONTINUE are not valid for the IBM 9313.
  - INSPECT 9313 support

    INSPECT 9313 support consists of allowing you to unconditionally assign alternate blocks.

    The INSPECT parameters CHECK, BLOCKRANGE, and MAP are not supported for IBM 9313
  - REFORMAT 9313 support

    REFORMAT 9313 support consists of being able to change the volume label of previously initialized volumes and minidisks.

For the IBM 9313 the following commands are not supported:

- ANALYZE
- CONTROL
- MAPALT

Other Changes

Updates have been added to Appendix C, "Device Support Facilities Messages (ICK)" on page 233.

Service changes have been added throughout the book.

Release 9.0

New Support

The IBM 3380 Models AJ4, AK4, BJ4, BK4, and CJ2 are now supported by the ANALYZE, BUILDIX, CONTROL, INIT, INSPECT, and REFORMAT commands.

The IBM 3990 Storage Control Models 1, 2, and 3 is now supported.

System Support

The IBM 3380 Models AJ4, AK4, BJ4, BK4, and CJ2 are supported in the VSE, MVS/370, MVS/XA, and stand-alone environments.

The IBM 3990 Storage Control Models 1, 2, and 3 are supported in the MVS/370, MVS/XA, and stand-alone environments. The VSE environment is supported by the IBM 3990 Storage Control Models 1 and 2.

Compatibility

Existing ICKDSF jobs will run under the IBM 3380 Models AJ4, AK4, BJ4, BK4, and CJ2 support with the following exception:

- The IBM 3375 and 3380 are not supported by the maximal INIT function (INIT command with CHECK, SKIP, or RECLAIM).

Command Changes

For IBM 3380 Models AJ4, AK4, BJ4, BK4, and CJ2 support, the following commands are changed:

- The IBM 3375 and 3380 are not supported by the maximal INIT function (INIT command with CHECK, SKIP or RECLAIM).
- A new parameter has been added to the INIT command. The DATA/NODATA parameters allow you to specify whether the factory functional verification data patterns are to be written during the validate process for the IBM 3375 and IBM 3380 devices.
- For DASD devices attached to an IBM 3990 Storage Control a new parameter, CLEARFENCE, has been added to the CONTROL command. The CLEARFENCE parameter of the CONTROL command allows you to clear a fence status.

You can now specify the volume table of contents in VSE by using either the DOSVTOC or VSEVTOC parameter.

The defaults for the SKIP/NOSKIP parameters of the INSPECT command have been updated. See the SKIP/NOSKIP parameters of the INSPECT (CKD) command for more information.

Other Changes

Before running the ANALYZE data verification test or a medial initialization, all IBM 3380 Models must be powered on with all devices in the ready condition and all covers closed for 2 hours 30 minutes to establish thermal stability.

The ANALYZE command has been enhanced to provide a more accurate reflection of the condition of the device in the event of a suspected drive problem. This data is provided for use by the service representative when assistance is required.

Service changes have been added throughout the book.

Updates have been added to Appendix C, "Device Support Facilities Messages (ICK)" on page 233.

Contents

Figures

Part 1. General Information

Overview

Device Support Facilities (ICKDSF) is a program you can use to perform various operations on direct-access storage devices. It can be used to perform service functions on a direct-access storage device, as well as error detection and media maintenance.

With ICKDSF, you can:

- Initialize a direct-access storage volume so that it can be used in an MVS or a VSE system.
- Inspect a volume for defective tracks.
- Modify the volume label and IPL bootstrap and program records.
- Examine a device with a nonremovable storage mechanism to determine if there are problems with the drive.
- Determine if there is a problem with reading or writing data stored on the volume.

Device Support Facilities supports a stand-alone and the following operating system versions:

- MVS/370.
- MVS/Extended Architecture (MVS/XA).
- VSE/SP.
- The stand-alone version of Device Support Facilities runs under an IBM System/370 as follows:
  - The IBM 308X, 309X, and 438X are supported in Extended Architecture mode.
  - The IBM 3033, 308X, 309X, 4331, 4341, 438X, and 9370 are supported in 370 mode.
- VM
  - You can execute ICKDSF under the CMS operating system of VM/SP, VM/SP HPO, and VM/XA 370 mode.

Abbreviations used for operating systems in this book:

- The term "MVS," as used in this book, includes the following systems: MVS/370 and MVS/XA.
- The term "VSE," as used in this book, refers to VSE/SP systems.
- The term "VM" as used in this book, includes VM/SP, VM/SP HPO, and VM/XA.

introduction to maintaining IBM 3380

For an introduction to maintaining IBM 3380 disk storage using Device Support Facilities, see *Device Support Facilities: A Primer for the User of IBM 3380 Direct Access Storage*, GC26-4498. This primer includes the following:

- Considerations on handling data checks
- Description of some of the functions of the ANALYZE, INITialize, INSPECT, and INSTALL commands
- Using Device Support Facilities for various tasks such as installation, HDA replacement, etc.
- General guidelines on when and how to perform media maintenance and the use of the INITialize, INSPECT, and INSTALL commands for performing media maintenance.

if you are experiencing problems

This publication is a reference for Device Support Facilities; however, if you are experiencing device errors or media maintenance problems, you should first refer to *Storage Subsystem Library: Maintaining IBM Disk Storage Media* to determine the correct procedure.

Things You Can Do with ICKDSF

Device Support Facilities performs two basic functions: volume formatting and disk surface maintenance.

- Volume formatting makes a volume usable by an operating system and provides a means of updating old volume formats. This is also known as "initializing" a volume.

  Newly installed volumes and replaced head disk assemblies (HDAs) will always require formatting.

- Disk surface maintenance aids in detecting problems with device operation and attempts to correct data errors that can be traced specifically to disk surface defects.

Device Support Facilities Commands

Device Support Facilities operates on both count-key-data (CKD) and fixed block architecture (FBA) devices. The commands, syntax, and functions are common across all versions of ICKDSF if the operating system supports the device and/or the function. The ICKDSF commands are:

- ANALYZE
- BUILDIX
- CONTROL
- INIT
- INSPECT
- INSTALL
- MAPALT
- REFORMAT
- REVALidate

The following is a description of each command:

- ANALYZE helps to detect and differentiate recording surface and drive-related problems on a volume. It can also scan data to aid in detecting the existence of possible media problems.
- BUILDIX changes a standard MVS-format VTOC into an indexed VTOC, or an indexed VTOC into a standard MVS-format VTOC.
- CONTROL command:
  - The FBA version of the CONTROL command with the RECLAIM(SYSAREA) parameter, allows you to reclaim primary and alternate blocks associated with the System Reserved Area of the IBM 9335 devices.
  - The CKD version of the CONTROL command with the ALLOWWRITE parameter allows subsequent writes to the devices controlled by an IBM 3880 or IBM 3990 Storage Control if that storage control has been WRITE INHIBITed by the error recovery procedures at the channel, director, or storage control level.

    The CKD version of the CONTROL command with the CLEARFENCE parameter (IBM 3990 only)allows you to clear a fence status of a path and/or a device that has been fenced off by the subsystem.
- INIT prepares a direct-access storage volume so that it can be used in an operating system environment or it reinitializes a volume that was previously prepared.
- INSPECT provides surface checking functions for blocks or tracks on a subset of a volume.
- INSTALL performs the procedures neccessary for installation, head-disk assembly (HDA) replacement, and physical movement of IBM 3380 DASD.
- MAPALT produces a report showing the assignment of primary blocks to alternate blocks on fixed block architecture devices.
- REFORMAT can be used to update or replace information on a previously initialized volume. For CKD devices, the volume label, owner ID, IPL bootstrap, and IPL program records can be updated. For FBA devices, the volume label and owner ID can be replaced.
- REVALidate combines the track validation functions of medial initialization with the problem determination and data verification functions of the ANALYZE command and also the INSPECT functions, if required.

Each command is described in:

- "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
- "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

Each command description is followed by examples, and further examples are presented in:
- "Examples of ICKDSF Tasks—CKD Devices" on page 37
- "Examples of ICKDSF Tasks—FBA Devices" on page 55

Figure 1 lists the ICKDSF commands and indicates the operating systems each command supports.

Figure 2 on page 7 lists the commands and indicates the devices each command supports.

In both figures, an X denotes support and a hyphen (-) denotes nonsupport.

	ANALYZE	BUILDIX	CONTROL	INIT	INSP	INSTALL	MAPALT	RFMT	REVALidate
MVS/370	X	X	X	X	X	X	-	X	X
MVS/XA	X	X	X	X	X	X	-	X	X
VSE	X	-	X	X	X	X	X	X	X
CMS(minidisk)	X	-	-	X	X	-	-	X	-
CMS(ATTACH'ed)	X	-	X	X	X	X	X	X	X
Stand-alone	X	-	X	X	X	X	X	X	X

Figure 1. Operating Systems Supported by Device Support Facilities Commands

Figure 2 lists the Device Support Facilities commands and indicates the devices each command supports. An X denotes support and a hyphen (-) denotes non-support.

CKD Devices	ANALYZE	BUILDIX	CONTROL	INIT	INSP	INSTALL	MAPALT	RFMT	REVALidate
2305-1,2	X	X	-	X	X	-	-	X	-
2311	X	-	-	X	X	-	-	X	-
2314	X	-	-	X	X	-	-	X	-
2319	X	-	-	X	X	-	-	X	-
3330-1,2,11	X	X	X	X	X	-	-	X	-
3333-1,2,11	X	X	X	X	X	-	-	X	-
3340	X	X	X	X	X	-	-	X	-
3344	X	X	X	X	X	-	-	X	-
3350	X	X	X	X	X	-	-	X	-
3375	X	X	X	X	X	-	-	X	-
3380 A04, B04, AA4	X	X	X	X	X	X	_	X	X
3380 AD4, BD4, AE4, BE4	X	X	X	X	X	X	_	X	X
3380 AJ4, AK4, BJ4, BK4, CJ2	X	X	X	X	X	X	_	X	X
FBA Devices									
3310	X	-	-	X	X	-	X	X	-
3370-1,2	X	-	-	X	X	-	X	X	-
9313	-	-	-	X	X	-	-	X	-
9332	-	-	-	X	X	-	-	X	-
9335	X	-	X	X	X	-	X	X	-

Figure 2. Devices Supported by Device Support Facilities Commands

- The stand-alone version of ICKDSF supports all the above devices. The support under the operating system versions is limited to the devices supported by each operating system. (For example, the MVS version does not support the IBM 3370.)
- For DASD operating as a paging subsystem, all Device Support Facilities commands must be directed to the base address.
- Device Support Facilities automatically bypasses caching functions in all channel programs requiring direct knowledge of DASD operations, if caching functions exist.

Devices You Can Use

Device Support Facilities supports the following direct access storage devices (DASDs) in various environments:

IBM Count-Key-Data (CKD) Devices

- 2305 Fixed Head Storage Models 1 and 2
- 2311 Disk Storage Model 1
- 2314 Direct Access Storage Facility
- 2319 Disk Storage Models A1, A2, and A3
- 3330 Disk Storage Models 1, 2, and 11
- 3333 Disk Storage and Control Models 1 and 11
- 3340 Direct Access Storage Facility
- 3344 Direct Access Storage
- 3350 Direct Access Storage
- 3375 Direct Access Storage
- 3380 Direct Access Storage Models A04, B04, and AA4
- 3380 Direct Access Storage Models AD4, BD4, AE4, and BE4
- 3380 Direct Access Storage Models AJ4, AK4, BJ4, BK4, and CJ2

IBM Fixed Block Architecture (FBA) Devices

- 3310 Direct Access Storage
- 3370 Direct Access Storage Models A1, A2, B1, and B2
- 9313 Direct Access Storage
- 9332 Direct Access Storage
- 9335 Direct Access Storage

How to Use ICKDSF

You can use Device Support Facilities in an operating system environment or in a stand-alone environment. ICKDSF is controlled by the use of commands (and their associated parameters) presented to it as part of the input stream. Briefly, to use Device Support Facilities you must do the following:

- In a stand-alone environment, you must IPL ICKDSF from a tape (or logical card reader on a VM system).

  See "How to IPL ICKDSF from a Stand-alone Tape" on page 15.

  ICKDSF commands are then entered from the input device.
- In an operating system environment, you execute Device Support Facilities using JCL (MVS) or job control statements (VSE).

See

- "Using Device Support Facilities in an MVS Environment" on page 18
- "Using Device Support Facilities in a VSE Environment" on page 21.

ICKDSF commands are then entered as part of the SYSIN data for MVS or part of the system data stream for VSE.

Also, in the MVS/XA environment, you can use the Interactive Storage Management Facility (ISMF) to schedule ICKDSF jobs. See "Using Device Support Facilities in an ISMF Environment" on page 20 for more information.

- In the CMS operating system environment, you execute ICKDSF under the control of CMS.

  See

  - "Using Device Support Facilities in a CMS Environment" on page 22

  You can then enter and execute the ICKDSF commands from the input device or from a CMS file.

- To specify the Device Support Facilities commands to perform the function you require see

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

  You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands are presented in

  - "Examples of ICKDSF Tasks—CKD Devices" on page 37
  - "Examples of ICKDSF Tasks—FBA Devices" on page 55

  Further examples are presented after each command in

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

- If required, you can control Device Support Facilities command execution using IF-THEN-ELSE statements.

  See "Controlling Command Execution" on page 25.

Installation Information

Instructions explaining how to install Device Support Facilities are contained in the program directory shipped with the machine-readable material.

Storage Requirements

Device Support Facilities requires the following estimated amounts of storage space:

MVS/370, and MVS/XA
Virtual storage: 512K bytes

VSE
Minimum partition size of 512K bytes

Stand-alone
512K bytes

CMS
512K byte user storage

Programming Interfaces for Customers

Programming interfaces for ICKDSF are those interfaces provided by ICKDSF by which a customer-written program is to request or receive functions or services of ICKDSF.

This section identifies a list of all programming interfaces provided by ICKDSF for use by customers.

No information other than that identified here should be construed as defining a programming interface

Product-Sensitive Programming Interfaces

Installation exits and other "product-sensitive" interfaces are provided to allow the customer installation to perform tasks such as product tailoring, monitoring, modification, or diagnosis. They are dependent on the detailed design or implementation of ICKDSF. Such interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new ICKDSF releases or versions, or as a result of maintenance. The product-sensitive interfaces provided by ICKDSF are as follows:

Section or Heading	Pages
Appendix H, "User Security Exit Module"	319–320

Using Device Support Facilities (ICKDSF)

To use Device Support Facilities, you execute a command processor. Each command is processed completely before processing of the next command begins.

This section describes how to use Device Support Facilities:

- In the stand-alone environment
- In an MVS environment (MVS/370, MVS/XA)
- In a VSE environment
- In a CMS environment
- In an Interactive Storage Management Facility (ISMF) environment

Using Device Support Facilities in a Stand-alone Environment

This section describes how to use Device Support Facilities in a stand-alone environment.

The stand-alone version of Device Support Facilities runs under an IBM System/370 in 370 mode or Extended Architecture mode. Under VM, each DASD device must be either ATTACHed to the VM virtual machine or LINKed as a minidisk. See Appendix B, "VM Support" on page 227.

To use Device Support Facilities in a stand-alone environment, you must do the following:

- You create a stand-alone IPL tape under MVS, CMS or VSE. For VM, you can use an already existing stand-alone tape, or you can use the CMS file "IPL DSF, IPL ICKDSF, or ICKSADSF COREIMAG" and punch it to your logical card reader.

  See

  - "Creating an ICKDSF Stand-alone IPL Tape using MVS" on page 12
  - "Creating an ICKDSF Stand-alone System Under CMS" on page 12
  - "Preparing to Run ICKDSF Stand-alone using VM" on page 13
  - "Creating an ICKDSF Stand-alone IPL Tape Using VSE" on page 15

- Ipl ICKDSF using the tape created in the last step.

  See "How to IPL ICKDSF from a Stand-alone Tape" on page 15.

- To specify the Device Support Facilities commands to perform the function you require, see

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

  You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands are presented in

  - "Examples of ICKDSF Tasks—CKD Devices" on page 37
  - "Examples of ICKDSF Tasks—FBA Devices" on page 55

Further examples are presented after each command in

- "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
- "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

- If required, you can control Device Support Facilities command execution using IF-THEN-ELSE statements.

  See "Controlling Command Execution" on page 25.

Creating an ICKDSF Stand-alone IPL Tape using MVS

For MVS, the stand-alone code is found in SYS1.SAMPLIB as ICKSADSF. The Device Support Facilities program is usually loaded from a file on tape, but can be loaded from a binary-card deck.

The following is sample MVS JCL that can be used to copy the stand-alone Device Support Facilities code to an unlabelled tape. There should only be one file on the tape. Note that BLKSIZE=80 is required on the tape data set.

```
//JOBNAME  JOB  JOB CARD PARAMETERS
//STEPNAME EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   DUMMY,DCB=BLKSIZE=80
//SYSUT1   DD   DSNAME=SYS1.SAMPLIB(ICKSADSF),UNIT=3330,
//              DISP=SHR,VOLUME=SER=XXXXXX,
//SYSUT2   DD   DSNAME=ICKDSF,UNIT=2400-3,LABEL=(,NL),
//              DISP=(,KEEP),VOLUME=SER=YYYYYY,
//              DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
```

Creating an ICKDSF Stand-alone System Under CMS

For CMS, the stand-alone code is found in the CMS file ICKSADSF COREIMAG. This stand-alone version of ICKDSF can be executed by punching the ICKSADSF COREIMAG to a virtual card reader and IPLing the file from the virtual reader. Alternatively, the stand-alone ICKDSF can also be copied to a tape and IPLed from the tape.

Creating the ICKDSF Stand-alone Tape under CMS

Figure 3 is a sample EXEC to copy the stand-alone program code to an unlabeled tape.

```
*THIS EXEC COPIES THE STAND-ALONE DEVICE SUPPORT FACILITIES
*FROM CMS FILE 'ICKSADSF COREIMAG' TO AN IPLABLE TAPE.
*THE TAPE MUST BE ATTACHED AS VIRTUAL ADDRESS '181'.
CP REWIND 181
&IF &RETCODE NE 0 &GOTO -NOTAPE
FILEDEF INMOVE DISK ICKSADSF COREIMAG E (LRECL 80 RECFM F
FILEDEF OUTMOVE TAP1 (LRECL 80 RECFM F BLOCK 80
MOVEFILE
&IF &RETCODE NE 0 &GOTO -BADMOVE
&BEGTYPE
*** STAND-ALONE ICKDSF SUCCESSFULLY GENERATED ***
&END
&EXIT 0
-BADMOVE
&TYPE ERROR LOADING CMS FILE TO TAPE
&EXIT 10
-NOTAPE
&TYPE TAPE 181 NOT ATTACHED OR NOT READY
&EXIT 20
```

Figure 3. Sample CMS EXEC to copy stand-alone ICKDSF to tape

Preparing to Run ICKDSF Stand-alone using VM

In the VM environment, the stand-alone code is available as a loadable file on a tape.[1] That file can be converted to card format and Device Support Facilities can be loaded from a virtual card reader as follows:

1. Read the file from the tape into your virtual machine as fname ftype.[1]
2. Spool your reader to *.
3. Punch fname ftype (noh

   This punches the file to your virtual card reader without any header information.
4. IPL your virtual card reader as you would the tape, and continue as specified below.

See Figure 4 on page 14 for a sample EXEC to invoke Device Support Facilities from your virtual card reader. This EXEC assumes that stand-alone Device Support Facilities has been copied into a CMS file named 'IPL DSF, IPL ICKDSF, or ICKSADSF COREIMAG'.

[1] The stand-alone version of Device Support Facilities may already be available as CMS file "IPL DSF, ICKSADSF COREIMAG, or IPL ICKDSF" on a system disk.

```
*
* THIS IS A SAMPLE EXEC TO INVOKE DEVICE SUPPORT FACILITIES
*
*
* IT IS INVOKED USING THE FOLLOWING COMMAND:
*
*      ICKDSF <FILENAME FILETYPE FILEMODE>
*
* WHERE THE FILE IDENTIFIER IS OPTIONAL.  IF SPECIFIED, THE FILE
* CONTAINS DEVICE SUPPORT FACILITIES CONTROL STATEMENTS THAT WILL
* BE AVAILABLE AS INPUT TO DEVICE SUPPORT FACILITIES.
*
* THIS EXEC WILL PUNCH A COPY OF THE STAND-ALONE DEVICE SUPPORT
* FACILITIES TO THE READER FOLLOWED BY THE DEVICE SUPPORT FACILITIES
* CONTROL STATEMENTS, IF SPECIFIED.  IT WILL THEN IPL THE READER TO
* INVOKE STAND-ALONE DEVICE SUPPORT FACILITIES.
*
* YOU MUST THEN IDENTIFY THE CONSOLE BY CAUSING AN ATTENTION INTERRUPT.
* ON A 3270 TYPE DISPLAY, THIS IS DONE BY PRESSING THE ENTER KEY.
*
* DEVICE SUPPORT FACILITIES RESPONDS WITH THE MESSAGE:
*
*   ICK005E DEFINE INPUT DEVICE 'DDDD,CUU' OR 'CONSOLE'
*
* IF YOU SPECIFIED A FILE WITH DEVICE SUPPORT FACILITIES CONTROL
* STATEMENTS, YOU ENTER:
*
*   2540,00C
*
* IF YOU WISH TO ENTER STATEMENTS DIRECTLY AT THE TERMINAL, YOU ENTER:
*
*   CONSOLE   (OR IF YOU ENTER A NULL LINE, CONSOLE IS INPUT DEVICE)
*
* DEVICE SUPPORT FACILITIES RESPONDS WITH THE MESSAGE:
*
*   ICK006E DEFINE OUTPUT DEVICE 'DDDD,CUU' OR 'CONSOLE'
*
* YOU MAY RESPOND 'CONSOLE' OR ENTER A NULL LINE
* TO SEE THE OUTPUT ON YOUR TERMINAL OR
* '1403,00E' TO SPOOL THE OUTPUT TO YOUR VIRTUAL PRINTER.
*
* WARNING:  THIS EXEC WILL RESET YOUR SPOOLED PUNCH TO CLASS A
*           NOHOLD NOCONT OFF, AND YOUR SPOOLED READER TO CLASS *.
*           THIS EXEC REQUIRES THAT NO CLASS I FILES EXIST IN YOUR
*           READER.
*
&TYPE **** SPOOLING ICKDSF TO YOUR READER
CP SPOOL PUNCH TO * NOHOLD CLASS I CONT
PUNCH IPL ICKDSF * (NOH)
&IF &RETCODE NE 0 &GOTO -ERROR
&IF &INDEX GT 0 PUNCH &1 &2 &3 (NOH)
&IF &RETCODE NE 0 &GOTO -ERROR
CP SPOOL PUNCH NOCONT CLOSE
CP SPOOL PUNCH OFF CLASS A
CP CLOSE READER
CP ORDER READER CLASS I
CP SPOOL READER NOCONT CLASS * NOHOLD
&TYPE **** IPLING ICKDSF
&TYPE **** WHEN ICKDSF IS FINISHED, RE-IPL CMS
CP IPL 00C CLEAR
-ERROR &R = &RETCODE
CP SPOOL PUNCH NOCONT PURGE
CP SPOOL PUNCH OFF CLASS A
&EXIT &R
```

Figure 4. Sample CMS EXEC to invoke Device Support Facilities

Creating an ICKDSF Stand-alone IPL Tape Using VSE

In the VSE/AF 2.1 environment, the stand-alone version of ICKDSF is automatically retrieved from the IJSYSRES.SYSLIB library. If the stand-alone version of ICKDSF (DSFSA.Z) is installed in any other sub-library, it must be copied to the IJSYSRES.SYSLIB library before a stand-alone tape is created.

```
// JOB BUILD A STAND-ALONE TAPE ON VSE/AF21
// ASSGN SYS006,280  OUTPUT TAPE
// MTC REW,SYS006
// EXEC LIBR
   BACKUP RESTORE=STANDALONE TAPE=SYS006 LIB=IJSYSRS
/*
/&

IF RESTORE = STANDALONE LIB= MUST BE SPECIFIED
```

How to IPL ICKDSF from a Stand-alone Tape

Because the procedures for loading a program are not the same for all models of processing units, you must load the Device Support Facilities program from the tape unit according to the procedures for your processing unit. The program contains the instructions necessary to load and store itself in the lower addresses of main storage. When the program has completely loaded, the processor enters the WAIT state. When the processor enters the WAIT state, the program status word (PSW) should contain the following:

```
* FF060000 XXFFFFFF (BC mode) or * 030E0000 XXFFFFFF (EC mode)
```

* An asterisk before the PSW indicates the high order bit may or may not be on (either X'80' or X'00') in the byte designated by 'XX'.

See "Error Conditions" on page 17 for other wait PSW conditions.

At this time, you continue by identifying the console, the input device, and the output device.

Identifying the Console

You identify the console to the program by pressing the REQUEST or ENTER key at the console. You must use one of the following consoles:

- 1052 Model 7 Printer-Keyboard
- 3036 Graphics Console
- 3056 Graphics Console
- 3138, 3148, 3158 Consoles in typewriter keyboard mode
- 3205 Console (supported as a 3278/3279)
- 3210 Models 1 and 2 Console Printer-Keyboard
- 3215 Model 1 Console Printer-Keyboard
- 3277 Graphics Console
- 3278 Graphics Console
- 3279 Graphics Console

Note: Remote devices are not supported.

Identifying the Input Device

After you have identified the console, message ICK005E prompts you to specify the input device and its address. (See Appendix C, "Device Support Facilities Messages (ICK)" on page 233 for the full message text.) You must use one of the following input devices:

- 1052 Model 7 Printer-Keyboard
- 1442 Model N1 Card Read Punch, Model N2 Card Punch
- 2501 Models B1 and B2 Card Reader
- 2520 Model B1 Card Read Punch, Models B2 and B3 Card Punch
- 2540 Model 1 Card Read Punch
- 2560 Multifunction Card Machine
- 3210 Models 1 and 2 Console Printer-Keyboard
- 3215 Model 1 Console Printer-Keyboard
- 3504 Models A1 and A2 Card Reader
- 3505 Models B1 and B2 Card Reader
- 3525 Card Punch (with card-read feature)
- 5424 Card Punch (80 columns only)
- 5425 Multifunction Card Unit (80 columns only)

Note: Remote devices are not supported.

You can specify the word CARD or one of the device types listed above (for example, "CARD,ccuu" or "5424,ccuu"). You may also specify the console as the input device by entering CONSOLE or a null line in response to message ICK005E.

When running under VM, if the stand-alone code is loaded from the virtual card reader, and the same card reader is specified as the input device, the Device Support Facilities commands should be in the same logical file as the stand-alone code. See Figure 4 on page 14.

If the command statements are in a separate file, stand-alone Device Support Facilities encounters an end-of-file condition. Cause an attention interrupt by pressing the ENTER key. This causes the stand-alone Device Support Facilities to restart. You will be prompted for the input device and processing will continue normally.

Identifying the Output Device

After you respond to message ICK005E, message ICK006E prompts you to specify the output device and its address. (See Appendix C, "Device Support Facilities Messages (ICK)" on page 233 for the full message text.) You must use one of the following output devices:

- 1052 Model 7 Printer-Keyboard
- 1403 Models 2, 7, and N1 Printer
- 1443 Model N1 Printer
- 3203 Models 1, 2, 4, and 5 Printer
- 3210 Models 1 and 2 Console Printer-Keyboard
- 3211 Model 1 Printer
- 3215 Model 1 Console Printer-Keyboard
- 3800 Printing Subsystem
- 3262 Line Printer
- 3289 Model 4 Printer
- 5203 Model 3 Printer

Note: Remote devices are not supported.

You can specify the word PRNT or one of the devices listed above (for example, "PRNT,ccuu" or "3800,ccuu"). You may also specify the console as the output device by entering CONSOLE or a null line in response to message ICK006E.

Specifying the Date and Time of Day

The date and time of day are automatically picked up from the TOD (time of day) clock of the processing unit. Note that the TOD clock would typically indicate Greenwich Mean Time, and this would be reflected in any listings.

In the event that the TOD clock is in error or not set, message ICK015E prompts you to enter the date. Then, message ICK016E prompts you to enter the time of day. In response to these messages, you can enter any string of up to 8 characters. These will appear in the title line of the output. On the other hand, you can press the ENTER key without specifying a date or time of day. In this case, the time of day and date are set to zeros. Note that these messages can appear again if a VTOC entry has to be created for a data set. It is necessary to furnish the information each time it is requested if the VTOC entry is to be correct.

Error Conditions

The stand-alone version of Device Support Facilities requires 512K bytes of real storage to execute successfully. In the event that the real storage available for Device Support Facilities is not sufficient, all the input is read and an error condition is posted in the PSW. This error condition is:

```
* 00020000 XX000033 (BC mode) * 000A0000 XX000033 (EC mode)
```

* In the above and following PSWs, an asterisk before the PSW indicates the high order bit may or may not be on (either X'80' or X'00') in the byte designated by 'XX'.

Other IPL error WAIT-state PSW codes are:

- * 00020000 XX000022 (BC mode) or * 000A0000 XX000022 (EC mode): Indicates that a unit check occurred on the IPL device during the IPL.
- * 00020000 XX000044 (BC mode) or * 000A0000 XX000044 (EC mode): Indicates that a channel error occurred on the IPL channel during the IPL
- * 00020000 XX000055 (BC mode) or * 000A0000 XX000055 (EC mode): Indicates that the IPL device became "not operational" during the IPL process

The PSW is also used to indicate the cause of errors that may occur during Device Support Facilities processing in a stand-alone environment. The processor is placed in a WAIT state and the PSW contains the following:

```
* FF060000 XXEEEEnn (BC mode) or * 030E0000 XXEEEEnn (EC mode)
```

Values of *nn* indicate the type of error:

- X'13' indicates an SVC interrupt
- X'14' indicates a program interrupt
- X'15' indicates insufficient main storage
- X'16' indicates an I/O error
- X'17' indicates data set not open
- X'18' indicates console not defined
- X'19' indicates no end-of-data routine

Notes:

1. A PSW code of * "FE020000 XX111111" (BC mode) or * "020A0000 XX111111" (EC mode) is loaded by Device Support Facilities whenever it is waiting for an I/O interrupt. If Device Support Facilities stops running, with this PSW code, a missing I/O interrupt condition has occurred. If this happens, re-IPL Device Support Facilities and rerun the failing command. If the problem persists, the cause of the missing interrupt should be investigated.
2. A PSW code of * "00020000 XX0000E2" (BC mode) or * "000A0000 XX0000E2" (EC mode) is loaded by Device Support Facilities whenever a machine check has occurred.

Program Termination

Execution of the Device Support Facilities program terminates in a stand-alone environment when the end of the command input stream is encountered. If the console was specified as the input device, you enter the characters EOF in response to a request for the next command. When the execution terminates, the processor is placed in a WAIT state, and the PSW should contain the following:

```
* FF060000 XXEEEEzz (BC mode) or * 030E0000 XXEEEEzz (EC mode)
```

where *zz* indicates the highest decimal condition code encountered during execution as shown by the hexadecimal values that appear next. See "Condition Codes to Control Command Execution" on page 25 for an explanation of condition codes.

- X'00' indicates 0
- X'04' indicates 4
- X'08' indicates 8
- X'0C' indicates 12
- X'10' indicates 16

Using Device Support Facilities in an MVS Environment

In the MVS environment, you can use Device Support Facilities under one of the following MVS operating systems:

- MVS/370
- MVS/XA

To use Device Support Facilities in the MVS environment, you execute ICKDSF as a job or job step as follows:

- Execute Device Support Facilities using JCL.

  See "Sample MVS JCL" on page 19.

- Specify the Device Support Facilities commands to perform the function you require. See:

  — "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71

  You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands are presented in

— "Examples of ICKDSF Tasks—CKD Devices" on page 37

Further examples are presented after each command in

— "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71

- If required, you can control Device Support Facilities command execution using IF-THEN-ELSE statements.

  See "Controlling Command Execution" on page 25.

Sample MVS JCL

To execute Device Support Facilities in the MVS environment, you invoke the ICKDSF program when you specify PGM=ICKDSF in the EXEC statement as follows:

```
//JOBNAME  JOB   YOUR INSTALLATION'S JOB-ACCOUNTING DATA
//JOBCAT   DD    DSNAME=YOUR.CATALOG,DISP=SHR
//STEP1    EXEC  PGM=ICKDSF
//MYVOL    DD    UNIT=(devicetype),DISP=OLD,VOL=SER=VOL123
//STEPCAT  DD    DSNAME=ANOTHER.CATALOG,DISP=SHR
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
  Device Support Facilities commands
/*
```

Sample MVS JCL Explanation

1. //JOBNAME is **required**. The JOB statement describes your job to the MVS system. Procedures at your installation may require you to supply user identification, accounting, and authorization information with the JOB statement's parameters.
2. //JOBCAT is optional. The JOBCAT DD statement identifies a user catalog that can be used by each of the job's steps. If your job uses only the master catalog, do not specify the JOBCAT DD statement since the master catalog is always open and available to all the jobs on the system.
3. //STEP1, an EXEC statement, is **required**. The EXEC statement invokes the Device Support Facilities processor to process the Device Support Facilities commands.
4. //STEPCAT is optional. The STEPCAT DD statement identifies a user catalog that can be used when processing the job step. If user catalogs are identified with JOBCAT and STEPCAT DD statements, only the catalog identified with the STEPCAT DD statement and the master catalog are used with the job step. If your job step uses only the master catalog, do not specify the STEPCAT DD statement, because the master catalog is always open and available to all the jobs on the system.
5. //SYSPRINT is **required**. The SYSPRINT DD statement identifies the output data set to which Device Support Facilities sends messages and output information.

6. //SYSIN is **required**. The SYSIN DD statement identifies the source of the input statements. An input statement to Device Support Facilities is a command. When you code SYSIN DD *, you identify the statements that follow as input.
7. The last input statement is followed by /* in the first two columns of the next input record.

Printed Output Data Set

The standard printed output data set is SYSPRINT. The default parameters for this data set are:

- Record format of VBA
- Logical-record length of 125 (121 + 4)
- Block size of 629 (5 x (121 + 4) + 4)

The first byte of each record is the American National Standards Institute (ANSI) control character. The minimum specifiable logical-record length (LRECL) is 121 (U-format records only). If a smaller size is specified, it will be overridden to 121.

You can alter the above defaults by specifying the desired values in the DCB parameter of the SYSPRINT statement. However, the record format cannot be specified as F or FB. If you specify either of these, it will be changed to VBA.

Using Device Support Facilities in an ISMF Environment

In the MVS/XA environment, you can execute ICKDSF as a job or job step as shown previously in "Using Device Support Facilities in an MVS Environment" on page 18. Under MVS/XA, you can also use Device Support Facilities under the MVS/Extended Architecture Interactive Storage Management Facility (ISMF), a component of the MVS/XA Data Facility Product (DFP), to build job streams for ICKDSF command functions using menu-driven panels.

The information you supply on ISMF panels is used to build and submit job streams like those you generate using JCL and ICKDSF commands. Using ISMF panels, though, you no longer have to remember ICKDSF keywords and syntax. You merely fill in the values you want on the panels and ISMF generates the job stream. You can then either submit the job or save the jobstream for later use.

Using ISMF panels, you can perform the following commands:

- INIT prepares a direct-access storage volume so that it can be used in an operating system environment or it reinitializes a volume that was previously prepared.

How to Invoke ICKDSF under ISMF

To use ICKDSF under ISMF requires you to do the following:

- Log on to TSO
- Select ISPF from TSO menu
- Select ISMF from ISPF menu
- Select the volume application from the ISMF primary menu
- Build a volume list from the volume selection entry panel

- Specify the required ICKDSF command against the volume
- Fill in the ICKDSF command panel(s) to perform the required function.

For more information on ICKDSF commands, see "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71. For more information on ISMF, see the ISMF online help panels, or *Integrated Storage Management Facility (ISMF) User's Guide*. *MVS/XA: Storage Administration Reference* contains more information on the tasks you can perform under ISMF.

Using Device Support Facilities in a VSE Environment

To use Device Support Facilities in the VSE environment, you execute ICKDSF as a job as follows:

- Execute Device Support Facilities as a job, using system control statements.

  See "Sample VSE System Control Statements"

- Specify the Device Support Facilities commands to perform the function you require. See:

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

  You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands are presented in

  - "Examples of ICKDSF Tasks—CKD Devices" on page 37
  - "Examples of ICKDSF Tasks—FBA Devices" on page 55

  Further examples are presented after each command in

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

- If required, you can control Device Support Facilities command execution using IF-THEN-ELSE statements.

  See "Controlling Command Execution" on page 25.

Sample VSE System Control Statements

To execute Device Support Facilities in the VSE environment, you invoke the Device Support Facilities program by specifying EXEC ICKDSF in the EXEC statement as follows:

```
// JOB      jobname   YOUR INSTALLATION'S JOB-ACCOUNTING DATA
// ASSGN    SYS010,353
// EXEC     ICKDSF,SIZE=AUTO
   Device Support Facilities commands
/*
/&
```

Sample VSE System Control Statements Explanation

1. // JOB is an optional statement used to describe your job to the VSE system.
2. // ASSGN causes the SYSNAME SYS010 to be assigned to the device whose channel and unit address is 353.
3. /* appears in the first two columns of the input record following the last input statement.
4. /& appears in the first two columns of the input record following the /* card to indicate end of job.

Printed Output

All printed output is directed to SYSLST.

Using Device Support Facilities in a CMS Environment

To use ICKDSF in the CMS environment, you must do the following:

- Invoke ICKDSF under the CMS operating system
- Identify the input and output device(s)
- Specify the ICKDSF commands to perform the function you require.

  While you are using ICKDSF under CMS, type "HELP ICKDSF" to invoke the online help panels.

Use ICKDSF under CMS as follows:

- To invoke ICKDSF when you are under control of the CMS operating system, type 'ICKDSF'
- After you have invoked ICKDSF, define the input device when you receive the following message:

  'ICK030E DEFINE INPUT DEVICE: FN FT FM, "CONSOLE", OR "READER"'

  - If you want to key in the ICKDSF commands and parameters at your console, type 'CONSOLE' or a null line (ENTER)
  - If you want to execute ICKDSF from a CMS file which contains the ICKDSF commands and parameters you want to execute, you type the file name, file type, and file mode (FN FT FM).
    - The logical record size of the file must be 132 or less.
    - The default for file type is 'INPUT'
    - The default for file mode is 'A'
  - If you want to execute ICKDSF commands and parameters from a CMS file which is in your virtual reader, type 'READER'
  - To invoke the online help panels type "?"

    Note: The first reader file is used when you specify 'READER'
- After you have defined the input device, you define the output device when you receive the message:

  'ICK031E DEFINE OUTPUT DEVICE: FN FT FM, "CONSOLE" OR "PRINTER"'

  - If you wish to receive the output at your console, type 'CONSOLE' or a null line (ENTER)
  - If you want your output stored in a CMS file, type the file name, file type, and file mode (FN FT FM).

  - If the file already exists, you receive a message and reply 'U' to allow the file to be erased.
  - The default for file type is 'OUTPUT'
  - The default for file mode is 'A'
- Type 'PRINTER' to route the output to your virtual printer.
- To invoke the online help panels type "?"

- If you specified "CONSOLE" as your input device, you receive the following message which allows you to start to enter and execute ICKDSF commands. The time (hh/mm/ss) and date (mm/dd/yy) in the message is supplied by the system:

```
ICKDSF - CMS DEVICE SUPPORT FACILITIES 10
Time: hh/mm/ss Date: mm/dd/yy  PAGE 1
ENTER INPUT COMMAND:
```

  you can now specify the Device Support Facilities commands to perform the function you require. These can be entered at your console, or as a CMS file stored on disk, or as a file in your virtual reader. For more information on ICKDSF commands, you can type "?" to view the online help panels or see:

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

  You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands are presented in

  - "Examples of ICKDSF Tasks—CKD Devices" on page 37
  - "Examples of ICKDSF Tasks—FBA Devices" on page 55

  Further examples are presented after each command in

  - "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
  - "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

- If required, you can control Device Support Facilities command execution using IF-THEN-ELSE statements.

  See "Controlling Command Execution" on page 25.

- Device Support Facilities is terminated when you type 'END' at your console, or the END-OF-FILE condition is detected from the CMS file or your virtual reader file

Invoking Device Support Facilities under CMS from a CMS EXEC

Device Support Facilities can be initialized and executed under CMS using an EXEC which contains the required statements to invoke ICKDSF, identify the input and output devices, and the ICKDSF commands and parameters for the function you want to perform.

Figure 5 is a sample CMS EXEC (using REXX) which invokes Device Support Facilities, identifies the console as the input device, and the printer as the output device. The ICKDSF statements are included which perform a minimal INIT followed by an ANALYZE SCAN. The 'PUSH 'U'' statement replies to ICKDSF message ICK003D which is received when the INIT is started. You must supply the UNITADRESS (ccuu) for both commands and also the VOLID (xxxxxx) for the INIT command.

```
/* Sample CMS exec to invoke and execute an ICKDSF INIT and ANALYZE*/
ADDRESS 'COMMAND'
PUSH 'END'
PUSH 'ANALYZE UNIT(ccuu) SCAN'
PUSH 'U'
PUSH 'INIT UNIT(ccuu) NVFY VOLID(xxxxxx)'
PUSH 'PRINTER'
PUSH 'CONSOLE'
ICKDSF
```

Figure 5. Sample CMS EXEC to invoke and execute an ICKDSF INIT and ANALYZE

Figure 6 is another sample CMS EXEC (using REXX) which invokes Device Support Facilities, uses a CMS file as the input device, and a CMS file as the output device. The ICKDSF statements are included which perform a minimum INIT. The 'PUSH 'U'' statement responds to ICKDSF message ICK003D which is received when the INIT is started. The CMS file 'ABC INPUT A' contains the ICKDSF command statement:

```
INIT UNIT(ccuu) NVFY VOLID(xxxxxx)
```

You must supply the UNITADDRESS (ccuu) and VOLID (xxxxxx) in this example. If an output file named 'XYZ OUTPUT A' already exists, you will have to add a second 'PUSH 'U'' to inform Device Support Facilities to erase the file. A 'PUSH 'END'' is not required, as ICKDSF is terminated when end-of-file is detected from the input file.

```
/* Sample CMS exec to invoke and execute an ICKDSF INIT*/
ADDRESS 'COMMAND'
PUSH 'U'
PUSH 'XYZ OUTPUT A'
PUSH 'ABC INPUT A'
ICKDSF
exit
```

Figure 6. Sample CMS EXEC to invoke and execute an ICKDSF INIT

Controlling Command Execution

Command execution can be controlled by the statements that follow:

- The **IF-THEN-ELSE** statement sequence tests a condition code and controls execution of subsequent commands according to the results of the test. The IF statement is followed by THEN and ELSE clauses, which specify alternative actions. DO and END statements are optionally part of THEN and ELSE clauses and specify the beginning and ending of a statement sequence. When you want to specify more than one Device Support Facilities command with either a THEN or an ELSE clause, include the DO and END statements.
- The **SET** statement sets condition codes.

Condition Codes to Control Command Execution

The condition codes compared in the IF-THEN-ELSE statement sequence are LASTCC and MAXCC, which take the following values:

0

indicates that the command was executed as specified. Some informational messages may have been issued.

4

indicates that some problem was met while executing the complete command, but Device Support Facilities processing continued. The continuation might not provide the user with exactly what was wanted, but no data or volume has been permanently harmed. A warning message was issued. An example of the kind of problem encountered is: The system ignored a parameter that was specified but did not apply.

8

indicates that a requested command was completed, but some actions were unavoidably bypassed. For example, Device Support Facilities was unable to reclaim an alternate track.

12

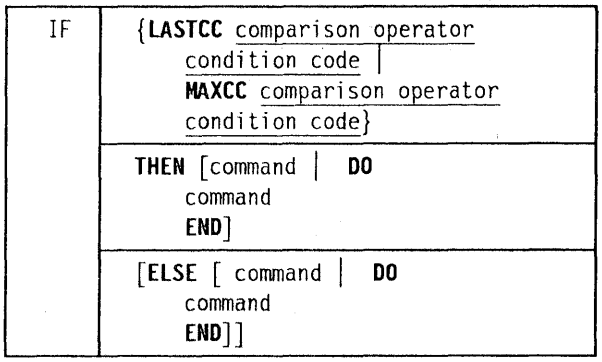
indicates that the requested command could not be executed. This condition code is a result of a logical error or a problem during execution. A logical error exists, for example, when inconsistent parameters are specified, when a too-small or too-large value is specified, or when required parameters are missing. A problem during execution exists when the volume label cannot be changed.

16

indicates a severe error occurred, causing the remainder of the command stream to be flushed. For example, this condition code might be a result of the following: the system output data set could not be opened (the SYSPRINT DD statement was missing, perhaps); an unrecoverable error occurred in a system data set; or Device Support Facilities encountered improper IF-THEN-ELSE statement sequences.

IF-THEN-ELSE Statement Sequence

You use the IF-THEN-ELSE statement sequence to control command execution. The syntax of the IF-THEN-ELSE statement sequence is:



where:

IF specifies that one or more Device Support Facilities commands are to be executed based on a test of a condition code value. The condition code was set by a SET command or reflects the completion status of previous Device Support Facilities commands.

LASTCC comparison operator condition code specifies that the condition code value resulting from the immediately previous Device Support Facilities command is to be compared to the condition code indicated by the comparison operator to determine whether the THEN or ELSE action is to be performed. See "Condition Codes to Control Command Execution" on page 25 for an explanation of condition codes.

For comparison operator, substitute one of the following six comparison operators to compare the condition code resulting from the previous Device Support Facilities command and the condition code you specified in the IF statement:

- Equal, specified as = or EQ
- Not equal, specified as ¬ = or NE
- Greater than, specified as > or GT
- Less than, specified as < or LT
- Greater than or equal, specified as > = or GE
- Less than or equal, specified as < = or LE

For condition code, substitute the decimal number that is to be compared with LASTCC. Condition codes greater than 16 are reduced to 16. See "Condition Codes to Control Command Execution" on page 25 for an explanation of condition codes.

MAXCC comparison operator condition code specifies that the maximum condition code value resulting from any previous Device Support Facilities command or a SET command is to be compared to the condition code indicated by the comparison operator to determine whether the THEN or ELSE action is to be performed. See "Condition Codes to Control Command Execution" on page 25 for an explanation of condition codes.

For comparison operator, substitute one of the following six comparison operators to compare the maximum condition code resulting from any Device Support Facilities command and the condition code you specified in the IF statement:

- Equal, specified as = or EQ
- Not equal, specified as ¬ = or NE
- Greater than, specified as > or GT
- Less than, specified as < or LT
- Greater than or equal, specified as > = or GE
- Less than or equal, specified as < = or LE

For condition code, substitute the decimal number that is to be compared with MAXCC. Condition codes greater than 16 are reduced to 16. See "Condition Codes to Control Command Execution" on page 25 for an explanation of condition codes.

THEN specifies that a single command (or a group of commands introduced by DO) is to be executed if the comparison is true. THEN can be followed by another IF statement.

ELSE specifies that a single command (or a group of commands introduced by DO) is to be executed if the comparison is false. ELSE can be followed by another IF statement.

When an IF statement appears in a THEN or ELSE clause, it is called a nested IF statement. The maximum number of nested levels allowed is 10, starting with the first time you specify IF.

Within nested IF statements, the innermost ELSE clause is matched to the innermost THEN clause, the next innermost ELSE clause is matched to the next innermost THEN clause, and so on. In other words, each ELSE clause is matched to the nearest preceding unmatched THEN clause. If there is an IF statement that does not require an ELSE clause, follow the THEN clause with a null ELSE clause, unless the nested structure does not require one.

DO command can be specified with the THEN and ELSE clauses to indicate that the group of commands following are to be treated as a single unit, that is, to be executed as a result of a single IF statement. Commands following a DO must begin on a new line. The group of commands is terminated by END.

For command, substitute the Device Support Facilities command that you want to process as a result of a single IF statement.

END specifies the end of a group of Device Support Facilities commands initiated by the nearest unended DO. END must be on a line by itself.

Null THEN and ELSE Clauses

When a THEN or ELSE clause is not followed by a continuation character or by a command in the same record, it results in no action.

For example, to specify a null THEN clause when you want a Device Support Facilities command to execute if the IF statement is false, you specify THEN without a continuation character or Device Support Facilities command. On the next line, specify ELSE followed by a Device Support Facilities command. If you want to specify a null THEN clause, specify:

```
IF ...
 THEN
 ELSE ...
```

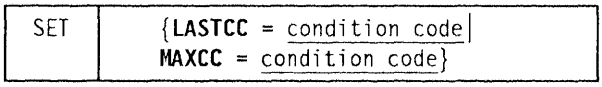
If you want to indicate a null ELSE clause, specify:

```
IF ...
 THEN ...
 ELSE
```

SET Statement

The SET statement sets a condition code. See "Condition Codes to Control Command Execution" on page 25 for an explanation of condition codes. You can terminate all processing simply by setting MAXCC or LASTCC to 16.

The syntax of the SET statement is:



where:

SET specifies that a condition-code value is to be set. If a SET statement is specified following a THEN or ELSE clause, and the clause is not executed because the comparison in the IF statement does not allow the clause to execute, the condition code specified with LASTCC or MAXCC is not set.

LASTCC = condition code specifies that the condition code to be set is the condition code set by the immediately preceding Device Support Facilities command.

For condition code, substitute a decimal number that is to be assigned to LASTCC. Condition codes greater than 16 will be reduced to 16. If the condition code assigned to LASTCC is a greater value than the condition code assigned to MAXCC, MAXCC assumes the condition code assigned to LASTCC.

MAXCC = condition code specifies that the condition code to be set is the maximum condition code. Setting MAXCC does not affect the condition code for LASTCC.

For condition code, substitute a decimal number that is to be assigned to MAXCC. Condition codes greater than 16 will be reduced to 16.

Continuation Lines

Use continuation lines cautiously when specifying IF-THEN-ELSE statements, SET statements, comments, or blank records in the input stream. Blank records or records ending with comments must end with a continuation mark when you specify these types of records in the middle of a command or between the THEN and ELSE clauses of an IF statement. Always specify a continuation mark when records end with partial comments. (See "Line Continuations" on page 68.)

You must be careful when you continue IF-THEN-ELSE statements so that you don't inadvertently specify a null THEN or ELSE clause. For information on this subject, see "Null THEN and ELSE Clauses" on page 28.

Examples of Statements to Control Command Execution

The examples that follow show the use of the IF-THEN-ELSE statement sequence and the SET statement.

IF-THEN-ELSE Statement Example

```
INIT DDNAME(VOLXYZ) VERIFY(VOLABC)
  IF LASTCC=0 -
    THEN REFORMAT DDNAME(VOL293) VOLID(SYX1) NOVERIFY
    ELSE REFORMAT DDNAME(VOL123) VOLID(SYX2) NOVERIFY
```

SET Statement Example

```
SET LASTCC=8
```

DO-END Statement Example

```
IF MAXCC=05 -
  THEN DO
         ANALYZE UNITADDRESS(ccuu) SCAN
         REFORMAT UNITADDRESS(ccuu) NOVERIFY
       END
```

ELSE-DO-END Statement Example

```
ANALYZE UNITADDRESS(ccuu)
  IF LASTCC>4 -
    THEN ...
    ELSE DO
           MAPALT UNITADDRESS(ccuu) SCAN
           INIT UNITADDRESS(ccuu) NOVERIFY VOLID(SYSX1)
         END
```

Volume and Data Set Security

Device Support Facilities commands are used to manage direct-access storage volumes. As there is some danger to data through inadvertent manipulation of a volume, it is important to have a means of protecting data. With Device Support Facilities, there are several degrees of volume and data set protection.

There are two environments in which Device Support Facilities can execute: under an operating system or in a stand-alone environment. Under MVS operating systems, two modes of operation are available: offline and online. In a stand-alone environment, only the offline mode is available. Under VSE, only the online mode is available. Volume and data set security in different environments and with different modes of operation are discussed in the following sections.

The ANALYZE and MAPALT commands do not expose data on the volume to unauthorized disclosure, modification, or destruction. These commands do not write on any user tracks and do not transmit any user data from the device. Therefore, no security functions are performed for these commands in any environment.

Volume Security

Volume security is limited to what is indicated below:

- The VERIFY parameter of the INIT, INSPECT, and REFORMAT commands can be specified to ensure that the volume serial number and owner identification supplied in the command match those found on the volume. If they match, execution of the command continues; if they do not match, execution of the command terminates. This check is operational under all environments and modes of operation.
- Under an MVS/370 or MVS/XA operating system, the RACF program can determine whether the volume is RACF-protected (RACF DASDVOL is active) and whether the user has authority to access the volume. If the user has insufficient access authority to a RACF-protected volume, command execution terminates. This is valid for INIT and INSP commands.

Data Set Security in an Offline Mode (MVS) or in a Stand-alone Environment

Data set security in an offline mode or in a stand-alone environment is limited to the following:

The PURGE parameter of the INIT command can be specified. Before the command starts processing the volume, the system operator is requested to confirm that the contents of the volume can be altered (see message ICK003D in Appendix C, "Device Support Facilities Messages (ICK)"). If the operator confirms that the volume contents can be changed, execution of the command continues and any data sets contained in the volume are purged.

Data Set Security in an Online Mode (MVS)

Because the resources of the operating system are available to Device Support Facilities in an online mode, data set security is greater. The data security function is only invoked for the INIT and INSPECT commands. The action taken depends on the type of data set, the expiration date, and whether the data set is RACF or password protected. The actions taken are as follows:

- *RACF-protected data sets*: For the INIT command, the RACF program is called for each RACF-protected data set on the volume to determine whether the user is authorized to access the data set contents. For the INSPECT command, the RACF program is called for each user-specified track that falls within any RACF-protected data set to determine whether the user has authority to access and alter the data set. With either command, if the user has the appropriate authority, command execution continues; otherwise, command execution terminates.
- *Non-VSAM password-protected data sets*: For the INIT command, all password-protected data sets on a volume are subject to the security function. For the INSPECT command, the security function is called for each track that falls within a password-protected data set. The user may provide passwords by using the PASSWORDS parameter on the INIT and INSPECT commands.

  When the first password-protected data set subject to the security function is encountered, Device Support Facilities invokes a user security exit module to determine what needs to be done (a default user security exit module (ICKUSER1) is provided with Device Support Facilities—for details see Appendix H, "User Security Exit Module" on page 319). This module returns a code to Device Support Facilities indicating one of the actions in Figure 7.

Return Code	Meaning
0	Bypass checking of password-protected data sets on the volume and do not call the exit routine again.
4	Bypass checking of the current password-protected data set but call the exit routine for the next password-protected data set.
8	Check the current password-protected data set normally and call the exit routine for the next password-protected data set.
12	Check all password-protected data sets on the volume normally and do not call the exit routine again. (The default for ICKUSER1.)
16	Reject access to this data set, but return to the exit routine for the next data set.
20	Terminate this command immediately.

Figure 7. Codes Returned by the User Security Exit Module for Password-protected Data Sets

If normal checking is to be done, Device Support Facilities looks for a user-supplied password for the current data set. If one is not found, the command is targeted for termination, and the next data set is checked. When a password is found, the system PASSWORD data set is checked to determine whether the user-supplied password provides authority to alter the data set. If it does not, the command is targeted for termination, and the next data set is checked.

As long as the user exit routine returns a code indicating that the routine should be called again, it is called for each password-protected data set. Even if the command is targeted for termination, password checking continues for all data sets on the volume before processing actually terminates. This ensures that all missing passwords are identified.

- *VSAM/unexpired data sets*: For the INIT command, all VSAM and unexpired data sets on a volume are subject to the security function. For the INSPECT command, the security function is called for each VSAM/unexpired data set that contains a track that is to be inspected. When the first VSAM/unexpired data set is encountered, Device Support Facilities invokes the user security exit module (ICKUSER1) to determine what needs to be done. This module indicates one of the actions in Figure 8.

Return Code	Meaning
0	Bypass checking of VSAM/unexpired data sets on the volume and do not call the exit routine again.
12	Process VSAM/unexpired data sets on the volume normally and do not call the exit routine again. (The default for ICKUSER1).
20	Terminate this command immediately.

Figure 8. Codes Returned by the User Security Exit Module for VSAM/Unexpired Data Sets

Normal processing for the INIT command is to check whether PURGE has been specified. If not, the command is targeted for termination and the next data set is checked.

Normal processing for the INSPECT command is to target the command for termination and to check the next data set.

- *VTOC and the VTOCIX data set*: For the INSPECT command when NOPRESERVE is specified, Device Support Facilities does not allow inspection of tracks inside a VTOC or a VTOCIX data set in an online (MVS) mode. Cylinder 0, track 0 is also not allowed to be inspected with NOPRESERVE in an online mode.

Authorized Program Facility (MVS/370 and MVS/XA only)

The authorized program facility (APF) limits the use of sensitive system services and resources to authorized system and user programs. See *OS/VS2 System Programming Library: Supervisor Services and Macro Instructions* for information about program authorization.

The Device Support Facilities load module is contained in SYS1.LINKLIB. The load module (ICKDSF) is link-edited with the SETCODE AC(1) attribute. This ensures that Device Support Facilities is correctly link-edited.

This authorization is established at the task level. If, during the execution of an APF-authorized job step, a load request is satisfied from an unauthorized library, the task will be abnormally terminated. It is the installation's responsibility to ensure that a load request cannot be satisfied from an unauthorized library during Device Support Facilities processing.

If a Device Support Facilities module is loaded from an unauthorized library, the Authorized Program Facility authorization of Device Support Facilities terminates.

Data Set Security and Protection (VSE)

The VERIFY parameter of the INIT, INSPECT, and REFORMAT commands can be specified to ensure that the volume serial number and owner identification supplied in the command match those found on the volume. If they match, execution of the command continues; if they do not match, execution of the command terminates.

The NOPURGE parameter of the INIT command can be specified. Execution of the command terminates if unexpired files, data-secured files, or VSAM files are found on the volume.

The PURGE parameter of the INIT command or the NOPRESERVE parameter of the INSPECT command can be specified. If unexpired files, data-secured files, or VSAM files are on the volume, data loss is prevented as follows:

- *Unexpired files*: The operator is requested to confirm that the contents of the volume can be purged (see message ICK001D in Appendix C, "Device Support Facilities Messages (ICK)"). Only one request is made, regardless of the number of unexpired files on the volume. A list of the unexpired files is not provided.
- *Data-Secured files*: If a file is a data-secured file, message ICK017D prompts the operator with the name of the secured file, and message ICK018D requests permission to purge the file. This sequence is repeated for each data-secured file that exists on the volume.
- *VSAM files*: If a single VSAM file is found on the volume, permission is requested for the purging of all the VSAM files on the volume (message ICK019D).

Processing in a Shared Environment (MVS)

The commands BUILDIX, INIT, INSPECT, INSTALL, REVALidate, and REFORMAT issue the RESERVE macro during processing to obtain control of the device. After processing on the device is completed, the DEQ macro is issued to release the device. This is done for offline as well as online processing.

This procedure prevents other processors from accessing the volume, but **cannot** guarantee exclusive access to the volume from the same processor.

When processing online there is always a potential danger that a particular track will be concurrently updated. In order to minimize this impact, the following steps are taken by ICKDSF.

Device Support Facilities will not permit a volume with more than one active user to be processed online when either of the following commands is specified:

- INIT
- REFORMAT with VOLID

At completion of any ICKDSF command that modifies the volume label, the VTOC, or the VTOC index, a MOUNT command should be issued from the other processors that are sharing the volume.

When the INSPECT command is specified to perform skip displacement processing (using the SKIP parameter), if a track is part of a data set, INSPECT will enqueue on the data set while processing that track.

Additionally, if the INSPECT command exposes user data for a relatively long period of time, ICKDSF enqueues for exclusive control of the data set containing the track which is to be processed. Data set enqueue takes place when

- The device is online to the operating system.
- The track is part of a non-VSAM data set.
- The SKIP parameter is specified.
- NOSKIP is specified, but primary surface checking indicates that skip displacement processing is necessary for the track.

If exclusive control of a data set cannot be obtained, a message is issued, and processing continues on the next track.

The TOLERATE(ENQFAIL) parameter is provided to indicate that processing is to continue if the data set enqueue fails when SKIP is specified.

If the enqueue is performed as a result of primary surface checking indicating the need for skip displacement surface checking, TOLERATE(ENQFAIL) is assumed to be in effect.

For tracks in VSAM data sets, TOLERATE(ENQFAIL) must always be specified when processing online.

Processing in a Shared Environment (VSE)

For all commands, processing continues only after the operator has replied to a console message (ICK020D) granting permission to process on the device that is being shared.

Device Support Facilities cannot guarantee exclusive access to an entire volume. When processing online there is always the danger of a particular track being concurrently updated; users who run Device Support Facilities online must be prepared to deal with this possibility.

In order to minimize the risks involved in online processing, exercise system controls when using Device Support Facilities.

Examples of ICKDSF Tasks—CKD Devices

The following examples show the commands and parameters needed for each task in a **stand-alone environment**. Default parameters are indicated only where they produce additional output; optional parameters for stand-alone environments are described briefly in the notes that follow. For complete descriptions of each command and examples of VSE, MVS/370, and MVS/XA operations, see the descriptions of the individual commands beginning with "ANALYZE Command—CKD" on page 73.

You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands using count-key-data devices are presented in this section. Examples of Device Support Facilities commands using fixed block architecture devices are presented in

- "Examples of ICKDSF Tasks—FBA Devices" on page 55.

Further examples are presented after each command in

- "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
- "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

Initializing a Replaced Head Disk Assembly (HDA)

Use the INIT command to prepare a replaced HDA for use. Volumes are initialized at the medial level.

```
INIT UNITADDRESS(ccuu) NOVERIFY VOLID(serial) OWNERID(owner) -
     VTOC(cylinder,track,extent) VALIDATE DATA
```

Notes:

- Data on the volume is destroyed.
- This example establishes a volume ID, an owner ID, and a VTOC for the volume.
- The VALIDATE parameter ensures that potentially erroneous data on all tracks is erased by initializing at the medial level for the entire volume.
- This process rewrites all home addresses and record zeros on the volume.
- To write IPL program records on the volume, add the IPLDD parameter to the end of the parameter list: IPLDD(SYSIN,ABSFORMAT)
- The DATA parameter specifies that factory functional verification data patterns (FFVDP) are written on the specified tracks. The DATA parameter is valid for the IBM 3375 and 3380 only. If the example is run against a device other than the 3375 or 3380, DATA is ignored.

Initializing VM Minidisks

Using ICKDSF under the CMS operating system of VM, you can use the INIT command to prepare a VM minidisk for use at the minimal level as follows.

```
INIT UNITADDRESS(ccuu) NOVERIFY VOLID(serial) OWNERID(owner) -
     VTOC(cylinder,track,extent)
```

Notes:

- This example establishes a volume ID, an owner ID, and a VTOC for a minidisk.

In the stand-alone environment, you can use the MIMIC(MINI) parameter of the INIT command to prepare a VM minidisk for use. VM minidisks (except 2314) must be initialized at the minimal level.

```
INIT UNITADDRESS(ccuu) NOVERIFY VOLID(serial) OWNERID(owner) -
     VTOC(cylinder,track,extent) MIMIC(MINI(15)) DEVICETYPE(3350)
```

Notes:

- This example establishes a volume ID, an owner ID, and a VTOC for the volume.
- Minidisks can only be initialized in the ICKDSF stand-alone version by using the MIMIC(MINI) parameter.

Initializing or Reinitializing a Device

Volumes should be reinitialized at the minimal level.

This example illustrates how to initialize or reinitialize a device at the minimal level. See "Initializing a Replaced Head Disk Assembly (HDA)" on page 37 for an example of medial initialization, and "Checking Disk Surfaces" on page 39 for an example of maximal initialization.

```
INIT UNITADDRESS(ccuu) NOVERIFY VOLID(serial) -
     OWNERID(owner) VTOC(cylinder,track,extent)
```

Notes:

- The VTOC is rewritten and access to data on the volume is lost.
- If you want to erase previous data on the volume, initialize at the medial level. See "Initializing a Replaced Head Disk Assembly (HDA)" on page 37.

Investigating Suspected Drive Problems

If you suspect a problem with a drive, use the ANALYZE command to assist in determining:

1. If the drive can perform basic operating functions correctly
2. If all data can successfully be read from the disk surface

```
ANALYZE UNITADDRESS(ccuu) SCAN
```

Notes:

- The data on the volume is not altered.
- The drive test option of the ANALYZE command is valid only for devices with nonremovable media: IBM 3344, 3350, 3375, 3380; or any emulated 3330 or 3340.
- Add the SPEED parameter to the parameter list to scan data one cylinder at a time rather than one track at a time.
- Add the LIMITS parameter to the parameter list to specify only a limited area for scanning.

Restoring Factory Functional Verification Data Patterns to a Volume

Use the following example to restore the factory functional verification data patterns on an IBM 3375 or 3380 volume. Because DATA is specified, the example is valid only for these devices.

```
INIT UNITADDRESS(ccuu) NOVERIFY VALIDATE VOLID(serial) DATA
```

Note:

- Data on the volume is erased.
- The factory functional verification data patterns are restored on the volume.
- Because DATA is specified, this example is valid only for the IBM 3375 and 3380. If the example is used for a device other than a 3375 or 3380, DATA is ignored.

Checking Disk Surfaces

Use the INIT command to check the surface of all tracks on a volume (other than an IBM 3375 or 3380) by writing test patterns and then reading them back for every track on the volume. An alternate track is assigned for any primary track that is indicated to be defective.

Note that the CHECK parameter is invalid for 3375 or 3380 devices.

Use the INSPECT command (see below for an example) to check the surface of a specific track or range of tracks of a volume for all devices including the IBM 3375 and 3380.

```
INIT UNITADDRESS(ccuu) NOVERIFY VOLID(serial) CHECK(n)
```

Notes:

- This example is not valid for the IBM 3375 or IBM 3380 because the CHECK parameter is not supported for these devices. See the INSPECT command below for an example of surface checking these devices.
- **Data on the volume is erased.** If data is to be retained, it should be dumped before processing begins.
- Tracks that were flagged defective when processing started will be surface checked, but not reclaimed. See "Reclaiming a Single Flagged Track" on page 41 and "Reclaiming Flagged Tracks" on page 42 if you want to try to reclaim defective tracks.
- **n** can be any number from 1 to 10. See Appendix G, "Surface Checking" on page 315 for recommended values of n for specific devices.
- This process rewrites all home addresses and record 0s and performs read/write checks on all surfaces.
- The default MAP prints a volume map of alternate tracks assigned during this process.

The following example uses the INSPECT command to check the surface of a volume. The example is valid for all devices including the IBM 3375 and 3380.

```
INSPECT UNITADDRESS(ccuu) NOVERIFY CHECK(n) -
        TRACKS(cccc,hhhh) PRESERVE
```

Notes:

- **n** can be any number from 1 to 10. See Appendix G, "Surface Checking" on page 315, for recommended values of n for specific devices.
- PRESERVE saves the data on the track by moving the data from the current track to an alternate track. NOPRESERVE causes the data on the track to be erased. Use this parameter only if there is no meaningful data on the track or if track errors prevent successful use of PRESERVE.
- When running online to an MVS operating system, security procedures are invoked.

Conditionally Assigning Alternate Tracks

Use the INSPECT command to check the surface of the specified track and to assign an alternate track if the surface checking process indicates that the track is defective. When PRESERVE is specified, the data is moved from the current track to the alternate track that is assigned.

```
INSPECT UNITADDRESS(ccuu) NOVERIFY CHECK(n) -
        TRACKS(cccc,hhhh) PRESERVE
```

Notes:

- **n** can be any number from 1 to 10. See Appendix G, "Surface Checking" on page 315, for recommended values of n for specific devices.
- PRESERVE saves the data on the current track by moving the data from the current track to an alternate track. NOPRESERVE causes the data on the track to be erased. Use this parameter only if there is no meaningful data on the track or if track errors prevent successful use of PRESERVE.
- When running online to an MVS operating system, security procedures are invoked.

Unconditionally Assigning Alternate Tracks

Use the INSPECT command to unconditionally assign an alternate track regardless of the condition of the primary track.

```
INSPECT UNITADDRESS(ccuu) VERIFY(serial,owner) NOCHECK -
        ASSIGN TRACKS(cccc,hhhh) PRESERVE
```

Notes:

- If the specified track is an alternate, the primary track currently assigned to it (if any) is assigned a different alternate track.
- When NOPRESERVE is specified, the data on the current track is lost.
- When running online to an MVS operating system, security procedures are invoked.

Printing a Track Assignment Map

Use the INSPECT command to prepare and print a report of alternate tracks assigned on the volume, for your information.

```
INSPECT UNITADDRESS(ccuu) NOVERIFY NOCHECK NOASSIGN -
        ALLTRACKS MAP
```

Note: Data on the volume is not altered.

Reclaiming a Single Flagged Track

Use the INSPECT command to check the surface of a specified track and to reclaim the track if it is currently flagged defective.

```
INSPECT UNITADDRESS(ccuu) NOVERIFY CHECK(n) -
        TRACKS(cccc,hhhh) RECLAIM PRESERVE
```

Notes:

- Data on the track is **not** erased. If the track is reclaimed, the data is rewritten to the reclaimed track.
- The maximum level of surface checking available for the device is done for the track. If the track condition is determined to be good, and the track was assigned an alternate when processing began, reclamation takes place. Conversely, if the track condition is determined to be defective, reclamation does not occur.
- If the above task is used for a track that is not flagged defective when processing begins, the results are the same as those described for "Conditionally Assigning Alternate Tracks" on page 40.
- See Appendix G, "Surface Checking" on page 315, for recommended values of n for specific devices.

Reclaiming Flagged Tracks

Use the INIT command to check the surface of all tracks on a volume (other than a 3375 or 3380) and to determine if tracks currently flagged as defective can be reclaimed.

```
INIT UNITADDRESS(ccuu) RECLAIM CHECK(n) NOVERIFY VOLID(serial)
```

Notes:

- **Data on the volume is erased. If data is to be retained, it should be dumped before processing begins.**
- Reclamation for a track that was flagged defective when processing started occurs only if surface checking indicates that the track is now defect-free.
- For skip displacement devices, skip displacement checking is done on currently flagged tracks (instead of primary checking) to ensure the current condition of the track.
- **n** can be any number from 1 to 10. See Appendix G, "Surface Checking" on page 315, for recommended values of n for specific devices.
- This process rewrites all home addresses and record 0s and performs read/write checks on all surfaces.
- The default MAP prints a volume map of alternate tracks assigned during this process.
- Note that the CHECK(n) parameter is not valid for IBM 3375 and 3380 devices. Use the INSPECT command to surface check and reclaim tracks for these devices

Rewriting Home Address/Record 0

Use the INIT command at the medial level to rewrite all home addresses and record zeros on the volume.

```
INIT UNITADDRESS(ccuu) NOVERIFY VALIDATE VOLID(serial)
```

Note: Data on the volume is erased. If data is to be preserved, it should be dumped before processing begins.

Emulating a CKD Device on an FBA Device

Preparing to Emulate a Device

Emulating a count-key-data device on a fixed block architecture device requires the following procedure:

1. Use Device Support Facilities to prepare the FBA disk for use. If you have not already done so, initialize the FBA disk with the FBAVTOC in the standard location (FBAVTOC(2,56,1024)) and write a volume label. See "Initializing a New Device or HDA" on page 55 for an example of initialization.
2. Execute the format emulated extent utility (INITEM). Refer to *VSE/Advanced Functions System Management Guide* for a description of the INITEM utility.
3. If it is not already active, activate the Direct Access Storage Compatibility Feature, described in *IBM 4331 Processor Compatibility Features*. Refer to this manual for a detailed description of:
   - Association of native device addresses with the emulated device addresses
   - Physical space boundaries of emulated space on native devices
   - Emulation restrictions
4. Use the Device Support Facilities INIT command to initialize the emulated CKD disk for use.

   For full emulated CKD devices, use:

```
INIT  UNITADDRESS(ccuu) NOVERIFY VALIDATE -
      VOLID(volser) MIMIC(EMUALL) DEVTYPE(2314)
```

   For partial emulated CKD devices, use:

```
INIT  UNITADDRESS(ccuu) NOVERIFY VALIDATE -
      VOLID(volser) MIMIC(EMU(n)) DEVTYPE(2314)
```

   Notes:

   a. For **n**, substitute the same number of cylinders that was specified as the size of the subdisk during the format emulated extent (INITEM) program/operator dialog. Device Support Facilities does not check the validity of **n**.

   b. INIT writes a special format-1 DSCB on the partial CKD device to ensure that the nonexistent cylinders are never accessed.

Media Maintenance on an Emulated Device

When you perform media maintenance because of surface defects, you should use INSPECT or the FBA maximal INIT against the base address. INSPECT can assign an alternate track to the emulated device. However, this has a greater performance impact than assigning alternates to blocks found defective on the base FBA device address.

Changing the Volume Serial Number and Owner Identification

Use the REFORMAT command to change the volume serial number and/or owner identification when the volume changes hands.

```
REFORMAT UNITADDRESS(ccuu) NOVERIFY VOLID(newvol) -
         OWNERID(newowner)
```

Notes:

- You may specify either VOLID, or OWNERID, or both.
- No other data on the volume is changed.
- You may also add the IPLDD(SYSIN) parameter to add IPL text to the volume at the same time.

Adding IPL Text to a Volume

Use the REFORMAT command to add IPL text to an already-initialized volume.

```
REFORMAT UNITADDRESS(ccuu) NOVERIFY BOOTSTRAP IPLDD(SYSIN)
```

Converting a Volume to Indexed Format

To convert an OS VTOC to indexed format, use the BUILDIX command:

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//VOLDD     DD UNIT=(3330,,DEFER),VOL=(PRIVATE,SER=VL3330),
//          DSN=SYS1.VTOCIX.VL3330,DISP=OLD
//SYSIN     DD *
 BUILDIX  DDNAME(VOLDD)  IX
/*
```

```
BUILDIX DDNAME(dname) IXVTOC
```

To convert an indexed VTOC back to OS format, use the BUILDIX command:

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//VOLDD     DD UNIT=(3330,,DEFER),VOL=(PRIVATE,SER=VL3330),
//          DSN=SYS1.VTOCIX.VL3330,DISP=OLD
//SYSIN     DD *
 BUILDIX  DDNAME(VOLDD)  OS
/*
```

```
BUILDIX DDNAME(dname) OSVTOC
```

Note: The BUILDIX command is valid only in MVS environments.

Surface Checking Part of a Volume

The following example performs primary surface checking, as part of an INSPECT command, on part of a volume, from cylinder 4, head 6, to cylinder 50 head 8.

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) NOSKIP -
     FROMRANGE(4,6) TORANGE(50,8) PRESERVE
```

Notes:

- Primary surface checking is performed beginning at cylinder 4 head 6 and ending at cylinder 50, head 8.
- The total combination of tracks that can be inspected is 2500.

The following example performs primary surface checking on all heads of a volume, from cylinder 50 through cylinder 100.

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) NOSKIP CYLRANGE(50,100) PRESERVE
```

The following example performs primary surface checking, on head 4 only, for all cylinders of a volume.

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) NOSKIP HEADRANGE(4,4) -
        PRESERVE
```

Notes:

- Primary surface checking is performed on cylinder 0, head 4, to the last alternate cylinder head 4.
- Data is saved at the backup location on each track processed.

The following example performs primary surface checking on heads 5 and 6 only, on cylinders 500 through 599:

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) NOSKIP -
     CYLRANGE(500,599) HEADRANGE(5,6) PRESERVE
```

The following example performs primary surface checking starting at cylinder 500, starting at head 14:

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) NOSKIP -
     CYLRANGE(500,9999) HEADRANGE(14,9999) PRESERVE
```

- In this example, all cylinders from cylinder 500 through the last alternate cylinder are processed, for heads 14 through the last head for this device type.
- For a 3380 A04, AA4, and AD4, ICKDSF processes cylinders 500 through 885, head 14 only.
- For a 3350, ICKDSF processes cylinders 500 through 559, heads 14 through 29.
- Total combination of tracks cannot exceed 2500.

The following example performs skip displacement surface checking of all tracks on the alternate cylinder of an IBM 3380 Model A04, AA4, and AD4.

```
INSPECT UNIT(ccuu) NOVFY CHECK(1) SKIP CYLRANGE(885,885)
```

Resume Processing from a Checkpointed Location

The examples below show processing being resumed after ICKDSF has automatically checkpointed itself during a medial INIT. CONTINUE specifies ICKDSF is to resume from the last checkpoint.

The following example (let's call it JOB1) validates and writes factory functional verification data patterns on a volume. Assume that ICKDSF automatically last took a checkpoint at cylinder 200, head 0, and processing is interrupted after that checkpoint.

```
INIT UNIT(ccuu) NOVFY VALIDATE DATA
```

If you ran the following, after processing has been interrupted as in JOB1 above, you can validate part of a volume.

```
INIT UNIT(ccuu) VALIDATE NODATA NOVERIFY -
     CYLRANGE(100,500) HEADRANGE(2,6) CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for about 1000 tracks, are validated and data is erased from those tracks. An IBM 3380 would be validated from cylinder 200 head 0 to cylinder 266 head 14.
- Validation (including erasing data) is also done for heads 2 through 6 for all cylinders beyond the last cylinder validated above, up to and including cylinder 500. For example, on a 3380, cylinders 267 to 500 are validated.
- Minimal initialization is performed.

If you ran the following, after processing has been interrupted as in JOB1 above, you can write factory functional verification data patterns on part of a volume. Note that the part of the volume specified had been totally completed (in JOB1) before the last checkpointed location.

```
INIT UNIT(ccuu) NOVFY DATA VALIDATE -
     CYLRANGE(0,100) CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for about 1000 tracks are validated and have factory functional verification data patterns written on them.
- Additional validation and writing of factory functional verification data patterns does not occur for the tracks in the specified range, because the specified range is totally contained before the last checkpointed location.
- Minimal initialization function is performed.

If you ran the following, after processing has been interrupted as in JOB1 above, you can perform a minimal initialization of the volume:

```
INIT UNIT(ccuu) NOVFY NOCHECK NOVALIDATE CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for approximately 1000 tracks, are validated and data is erased.
- Then the requested minimal initialization is performed.

If you ran the following, after processing has been interrupted as in JOB1 above, you can write factory functional verification data patterns on part of the volume, regardless of where previous processing was interrupted (NOCONTINUE).

```
INIT UNIT(ccuu) NOVFY VALIDATE DATA -
     CYLRANGE(0,100) NOCONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for about 1000 tracks, are validated and factory functional verification data patterns are written on them.
- Tracks are validated and factory functional verification data patterns are written for the specified cylinders for all heads on the device:
  - For a 3380 A04, AA4, and AD4, ICKDSF processes heads 0 through 14 for cylinders 0 through 100
- Minimal initialization is performed.

If you ran the following, after processing has been interrupted as in JOB1 above, you can validate part of the volume from a specific point.

```
INIT UNIT(ccuu) NOVFY VALIDATE -
     FROMRANGE(500,6)  CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for approximately 1000 tracks, are validated and data is erased.
- All tracks are then validated starting from cylinder 500 head 6, through the last head on the last alternate cylinder.
  - For a 3380 A04, AA4, and AD4, ICKDSF processes all tracks on each cylinder starting from cylinder 500, head 6 through cylinder 885, head 14.
  - Minimal initialization is performed.

If you ran the following, after processing has been interrupted as in JOB1 above, you can validate part of the volume to a specific point.

```
INIT UNIT(ccuu) NOVFY VALIDATE -
     TORANGE(500,6) CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for approximately 1000 tracks, are validated and data is erased.
- Starting from cylinder 200 head 0 through cylinder 500 head 6, all tracks on each cylinder are validated and data is erased from those tracks.

If you ran the following, after processing has been interrupted as in JOB1 above, you can validate and write factory functional verification data patterns for part of the volume starting at a specific cylinder for only one head.

```
INIT UNIT(ccuu) NOVFY VALIDATE DATA -
     CYLRANGE(500,9999) HEADRANGE(6,6) CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for approximately 1000 tracks, are validated and factory functional verification data patterns are written.
- Tracks are validated and factory functional verification data patterns are written on head 6 only, from cylinder 500 through the last alternate cylinder.
  - For a 3380 A04, AA4, and AD4, ICKDSF processes head 6 only, from cylinder 500 through cylinder 885.
- Minimal initialization is performed.

If you ran the following, after processing has been interrupted as in JOB1 above, you can validate the last few heads of a device.

```
INIT UNIT(ccuu) NOVFY VALIDATE -
     HEADRANGE(8,99) CONTINUE VOLID(volser)
```

Notes:

- All tracks starting at cylinder 200, head 0 (the last checkpointed location), and continuing for approximately 1000 tracks, are validated and data is erased.
- Validation is performed and data is erased for all cylinders on the device, for heads 8 through n (where n is the maximum head number for the device).
  - For a 3380 A04, AA4, and AD4, ICKDSF processes heads 8 through 14, from cylinder 200 through cylinder 885.
  - Minimal initialization is performed.

Recovering after an INSPECT Job Failure

The following examples show how to recover after an INSPECT job failure. When PRESERVE is specified, ICKDSF saves a copy of the data on the track at a backup location as well as in storage. If processing is not completed, ICKDSF can find and restore the track that was being processed when the job was interrupted.

The following job ensures that data is saved during surface checking for all tracks on cylinder 200. When PRESERVE is specified, ICKDSF saves a copy of the data on the track at a backup location as well as in storage.

```
INSPECT UNIT(ccuu) NOVFY CHECK(1) SKIP -
        CYLRANGE(200,200) PRESERVE
```

The following example (let's call it JOB2) performs skip displacement surface checking on part of a volume. When PRESERVE is specified, ICKDSF saves a copy of the data on the track being processed at a backup location as well as in storage.

Assume that ICKDSF is executing on a 3380, and processing is interrupted on cylinder 201, head 1.

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) SKIP -
        FROMRANGE(200,4) TORANGE(201,3) PRESERVE
```

If you ran the following job after processing has been interrupted as in JOB2 above, you can surface check a different portion of the volume:

```
INSPECT  UNIT(ccuu) NOVFY CHECK(2) NOSKIP -
         CYLRANGE(100,500) HEADRANGE(2,6) KEEPIT
```

Notes:

- Cylinder 201, head 1 is validated and primary surface checked.
- If primary surface checking for cylinder 201, head 1 fails, skip displacement surface checking is invoked.
- If any data existed for cylinder 201, head 1 when processing was interrupted, the data is rewritten on cylinder 201, head 1 from the backup location.
- Primary surface checking is then performed for all the tracks within the new range.
  - Heads 2 through 6 on cylinders 100 through 500.

If you ran the following job after processing has been interrupted as in JOB2 above, you can surface check the specified heads for part of the volume:

```
INSPECT  UNIT(ccuu) NOVFY CHECK(2) NOSKIP -
         CYLRANGE(100,500) HEADRANGE(2,6) PRESERVE
```

Notes:

- Cylinder 201, head 1 is validated and primary surface checked.
- If primary surface checking for cylinder 201, head 1 fails, skip displacement surface checking is invoked.
- If any data existed for cylinder 201, head 1 when processing was interrupted, the data is rewritten on cylinder 201, head 1.

- Primary surface checking is then performed for all the tracks within the new range.

If you ran the following job after processing has been interrupted as in JOB2 above, you can use INSPECT to restore any data that may exist on the backup location, and ensure the usability of the volume.

```
INSPECT  UNIT(ccuu) NOVFY
```

Notes:

- Cylinder 201, head 1 is validated and primary surface checked.
- If primary surface checking for cylinder 201, head 1 fails, skip displacement surface checking is invoked.
- If any data existed for cylinder 201, head 1 when processing was interrupted, the data is rewritten on cylinder 201, head 1.

Scanning Data on Part of a Volume

The following examples show parts of a volume being specified for the ANALYZE command

In this example, the data verification test is performed for heads 5 and 6 only, on cylinders 500 through cylinder 599.

```
ANALYZE UNIT(ccuu) SCAN NODRIVE -
        CYLRANGE(500,599) HEADRANGE(5,6)
```

In this example, after the drive test, the data verification test is performed for all heads starting at head 14, for all cylinders starting at cylinder 500.

```
ANALYZE UNIT(ccuu) SCAN -
        CYLRANGE(500,9999) HEADRANGE(14,9999)
```

Notes:

After the drive test, all cylinders from cylinder 500 until the last primary cylinder are scanned, for heads 14 through the last head for this device type.

- For a 3380 A04, AA4, and AD4, ICKDSF processes head 14 only, on cylinders 500 through 884 (alternate cylinder not processed).
- For a 3350, ICKDSF processes heads 14 through 29 on cylinders 500 through 554 (alternate cylinder not processed).

How to Find Where a Job Failed Using ANALYZE

If you specified the following INSPECT job (let's call it JOB3), to perform skip displacement surface checking on part of a volume and the job did not finish, you could use ANALYZE to locate the track that was being processed when execution terminated. Assume that ICKDSF was executing on a 3380, and processing was interrupted on cylinder 201 head 1.

```
INSPECT UNIT(ccuu) NOVFY CHECK(2) SKIP -
       FROMRANGE(200,4) TORANGE(201,3) PRESERVE
```

If you ran the following job after processing has been interrupted as in JOB3 above, ANALYZE will determine and report the location of the track that was being processed when execution terminated. Note that the DRIVETEST parameter is valid only for direct-access storage devices with nonremovable storage media. See "ANALYZE Command—CKD" on page 73 for more information.

```
ANALYZE UNIT(ccuu) DRIVETEST
```

Placing a VTOC at the End of a Volume

In the following example, ICKDSF puts the VTOC at the end of the volume. This example allows ICKDSF to choose the location and size of the VTOC and is valid for any device.

```
INIT UNIT(ccuu) NOVFY VOLID(volser) VSEVTOC(END)
```

Note:

- ICKDSF places the VTOC on the last primary cylinder of the volume. The VTOC is one cylinder in length. This example is valid for all device types.

The following example allows ICKDSF to determine the location and size of the VTOC for an MVS system.

```
INIT UNIT(ccuu) NOVFY VOLID(volser) VTOC(END)
```

Note:

- ICKDSF places the VTOC on the last primary cylinder of the volume and the VTOC is one cylinder in length.

Clearing a Storage Path Fence Status

In the following example, ICKDSF clears a fence status. When this command is issued, all paths to all devices on the subsystem are cleared.

```
CONTROL  UNITADDRESS(162) CLEARFENCE
```

Notes:

- When you issue the CLEARFENCE parameter of the CONTROL command, it is assumed the path and/or the device which has been fenced off by the subsystem has been repaired.
- The CONTROL command with the CLEARFENCE parameter clears ALL paths to ALL devices on the subsystem.

Examples of ICKDSF Tasks—FBA Devices

The following examples show the commands and parameters needed for each task in a **stand-alone environment**. Default parameters are indicated only where they produce additional output; optional parameters for stand-alone environments are described briefly in the notes that follow. For complete descriptions of each command and examples of VSE operations, see the descriptions of the individual commands beginning with "ANALYZE Command—FBA" on page 175.

You may find it useful to select an example that most closely matches the task(s) you want to perform and use it as a model for your ICKDSF job. Examples of Device Support Facilities commands using fixed block architecture devices are presented in this section. Examples of Device Support Facilities commands using count-key-data devices are presented in

- "Examples of ICKDSF Tasks—CKD Devices" on page 37.

Further examples are presented after each command in

- "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71
- "Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices" on page 173

Initializing a New Device or HDA

Use the INIT command to prepare a new device or a replaced HDA for use. To surface check the volume for possible defects (not normally required for a new device), see "Checking Disk Surfaces" on page 57.

```
INIT UNITADDRESS(ccuu) NOVERIFY FBAVTOC(rbn,extent,cisize) -
     VOLID(serial) OWNERID(owner) NOMAP
```

Notes:

- Data on the volume is destroyed.
- This example establishes a volume ID, an owner ID, and a FBA VTOC on the volume.
- If you operate in VSE, add the PURGE parameter.

Initializing VM FBA Minidisks

Using ICKDSF under the CMS operating system of VM, or using the stand-alone version of ICKDSF you can use the INIT command to prepare a VM FBA minidisk for use.

```
INIT UNITADDRESS(ccuu) NOVERIFY FBAVTOC(rbn,extent,cisize) -
    VOLID(serial) OWNERID(owner)
```

Notes:

- This example initializes a minidisk at the minimal level, by establishing a volume ID, an owner ID, and a FBA VTOC on the minidisk.
- This example is initialized at the minimal level in the CMS or stand-alone environment.

Reinitializing a Volume

Volumes should be reinitialized at the minimal level.

```
INIT UNITADDRESS(ccuu) NOVERIFY VOLID(serial) -
     OWNERID(owner) FBAVTOC(rbn,extent,cisize) NOMAP
```

Note: The VTOC is rewritten and access to data on the volume is lost.

Investigating Suspected Drive Problems

If you suspect a problem with a drive, use the ANALYZE command to assist in determining:

1. If the drive can perform basic operating functions correctly
2. If all data can successfully be read from the disk surface

```
ANALYZE UNITADDRESS(ccuu) SCAN
```

Notes:

- The data on the volume is not altered.
- Add the SPEED parameter to scan the maximum number of data blocks on each pass. This option is not recommended for VM users; although SPEED makes the scanning process faster, it also keeps the drive busy and therefore degrades the performance of other VM users.
- Add the LIMITS parameter to specify only a limited area for scanning.
- Refer to "ANALYZE Command—FBA" on page 175 for information on ANALYZE output.

Checking Disk Surfaces

Use the INIT command to check the surface of all blocks during initialization by writing test patterns on the new volume and then reading them back. Any block flagged defective during this analysis is then assigned an alternate block (see also "Conditionally Assigning Alternate Blocks"). No attempt is made to reclaim defective blocks; refer to "Reclaiming Flagged Blocks" on page 58 for the recommended procedure.

```
INIT UNITADDRESS(ccuu) VERIFY(serial,owner) -
     FBAVTOC(rbn,extent,cisize) CHECK(3)
```

Notes:

- **The data on the volume is erased. If data is to be retained, it should be dumped before processing begins.**
- The CHECK(3) parameter ensures that the three data patterns available for surface checking are all used. A value higher than 3 merely repeats the three basic patterns. A value of 1 or 2 may be substituted and requires less run time; however, the surface checking is less thorough.
- If you operate in VSE, add the PURGE parameter.
- The default, MAP, prints a volume map of alternate blocks assigned during this process.

Conditionally Assigning Alternate Blocks

Use the INSPECT command to assign an alternate block only when the primary block is indicated to be defective as the result of surface checking. When PRESERVE is specified, the data is moved from the current block to the alternate block that is assigned.

```
INSPECT UNITADDRESS(ccuu) VERIFY(serial,owner) BLOCKS(rbn) -
        CHECK(3) NOPRESERVE|PRESERVE
```

Notes:

- The CHECK(3) parameter ensures that the three data patterns available for surface checking are all used. A value higher than 3 merely repeats the three basic patterns. A value of 1 or 2 may be substituted and requires less run time; however, the surface checking is less thorough.
- You may specify up to 20 blocks for conditional assignment by adding them to the BLOCKS parameter (separated by commas).
- NOPRESERVE causes the data on the block to be erased. Use this parameter only if there is no meaningful data on the block or if block errors prevent successful use of PRESERVE.

Unconditionally Assigning Alternate Blocks

Use the INSPECT command to unconditionally assign an alternate block regardless of the condition of the primary block. When PRESERVE is specified, the data is moved from the current block to the alternate block that is assigned.

```
INSPECT UNITADDRESS(ccuu) VERIFY(serial,owner) BLOCKS(rbn) -
        ASSIGN NOCHECK
```

Notes:

- If the inspected block has a current alternate, a new alternate is assigned.
- You may specify up to 20 blocks for unconditional assignment by adding them to the BLOCKS parameter (separated by commas).

Reclaiming Flagged Blocks

In general, a block should only be flagged defective if a known defect is found on it. If a volume contains multiple flagged blocks that no longer need to be flagged, the defective blocks may be reclaimed.

To reclaim blocks, first verify correct device operation using the ANALYZE command. Then, if there are no problems with the device, use the INIT command to reclaim previously flagged blocks.

```
ANALYZE UNITADDRESS(ccuu)

IF LASTCC ≤ 4 THEN -
INIT UNITADDRESS(ccuu) VERIFY(serial,owner) -
     FBAVTOC(rbn,extent,cisize) CHECK(3) RECLAIM
```

Notes:

- The INIT command **destroys** the data on the volume. If you want to save the data, dump it before processing begins.
- After all assigned alternate blocks (except factory-flagged blocks) become unassigned, the INIT command makes thorough surface check of all nondefective primary blocks and factory-assigned alternate blocks. Any block found defective and any block previously flagged defective at the factory are assigned alternate blocks.
- The CHECK(3) parameter ensures that the three data patterns available for surface checking are all used. A value higher than 3 merely repeats the three basic patterns. A value of 1 or 2 may be substituted and requires less run time; however, the surface checking is less thorough.
- If you operate in VSE, add the PURGE parameter.
- The default, MAP, prints a volume map of alternate blocks assigned during this process.
- If a maximal initialization, with RECLAIM specified, abends or terminates with an I/O error before reclaim processing completes, the volume being initialized may have one or more defective primary blocks with invalid alter-

nate block pointers. If this occurs, resolve the cause of the early termination and rerun the maximal initialization job, respecifying the RECLAIM parameter. Any future attempt to initialize the volume will force the maximal INIT RECLAIM to continue.

Printing a Block Assignment Map

Use the MAPALT command to prepare and print a detailed report of alternate block assignment status for the volume.

```
MAPALT UNITADDRESS(ccuu) DETAIL
```

Notes:

- The data on the volume is not altered.
- Add the LIMITS parameter to restrict the report to a specific range of blocks.
- To produce only a summary report, remove the DETAIL parameter.
- The INIT or INSPECT command can also be used to get a map of alternate blocks assigned.

Changing the Volume Serial Number and Owner Identification

Use the REFORMAT command to change the volume serial number and/or owner identification when the volume changes hands.

```
REFORMAT UNITADDRESS(ccuu) VERIFY(serial,owner) -
         VOLID(newserial) OWNERID(newowner)
```

Notes:

- You may specify either VOLID, or OWNERID, or both.
- No other data on the volume is changed.

Emulating a CKD Device on an FBA Device

Refer to "Emulating a CKD Device on an FBA Device" on page 43.

Resume Processing From a Checkpointed Location

The following examples assume you were INITIALIZING a volume with ICKDSF automatically taking checkpoints and processing is interrupted. Some examples of how you might resume processing from the last checkpointed location are illustrated. Checkpointing is assumed to occur every 20000 blocks.

The following example (let's call it JOB4) is surface checking an FBA device, and a checkpoint is taken last at block 5000 before processing is interrupted.

```
INIT UNIT(ccuu) NOVFY CHECK(3)
```

If you ran the following job next, after processing has been interrupted as in JOB4 above, you can surface check part of the volume:

```
INIT UNIT(ccuu) NOVFY CHECK(3) -
     BLOCKRANGE(3000,24000) CONTINUE VOLID(volser) NOMAP
```

Notes:

- The blocks starting at block 5000, and continuing for 20000 blocks are surface checked.
- No additional blocks are processed.
- Minimal initialize functions will be performed.

If you ran the following job next, after processing has been interrupted as in JOB4 above, you can perform a minimal initialization of the volume:

```
INIT UNIT(ccuu) NOVFY NOCHECK CONTINUE VOLID(volser) NOMAP
```

Notes:

- Blocks 5000 through 24999 are processed to ensure that the data fields are readable.
- Minimal initialize functions are then be performed.

If you ran the following job next, after processing has been interrupted as in JOB4 above, you can ensure processing starts from the beginning of the specified range:

```
INIT UNIT(ccuu) NOVFY CHECK(3) -
     BLOCKRANGE(3000,24000) NOCONTINUE VOLID(volser) NOMAP
```

Notes:

- Blocks 3000 through 24000 are surface checked.
- Blocks 24001 through 24999 are processed to ensure that the data fields are readable.
- Minimal initialize functions are then performed.

Preventing Accidental Destruction of Data

The following example (let's call it JOB5) surface checks an FBA device, and processing is interrupted on block 13656. PRESERVE specifies that the data is to be saved during surface checking and saves a copy of the data at the backup location as well as in storage.

```
INSPECT UNIT(ccuu) NOVFY CHECK(3) -
        BLOCKRANGE(5000,25000) PRESERVE
```

If you ran the following job next, after processing has been interrupted as in JOB5 above, you can surface check a different part of the volume:

```
INSPECT  UNIT(ccuu) NOVFY CHECK(3) -
         BLOCKRANGE(50000,100000) PRESERVE NOMAP
```

Notes:

- Block 13656 is surface checked. If any data existed for block 13656 when processing was interrupted, the data is rewritten on block 13656.
- Surface checking is then performed for all the blocks within the new range.

If you ran the following job next, after processing has been interrupted as in JOB5 above, you can ensure the usability of the block and recover the data.

```
INSPECT  UNIT(ccuu) NOVFY
```

Note:

- Block 13656 is surface checked. If any data existed for block 13656 when processing was interrupted, the data is rewritten on block 13656.

Specifying Part of a Volume

The following example surface checks all blocks from the beginning of the volume through block 25000. PRESERVE specifies that the data is to be saved during surface checking and saves a copy of the data at the backup location as well as in storage.

```
INSPECT UNIT(ccuu) NOVFY CHECK(1) -
        BLOCKRANGE(0,25000) PRESERVE NOMAP
```

The following example performs the data verification test for blocks 0 to 100000. The drive test is not performed. As BLOCKRANGE is identical to LIMITS you can substitute LIMITS for BLOCKRANGE.

```
ANALYZE UNIT(ccuu) SCAN NODRIVE BLOCKRANGE(0,100000)
```

How to Find Where a Job Failed, Using ANALYZE

The following example (let's call it JOB6) surface checks an FBA device, and processing is interrupted on block 13656. PRESERVE specifies that the data is to be saved during surface checking and saves a copy of the data at the backup location as well as in storage.

```
INSPECT UNIT(ccuu) NOVFY CHECK(3) -
       BLOCKRANGE(5000,25000) PRESERVE
```

If you ran the following job after processing has been interrupted as in JOB6 above, you can use ANALYZE to determine the location of the block that was being processed when execution was terminated. Note that DRIVETEST is only valid for devices which have non-removable storage media. See "ANALYZE Command—FBA" on page 175 for more information.

```
ANALYZE UNIT(ccuu) DRIVETEST
```

How to Place an FBAVTOC at the End of a Volume

In the following example, ICKDSF puts the FBAVTOC at the end of the volume. This example allows ICKDSF to choose the location and size of the FBAVTOC and is valid for any device.

```
INIT UNIT(ccuu) NOVFY VOLID(volser) FBAVTOC(END) NOMAP
```

Reclaiming the System Reserved Area of an IBM 9335 device

In this example, the primary and alternate blocks of the System Reserved Area of an IBM 9335 device are reclaimed.

```
CONTROL UNITADDRESS(141) RECLAIM(SYSAREA)
```

Notes on Using Device Support Facilities

Determining the level of your Device Support Facilities

To check that the level of your documentation matches the level of your code, Device Support Facilities displays the code level at the top line of each page that is printed on the output device. The top line is printed as follows:

```
ICKDSF-xxx DEVICE SUPPORT FACILITIES yyy
```

where:

```
xxx=VS2 (MVS/370)
    VSE (VSE/SP)
    XA (MVS/XA)
    SA (stand alone)
    CMS (ICKDSF/CMS)
yyy=Release (for example, 10)
```

Surface Checking Considerations

The surface checking functions performed by Device Support Facilities are not equivalent to the surface checking that is performed on a volume at the factory.

Marginal defects that exist on the surface of a track cannot be detected consistently by Device Support Facilities because of their intermittent nature. Therefore, different results can be obtained for the same track and/or volume from multiple runs of Device Support Facilities. Also, Device Support Facilities output can differ from other surface checking products because of different checking algorithms. See Appendix G, "Surface Checking," for more information.

Disk Surface Error Diagnosis

A prime use of Device Support Facilities is to maintain disk surfaces without requiring the assistance of a customer engineer. To do this efficiently, and as an aid in problem detection and analysis, you should be familiar with a few terms.

Data Errors

Data errors are errors detected in the bit pattern read from a disk.

When a data check occurs, there are two methods that may be attempted for automatic recovery:

- The data may actually be corrected.
- The data may be read correctly when the operation is retried.

All DASDs supported by Device Support Facilities add error detection information to each count, key, and data area of a record when it is written. Later, when the record is read, the information is used to detect errors that may be present and to correct the data where possible.

Error detection and correction information is a pattern of bits, coded to define a validity check for the full data in the recorded area. It is referred to as error correction code (ECC). When **ECC correctable**, the data is reconstructed to be the same as the data originally transferred to disk storage. The data correction may be done by the subsystem as data is transferred to main storage, or it may be done by the system after the data is in main storage. Although the data in main storage is correct for use in processing, the data on the disk is not changed. The next attempt to read the data from the disk may again result in a data error.

The other technique used for recovery from data errors is to retry the read operation by reissuing the command. Retry is sometimes done by the subsystem and sometimes by the system. The **retry** is done repeatedly in attempts to read the data correctly. If retry attempts are made a certain number of times without successfully reading the data, some disk storage types physically adjust the access mechanism. This causes the head to move to different positions across the track in attempts to better read the data signals. This technique is referred to as **retry with head offset**.

The method used for data error recovery depends on the disk storage type and the area that contains the error. (The error may be in the home address area of the track, in the count, key, or data area of a record, in a block, or in an identification field.)

Temporary or Permanent Errors

The terms **temporary** and **permanent** describe whether or not an error is recovered as seen by the program issuing the I/O operation. This error can be recovered either by the subsystem or by the operating system error recovery procedures (ERPs).

A temporary error is one that is recovered by correcting the data or by retrying the operation. In both cases, the operation in progress is completed.

A permanent error is one that is not correctable and cannot be recovered by retrying the operation.

Maintaining Disk Surfaces

Device Support Facilities surface checking procedures bypass ALL system recovery procedures, and as many of the subsystem recovery procedures as can be bypassed (this varies by subsystem). This ensures that errors are as visible as possible to the Device Support Facilities surface checking procedures; that is, ECC correctable data field errors for count-key-data devices are visible to Device Support Facilities, and subsystem retry with offset in the data field is bypassed for 3375, and 3380 devices.

Device Support Facilities actions during surface checking procedures vary with the characteristics of the device being processed, but it always executes with data error recovery bypassed. Because of the high visibility of data errors being enforced, Device Support Facilities may classify a data error as uncorrec-

table when the subsystem retry may actually be able to correct the error. This might include data errors that could be fixed by subsystem retry and/or subsystem offset, and also errors that could be fixed by system error recovery procedures. All other data error I/O failures are classified as correctable.

Count Key Data (CKD) Device Characteristics

Skip Displacement Devices

The IBM 3340, 3344, 3350, 3375, and 3380 devices contain "skip displacement areas" reserved on every track, and are called *skip displacement devices*.

Device Support Facilities can use skip displacement checking on tracks on these device types to allow the defective area of a track to be skipped over without affecting track capacity or performance. A track is not considered defective unless the entire skip displacement area for that track has been used and another defect is then detected.

For skip displacement devices, Device Support Facilities can perform extensive analysis on a track and possibly prevent the need for assigning alternates.

Nonskip Displacement Devices

The IBM 2305, 2311, 2314, 2319, 3330, and 3333 devices do not have the ability to skip defective track areas. These are referred to as *nonskip displacement devices*.

Device Support Facilities can perform read/write analysis for a track to verify that a defect exists. If an uncorrectable error is detected, the track is considered defective. If the error is correctable, a note of this fact is made on the volume map. Because software can correct the problem faster than the hardware can seek to an alternate track, Device Support Facilities will never consider a correctable error as a defective condition for nonskip displacement devices.

Fixed Block Architecture (FBA) Device Characteristics

For IBM 3370 FBA devices, assignment of an alternate block is generally done on the same physical cylinder on which the defective block resides, thereby minimizing any performance degradation experienced when seeking to an independent alternate cylinder.

A block is therefore considered defective for correctable data checks as well as for uncorrectable data checks when the correctable data check is made visible to Device Support Facilities.

For IBM 3310 FBA devices, correctable data checks cannot be made visible to Device Support Facilities. Uncorrectable data checks cause the block to be declared defective.

Emulated Devices

Device Support Facilities operates on count-key-data devices that might be emulated on other count-key-data devices (for example, a 3330 emulated on a 3350), and on count-key-data devices that might be emulated on fixed block architecture devices (for example, a 3330 emulated on a 3370).

CKD Emulated on CKD

For count-key-data devices emulated on other count-key-data devices, the emulation is transparent to the Device Support Facilities user. However, an emulated device works in the same manner as the device which is being emulated, not the native device. For example, if a 3330 is being emulated on a 3350, skip displacement is not performed as the 3330 does not support skip displacement.

In addition, the drive test function of the ANALYZE command, which would not work on the native version of the device (for example, a native 3330-1), will operate on the emulated version (for example, a 3330-1 emulated on a 3350). No special input, other than that required for any other invocation of ANALYZE, is required to run on an emulated device.

CKD Emulated on FBA

For count-key-data devices emulated on fixed block architecture devices, Device Support Facilities must be made aware of the situation. This is done by use of the MIMIC(EMUALL) or MIMIC(EMU(n)) parameter on the INIT or INSPECT command. There are also multiple steps that the user must perform *before* the emulated device is ready for Device Support Facilities. See "Emulating a CKD Device on an FBA Device" on page 43.

The drive test function of the ANALYZE command does not operate on count-key-data devices being emulated on fixed block architecture devices.

The CONTINUE function of the INIT command and the recovery backup function of the INSPECT command are not supported on a 3340 emulated on a 3370.

Syntax of the Device Support Facilities Commands

The illustrations of command formats in this book use the following conventions:

- Brackets ([]) indicate an optional field or parameter.
- Braces ({}) indicate that one of alternative items must be selected.
- Items separated by a vertical bar (|) represent alternative items. Only one item may be selected.
- An ellipsis (•••) indicates that multiple entries of the type immediately preceding the ellipsis are allowed.
- **BOLDFACE UPPERCASE** type indicates the exact characters to be entered. Such items must be entered as shown or in the abbreviated form allowed.
- **UNDERSCORED BOLDFACE UPPERCASE** type indicates a default option. If the parameter is omitted, the default value is assumed.
- *italics* indicates fields to be supplied by you.
- Parentheses, commas, slashes, spaces, and other punctuation must be entered as shown.
- Numeric parameters can be coded in two ways: decimal or hexadecimal (X'n'). This applies to any parameter that specifies a substitution of decimal numbers.
- Character string parameters can be coded either with or without enclosing them in single quotation marks. However, if delimiters such as commas, blanks, or parentheses appear within a character string, the string must be enclosed within single quotation marks.

You can specify values with some keyword parameters. The value is entered within parentheses following the keyword parameter. A typical keyword with a value appears in this book as:

VOLID(*serial***)**

A value cannot contain commas, semicolons, blanks, parentheses, or slashes unless the entire value is enclosed within single quotation marks. (A single quotation mark in a field enclosed within quotation marks must be coded as two single quotation marks.)

Positional Parameters: In the explanations of commands within this book, the positional parameters are shown in *italics*. When a positional parameter consists of a list of items, you must enclose the list in parentheses.

Note: When you are using ICKDSF in the MVS/370, MVS/XA, or VSE environment the commands and comments must be entered in uppercase.

Abbreviating Commands and Parameters

Some of the commands and parameters can be abbreviated. Abbreviations for commands are listed just before the syntax for each command. Abbreviations for parameters are shown under the heading "Abbreviations." The full form of the parameter is given first, followed by acceptable abbreviations. A vertical bar is used to separate the parameter and its abbreviations.

Delimiters

When you issue a command, you must separate the command name from the first parameter by one or more blanks. You must separate parameters from each other by one or more blanks or a comma. Do **not** use a semicolon as a delimiter, because any characters that follow a semicolon are ignored.

Line Continuations

A command, IF statement, or comment can be entered using one or more lines. To continue a command or comment that requires more than one line, each line, except the last, must be terminated by a plus (+) or a minus (-) character. This line continuation character must occur in the last nonblank column at or before column 72, and must be preceded by a space. The minus character causes leading blanks in the following line to be read as part of the command. The plus character causes leading blanks in the following line to be ignored.

Note: When using an 80-column card format, the Device Support Facilities commands must be entered within columns 2 and 72. If the length of the command exceeds column 72, a continuation character must be entered, and the command continued on the next line. When commands are entered at the console in a stand-alone environment, they can begin at the left margin and are also subject to the 72-column maximum.

Comments

You can add comments to any of the commands wherever a blank character can appear. To distinguish your comments from the commands, enter them within the comment delimiters /* and */. A comment may be continued onto the next line by using a line continuation character at the end of the line.

Terminator

The terminator indicates the end of the command. The terminator can be either an end-of-command condition (that is, no continuation character) or a semicolon (;). If you use the semicolon as the terminator, the semicolon cannot be enclosed in quotation marks or be embedded in a comment. Everything to the right of the semicolon is ignored. If there is information to the right of the semicolon that is continued to another record, all such information, including the continued information, is ignored.

For example, if you code:

```
REFORMAT UNITADDRESS(353) VOLID(AA2314); VFY(BB2314,SMITH) -
INIT
REFORMAT
```

characters following the semicolon terminator are ignored. The continuation (minus) character at the end of the first record causes the INIT command to be ignored also. The two REFORMAT commands are the only commands that are recognized.

Conflicting Keyword Parameters

Conflicting keyword parameters are identified by a logical OR sign (|) representing alternative items. If conflicting keyword parameters are entered, execution of the command terminates.

Using the Device Support Facilities Commands

A command consists of a command name followed by one or more operands or parameters. Hereafter, all operands described in this book are called parameters. They provide the specific information required for the command to perform the requested function.

For example, some parameters associated with the INSPECT command indicate where the volume is mounted, which tracks are to be inspected, and whether the volume serial number and owner identification should be verified.

The following is an example of this command:

```
INSPECT DDNAME(VOL1) TRACKS((4,6),(4,10)) VERIFY(DSF123,SMITH)
```

where:

- DDNAME specifies the DD statement that contains the description of the device.
- TRACKS specifies that cylinder 4, tracks 6 and 10 are to be inspected.
- VERIFY specifies that volume serial number DSF123 and owner identification SMITH are to be verified before inspection continues.

Several of the parameters take default values if they are not specified when the command is issued. In the foregoing example, the following parameters were not specified, but are defaulted:

- CHECK(1) indicates that track surfaces are checked for media defects.
- NORECLAIM indicates that no attempt is made to reclaim tracks that were previously flagged defective.
- PRESERVE indicates that existing data on the inspected tracks is saved and rewritten.
- MAP indicates that a volume map is printed after the command processes.
- ASSIGN indicates that alternate tracks are assigned for the specified tracks if they are shown to be defective as a result of surface checking.

Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices

This part of the book is intended as a reference when you specify the Device Support Facilities commands for count-key-data (CKD) devices. See "Devices You Can Use" on page 8 for a list of count-key-data devices supported by Device Support Facilities (ICKDSF).

"Examples of ICKDSF Tasks—CKD Devices" on page 37 and this part of the book are guides to help you determine why, when, and how to run Device Support Facilities.

This part describes the Device Support Facilities commands for CKD devices. The commands are listed in alphabetic order. "Syntax of the Device Support Facilities Commands" on page 67 describes the syntax of the commands. The presentation of each command includes:

- A general description of the command and its use
- A table summarizing the syntax of the command
- An explanation of required and optional parameters
- Examples

Device Support Facilities Functions

The following Device Support Facilities commands operate on count-key-data (CKD) devices: ANALYZE, BUILDIX, CONTROL, INIT, INSPECT, INSTALL, and REFORMAT.

- ANALYZE helps to detect and differentiate recording surface and drive-related problems on a volume. It can also scan data to aid in detecting the existence of possible media problems.
- BUILDIX changes a standard MVS-format VTOC into an indexed VTOC, or an indexed VTOC into a standard MVS-format VTOC.
- The CONTROL command with the ALLOWWRITE parameter allows subsequent writes to the devices controlled by an IBM 3880 or IBM 3990 Storage Control if that storage control has been WRITE INHIBITed by the error recovery procedures at the channel, director, or storage control level.

  The CONTROL command with the CLEARFENCE parameter (IBM 3990 only) allows you to clear a fence status of a path and/or device that has been fenced off by the subsystem.
- INIT prepares a direct-access storage volume so that it can be used in an operating system environment or it reinitializes a volume that was previously prepared.
- INSPECT provides surface checking functions for defective tracks on a volume.
- INSTALL performs the procedures neccessary for installation, head-disk assembly (HDA) replacement, and physical movement of IBM 3380 DASD.
- REFORMAT can be used to update or replace information on a previously initialized volume. The volume label, owner ID, IPL bootstrap, and IPL program records can be updated.
- REVALidate combines the track validation functions of medial initialization with the problem determination and data verification functions of the ANALYZE command and also the INSPECT functions, if required.

Device Support Facilities performs two basic functions: volume formatting and disk surface maintenance.

- Volume formatting makes a volume usable by an operating system and provides a means for updating old volume formats. This is also known as "initializing" a volume.

  Newly installed volumes and replaced head disk assemblies (HDAs) will always require formatting.
- Disk surface maintenance aids in detecting problems with device operation and attempts to correct data errors that can be traced specifically to disk surface defects.

ANALYZE Command—CKD

The ANALYZE command is used to examine the drive and/or the user's data on a volume to help determine if errors exist. The output is intended to aid in distinguishing between drive problems (on nonremovable media) and media problems, in addition to providing assistance in locating and fixing the problem.

There are two basic functions of the ANALYZE command, the drive test, and the data scan.

The Drive Test

The drive test provides a general exercising of the storage device by issuing SEEK, READ, and WRITE commands. (Note that WRITE commands are issued to the CE cylinder only.)

It is to be used with IBM direct-access storage devices (DASD) that have nonremovable storage media. These include the IBM 3344, 3350, 3375, 3380 and any emulated version of the IBM 3330 or 3340. Note that the drive test does not support the IBM 2305.

The drive test is not supported if you are running ICKDSF under VM unless the device is ATTACHed.

The drive test runs with all system error recovery procedures disabled.

ANALYZE determines that the drive is fully operational by performing I/O operations on the CE track which test the drive's functional capabilities. Each I/O operation, called a channel command word (CCW) chain, tests a specific function of the drive and logical volume. Tests are executed in order of increasing complexity.

Drive problems that can be detected during initial ANALYZE tests are access arm positioning errors and the ability to read and write (on the CE cylinder). The drive is also checked to ensure that it can detect an address mark, can switch read/write heads as a result of a multitrack command, can sense the disk's rotational position, and can detect and skip over defective areas on a track.

If PRESERVE or CONTINUE data exists on the CE cylinder, ANALYZE does not destroy this data (this data is written by an INSPECT or INIT). ANALYZE bypasses writing over the data, and reports the presence of the data. As this feature is part of the drive test, it supports only nonremovable storage media.

The drive test is optional. It is controlled by the DRIVETEST/NODRIVETEST parameters.

The Data Verification Test

The data verification test reads (without data transfer across the channel to the processor) all data records on the volume (minidisk if you are running ICKDSF under CMS).

Before running the ANALYZE data verification test all IBM 3380 Models must be powered on with all devices in the ready condition and all covers closed for 2 hours 30 minutes to establish thermal stability.

Options for the data verification tests include:

- Specifying data verification of the whole volume.

  If you are running ICKDSF under CMS, you can verify the whole minidisk.

- Selecting a range of data to be verified:
  - By cylinders (LIMITS or CYLRANGE)
  - CCHH to CCHH (FROMRANGE,TORANGE)
  - All heads within a range of cylinders, for example, CC to CC (CYLRANGE or LIMITS)
  - All cylinders for certain heads, for example, head 3 for all cylinders (HEADRANGE).
  - A range of heads within a range of cylinders, for example, heads 2 through 4 on cylinders 10 through 20 (CYLRANGE and HEADRANGE). Similarly, if you are running ICKDSF under CMS, you can select a range to be verified within a minidisk.
- Selecting the SPEED or NOSPEED parameter on the data verification tests.

  SPEED specifies:
  - Perform one I/O per cylinder for the count-key-data devices.

  NOSPEED specifies:
  - Perform one I/O per track for the count-key-data devices.

The intent of the data verification test is to scan the CURRENT DATA on the volume for data checks.

Data verification does not ensure that the volume is in standard IBM format.

ANALYZE data verification executes to a successful completion regardless of whether, one or more of the following is true:

- No data exists on the volume.
- Data on the volume is written as nonstandard record zeros.
- There are no record zeros on the volume.

Data Verification Output

ANALYZE will report the cylinder and head location of ALL ECC correctable data checks, in addition to the cylinder and head location of ECC uncorrectable data checks, followed by the CCW, CSW, and sense bytes of the failing I/O.

The data verification portion of ANALYZE executes without system error recovery procedures (regardless of the environment you are running in), and with storage control retry inhibited where possible.

If the surface defect is of the type that can be corrected by storage control retry (that is, an error in a count field or a key field), ANALYZE will report the error as uncorrectable.

Subsequent I/O against the data in a standard operating system environment might detect the error as correctable, or might not detect the error at all.

"Unexpected" I/O errors

If the data verification test encounters an "unexpected" I/O error (that is, an equipment check, environmental data, and so forth) depending upon the severity of the error, ANALYZE might:

- Issue a message and stop processing (for example, equipment check)
- Issue a message and continue (for example, invalid track format)
- Ignore the error and continue (for example, environmental data)

RANGE Parameters: Specifying Part of a Volume

LIMITS, CYLRANGE/HEADRANGE, and FROMRANGE/TORANGE are optional parameters, used to limit the areas of a volume or minidisk that are to be scanned. Use the CYLRANGE/HEADRANGE pair to specify the starting and ending cylinder(s) and the starting and ending head(s) to be scanned. Use the FROMRANGE/TORANGE pair to specify the starting track (cylinder and head) and the ending track (cylinder and head) to be scanned.

The following notes apply to specifying part of a volume:

- LIMITS, CYLRANGE/HEADRANGE, and FROMRANGE/TORANGE are mutually exclusive parameters. LIMITS and CYLRANGE can be used interchangeably.
- LIMITS, CYLRANGE/HEADRANGE, and FROMRANGE/TORANGE are only valid if SCAN is specified.
- If you specify CYLRANGE and do not specify HEADRANGE, all the heads of the specified cylinders are processed.
- If you specify HEADRANGE and do not specify CYLRANGE, the specified heads for all the cylinders on the volume (minidisk under CMS) are processed.
- Both the starting and ending values are required for CYLRANGE and HEADRANGE. If you specify an ending value greater than the maximum for the device, the device (minidisk under CMS) maximum is used for the ending value.
- If you specify FROMRANGE and do not specify TORANGE, TORANGE is defaulted to the last cylinder of the device (minidisk under CMS).

- If you specify TORANGE and do not specify FROMRANGE, FROMRANGE is defaulted to cylinder 0 track 0 (starting cylinder and track of the minidisk under CMS)
- If you do not specify LIMITS or CYLRANGE, the default is ALL if SCAN is specified. If you are running under CMS and do not specify a range, the default is all of the specified minidisk.

Using the ANALYZE Command under CMS

When you run ICKDSF under CMS, you can run in two modes:

- Minidisk
- ATTACHed

Minidisk Support under CMS

Under CMS you can use the ANALYZE command to perform the data verification test.

The following parameters are valid for minidisks under CMS:

- ALL, CYLRANGE, FROMRANGE, HEADRANGE, LIMITS, NODRIVETEST, SCAN/NOSCAN, SPEED/NOSPEED, TORANGE, and UNITADDRESS

The following parameters are valid *only* when you are under CMS and have DEVMAINT authority (as defined in the CP directory):

- REALADDR, and USERID

You can use these parameters as follows:

USERID: When you are under CMS and have DEVMAINT authority and specify the USERID parameter you can perform the data verification test on another user's minidisk. The following parameters are valid in this mode:

- ALL, CYLRANGE, FROMRANGE, HEADRANGE, LIMITS, NODRIVETEST, SCAN/NOSCAN, SPEED/NOSPEED, TORANGE, UNITADDRESS, and USERID

REALADDR: When you are under CMS and have DEVMAINT authority you can perform the data verification test on a volume when you specify the real device address using the REALADDR parameter. The following parameters are valid in this mode:

- ALL, CYLRANGE, FROMRANGE, HEADRANGE, LIMITS, NODRIVETEST, SCAN/NOSCAN, SPEED/NOSPEED, and TORANGE

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed to your user ID. When you use ICKDSF in this mode, you can perform any ICKDSF function which can be performed when you are running ICKDSF stand-alone.

Syntax

Command	Required Parameters
ANALYZE	**DDNAME(***dname***)|SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)|REALADDR(***ccuu***)**
	Optional Parameters
	ALL **DRIVETEST|NODRIVETEST** **FROMRANGE(***cccc,hhhh***)** **HEADRANGE(***hhhh,hhhh***)** **LIMITS(***scanlo,scanhi***)|CYLRANGE(***cccc,cccc***)** **MSS** **SCAN|NOSCAN** **SPEED|NOSPEED** **TORANGE(***cccc,hhhh***)** **USERID(***user's ID***)**

Required Parameters

DDNAME|SYSNAME|UNITADDRESS|REALADDR: Identifying the Device on Which the Volume is Mounted

Explanation: **DDNAME|SYSNAME|UNITADDRESS|REALADDR** are required parameters identifying the volume (or minidisk under CMS) that is to be analyzed.

DDNAME(*dname*)
Required when analyzing a volume that is mounted online. The device on which the volume is mounted must be online.

For *dname*, specify the MVS JCL statement that identifies the device on which the volume is mounted.

SYSNAME(*sysxxx*)
Required when analyzing a volume in the VSE environment. For *sysxxx*, specify the SYSNAME in the ASSGN system control statement.

UNITADDRESS(*ccuu*)
Required when analyzing a volume that is mounted offline or when analyzing a volume in a stand-alone environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: There must be an available path to the device.

Under CMS use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual minidisk address.

REALADDR(*ccuu*)
Required to specify the real address of a volume. This parameter is only valid when you are running ICKDSF under CMS and you have DEVMAINT authority.

For *ccuu*, specify the real address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Abbreviations:

DDNAME|DNAME
UNITADDRESS|UNITADDR|UNIT
REALADDR|RADDR

Default: None

Restrictions: DDNAME or SYSNAME cannot be specified when analyzing a volume in a stand-alone environment or CMS environment. UNITADDRESS cannot be specified when analyzing a volume in an online or VSE environment.

REALADDR is mutually exclusive with UNITADDRESS, DDNAME, SYSNAME, and USERID.

Optional Parameters

ALL: Specifying the Area of Data Verification

Explanation: **ALL** is an optional parameter that specifies that all cylinders are to be read during the data verification tests.

When you are running ICKDSF under CMS, ALL specifies all cylinders of the minidisk.

Abbreviations: None

Default: ALL is the default if LIMITS or CYLRANGE is not specified.

Restrictions: ALL applies only when the SCAN parameter has been specified.

ALL cannot be specified with FROMRANGE/TORANGE, CYLRANGE/HEADRANGE, or LIMITS.

DRIVETEST|NODRIVETEST

Explanation: **DRIVETEST**|**NODRIVETEST** are optional parameters specifying whether or not the drive test function of the command should be executed.

DRIVETEST
Specifies that the drive test should be executed. This parameter is valid only on IBM direct-access storage devices (DASD) that have nonremovable storage media. These include the 3344, 3350, 3375, 3380; and any emulated version of the 3330 or 3340.

If DRIVETEST is specified for a nonsupported device, a warning message is issued and processing continues.

DRIVETEST is bypassed if you are using ICKDSF under CMS (unless the device is ATTACHed).

NODRIVETEST
Specifies that the drive test should be bypassed.

Abbreviations:

DRIVE
NODRIVE

Default: DRIVETEST

Restrictions: NODRIVE NOSCAN performs no functions but will complete without error.

When executing in a VM environment, DRIVETEST is valid only for devices that are ATTACHed to the current userid. I/O errors will occur if the drive test is attempted for LINKed devices or minidisks.

FROMRANGE: Specify a Specific Starting Location

Explanation: **FROMRANGE**(*cccc,hhhh*) is an optional parameter used to specify that part of a volume is to be scanned. Use this parameter to specify the starting track (cylinder and head) to be scanned.

For *cccc,hhhh* substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the cylinder and head at which processing is to start.

If you specify FROMRANGE and do not specify TORANGE, TORANGE is defaulted to the last cylinder of a volume or minidisk.

For information on specifying the ending track (TORANGE) to go with the starting track, see "TORANGE: Specify a Specific Ending Location" on page 82.

For information on specifying part of a volume to be examined, see "RANGE Parameters: Specifying Part of a Volume" on page 75.

Abbreviations:

FROMRANGE|FROM

Default: None

Restrictions: FROMRANGE cannot be specified with CYLRANGE/HEADRANGE.

FROMRANGE applies only when SCAN is specified.

HEADRANGE: Specify a Range of Heads

Explanation: **HEADRANGE**(*hhhh,hhhh*) is an optional parameter used to specify that part of a volume is to be scanned. Use the HEADRANGE parameter to specify the starting and ending head(s) to be scanned.

For *hhhh,hhhh* substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the starting and ending heads to be examined.

If you specify HEADRANGE and do not specify CYLRANGE, the specified heads for all the cylinders on the volume are processed. If you are running ICKDSF under CMS and default CYLRANGE, all the cylinders on the minidisk are processed.

For information on specifying the starting and ending cylinder(s) (CYLRANGE) to go with the starting and ending heads, see "LIMITS|CYLRANGE: Specifying the Area of Data Verification" on page 80.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Part of a Volume" on page 75.

Abbreviations:

HEADRANGE|HDRANGE|HD

Default: None

Restrictions: Both the starting and ending values are required for HEADRANGE. If you specify an ending value greater than the maximum for the device, the device maximum is used for the ending value.

HEADRANGE applies only when you specify SCAN.

HEADRANGE cannot be specified with FROMRANGE/TORANGE.

Because SPEED operates on a cylinder at a time, it is not valid with HEADRANGE, and is ignored if both HEADRANGE and SPEED is specified.

LIMITS|CYLRANGE: Specifying the Area of Data Verification

Explanation: **LIMITS(***scanlo,scanhi***) |CYLRANGE(***cccc,ccc***)** are optional parameters to specify the area of the disk where data verification is to be performed. Specify the range of cylinders that are to be read during the data verification tests.

LIMITS (*scanlo,scanhi***)**

scanlo specifies the relative cylinder number for the start of the data verification tests. *scanlo* can be expressed in decimal (n) or hexadecimal (X'n') form.

scanhi specifies the relative cylinder number of the last cylinder of the data verification tests. *scanhi* can be expressed in decimal (n) or hexadecimal (X'n') form.

Also, the numbers *scanlo* and *scanhi* must be equal or in ascending order.

CYLRANGE(*cccc,cccc***)**

For *cccc,cccc*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the starting and ending cylinders to be examined.

If you specify CYLRANGE and do not specify HEADRANGE, all the heads of the specified cylinders are processed.

For information on specifying the starting and ending heads (HEADRANGE) to go with the starting and ending cylinders, see "HEADRANGE: Specify a Range of Heads" on page 79.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Part of a Volume" on page 75.

Abbreviations:

```
LIMITS|LIMIT|LIMS|LIM
CYLRANGE|CYL
```

Default: If you do not specify LIMITS or CYLRANGE, ALL is the default (if SCAN is specified). If you are running under CMS on a minidisk and do not specify a range, the default is all the specified minidisk.

Restrictions: The LIMITS/CYLRANGE parameters apply only when the SCAN parameter has been specified.

Both the starting and ending values are required for LIMITS or CYLRANGE. If you specify an ending value greater than the maximum for the device, the device maximum is used for the ending value.

LIMITS or CYLRANGE cannot be specified with FROMRANGE/TORANGE.

MSS:

Explanation: **MSS** is an optional parameter to be specified only when ANALYZE is executed against a 3330 staging volume that is currently offline to the MSS.

By specifying this parameter, the high cylinder value of the LIMITS parameter (and the default high cylinder value if ALL is specified) can be expanded to include one additional track, as follows:

- 408 for a 3330-1
- 808 for a 3330-2,11

Abbreviations: None.

Default: None.

If MSS is not specified for a staging volume, the last cylinder is not scanned.

If both MSS and NOSCAN are specified, MSS is ignored.

SCAN|NOSCAN: Run Data Verification Tests

Explanation: **SCAN**|**NOSCAN** are optional parameters specifying whether the data verification test is to be run.

Data verification is run if SCAN is specified and any of the following is true:

- The drive test completed successfully.
- The drive test is not supported for the input device type.
- The drive test has been bypassed

SCAN
Indicates that you want the data verification test to be run.

NOSCAN
Indicates that you do not want the data verification test to be run.

Abbreviations:

SCAN|SCN
NOSCAN|NOSCN|NSCAN|NSCN

Default: NOSCAN

SPEED|NOSPEED: Specify Data Verification Tests

Explanation: **SPEED**|**NOSPEED** are optional parameters which specify how much is read by each I/O for the data verification tests.

SPEED
One cylinder is read at a time.

NOSPEED
One track is read at a time.

Abbreviations:

NOSPEED|NSPEED.

Default: NOSPEED.

Restrictions: This option is not recommended for VM users, because this may degrade the performance of other users.

Because SPEED operates one cylinder at a time, it improves the performance of the scanning process, but, at the same time, it can severely degrade the performance of other users accessing the channel. This impact should be considered. SPEED is only valid when SCAN is specified.

TORANGE: Specify a Specific Ending Location

Explanation: **TORANGE**(*cccc,hhhh*) is an optional parameter used to specify that part of a volume is to be surface checked. Use this parameter to specify the ending track (cylinder and head) to be examined before minimal volume initialization is performed.

For *cccc,hhhh*, substitute the 1 to 4 decimal (n) or hexadecimal (X'n') digits to identify the cylinder and head where processing is to end.

If you specify TORANGE and do not specify FROMRANGE, FROMRANGE is defaulted to cylinder 0 track 0. If you are running under CMS and do not specify FROMRANGE, the default is cylinder zero.

For information on specifying the starting track of part of a volume (FROMRANGE) see "FROMRANGE: Specify a Specific Starting Location" on page 79.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Part of a Volume" on page 75.

Abbreviations:

TORANGE|TO

Default: None

Restrictions: None

USERID: Specify Another User's Minidisk

Explanation: **USERID**(*user's ID*) is an optional parameter used to specify that the ANALYZE data verification test is to be performed on another user's minidisk.

For *user's ID*, substitute the 1 to 8 characters of the user ID of the owner of the minidisk you want to verify.

Abbreviations:

USERID|UID

Default: If USERID is not specified your own minidisk is verified.

Restrictions: USERID can only be specified if you are running ICKDSF under CMS and have DEVMAINT authority. USERID is ignored in all other system environments.

USERID is mutually exclusive with REALADDR

Examples of the ANALYZE Command

The following examples show different ways to use the ANALYZE command.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Analyzing a Volume in a Stand-alone Environment

In this example, only the Direct Access Storage drive tests are performed.

```
ANALYZE UNITADDRESS(141)
```

Using keyword abbreviations, you could have specified:

```
  ANALYZE UNIT(141)
```

Analyzing a Volume in a Stand-alone Environment

In this example, the Direct Access Storage drive tests are performed and are followed by the data verification tests for the entire volume. ALL is the default.

```
ANALYZE UNITADDRESS(141) SCAN
```

Analyzing a Volume in a Stand-alone Environment

In this example, the Direct Access Storage drive tests are bypassed. A partial data verification test, starting at relative cylinder 6 and ending at relative cylinder 9, is performed.

```
ANALYZE UNITADDRESS(0351) LIMITS(6,9) SCAN NODRIVETEST
```

Analyzing a Partial Volume

In this stand-alone environment example, the data verification test is performed for all heads starting at cylinder 500 head 14, to the end of the volume.

```
ANALYZE UNITADDRESS(351) SCAN -
        FROMRANGE(500,14) TORANGE(9999,9999)
```

Analyzing Another User's Minidisk under CMS

In this CMS environment example, the data verification test is performed for another user's minidisk. You specify the USERID parameter to ANALYZE the minidisk which is owned by user SMITH at the user's virtual address 0351. You must have DEVMAINT authority to specify the USERID parameter.

```
ANALYZE UNIT(0351) SCAN USERID(SMITH) NODRIVETEST
```

Analyzing a Volume under CMS Specifying a Real Address

In this CMS environment example, the data verification test is performed for a volume at real address 290 by using the REALADDR parameter. You must have DEVMAINT authority to specify the REALADDR parameter.

```
ANALYZE REALADDR(290) SCAN NODRIVETEST
```

Analyzing an Emulated Volume in Online MVS

In this example, VOLUMEG is a 3330 emulated on a 3350. VOLUME2 is a native 3330.

For VOLUMEG, the drive test is performed, and, if the drive test is successful, data verification of cylinders 2 through 5 is performed.

For VOLUME2, the drive test is automatically bypassed, and data verification of cylinders 2 through 5 is performed.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//VOLUMEG  DD    UNIT=3330,DISP=SHR,VOL=SER=PAY345
//VOLUME2  DD    UNIT=3330,DISP=SHR,VOL=SER=PAY320
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 ANALYZE DDNAME(VOLUMEG) SCAN LIMITS(2,5)
 ANALYZE DDNAME(VOLUME2) SCAN -
         LIMITS(2,5)
/*
```

Analyzing a Volume in a VSE Environment

In this example, drive tests are performed on volume SYS001. Specification of the NOSCAN parameter indicates that data verification tests are not to be performed.

```
// JOB   jobname
// ASSGN SYS001,150
// EXEC  ICKDSF,SIZE=AUTO
   ANALYZE SYSNAME(SYS001) NOSCAN
/*
/&
```

Analyzing an MSS Volume in an MVS Environment (Offline)

In this example, the data verification test is being done on an MSS staging volume that is offline to the MSS. Data verification includes what would otherwise be the first alternate cylinder.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
  ANALYZE UNIT(164) SCAN MSS NODRIVE
/*
```

Diagnostic Messages

ANALYZE prints diagnostic messages to the output device. They can be used by a programmer (and by the customer engineer or service representative) to aid in determining if and what type of problem might exist on the volume.

The informational and diagnostic messages are listed and explained in Appendix C, "Device Support Facilities Messages (ICK)" on page 233.

In addition, ANALYZE produces a movable head and/or fixed head error table that summarizes errors associated with the read/write circuitry and/or data verification. See Figure 9 on page 86 or Figure 10 on page 87, respectively.

Note: The fixed head error table may be printed although the drive under test does not include the fixed-head feature. In that case, Figure 11 on page 88 can be used to map the fixed head numbers to the physical movable cylinder and head.

MOVABLE HEAD ERROR TABLE

HEAD NUMBER	DATA CHK	SEEK VERIFY CHECK	WRITE CHK	DATA CHK CE CYL	DATA COMP ERROR
00	---TUC----				
01					
02					
03	----U-----				
04					
05	---T-C----				
06					
07	-----C----				
08					
09	---TU-----				
11					
12					
13	---T------				
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28	-----C----	------X------	-----X-----	----X-----	-----X----
29	----U-----	------X------	-----X-----	----X-----	-----X----

In the data check column:

"T" indicates which head(s) exceeded the data check threshold error rate.

"U" indicates that at least one ECC uncorrectable error occurred on this head.

"C" indicates that at least one ECC correctable error occurred on this head.

"X" indicates that at least one error, as indicated by the column heading, has occurred on this head.

The actual number of errors that occurred for a specific head can be obtained by examining previous messages provided in the ANALYZE output.

Figure 9. The Movable Head Error Table—CKD

FIXED HEAD ERROR TABLE

HEAD NUMBER	DATA CHECK	SEEK VERIFY CHECK	HEAD NUMBER	DATA CHECK	SEEK VERIFY CHECK
00	U	X	30	X	X
01	C	X	31	X	X
02			32		
03			33		
04			34		
05			35		
06			36		
07			37		
08			38		
09			39		
10			40		
11			41		
12			42		
13			43		
14			44		
15			45		
16			46		
17			47		
18			48		
19			49		
20			50		
21			51		
22			52		
23			53		
24			54		
25			55		
26			56		
27			57		
28			58		
29			59		

In the data check column:

"U" indicates that an ECC uncorrectable error occurred on this head.

"C" indicates that an ECC correctable error occurred on this head.

"X" indicates that at least one error, as indicated by the column heading, has occurred on this head.

Figure 10. The Fixed-Head Error Table

Note: The fixed-head error table may be printed although the drive under test does not include the fixed-head feature. In that case, the table applies to movable heads 00 through 29.

The fixed-head error table could be printed for a 3350 in native or compatibility mode regardless of whether the fixed-head feature exists. This may happen if an intermittent error occurred on the movable heads during the fixed head-test. If this occurs, see the conversion table in Figure 11 on page 88 to determine which physical movable-head number corresponds with the fixed-head number in the fixed-head error table.

Head Number From Fixed Head Error Table	Physical Movable Head Number		Head Number From Fixed Head Error Table	Physical Movable Head Number	
	3350[1]	3330 (compatibility modes)[2]		3350[3]	3330 (compatibility modes)[4]
0	0	20	30	0	11
1	1	21	31	1	12
2	2	22	32	2	13
3	3	23	33	3	14
4	4	24	34	4	15
5	5	25	35	5	16
6	6	26	36	6	17
7	7	27	37	7	18
8	8	28	38	8	0
9	9	29	39	9	1
10	10	20	40	10	2
11	11	21	41	11	3
12	12	22	42	12	4
13	13	23	43	13	5
14	14	24	44	14	6
15	15	25	45	15	7
16	16	26	46	16	8
17	17	27	47	17	9
18	18	28	48	18	10
19	19	0	49	19	11
20	20	1	50	20	12
21	21	2	51	21	13
22	22	3	52	22	14
23	23	4	53	23	15
24	24	5	54	24	16
25	25	6	55	25	17
26	26	7	56	26	18
27	27	8	57	27	Unused
28	28	9	58	28	Unused
29	29	10	59	29	Unused

[1] The head numbers listed below are on physical cylinder 1.

[2] The first 10 head numbers listed below (20 through 29) are on physical cylinder 0. The next 20 head numbers (20 through 28 and 0 through 10) are on physical cylinder 1.

[3] The head numbers listed below are on physical cylinder 2.

[4] The first 8 head numbers listed below (11 through 18) are on physical cylinder 1. The next 19 head numbers (0 through 18) are on physical cylinder 2.

Figure 11. Mapping of Fixed-Head Numbers to Physical Movable-Head Numbers if Fixed-Head Feature Not Present

BUILDIX Command—CKD

The BUILDIX command builds a VTOC index. Direct access volumes in the MVS environment may optionally be changed from the OS format VTOC (OSVTOC) to the indexed format VTOC (IXVTOC) by invoking the BUILDIX command. BUILDIX will build a VTOC index data set and prepare the VTOC to show that the volume is in IXVTOC format. BUILDIX will also change a volume in IXVTOC format to OSVTOC format.

To build an index data set on a volume, the BUILDIX command requires that the host system contain indexed VTOC programming support. Execution of the command will terminate without action if this support is not present.

Appendix E, "The VTOC Index" on page 305, contains more information about indexed format VTOCs.

Restrictions

Volumes containing split cylinder extents are not supported by the BUILDIX command. VSE volumes will be converted to OSVTOC format or to IXVTOC format, according to the parameters specified.

The BUILDIX command is valid only in the MVS/370 and MVS/XA environments.

When the Storage Management Subsystem (SMS) is active, the BUILDIX command is not supported for SMS-managed volumes.

Processing in a Shared Environment

When using BUILDIX to convert a VTOC on a volume shared between systems, follow these guidelines:

- Vary the device offline to the other systems.
- Run BUILDIX.
- Vary the device back online to the other systems.

The other systems will then recognize the new format of the VTOC and continue with normal processing.

Converting an MVS VTOC to an Indexed VTOC

If changing to an indexed VTOC, before invocation of BUILDIX you must preallocate the index data set in a separate job or job step; or you must provide a DD card describing the index data set in the same job step so that the scheduler will cause allocation of the index before command execution begins.

You must ensure that the name of the index data set begins with the characters "SYS1.VTOCIX." and must provide a unique third-level qualifier for each index data set in your installation. The recommended convention is "SYS1.VTOCIX.volser", which ensures that all online volumes have uniquely identified index data set names. This prevents ENQ lockouts on all other

volumes needing IXVTOC services. In those cases where the first character of the volser is numeric, some other convention must be used. The convention used for the INDEX parameter of the INIT command, is to replace the first character of the volser with the letter "V". If this convention is used, the last five characters of all volser's starting with a numeric character must be unique. The name "SYS1.VTOCIX." is a reserved name in systems supporting the indexed VTOC, and only one data set per volume may begin with this prefix.

Space for the index must be reserved in one continuous extent. This can be accomplished by allocating the index by absolute track or by specifying the CONTIG subparameter for all other requests. The amount of space to reserve for the index is device-dependent, as well as dependent upon the size (in tracks) of the VTOC. A reference table to aid in determining how much space to reserve for the index is found in Appendix E, "The VTOC Index" on page 305.

Converting an Indexed VTOC to an MVS VTOC

This function will allow you to change an indexed VTOC to an OSVTOC.

You are provided an option to leave the index allocated (NOPURGE) or to delete it (PURGE). The option to leave it ensures that, if the volume should be changed again to the IXVTOC format, no calculation as to size and optimum location would be necessary. This convenience is realized when an IXVTOC volume is temporarily transported to a location that does not have programming support for the indexed VTOC and is then returned to a location having such support.

The procedure to follow when transporting IXVTOC volumes to nonindexed VTOC systems is to prepare the volumes prior to their removal from the indexed VTOC system by executing BUILDIX and specifying the OSVTOC parameter. Upon return of a volume, BUILDIX is again executed, if desired, to rebuild the index.

Care must be exercised if IXVTOC volumes are to be temporarily transported to nonindexed VTOC systems. If the volumes are not changed to OSVTOC format prior to being transported, serious errors may result when the volumes are returned to the indexed VTOC system. DADSM functions performed by the nonindexed VTOC system on the VTOC do not always force reconstruction of the VTOC to OSVTOC format. VTOC changes not causing reconstruction of the VTOC are not recorded in the index and, in effect, invalidate the index.

Syntax

Command	Required Parameters
BUILDIX	DDNAME(*dname*) IXVTOC|OSVTOC
	Optional Parameters
	PURGE|NOPURGE

Required Parameters

DDNAME: Identifying the Device on Which the Volume is Mounted

Explanation: **DDNAME**(*dname*) must be specified to identify the volume whose VTOC is to be changed.

DDNAME

Required parameter. The device on which the volume is mounted must be online.

For *dname*, specify the MVS JCL DD statement that identifies the device on which the volume is mounted.

Abbreviations:

DDNAME|DNAME

Default: None.

IXVTOC|OSVTOC: Identifying the BUILDIX Function to Be Executed

Explanation: **IXVTOC** or **OSVTOC** must be specified to identify the BUILDIX function to be executed.

IXVTOC

Causes an MVS format VTOC to be changed to the indexed format.

OSVTOC

Causes an indexed VTOC to be changed to an MVS format.

Abbreviations:

IXVTOC|IX
OSVTOC|OS

Default: None.

Restrictions: **IXVTOC** may only be specified if indexed VTOC programming support is part of the host system. Command execution will terminate with no action taken, if this support is not available.

Optional Parameters

PURGE|NOPURGE: Specifying Disposition of the Index Data Set

Explanation: **PURGE|NOPURGE** are optional parameters describing the final disposition of the index data set when changing an indexed VTOC to an MVS format VTOC.

PURGE
Causes deletion of the index.

NOPURGE
Causes the index to be left on the volume.

Abbreviations:

PURGE|PRG
NOPURGE|NOPRG

Default: NOPURGE.

Restrictions: This parameter is valid only when changing from an indexed VTOC to an MVS VTOC. If specified with the parameter **IXVTOC**, the option is ignored.

Examples of the BUILDIX Command

The following examples illustrate ways in which the BUILDIX command might be coded in a variety of situations.

The examples show the job control language statements you will need to use when invoking the BUILDIX command.

Note that the content of the data definition statement varies according to the function to be executed. The examples show various specifications of the statement.

Changing an IXVTOC to an OSVTOC

In the following example, a 3330 volume having the volume identification VL3330 is to be changed to OSVTOC format. The DD statement simply identifies the volume.

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//DDCARD    DD UNIT=(3330,,DEFER),VOL=(PRIVATE,SER=VL3330),
//          DISP=OLD
//SYSIN     DD *
 BUILDIX  DDNAME(DDCARD)  OS  NOPURGE
/*
```

- UNIT=(3330,,DEFER) specifies that mounting of the volume on the 3330 device type is to be deferred.
- VOL=(PRIVATE,SER=VL3330) identifies the volume and specifies a private mounting to prevent use of the volume for temporary data sets. The volume must have been previously mounted as PRIVATE.

- DISP=OLD specifies that allocation routines will check the mount status of VL3330 and issue appropriate messages to the operator.
- Specification of NOPURGE in the BUILDIX command statement indicates that the index is to be left allocated. If NOPURGE were not specified, the command statement would have the same effect, since NOPURGE is the default. Specification of PURGE, as in the following command statement, would cause the index to be deleted.

```
BUILDIX  DDNAME(DDCARD)  OSVTOC PURGE
```

Note: Whether the index is purged or left allocated is controlled by command parameters. Disposition of the index data set must not be specified in the DD statement.

Building a VTOC Index on a Volume without an Index Allocated

This option requires additional parameters on the DD statement to cause allocation of the index. The statement must contain data set information for the index.

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//VOLDD     DD UNIT=(3330,,DEFER),VOL=(PRIVATE,SER=VL3330),
//          DSN=SYS1.VTOCIX.VL3330,DISP=(NEW,KEEP),
//          SPACE=(ABSTR,(10,1))
//SYSIN     DD *
 BUILDIX  DDNAME(VOLDD)  IXVTOC
/*
```

- DSN=SYS1.VTOCIX.VL3330 specifies the name of the index data set. The third-level qualifier, VL3330 (the volume serial), follows the recommended convention for naming the index.
- DISP=(NEW,KEEP) directs the system allocation routines to allocate the data set prior to execution of Device Support Facilities commmands and to retain it upon termination of the task.
- SPACE=(ABSTR,(10,1)) directs the allocation routines to allocate a ten-track index starting at track one. ABSTR is specified in the space request to ensure that the index space is a single continuous extent and is in the location desired by the user. If location is not a primary concern, space can be reserved by specifying SPACE=(TRK,10,,CONTIG). This causes ten contiguous tracks to be reserved at some location.

Building a VTOC Index on a Volume with an Index Allocated

In this example, the volume identified as VL3330 was changed to an MVS format at some time, either by the user or by a system routine that disabled the index and changed the VTOC to OSVTOC format. The index data set was left allocated on the volume. The DD statement required to change again to an IXVTOC is shown.

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//VOLDD     DD UNIT=(3330,,DEFER),VOL=(PRIVATE,SER=VL3330),
//          DSN=SYS1.VTOCIX.VL3330,DISP=OLD
//SYSIN     DD *
 BUILDIX  DNAME(VOLDD)  IX
/*
```

- DISP=OLD specifies that the data set already exists.

CONTROL Command—CKD

The CONTROL command provides a means of resetting certain subsystem conditions that have been previously set. The CONTROL command is used after the condition has been repaired to reset that previous condition. For example, the CONTROL command can be used to reset a device that has been WRITE INHIBITed, or to clear a fence status of a path and/or a device.

Using the CONTROL Command

The CONTROL command allows you to perform different functions. When you specify the ALLOWWRITE parameter you can clear a storage control that has been WRITE INHIBITed. The CLEARFENCE parameter allows you to clear a fence status of a path and/or a device that has been fenced off by the subsystem. These functions are described below.

ALLOWWRITE Function

The CONTROL command with the ALLOWWRITE parameter is provided for MVS/370, MVS/XA, CMS, (ATTACHed mode only) and stand-alone users only. This command and parameter combination allows subsequent writes to the devices controlled by an IBM 3880 or IBM 3990 Storage Control if the storage control has been WRITE INHIBITed by the error recovery procedures at the channel, director, or storage control level.

After WRITE INHIBIT has occurred, subsequent write operations to any device attached to the WRITE INHIBITed storage control will fail. The CONTROL command with the ALLOWWRITE parameter causes **all** storage controls to which a specified device is attached to be cleared for subsequent write operations. This command gives you the ability to clear a storage control that has been WRITE INHIBITed without having to perform an IMPL.

There is the possibility, during normal processing in an MVS environment, that the system error recovery procedures (ERPs) may detect an error condition in the storage control such that future write operations to some or all of the devices through the storage control may fail.

When the ERPs detect such a condition, they can "write-inhibit" the storage control (at different levels), thereby inhibiting any further write operations to the devices through the failing storage control. (If an alternate path exists through a different storage control, the device(s) can remain online and functional.)

Maintenance is required on the failing storage control.

After the failing storage control has been repaired, the CONTROL command is the means by which the storage control is "write-allowed," thereby again allowing write operations through that storage control.

Notes:

1. Because CONTROL operates on a device basis, if more than one storage control (that is, path) to a device is failing, **all** such storage controls must be repaired before executing the CONTROL command.
2. When executing stand-alone ICKDSF in XA mode, all storage controls attached to the specified device are cleared for subsequent write operations. When executing the stand-alone ICKDSF in 370 mode, only the storage control attached to the specified device address is write-allowed. To cause all storage controls attached to the device to be cleared, the ALLOWRITE must be executed on each path, specifying the *ccuu* for the path to be cleared.

   Maintenance on the failing storage control should be initiated following message IEA467 or IEA468 at the system console.

CLEARFENCE Function

When you are using an IBM 3990 Storage Control in DLSE mode, an error may occur causing the subsystem to fence off one particular path to a device or group of devices. The subsystem continues to function with the remaining path(s).

Maintenance is required on the failing storage control and/or the failing device to repair the condition that caused the FENCE. After the failing unit has been repaired, the CONTROL command with the CLEARFENCE parameter is provided to clear the fenced condition for the path.

The CONTROL command with the CLEARFENCE parameter will clear ALL paths to ALL devices on the subsystem. The specified device can be any device on the subsystem.

The CLEARFENCE function is available in the CMS, VSE, MVS/370, MVS/XA, and stand-alone environments.

Maintenance is normally performed after message IEA463I (MVS environment) is issued at the console when a fence condition has occurred.

Using the CONTROL Command under CMS

When you are running ICKDSF under CMS, the CONTROL command is valid only with ATTACHed devices (that is, no minidisk support).

Syntax

Command	Required Parameters
CONTROL	**UNITADDRESS(***ccuu***)|DDNAME(***dname***) |SYSNAME(***sysxxx***)**
	Optional Parameters
	ALLOWRITE **CLEARFENCE**

Required Parameters

DDNAME|SYSNAME|UNITADDRESS: Identifying the Device or Volume

Explanation: **DDNAME(***dname***)|SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** are required parameters identifying the device or volume that is to be reset.

DDNAME

Required when resetting a device or volume that is mounted online in the MVS environment. The device on which the volume is mounted must be online.

For *dname*, specify the MVS JCL DD statement that identifies the device on which the volume is mounted.

SYSNAME

Required when resetting a device or volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN control statement.

UNITADDRESS

Required when resetting a device or volume that is mounted offline in MVS, or when resetting a device or volume in the stand-alone environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits), of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS, use the UNITADDRESS parameter to specify the virtual address of the ATTACHed volume.

Abbreviations:

```
DDNAME|DNAME
UNITADDRESS|UNIT
```

Default: None.

Restrictions: The DDNAME parameter cannot be specified when the device or volume is in the stand-alone, VSE, or CMS environment.

The SYSNAME parameter cannot be specified when the device or volume is in the stand-alone, MVS, or CMS environment.

The UNITADDRESS parameter cannot be specified when the device or volume is mounted online or running under VSE.

Optional Parameters

ALLOWWRITE

Explanation: **ALLOWWRITE** must be specified if you want to clear a storage control that has been WRITE INHIBITed (**MVS/370, MVS/XA, CMS, and stand-alone only**).

ALLOWWRITE
Specifies that a storage control is to be cleared.

Abbreviations: ALLOWWRITE|ALLOWWR

Default: If you are running in the MVS/370, MVS/XA, or stand-alone environment, ALLOWWRITE is the default.

If you want to execute ALLOWWRITE and CLEARFENCE, both parameters must be specified.

Restriction: Valid for MVS/370, MVS/XA CMS, and stand-alone only.

The CONTROL command with the ALLOWWRITE parameter is valid only for DASD devices attached to an IBM 3880 or IBM 3990 Storage Control

CLEARFENCE

Explanation: **CLEARFENCE** must be specified if you want to clear a fence status of a path and/or a device that has been fenced off by the subsystem.

CLEARFENCE
Specifies that a fence status of a path and/or a device is to be cleared.

Abbreviations: CLEARFENCE|CLEARF|CLRF

Default: None. CLEARFENCE must be specified.

If you want to execute ALLOWWRITE and CLEARFENCE, both parameters must be specified.

Restriction: The CONTROL command with the CLEARFENCE parameter is valid only for DASD devices attached to an IBM 3990 Storage Control.

Examples of the CONTROL Command

The following examples illustrate how the CONTROL command can be coded.

The examples show the job control language statements you will need to use when invoking the CONTROL command in an operating system or the CONTROL command statement needed for a stand-alone environment.

Clearing a WRITE INHIBITed storage control

In the following example, a WRITE INHIBITed IBM 3880 or IBM 3990 Storage Control, attached to an IBM 3380 DASD volume with a volume serial number of ABCDEF, is to be cleared. The DD statement identified in the input stream identifies an online DASD volume.

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//DDCARD    DD UNIT=3380,DISP=OLD,VOL=SER=ABCDEF
//SYSIN     DD *
 CONTROL  ALLOWWR DDNAME(DDCARD)
/*
```

Clearing a Storage Path Fence Status

The following example shows the job control language and CONTROL command used to reset a fence status. It is assumed the path and/or the device which has been fenced off by the subsystem has been repaired. The DD statement identified in the input stream identifies an online DASD volume.

```
//jobname   JOB . . .
//stepname  EXEC PGM=ICKDSF
//SYSPRINT  DD SYSOUT=A
//DDCARD    DD UNIT=3380,DISP=OLD,VOL=SER=ABCDEF
//SYSIN     DD *
 CONTROL  CLEARFENCE DDNAME(DDCARD)
/*
```

Clearing a Storage Path Fence Status in a CMS or Stand-alone Environment

The following example shows the CONTROL command used to reset a fence status. It is assumed the path and/or the device which has been fenced off by the subsystem has been repaired. The example is for a CMS or stand-alone environment.

```
 CONTROL  CLEARFENCE UNITADDRESS(162)
```

INIT Command—CKD

The INIT command initializes a volume. A direct-access storage volume must be initialized before an operating system can use it. The INIT command writes a volume label and a VTOC (optionally, with an index) on the volume for use by MVS or VSE operating systems. This is the minimal level of initialization. Volume validation (medial initialization) and surface checking (maximal initialization) can also be performed. These levels are described in the sections that follow.

For INIT support of the Mass Storage System (MSS), VM minidisks, and emulated devices respectively, see:

- "MIMIC: Specifying a Special Volume Usage Format" on page 119
- Appendix B, "VM Support" on page 227
- "Emulated Devices" on page 66

Processing in a Shared Environment

When running INIT on a volume shared between systems, follow these guidelines:

- Vary the device offline to the other systems.
- Run the INIT command.
- Vary the device back online to the other systems.

The other systems will then recognize the volume serial number and VTOC location of the volume. Should you choose **not** to follow these guidelines, other systems accessing the volume may experience unpredictable results.

Initializing a Volume at the Minimal Level

A volume is always initialized at least to the minimal level. When both the NOVALIDATE and NOCHECK parameters are specified, minimal initialization is all that is performed.

Minimal initialization creates the contents of cylinder 0, track 0 by:

- Writing IPL bootstrap records as records 1 and 2
- Writing a volume label as record 3
- Optionally, writing the IPL text

Initialization also reserves and formats tracks for the VTOC at the location specified by the user and for the number of tracks specified. If no location is specified, tracks are reserved at the default location. The size and location can also be defaulted to the end of the volume.

If the INDEX parameter of the INIT command is specified, a format-1 DSCB is built in the VTOC with the default data set name of SYS1.VTOCIX.volser. If the *volser* begins with a numeric character, the default data set name will be SYS1.VTOCIX.Vxxxxx, where xxxxx is the volser with the first character overlaid by V. For example, if the *volser* is 333001, the default data set name written in

the format-1 DSCB is SYS1.VTOCIX.V33001. Appendix E, "The VTOC Index" on page 305, contains more information about indexed format VTOCs.

If the IPLDD parameter is specified, the IPL text is added to the volume.

At the completion of the minimal initialization, defective primary tracks that were using an alternate are now assigned alternates in sequential order within the alternate track area.

Initializing a Volume at the Medial Level

A volume is initialized at the medial level when both the VALIDATE and NOCHECK options are specified.

Medial initialization includes the actions of minimal initialization, as well as validation of the existence and content of the track's home address and record 0 for every track in the specified range. Validation ensures that the proper track address (cylinder and head) appears in both the home address and record 0 on the track, and that a standard record 0 exists on the track. Validation rewrites the home address and record 0.

At the completion of the medial initialization processing for the requested tracks, defective primary tracks that were using an alternate are now assigned alternates in sequential order within the alternate track area.

If a medial initialization does not run to completion, the CONTINUE function is provided to allow you to recover in a minimum amount of time, and to ensure the volume is usable. See "Resuming after a Job or System Failure" on page 103 for more information concerning continuing after a failure.

Before running a medial initialization, all IBM 3380 Models must be powered on with all devices in the ready condition and all covers closed for 2 hours 30 minutes to establish thermal stability.

Initializing a Volume at the Maximal Level

A volume is initialized at the maximal level when the CHECK parameter is specified. Note that this level of surface checking is supported for the IBM 2305, 2314, 3330, 3340, 3344, and 3350 devices only.

Maximal initialization includes the actions of minimal and medial initialization, as well as checking of track surfaces for each track in the specified range.

If surface checking detects a data check on the volume, skip displacement checking is performed for supported devices. The primary track continues to be used until the number of defects exceeds the maximum skips allowed for the device, after which an alternate track is assigned. In addition, skip displacement checking is always performed for cylinder 0 track 0 and the first track of the VTOC (before the volume label or the format-4 DSCB is written) if they are in the specified range.

See Appendix G, "Surface Checking" on page 315, for more information on surface checking.

At the completion of maximal initialization processing for the requested tracks, defective primary tracks that were using an alternate are now assigned alternates in sequential order within the alternate track area.

If a maximal initialization does not run to completion, the CONTINUE function is provided to allow you to recover in a minimum amount of time, and to ensure the volume is usable. See "Resuming after a Job or System Failure" for more information concerning continuing after a failure.

Controlling the Level of Surface Checking

The SKIP/NOSKIP parameters specify the level of surface checking that is to be performed for a maximal initialization. Note that the SKIP parameter is valid for the IBM 3340, 3344, and 3350 devices only. See Appendix G, "Surface Checking" on page 315, for more information on surface checking.

NOSKIP indicates that primary surface checking is to be performed for all tracks in the specified range, and skip displacement surface checking is performed only if a data check is detected.

SKIP indicates that skip displacement surface checking is to be performed for *every* track in the specified range. Although SKIP provides a more extensive check of the surface of a track, it runs for long periods of time per track. If SKIP is specified, the total number of tracks in the specified range is limited to the number of tracks in a cylinder for the particular device type.

Resuming after a Job or System Failure

If for any reason a medial or maximal initialization does not run to completion, subsequent processing can start from a checkpoint location. ICKDSF checkpoints itself automatically during processing and remembers where to resume processing.

After a job or system failure, you can submit a new job:

- At the minimal, medial or maximal level
- For the entire volume, or for a different specified range, or the same specified range (for medial or maximal INIT)
- To resume from the point of failure, or to ignore the previous processing and start to process a different range

Regardless of the the job you choose to run after a failure, when processing completes, all tracks on the volume that ICKDSF might have operated on are in a usable condition. Note that, while ICKDSF will fix any tracks that were left unusable because of the failure, it cannot ensure the usability of any track that is not in the current specified range or that was not processed before the point of failure.

Because INIT processing always destroys the volume label, a subsequent invocation must include the VOLID parameter.

INIT checkpoints its current location not more than every five minutes. When INIT is run after a previous failure, it detects the checkpointed information, and

then validates or skip analyzes (depending on the device type that failed) all the tracks from the last checkpoint. This is done for *all* levels of initialize, and regardless of the CONTINUE/NOCONTINUE parameters. This ensures that ICKDSF has not left any tracks in an unusable condition because of the failure. Note that this adds additional run time to INIT, and is most noticeable for minimal INIT.

After the tracks from the previous point of failure have been processed, if CONTINUE is specified, processing continues from the point of failure *in correlation with* the current specified range as follows:

- If the new range is totally before the previous point of failure, no new tracks are processed.
- If the new range is totally beyond the adjusted point of failure, then the entire new range is processed.
- The point of failure is adjusted if the new job specifies VALIDATE, since we have already validated the tracks to the next checkpoint location.
  - If the new job specifies a range that is entirely before where the next checkpoint should occur, there are no more tracks to process.
  - If the new job specifies a range that overlaps the point of failure, then processing begins from the point of failure.
- If NOCONTINUE is specified, processing begins with the first track in the current specified range.

RANGE Parameters: Specifying Part of a Volume

CYLRANGE/HEADRANGE and FROMRANGE/TORANGE are optional parameters, generally specified in pairs, to limit the areas of a volume that are to be processed. Use the CYLRANGE/HEADRANGE pair to specify the starting and ending cylinder(s) and the starting and ending head(s) to be examined before minimal initialization functions are performed. Use the FROMRANGE/TORANGE pair to specify the starting track (cylinder and head) and the ending track (cylinder and head) to be examined before standard volume initialization is performed.

Note that the VTOC and volume label are still written even when range parameters are specified.

Where the term 'specified range' is used in this book, it describes the tracks that are the result of:

- The combination of FROMRANGE and/or TORANGE
- The combination of CYLRANGE and/or HEADRANGE
- The entire volume, if all the range parameters are omitted

The following notes apply to specifying part of a volume:

- CYLRANGE/HEADRANGE and FROMRANGE/TORANGE are mutually exclusive pairs of parameters.
- If you specify CYLRANGE and do not specify HEADRANGE, all the heads of the specified cylinders are processed.
- If you specify HEADRANGE and do not specify CYLRANGE, the specified heads for all the cylinders on the volume are processed.

- Both the starting and ending values are required for CYLRANGE and HEADRANGE. If you specify an ending value greater than the maximum for the device, the device maximum is used for the ending value.
- If you specify FROMRANGE and do not specify TORANGE, TORANGE is defaulted to the last alternate cylinder
- If you specify TORANGE and do not specify FROMRANGE, FROMRANGE is defaulted to cylinder 0 track 0.
- If CHECK is specified, surface checking is performed only for those tracks within the range. (The SKIP parameter specifies the level of surface checking to be performed.)
- IF NOCHECK VALIDATE is specified, the home address and record zero are rewritten only for those tracks within the range.
- If NOCHECK NOVALIDATE is specified, the range parameters are ignored.
- If none of the range parameters are specified, medial and maximal initialize operate on the entire volume.
- Primary tracks within the range specified have alternate tracks assigned as necessary. Primary tracks outside the range specified are not read or validated and do not have their alternate track pointers verified.
- If there is a need to verify the alternate tracks on a volume (that is, primary/alternate pairs are valid), the range parameters can be specified for the alternate cylinders.

Using the INIT Command under CMS

When you run ICKDSF under CMS, you can run in two modes:

- Minidisk
- ATTACHed

Minidisk Support under CMS

When you run ICKDSF under CMS the following is supported for minidisks:

- Minimal INIT *only* is supported

When you initialize a minidisk at the minimal level, you can specify the minidisk volume label (VOLID), the VTOC, the index data set (INDEX), the owner identification (OWNERID), and whether you are supplying or the system is to supply the IPL bootstrap records.

The following parameters are supported when you are running under CMS

- UNITADDRESS, BOOTSTRAP/NOBOOTSTRAP, DOSVTOC, INDEX, IPLDD, LABELS, OWNERID, PASSWORD, PURGE, VERIFY, VOLID, and VTOC.

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed to your user ID. When you use ICKDSF in this mode, you can then do any ICKDSF function which can be performed when you are running ICKDSF stand-alone

Restoring Factory Functional Verification Data Patterns to a Volume

If there is a need to ensure that data exists on every track of an IBM 3375 or 3380 volume (for example, for a subsequent ANALYZE SCAN after a volume has been moved elsewhere), the DATA parameter can be used. When you specify DATA, records are written which correspond to the factory functional verification data patterns which are written on the volume at the factory.

Initializing an SMS-Managed Volume

You can initialize a volume which is to be managed by the Storage Management Subsystem (SMS) by specifying the STORAGEGROUP parameter.

ICKDSF marks the volume as being managed by SMS during the initialization process and informs the operating system that the volume has been initialized for SMS use. If the volume you are initializing is online, ICKDSF will:

- Bypass RACF DASDVOL checks
- Check that there are no SMS data sets on the volume

Volumes which have been previously SMS managed, can be initialized as SMS managed again or initialized as non-SMS-managed volumes.

Syntax

Command	Required Parameters
INIT	**DDNAME(***dname***)|SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** **VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	BOOTSTRAP|NOBOOTSTRAP **CHECK(***n***)|NOCHECK** **CONTINUE|NOCONTINUE** **CYLRANGE(***cccc,cccc***)** **DATA|NODATA** **DEVICETYPE(***devtype***)** **DOS[VSE]VTOC(END|***cylinder,head[,extent]***)** **FROMRANGE(***cccc,hhhh***)** **HEADRANGE(***hhhh,hhhh***)** **INDEX(***cylinder,head[,extent]***)** **IPLDD(**{*dname*|*dlblname*} *[*,**OBJFORMAT|,ABSFORMAT***])** **LABELS(***n***)** **MAP|NOMAP** **MIMIC(***type***)** **OWNERID(***owner***)** **PASSWORDS((***dsname/password***),•••)** **PURGE|NOPURGE** **RECLAIM|NORECLAIM** **SKIP|NOSKIP** **STORAGEGROUP** **TORANGE(***cccc,hhhh***)** **VALIDATE|NOVALIDATE** **VOLID(***serial***)** **VTOC(END|***cylinder,head[,extent]***)**

Required Parameters

Note that various combinations of parameters can cause INIT to function differently. These differences are outlined below.

DDNAME|SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **DDNAME(***dname***)\|SYSNAME(***sysxxx***)\|UNITADDRESS(***ccuu***)** are required parameters identifying the volume (or minidisk under CMS) that is to be initialized. When you initialize a volume for the first time under MVS, the volume must be mounted offline because it contains no volume label and is not acceptable to an operating system.

DDNAME

Required when initializing a volume that is mounted online in the MVS environment. The device on which the volume is mounted must be online and the volume mounted as PRIVATE.

For *dname*, specify the MVS JCL DD statement that identifies the device on which the volume is mounted.

SYSNAME
Required when initializing a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN control statement.

UNITADDRESS
Required when initializing a volume that is mounted offline in MVS, or when initializing a volume in a stand-alone environment.

For *ccuu*, specify the address, in hexadecimal (3 or 4 digits), of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS, use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual minidisk address.

Abbreviations:

```
DDNAME|DNAME
UNITADDRESS|UNIT
```

Default: None.

Restrictions: The DDNAME parameter cannot be specified when initializing a volume that has not been previously initialized or when the volume is in a stand-alone environment or the VSE environment.

The SYSNAME parameter cannot be specified when initializing a volume in the stand-alone, CMS or MVS environment.

The UNITADDRESS parameter cannot be specified when initializing a volume that is mounted online or running under VSE.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(***serial[,owner]***)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before initializing the volume or minidisk.

VERIFY
Required when you want to verify the volume serial number and owner identification before initializing the volume or minidisk. If the volume serial number or owner identification does not match that found on the volume or minidisk the INIT command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY
Required when you want to bypass verification of the volume or minidisk serial number and owner identification.

Abbreviations:

VERIFY|VFY
NOVERIFY|NOVFY|NVFY

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number; both the volume serial number and the owner identification must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, the command terminates.

Optional Parameters

BOOTSTRAP|NOBOOTSTRAP: Writing IPL Bootstrap Records on the Volume

Explanation: **BOOTSTRAP|NOBOOTSTRAP** are optional parameters specifying whether you are supplying the IPL bootstrap records that are to be written on the volume during initialization.

BOOTSTRAP
Indicates that you are supplying the IPL bootstrap records that are written on the volume or minidisk during initialization. These records must be the first four records in the IPL program text. The first text record must have IPL1 in the first four columns, followed by 24 bytes of data. These 24 bytes have the same format as the first IPL record described under ABSFORMAT in "IPLDD: Writing a User-Supplied IPL Program on the Volume" on page 115. The next three text records must have IPL2 in their first four columns, followed by 68 bytes of data each in the second and third records and 8 bytes of data in the last record. These 144 bytes of data have the same format as IPL record 2 described under ABSFORMAT.

NOBOOTSTRAP
Indicates that you want the system to supply the IPL bootstrap records that are written on the volume during initialization.

Abbreviations:

BOOTSTRAP|BOOT
NOBOOTSTRAP|NOBOOT|NBOOT

Default: The default is NOBOOTSTRAP.

Restrictions: The BOOTSTRAP and NOBOOTSTRAP parameters apply only when the IPLDD parameter is specified. This parameter is ignored when the IPL program is supplied in absolute format.

CHECK|NOCHECK: Surface Checking of the Volume

Explanation: **CHECK**(*n*)|**NOCHECK** are optional parameters specifying whether the tracks are to be checked for recording errors (maximal initialization).

If CHECK is specified, for each track, the home address and record zero are validated and rewritten, the track is surface checked by writing and reading

specially patterned records (as a long record zero), and the standard record zero is then rewritten onto the track.

If errors occur that cannot be corrected with skip displacement, the track is flagged, and an alternate track is assigned to it. If an alternate track is identified as defective, it is flagged as defective.

CHECK
Indicates that maximal initialization is to take place. Each track surface is checked for recording errors during initialization. This parameter erases the contents of each track during initialization.

For *n*, substitute a decimal number from 1 through 10 for the number of times you want each track to be checked. See Appendix G, "Surface Checking" on page 315, for information about n values. The CHECK parameter is valid only for the IBM 2305, 2314, 3330, 3340, 3344, and 3350.

NOCHECK
Indicates that you do not want the tracks to be surface checked for recording errors during initialization. Initialization will be done at the medial or minimal level only, depending on the VALIDATE/NOVALIDATE specification.

Abbreviations:

CHECK|CHK
NOCHECK|NOCHK|NCHK

Default: The default is NOCHECK.

Restrictions: The maximum number you can specify with the CHECK parameter is 10.

For VM minidisks in a stand-alone environment, CHECK(n) is valid only for 2311, 2314, and 2319 device types.

The CHECK parameter is not valid for the IBM 3375 and 3380 devices.

Note that, if CHECK is specified, the functions of medial INIT have been included. The VALIDATE|NOVALIDATE parameter is ignored.

Use CHECK with SKIP/NOSKIP to control the level of surface checking.

CHECK is not valid for minidisks when you are running ICKDSF under CMS.

CONTINUE|NOCONTINUE: RESUME FROM A CHECKPOINT

Explanation: **CONTINUE|NOCONTINUE** are optional parameters specifying whether processing is to resume from the last checkpointed location (CONTINUE) or to start processing the specified range or full volume (NOCONTINUE).

These parameters are interrogated only if processing was previously interrupted during a medial or maximal initialization.

CONTINUE
Indicates that processing is to resume from the last checkpointed location.

A message is issued to indicate where processing is resumed.

NOCONTINUE
Indicates that you do not want to start from the last checkpointed location, but want to start processing the specified range or volume.

Note that ICKDSF still goes to the backup location even if NOCONTINUE is specified. This is done to ensure no tracks are left in a non-standard format.

Default: The default is CONTINUE

Restriction: The CONTINUE function is not supported for the IBM 2305, 2311, 2314, 2319, or for a 3340 emulated on a 3370 FBA device.

CONTINUE is ignored for minidisks.

CYLRANGE: Specify a Range of Cylinders

Explanation: **CYLRANGE**(*cccc,cccc*) is an optional parameter used to specify that part of a volume is to be surface checked. Use the CYLRANGE parameter to specify the starting and ending cylinder(s) to be examined before standard volume initialization is performed.

For *cccc,cccc*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the starting and ending cylinders to be examined.

If you specify CYLRANGE and do not specify HEADRANGE, all the heads of the specified cylinders are processed.

For information on specifying the starting and ending heads (HEADRANGE) to go with the starting and ending cylinders, see "HEADRANGE: Specify a Range of Heads" on page 114.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Part of a Volume" on page 104.

Abbreviations:

CYLRANGE|CYL

Default: None

Restrictions: Both the starting and ending values are required for CYLRANGE. By specifying an ending value greater than the maximum for the device, processing is performed to the maximum value for that device.

CYLRANGE cannot be specified with FROMRANGE/TORANGE.

CYLRANGE applies only when CHECK or VALIDATE is specified.

CYLRANGE is not valid when you are processing minidisks

DATA|NODATA: Writing factory functional verification data patterns on a Volume

Explanation: **DATA|NODATA** are optional parameters specifying whether the factory functional verification data patterns (FFVDP) are to be written during the validate process for the IBM 3375 and 3380 devices.

DATA
Indicates that three records of factory functional verification data patterns are to be written on the specified tracks. When DATA is specified for devices other than the IBM 3375 and 3380, the parameter is ignored and no data is written.

NODATA
Indicates that you do not want to write factory functional verification data patterns on the volume.

Default: The default is NODATA

Restriction: The DATA parameter applies only when the VALIDATE parameter has been specified. If DATA is specified with NOVALIDATE, the function is terminated with a condition code of 12.

DATA is valid for the IBM 3375 and 3380 only.

DATA is not valid when you are processing minidisks.

DEVICETYPE: Identifying the Type of Device

Explanation: **DEVICETYPE**(*devtype*) is required in a stand-alone environment when the MIMIC(MINI) or MIMIC(EMU) parameter is specified.

For *devtype*, substitute one to eight characters for the type of device on which the volume is mounted. See Figure 2 on page 7 for valid device types.

Abbreviations:

```
DEVICETYPE|DEVTYPE|DEVTYP
```

Default: None.

Restrictions: The DEVICETYPE parameter applies only when you are initializing a volume in a stand-alone environment and MIMIC(MINI) or MIMIC(EMU) is specified. If DEVICETYPE is specified when it does not apply, it will be ignored.

When using an IBM 3350 Direct Access Storage volume in 3330-compatibility mode, the device type must be specified as 3330-1 or 3330-11, as required. The IBM 3333 Disk Storage and Control Models 1 and 11 must be specified as 3330-1 and 3330-11, respectively.

DOSVTOC|VSEVTOC: Specifying the Volume Table of Contents in VSE

Explanation: **DOSVTOC(END|***cylinder,head[,extent]***)** and **VSEVTOC(END|***cylinder,head[,extent]***)** are optional parameters specifying the location and size of the volume or minidisk VTOC for a VSE system. You can specify either parameter and the results are the same.

For *cylinder,head*, specify one to four decimal (n) or hexadecimal (X'n') numbers to identify the cylinder and one to four decimal (n) or hexadecimal

(X'n') numbers to identify the head where the volume table of contents is to be placed.

For *extent*, specify one to five decimal (n) or hexadecimal (X'n') numbers for the number of tracks that are to be reserved for the volume or minidisk VTOC. If extent is omitted, one track is reserved.

When you specify DOSVTOC(END) or VSEVTOC(END), the VTOC is put on the last primary cylinder of the volume or minidisk and is one cylinder in length.

Abbreviations:

DOSVTOC|DVTOC
VSEVTOC

Default: If neither VTOC nor DOSVTOC (or VSEVTOC) is specified, the default taken is an MVS VTOC at cylinder 0, track 1, except for MSS, where it is cylinder 0, track 2.

If neither VTOC nor DOSVTOC (or VSEVTOC) is specified and the device is not a minidisk, the size is defaulted to the number of tracks in a cylinder minus one.

Restrictions: A VTOC cannot be placed at cylinder 0, track 0.

FROMRANGE: Specify a Specific Starting Location

Explanation: **FROMRANGE(***cccc,hhhh***)** is an optional parameter used to specify that part of a volume is to be surface checked. Use this parameter to specify the starting track (cylinder and head) to be examined before minimal volume initialization is performed.

For *cccc,hhhh*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the cylinder and head, where processing is to start.

If you specify FROMRANGE and do not specify TORANGE, TORANGE is defaulted to the last alternate cylinder.

For information on specifying the ending track (TORANGE) to go with the starting track, see "TORANGE: Specify a Specific Ending Location" on page 123.

For information on specifying part of a volume to be examined, see "RANGE Parameters: Specifying Part of a Volume" on page 104.

Abbreviations:

FROMRANGE|FROM

Default: None

Restrictions: FROMRANGE cannot be specified with CYLRANGE/HEADRANGE.

FROMRANGE applies only when CHECK or VALIDATE is specified.

HEADRANGE: Specify a Range of Heads

Explanation: **HEADRANGE(*hhhh,hhhh*)** is an optional parameter used to specify that part of a volume is to be surface checked. Use the HEADRANGE parameter to specify the starting and ending head(s) to be examined before standard volume initialization is performed.

For *hhhh,hhhh*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the starting and ending heads to be examined.

If you specify HEADRANGE and do not specify CYLRANGE, the specified heads for all the cylinders on the volume are processed.

For information on specifying the starting and ending cylinder(s) (CYLRANGE) to go with the starting and ending heads, see "CYLRANGE: Specify a Range of Cylinders" on page 111.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Part of a Volume" on page 104.

Abbreviations:

HEADRANGE | HDRANGE | HD

Default: None

Restrictions: Both the starting and ending values are required for HEADRANGE. By specifying an ending value greater than the maximum for the device, processing is performed to the maximum value for that device.

HEADRANGE cannot be specified with FROMRANGE/TORANGE.

HEADRANGE applies only when CHECK or VALIDATE is specified.

HEADRANGE is not valid when you are processing minidisks.

INDEX: Creating a VTOC Index

Explanation: **INDEX(*cylinder,head[,extent]*)** is an optional parameter specifying the starting location and size of the index data set. Values for subparameters *cylinder* and *head* are required; specification of *extent* is optional.

For *cylinder* and *head*, specify one to five decimal (n) or hexadecimal (X'n') digits to identify the cylinder number, and one to five decimal or hexadecimal digits to identify the track number within the cylinder where the index is to start.

For *extent*, specify one to five decimal (n) or hexadecimal (X'n') digits for the number of tracks that are to be reserved for the index data set.

A format-1 DSCB is built in the VTOC with the default data set name of SYS1.VTOCIX.volser. If the *volser* begins with a numeric character, the default data set name will be SYS1.VTOCIX.Vxxxxx. For example, if the *volser* is 333001, the default data set name written in the format-1 DSCB is SYS1.VTOCIX.V33001. Appendix E, "The VTOC Index" on page 305 contains more information about indexed format VTOCs.

Example : INDEX(10,11,12) This specification will cause the index to be placed starting at cylinder 10, track 11 for a total of 12 tracks. Expressed in hexadecimal: INDEX(X'A',X'B',X'C')

Abbreviations: None.

Default: No default will be taken for *cylinder* or *track*. The default for *extent* is an internal calculation based on the number of tracks in the VTOC.

Restrictions: The 2311, 2314, and MSS staging volumes are not supported by the INDEX parameter. All other devices, including VM minidisks of the proper device type, are supported.

This parameter is ignored in the VSE environment.

Note: The system generates SYS1.VTOCIX.volser as the name of VTOC index unless the volser begins with a numeric character, in which case the convention is SYS1.VTOCIX.Vxxxxx, where xxxxx is the volser with the first character overlaid by V.

When you specify the STORAGEGROUP parameter, INDEX must also be specified.

IPLDD: Writing a User-Supplied IPL Program on the Volume

Explanation: **IPLDD** (*{dname|dlblname}[* ,**OBJFORMAT**|,**ABSFORMAT***]*) is an optional parameter that allows you to supply an IPL program to be written on the volume or minidisk during initialization.

The IPLDD and BOOTSTRAP parameters are used to specify one of the following:

- If you do not specify IPLDD, ICKDSF writes special bootstrap records that cause the processing unit to be placed in a WAIT state if the volume or minidisk is specified during an attempt to load the system.
- If you specify IPLDD but do not specify BOOTSTRAP, ICKDSF supplies an IPL bootstrap which is written on the volume or minidisk during initialization together with the IPL text you supply.
- If you specify IPLDD and BOOTSTRAP, ICKDSF uses the IPL bootstrap and the IPL text you supply. If necessary, ICKDSF updates the bootstrap records to allow for possible user labels.

The maximum size permitted for the IPL program record depends upon the type of volume being initialized. The maximum sizes allowed are:

Volume Type	Maximum Bytes for IPL Program
2305-1	12,180
2305-2	13,616
2311	3,110
2314	6,514
2319	6,514
3330-1	12,117
3330-11	12,117
3340	7,286
3344	7,286
3350	17,902
3375	33,984
3380	44,948

Also, the number of additional user volume labels can further limit the size allowed for the IPL program. If you supply an IPL program that exceeds the size allowed, you receive an error message but initialization continues.

On MVS, CMS, and stand-alone systems, use *dname*.

In MVS, for *dname*, substitute one to eight alphameric characters for the DD statement identifying the data set that contains the IPL program you want written on the volume (or minidisk under CMS) being initialized. The IPL program can be included within the MVS JCL input stream (SYSIN). If the IPL program is in the MVS JCL input stream, the data must be included immediately after the INIT command.

In a CMS or stand-alone environment, *dname* must be specified as SYSIN, and must be the same device as the input device (that is, a CMS file, the card reader, or the console). The data must immediately follow the INIT command and be terminated by an ENDIPLTEXT card. In a VM environment, the IPLDD data must be contiguous to the INIT command.

On VSE systems, use *DLBL name*. For *DLBL name*, substitute 1 to 7 alphameric characters. These represent the file name that appears in the DLBL statement and identify the file that contains the IPL program you want written on the volume being initialized. The IPL program can be included in the VSE JCS input stream (SYSIN). It must then be included immediately after the INIT command.

OBJFORMAT

Specifies that IPL data is being supplied in object deck format; that is, cards will have one of the following strings of EBCDIC characters in columns 2 through 4:

```
TXT
RLD
ESD
END
```

Note that only cards with TXT will be processed. All others will be ignored.

OBJFORMAT may be abbreviated as OBJECT or OBJ.

OBJFORMAT is the default if IPLDD is specified.

ABSFORMAT

Specifies that IPL data is being supplied as variable-length records that contain executable instructions. When IPL data is supplied in this format, it is necessary to provide a minimum of three records. The maximum number of records supplied and their lengths are only limited by the track capacity of the volume or minidisk on which these records are to be written.

The first two records supplied must be the bootstrap records and are restricted to lengths of 24 and 144 bytes, respectively.

The contents of the IPL records and the contents of the program are not checked by ICKDSF. It is the user's responsibility to ensure that the IPL records can load an executable program. The first IPL record must contain a PSW followed by two CCWs (channel command words). The CCWs should have the following hexadecimal formats:

```
First CCW    06xxxxxx 60000090
Second CCW   08xxxxxx 00000000
```

The first CCW is a command to read in the second IPL record at main storage address xxxxxx. The second CCW is a transfer-in-channel command (a branch) to the CCW that begins the second IPL record.

The second IPL record must be 144 bytes long. Bytes 32 through 42 (starting from byte 0) cannot be used. Bytes 32 through 42 are used by ICKDSF for the seek address (bytes 32 through 37) and the CCHHR (bytes 38 through 42) of the third IPL record.

The 3rd through nth records that are supplied are assumed to be IPL program records and will be written on the volume, without any modification, after the standard volume label and any existing user volume labels. Note that the IPL program will only be written on the first track of the pack. If the records supplied exceed in length the remaining space on the first track, the function will terminate with an error message.

ABSFORMAT may be abbreviated as ABSOLUTE or ABS.

Abbreviations:

```
IPLDD|IPL
```

Default: The default is OBJFORMAT.

The system provides special IPL bootstrap records if you specify the IPLDD parameter without specifying the BOOTSTRAP parameter.

Restrictions: The IPLDD parameter is ignored when the MIMIC(MSS) parameter is specified.

In a CMS or stand-alone environment, the *dname* subparameter of IPLDD must be SYSIN.

When an IPL program is included in the SYSIN stream, it must immediately follow the INIT command and be terminated by an ENDIPLTEXT card. The ENDIPLTEXT card is optional when the IPL program is in a data set other than the one specified by SYSIN, or when the end-of-file indicator (/*) immediately follows the data for the IPL program.

LABELS: Reserving Space for Additional Volume Labels

Explanation: **LABELS**(*n*) is an optional parameter specifying that space on cylinder 0, track 0 is to be reserved for additional volume labels, also known as user volume labels. In addition to the label that must be written on the volume, a maximum of nine additional volume labels can be specified.

To specify *n*, substitute a decimal number from 1 through 9 for the number of additional user volume labels to be placed on the volume during initialization.

Abbreviations:

LABELS|LABEL|LBL

Default: None.

Restrictions: If LABELS is not specified, user volume labels that existed from a previous initialization are destroyed.

If a value greater than 9 is specified with the LABELS parameter, the INIT command terminates.

The LABELS parameter is ignored when the MIMIC(MSS) parameter is specified.

MAP|NOMAP: Printing a Volume Map

Explanation: **MAP**|**NOMAP** are optional parameters specifying whether to print a volume map during initialization. The volume map lists tracks that are found to be defective or inconsistent during initialization.

MAP
Indicates that you want to print a volume map during initialization.

NOMAP
Indicates that you do not want to print a volume map during initialization.

Abbreviations: None.

Default: The default is MAP.

Restrictions: Potential entries in the MAP are limited to the scope of tracks that were referenced during processing. For minimal initialization, referenced tracks include only the alternate tracks and any primary tracks that might be associated with them. For medial and maximal initialization, this also includes all tracks in the specified range. Note that if there are primary tracks incor-

rectly associated with alternates, that information is not available in a minimal initialization MAP.

MAP is ignored when processing minidisks under CMS.

MIMIC: Specifying a Special Volume Usage Format

Explanation: **MIMIC**(*type*) is an optional parameter specifying a special usage format for the volume being initialized. For *type* you can specify MINI(n), MSS, EMU(n), or EMUALL.

The MIMIC parameter is not supported under CMS. As the MIMIC minidisk functions can be performed under CMS, it is recommended that you process minidisks under CMS.

MINI(n)

Indicates that a VM minidisk is to be initialized for a VM environment. n represents the number of cylinders that are to comprise the minidisk. Details of this support are in Appendix B, "VM Support" on page 227 under "Minidisk Support in the Stand-alone Environment" on page 230

The MINI(n) parameter is valid in a stand-alone environment only, and requires the DEVICETYPE parameter to be specified.

MINI(n) should only be used with minimal INIT for all devices except 2311, 2314, and 2319.

A device that is LINKED to a userid operates as if it were a full volume minidisk. If media maintenance functions are required on a device in the VM environment, **the device must be ATTACHed to the USERID** that is executing ICKDSF, and MIMIC(MINI(n)) should not be used.

MSS

Indicates that the volume is to be formatted as a Mass Storage System staging pack. A one-track volume table of contents is created on cylinder 0, track 2 with a format-5 data set control block (DSCB) indicating that no tracks are available for allocation.

When the MIMIC(MSS) parameter is specified, the VTOC, LABELS, and IPLDD parameters do not apply and are ignored.

With the MIMIC(MSS) parameter, 3330-1 and 3330-11 are the only valid device types.

When initializing a 3330-1 as a Mass Storage System staging pack, cylinders 0 through 408 are identified as primary cylinders, and cylinders 409 and 410 are identified as alternate cylinders.

When initializing a 3330-11 as a Mass Storage System staging pack, cylinders 0 through 808 are identified as primary cylinders, and cylinders 809 through 814 are identified as alternate cylinders.

Note: Before Device Support Facilities can be run against a staging device, and before the drive can be accessed, the device must be in offline real status. Also, all paths to the staging pack must be VARY'd offline (except for the path used to VARY the device offline). See *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS* for details of the VARY OFFLINE, TEST command that is used to set up the device for nonstaging access.

EMU(n)
Indicates that a partial CKD disk emulated on a fixed block device is to be initialized. In a stand-alone environment, you must also specify the DEVICETYPE parameter.

EMUALL
Indicates that a full CKD disk emulated on a fixed block device is to be initialized.

Abbreviations: None.

Default: None.

Restrictions: A volume being initialized as a Mass Storage System staging pack must be initialized offline at the maximal level with the RECLAIM parameter specified.

If EMU(n) or EMUALL is specified for an emulated 3330, validation and surface checking functions are bypassed. Data checks should be handled using the INIT command against the base FBA device.

MIMIC is not valid under CMS.

OWNERID: Specifying the Owner Identification

Explanation: **OWNERID(*owner*)** is an optional parameter specifying the owner identification to be written in the volume or minidisk label.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification to be written in the volume or minidisk label.

Abbreviations:

```
OWNERID|OWNER
```

Default: The default for *owner* during a first-time initialization is 14 blanks.

If you do not specify OWNERID when reinitializing a previously initialized volume, the owner identification remains unchanged.

Passwords: Providing Passwords for Data Set Security

Explanation: PASSWORDS((*dsname/password*),•••) is an optional parameter specifying passwords for non-VSAM password-protected data sets. The supplied passwords will be used to determine if the user has authority to alter the data sets.

For *dsname*, substitute the fully qualified name of a password-protected data set. For *password*, substitute the password you wish to apply to this data set.

Abbreviations:

```
PASSWORD|PASSWD|PWD|PD
```

Default: None.

Restrictions: Up to a maximum of 512 passwords may be specified.

This parameter is ignored in the CMS, VSE, and stand-alone environments.

PURGE|NOPURGE: Writing Over Data on a Previously Initialized Volume

Explanation: **PURGE|NOPURGE** are optional parameters that specify whether you want to write over the data in the following types of data sets during initialization:

- Unexpired
- VSAM
- Password-protected
- RACF-protected

PURGE specifies that you want to write over the data. NOPURGE specifies that you do not want to write over the data. For more information on execution of the PURGE and NOPURGE parameters, see "Volume and Data Set Security" on page 31.

A volume that appears to contain real data cannot be initialized unless the PURGE parameter is specified.

PURGE
Indicates that you want to write over existing data during initialization. For more information about the actions taken when certain types of data are found on the volume, see "Volume and Data Set Security" on page 31.

NOPURGE
Indicates that you do not want to write over existing data. If a volume appears to contain real data, it cannot be initialized unless the PURGE parameter is specified.

Abbreviations:

```
PURGE|PRG
NOPURGE|NOPRG|NPRG
```

Default: The default is NOPURGE.

Restrictions: The NOPURGE parameter does not apply in offline mode or in a CMS or stand-alone environment. If you initialize a volume in offline mode or in a stand-alone environment, all existing data on the volume is purged, regardless of the data set security attributes.

RECLAIM|NORECLAIM: Reclaiming Tracks Previously Flagged Defective

Explanation: **RECLAIM|NORECLAIM** are optional parameters specifying whether to reclaim primary or alternate tracks that were flagged as defective in input but appear usable after surface checking.

Reclamation occurs only if surface checking is performed without errors for that track. For devices that support it, skip displacement surface checking is done on all tracks that are currently flagged defective before reclamation takes place. For devices that do not have skip displacement areas, later use of a track that is reclaimed can show defects that were undetected in the primary checking process. See Appendix G, "Surface Checking" on page 315, for more information.

RECLAIM
Indicates that you want tracks that are identified as usable by surface checking to be reclaimed during initialization.

NORECLAIM
Indicates that you want to suppress track reclamation during initialization.

Abbreviations:

```
RECLAIM|RCLM
NORECLAIM|NORCLM|NRCLM
```

Default: The default is NORECLAIM.

Restrictions: The RECLAIM parameter applies only when the CHECK parameter has been specified. If RECLAIM is specified with NOCHECK, it will be ignored.

The RECLAIM parameter is not valid for the IBM 3375 and 3380.

The RECLAIM parameter does not apply when initializing IBM 2305 Fixed Head Storage Models 1 and 2.

RECLAIM is not valid when using minidisks.

SKIP|NOSKIP: Specify the Level of Surface Checking

Explanation: **SKIP**|**NOSKIP** are optional parameters used to specify the level of surface checking to be performed when CHECK is specified.
Appendix G, "Surface Checking" on page 315, describes surface checking.

SKIP
Specifies that skip displacement checking is to be performed on all tracks within the specified range.

IF SKIP is specified, the maximum total number of tracks within the combination of range parameters cannot exceed the number of tracks contained in one cylinder for the specified device type. Because of this limitation, when you specify SKIP, you must also use the range parameters to specify part of a volume.

Because SKIP does extensive surface checking for all tracks within the range, the run time can be excessive, even with the one cylinder limit. (Extensive surface checking can take up to 5 minutes per track or more, depending on the device type and system conditions.)

SKIP is valid for the IBM 3340, 3344, and 3350.

NOSKIP
Specifies that no skip displacement checking is to be performed. Primary checking only is to be performed on all tracks within the specified range.

If a data check is detected on a track during primary checking, skip displacement checking is performed for that track.

Default: NOSKIP

Restrictions: The SKIP/NOSKIP parameter is ignored if NOCHECK is specified

If SKIP is specified, the value of n in CHECK(n) is ignored.

SKIP is not valid for the IBM 3375 and 3380.

Not valid when using minidisks under CMS.

STORAGEGROUP: Initialize a Volume that is managed by SMS

Explanation: **STORAGEGROUP** is an optional parameter which specifies that the volume to be initialized is to be a Storage Management Subsystem (SMS) managed volume.

STORAGEGROUP
Specifies that the volume is to be initialized for use by SMS.

Abbreviations:

STGR|SG

Default: None

Restrictions: The STORAGEGROUP parameter and the INDEX parameter must both be specified to indicate an SMS-managed volume.

TORANGE: Specify a Specific Ending Location

Explanation: **TORANGE**(*cccc,hhhh*) is an optional parameter used to specify that part of a volume is to be surface checked. Use this parameter to specify the ending track (cylinder and head) to be examined before minimal volume initialization is performed.

For *cccc,hhhh* substitute the 1 to 4 decimal (n) or hexadecimal (X'n') digits to identify the cylinder and head where processing is to end.

If you specify TORANGE and do not specify FROMRANGE, FROMRANGE is defaulted to cylinder 0 track 0.

For information on specifying the starting track of part of a volume (FROMRANGE) see "FROMRANGE: Specify a Specific Starting Location" on page 113.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Part of a Volume" on page 104.

Abbreviations:

TORANGE|TO

Default: None

Restrictions: TORANGE applies only when CHECK or VALIDATE is specified.

TORANGE is not valid when using minidisks.

VALIDATE|NOVALIDATE: Validating the Home Address and Record 0

Explanation: **VALIDATE|NOVALIDATE** are optional parameters specifying whether to validate the home address and record 0 for each track in the specified range. VALIDATE|NOVALIDATE determines if the device is to be initialized at the medial level (when NOCHECK is specified). Validation is less time-consuming than surface checking, but does read, validate, and rewrite the

home address and record 0 on each track. Validation causes the remainder of the data on a track to be erased.

VALIDATE
Indicates that a medial initialization is to be performed. The home address and record 0 for each track in the specified range are read, validated, and rewritten. Primary tracks that are flagged defective when the home address is read will have an alternate assigned to them.

NOVALIDATE
Indicates that you do not want to validate the home address and record 0 for each track during initialization. When you specify NOVALIDATE with the NOCHECK parameter, you initialize the volume at the minimal level.

Abbreviations:

```
VALIDATE|VAL
NOVALIDATE|NOVAL|NVAL
```

Default: The default is NOVALIDATE.

Restrictions: The VALIDATE and NOVALIDATE parameters are ignored when the CHECK parameter is specified.

VALIDATE is not valid when using minidisks.

VOLID: Specifying the Volume Serial Number

Explanation: **VOLID**(*serial*) is an optional parameter specifying the volume serial number to be written in the volume or minidisk label.

For *serial*, substitute one to six alphameric characters for the volume serial number. If fewer than six characters are specified, the serial is left-justified, and the remainder of the field is padded with blanks (X'40').

Abbreviations: None.

Default: When you initialize a volume or minidisk that was previously initialized and do not specify the VOLID parameter, the old volume serial number remains unchanged.

Restrictions: When you initialize a volume or minidisk for the first time, you must specify the VOLID parameter, or the INIT command terminates.

If you change the volume serial number of a volume that is mounted online, thereafter you must reference the volume by the new volume serial number, even though the reference may occur in the same step.

VTOC: Specifying the Volume Table of Contents

Explanation: **VTOC(END|***cylinder,head[,extent]***)** is an optional parameter specifying the location and size of the volume table of contents.

For *cylinder,head*, specify one to four decimal (n) or hexadecimal (X'n') numbers to identify the cylinder and one to four decimal (n) or hexadecimal (X'n') numbers to identify the head where the volume table of contents is to be placed.

For *extent*, specify one to five decimal (n) or hexadecimal (X'n') numbers for the number of tracks that are to be reserved for the volume table of contents. Refer to Appendix E, "The VTOC Index" on page 305 for information on specifying the size of the VTOC index.

The default for *extent* is one track.

When you specify VTOC(END), the VTOC is put on the last primary cylinder of the volume (or minidisk under CMS) and is one cylinder in length.

Abbreviations: None.

Default: The default for *cylinder,head* is 0,1; when initializing a Mass Storage System staging pack, however, the default is 0,2.

If neither VTOC nor DOSVTOC (or VSEVTOC) is specified, the default taken is an MVS VTOC at cylinder 0, track 1.

If neither VTOC nor DOSVTOC (or VSEVTOC) and the device is not a minidisk, the size is defaulted to the number of tracks in a cylinder minus one.

Restrictions: If VTOC is specified with the MIMIC(MSS) parameter, the VTOC parameter is ignored.

A VTOC cannot be placed at cylinder 0, track 0.

Examples of the INIT Command

The following examples show different ways to code the INIT command. Most are for the offline and online modes of MVS.

To initialize a volume in a stand-alone environment, do the following:

- Eliminate the MVS JCL.
- Substitute the UNITADDRESS parameter for the DDNAME parameter.
- Specify the DEVICETYPE parameter if the MIMIC(MINI) or MIMIC(EMU) parameter is also specified.
- Substitute SYSIN for *dname* if the IPLDD parameter is specified.

To initialize a volume in the VSE environment, do the following:

- Replace the MVS JCL statements with VSE system control statements.
- Substitute the SYSNAME parameter for the DDNAME parameter.
- Substitute DLBL name for *dname* if the IPLDD parameter is specified and the data set is located on a direct access volume. If the data set is included in stream, replace the *dname* with either SYSIN or SYSIPT.
- Introduce the VSEVTOC or DOSVTOC parameter if a VSE type VTOC is required on the volume.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Initializing a Volume for the First Time in Offline Mode (MVS)

In this example, a volume is initialized at the minimal level because neither the CHECK nor VALIDATE parameter is specified. Because the volume is being initialized for the first time, it must be mounted offline, and the volume serial number must be specified. Because the VTOC parameter is not specified, the program supplies a default volume table of contents at cylinder 0, track 1 that occupies one track.

```
//EXAMPLE   JOB
//          EXEC  PGM=ICKDSF
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
 INIT UNITADDRESS(353) NOVERIFY VOLID(VOL123) -
      OWNERID(PAYROLL)
/*
```

Initializing a Volume at the Minimal Level in Online Mode (MVS)

In this example, a volume is reinitialized at the minimal level because neither the CHECK nor VALIDATE parameter is specified. Because the VOLID and OWNERID parameters are not specified, the volume serial number and owner identification that exist on the volume remain unchanged on the reinitialized volume. Because the VTOC parameter is not specified, the program creates a default volume table of contents at cylinder 0, track 1 that occupies one track. By accepting the default parameter NOPURGE, the command terminates if unexpired or password-protected data sets are found. Volume PAY456 was previously mounted as PRIVATE.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//XYZ987   DD    UNIT=3330,DISP=OLD,VOL=SER=PAY456
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 INIT DDNAME(XYZ987) NOVERIFY
/*
```

Initializing a Volume at the Medial Level in a Stand-alone Environment

In this example, a previously initialized volume is reinitialized at the medial level because the VALIDATE parameter is specified and the CHECK parameter is not. The home address and record 0 on each track are read and validated. The volume serial number and owner identification are verified before initialization continues and, because a new volume serial number or an owner identification is not specified, they remain unchanged on the reinitialized volume. Specification of the VTOC parameter causes a volume table of contents to be created at cylinder 42, track 12, that occupies three tracks. Because the volume is being reinitialized in a stand-alone environment, all data sets are purged, regardless of whether they are unexpired or password protected.

```
INIT UNITADDRESS(272) -
    VERIFY(VOL123,PAYROLL) VALIDATE VTOC(42,X'C',3)
```

Initializing a 3375 or 3380 Volume at the Medial Level in a Stand-alone Environment

In this example, a previously initialized 3375 or 3380 volume is reinitialized at the medial level because the VALIDATE parameter is specified. DATA is specified, causing the three records of factory functional verification data patterns to be written on the volume. The home address and record 0 on each track are read and validated. The volume serial number and owner identification are verified before initialization continues and, because a new volume serial number

or an owner identification is not specified, they remain unchanged on the reinitialized volume. Because the volume is being reinitialized in a stand-alone environment, all data sets are purged, regardless of whether they are unexpired or password protected.

```
INIT UNITADDRESS(272) VOLID(338001) VALIDATE DATA -
    VERIFY(VOL222,MASTER)
```

Initializing a Volume at the Maximal Level in a Stand-alone Environment

In this example, a 3330 volume is initialized at the maximal level because the CHECK parameter is specified. If the results of surface checking show tracks are nondefective but were previously marked defective, the tracks are reclaimed. The volume serial number and owner identification remain unchanged. The volume table of contents begin at cylinder 2, track 0 and occupy one track. Because the IPLDD parameter is specified and the IPL program data is supplied (indicated by the bullets in the example), the IPL bootstrap records will be supplied by the program by default. Because the volume is being reinitialized in a stand-alone environment, all existing data is purged, regardless of whether it is unexpired or password protected.

Note that this example is not valid for the IBM 3375 and 3380.

```
INIT UNITADDRESS(350) CHECK(3) RECLAIM -
    NOVERIFY IPLDD(SYSIN) VOLID(333010) VTOC(2,0,1)
    •
    •
    •
```

Surface Checking Part of a Volume in a Stand-alone Environment

In this example, primary surface checking is performed for a 3350 beginning at cylinder 4, head 6 and ending at cylinder 50, head 8. At the completion of surface checking, minimal initialization is performed.

```
INIT UNIT(ccuu) VOLID(335001) NOVFY CHECK(2) NOSKIP -
     FROMRANGE(4,6) TORANGE(50,8)
```

In this example, primary surface checking is performed on all heads of the 3350 volume, beginning at cylinder 50 and ending at cylinder 100.

```
INIT UNIT(ccuu) VOLID(335001) NOVFY CHECK(2) NOSKIP CYLRANGE(50,100)
```

In this example, primary surface checking is performed on head 4 only for all cylinders of the 3350 volume. Surface checking is performed beginning at cylinder 0, head 4, and ending at the last alternate cylinder, head 4.

```
INIT UNIT(ccuu) VOLID(335001) NOVFY CHECK(2) NOSKIP HEADRANGE(4,4)
```

Initializing Two Volumes in Offline Mode (MVS)

In this example, the VALIDATE parameter without the CHECK parameter in the first INIT command causes medial level initialization. The volume serial number is verified before initialization continues and is then replaced by the number supplied in the command. A default volume table of contents begins at cylinder 0, track 1 and occupies one track. Four user volume labels are written in addition to the volume label. Because the volume is being reinitialized in offline mode, all existing data is purged, regardless of whether it is unexpired or password protected. The IPLDD parameter specifies that a user-supplied IPL program is to be written, and the BOOTSTRAP parameter specifies that the

bootstrap records are also user-supplied. (The IPL program and bootstrap records are indicated by the vertical ellipsis.)

The second INIT command initializes a volume at the minimal level. The volume serial number and owner identification remain unchanged. A default volume table of contents is written at cylinder 0, track 1 and occupies one track. Because the volume is being reinitialized in offline mode, all existing data is purged, regardless of whether it is unexpired or password protected.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//IPLDECK  DD    *
•
•
•
/*
//SYSPRINT DD    SYSOUT=A
//SYSIN     DD    *
 INIT UNITADDRESS(353) VALIDATE VOLID(AAA789) -
      OWNERID(SMITH) VERIFY(XYZ123) -
      IPLDD(IPLDECK) BOOTSTRAP LABELS(4) NOMAP
 INIT UNITADDRESS(271) NOVERIFY
/*
```

Initializing a Volume at the Maximal Level in Offline Mode (MVS) as a Mass Storage System Staging Pack

In this example, the RECLAIM parameter with the CHECK parameter causes maximal level initialization. The volume serial number and owner identification are not changed. A default volume table of contents begins at cylinder 0, track 2 and occupies one track. Because the volume is being reinitialized in offline mode, all existing data is purged, regardless of whether it is unexpired or password protected.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 INIT UNITADDRESS(253) NOVERIFY CHECK RECLAIM -
      MIMIC(MSS)
/*
```

Initializing a Minidisk in a Stand-alone Environment

In this example, a mini 3330 is initialized with an IPL program. It provides 30 primary and no alternate cylinders on unit 151. The VTOC is written on a default location of cylinder 0, track 1 for a length of one track. The volume is labeled 333000. The IPL program supplied as the input deck is written on cylinder 0, track 0. The label and VTOC are written in OS format.

```
INIT  UNITADDRESS(151) NOVERIFY DEVICETYPE(3330) -
    MIMIC(MINI(30) VOLID(33300) IPLDD(SYSIN)
```

Initializing a Minidisk in a CMS Environment

In this example, a minidisk is initialized at the minimal level. It establishes a VOLID, OWNERID, and the VTOC on a minidisk with a virtual address of 391. The VTOC is written at the default location of cylinder 0, track 1 of the minidisk for a length of one track. The minidisk is labeled 338012 and the owner ID is PAYROLL. The VTOC is written in MVS format.

After performing the INIT, the minidisk is ready for future use in an MVS operating system environment.

```
INIT  UNITADDRESS(391) NOVERIFY VOLID(338012) OWNERID(PAYROLL)
```

Initializing a Volume at Minimal Level in the VSE Environment

In this example, a volume is initialized at a minimal level under VSE. A VSE format VTOC is written on cylinder 32, track 0 for a length of 20 tracks. The volume is labeled 231401.

```
// JOB      jobname
// ASSGN    SYS002,151
// EXEC     ICKDSF,SIZE=AUTO
     INIT SYSNAME(SYS002) NOVERIFY -
          VSEVTOC(X'20',X'0',X'14') VOLID(231401)
/*
/&
```

Initializing an Emulated Partial Disk at Minimal Level in the VSE Environment

In this example, an emulated partial disk is initialized under VSE. A VSE format VTOC is written at cylinder 0, track 1 for a length of 1 track. The volume is labeled AA2311.

```
// JOB      jobname
// ASSGN    SYS000,353
// EXEC     ICKDSF,SIZE=AUTO
    INIT SYSNAME(SYS000) NVFY VSEVTOC(0,1,1) -
         VOLID(AA2311) MIMIC(EMU(20))
/*
/&
```

Initializing a Volume with an Index (MVS)

This example performs an online minimal initialization, and as a result of the command, an index to the VTOC is created.

```
//         JOB
//         EXEC  PGM=ICKDSF
//XYZ987   DD    UNIT=3375,DISP=OLD,VOL=SER=PAY456
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 INIT DDNAME(XYZ987) NOVERIFY INDEX(X'A',X'B',X'2')
/*
```

Initializing a Volume to Be Managed by SMS

In the following example, a volume which is to be managed by the Storage Management Subsystem (SMS) is initialized. The volume is initialized in offline mode at the minimal level. The VTOC is placed at cylinder 2 track 1, and occupies ten tracks, and is followed by the VTOC index. The STORAGEGROUP parameter indicates the volume is to be SMS managed.

```
 INIT UNIT(353) NOVERIFY VOLID(VOL123) STORAGEGROUP -
      OWNERID(PAYROLL) VTOC(2,1,10) INDEX(2,11,5)
```

Initializing a Previously SMS-Managed Volume

The following example initializes a volume which has been previously initialized for Storage Management Subsystem (SMS) use. The volume is initialized at the minimal level in online mode. The VTOC is placed at cylinder 10 track 1, and is allotted twenty tracks. The VTOC index is placed at cylinder 20 track 1 and is allotted 10 tracks. The STORAGEGROUP parameter specifies that the volume is to be an SMS volume. When the VOLID and OWNERID are specified, the volume serial and owner are changed. If there are any data sets on the volume, a re-initialize cannot be performed.

To initialize a previously SMS-managed volume to a non-SMS-managed volume, do not include the STORAGEGROUP parameter in this example.

```
//EXAMPLE     job
//            EXEC   PGM=ICKDSF
//DDN1        DD     UNIT=3380,DISP=OLD,VOL=SER=XXX001
//SYSPRINT    DD     SYSOUT=A
//SYSIN       DD     *
  INIT DDNAME(DDN1) NOVERIFY VOLID(IMS123) STORAGEGROUP -
       OWNERID(DBGROUP) VTOC(10,1,20) INDEX(10,21,10)
/*
```

INSPECT Command—CKD

The INSPECT command inspects a subset of a volume and can:

- Check the surface of a track to determine if there is a defect
- Skip a defect (for specified devices)
- Assign an alternate track
- Reclaim a track that has been flagged defective
- Print a map of defective tracks on a volume.

Using the INSPECT Command

Several options are available for inspection:

- The range of tracks (CYLRANGE/HEADRANGE or FROMRANGE/TORANGE) and TRACKS parameters are provided to surface check individual tracks or groups of tracks.
- The CHECK parameter is provided to check each track surface and to permit optional specification of the number of checking cycles that are to be performed on each track being inspected. Alternate track assignment is done only after an inspection of the surface reveals defects. The NOCHECK parameter suppresses such checking and, unless alternate track assignment is suppressed, an alternate will be unconditionally assigned to the primary track.

  For the IBM 3340, 3344, 3350, 3375, and 3380 when CHECK is specified, surface checking locates and skips over defective areas on the track. Only when the primary track contains more than the maximum number of allowable defects is it considered defective.
- The SKIP parameter allows you to control the level of surface checking performed and, consequently, the total execution time involved with inspecting individual or groups of tracks.
- The PRESERVE, KEEPIT, and HOLDIT parameters are provided to save data that would normally be destroyed during track inspection. The NOPRESERVE parameter allows inspection of tracks when the data cannot be read.

  losing data warning

  When the HOLDIT parameter is specified, the data is only kept in storage. If processing terminates before the data is rewritten, the track for which the data was preserved will **not** contain a standard record zero, and **the data will be lost**.

  When the PRESERVE option is specified, a copy of the data in storage is written at a backup location as well as kept in storage. If processing ends abnormally, the next use of INSPECT detects the data at the backup location and automatically restores the data to the track on which processing previously ended abnormally.

- The RECLAIM parameter is provided to reclaim a track that was previously flagged defective if track inspection has determined it is currently defect-free. The NORECLAIM parameter suppresses automatic track reclamation.
- The ASSIGN parameter allows alternates to be assigned if required. Assignment of alternates is suppressed by the NOASSIGN parameter.
- The MAP parameter provides printed output of the detailed status of each track being inspected. The NOMAP parameter suppresses the printed output.
  - Specify NOCHECK NOASSIGN ALLTRACKS for a map.

Data Integrity When Using the INSPECT Command

When an INSPECT is being performed, the RESERVE macro is issued to obtain control of the device. DEQ is issued to release the device after processing is completed.

When the INSPECT command is used to perform skip displacement processing (using the SKIP parameter), if a track is part of a data set, INSPECT will enqueue on the data set while processing that track.

Additionally, if the INSPECT command exposes user data for a relatively long period of time, ICKDSF enqueues for exclusive control of the data set containing the track which is to be processed. Data set enqueue takes place when:

- The device is online to the operating system.
- The track is part of a non-VSAM data set.
- The SKIP parameter is specified.
- NOSKIP is specified, but primary surface checking indicates that skip displacement processing is neccessary for the track.

If exclusive control of a data set cannot be obtained, a message is issued, and processing continues on the next track.

The TOLERATE(ENQFAIL) parameter is provided to indicate that processing is to continue if the data set enqueue fails when SKIP is specified.

If the enqueue is performed as a result of primary surface checking indicating the need for skip displacement surface checking, TOLERATE(ENQFAIL) is assumed to be in effect.

For tracks in VSAM data sets, TOLERATE(ENQFAIL) must always be specified when processing online.

Controlling the Level of Surface Checking

When you specify CHECK(n) the level of surface checking which is performed for each track in a specified range is controlled by the SKIP/NOSKIP parameters. If you specify NOCHECK, no surface checking is performed. See Appendix G, "Surface Checking" on page 315, for more information on surface checking.

NOSKIP indicates that primary surface checking is to be performed for all tracks in the specified range, and skip displacement surface checking is performed only if a data check is detected. NOSKIP is the default if you specify any of the range parameters.

SKIP indicates that skip displacement surface checking is to be performed for *every* track in the specified range. Although SKIP provides a more extensive check of the surface of a track, it runs for long periods of time per track. If SKIP is specified, the total number of tracks in the specified range is limited to the number of tracks in a cylinder. SKIP is the default if you specify the TRACKS parameter.

RANGE Parameters: Specifying Discrete Tracks

CYLRANGE/HEADRANGE or FROMRANGE/TORANGE are optional parameters, generally specified as a pair, to limit the areas of a volume that are to be inspected. Use the CYLRANGE/HEADRANGE pair to specify the starting and ending cylinder(s) and the starting and ending head(s) to be inspected. Use the FROMRANGE/TORANGE pair to specify the starting track (cylinder and head) and the ending track (cylinder and head) to be inspected.

The following notes apply to specifying part of a volume:

- CYLRANGE/HEADRANGE and FROMRANGE/TORANGE are mutually exclusive pairs of parameters. Either pair is mutually exclusive with TRACKS.
- If you specify CYLRANGE and do not specify HEADRANGE, all the heads of the specified cylinders are processed.
- If you specify HEADRANGE and do not specify CYLRANGE, the specified heads for all the cylinders on the volume are processed.
- Both the starting and ending values are required for CYLRANGE and HEADRANGE. By specifying an ending value greater than the maximum for the device, processing is performed to the maximum value for that device.
- If you specify FROMRANGE and do not specify TORANGE, TORANGE is defaulted to the last alternate cylinder.
- If you specify TORANGE and do not specify FROMRANGE, FROMRANGE is defaulted to cylinder 0 track 0.
- If NOCHECK is specified with any of the range parameters, the home address and record zero of all tracks in the specified range are read. This can be used to verify alternate/primary track assignments of a subset of a volume.
- If you want to verify the alternate tracks on a volume (that is, primary/alternate pairs are valid), specify the range parameters for the alternate cylinders.
- Range parameters used with NOPRESERVE can destroy large areas of a volume.
- NOCHECK ASSIGN is not valid when used with the range parameters. For unconditional track assignment, TRACKS must be used.
- The total number of tracks that can be inspected is 2500 if NOSKIP is specified. If SKIP is specified, the total cannot exceed the number of tracks in a cylinder for the specified device type.

Recovering Data After a Job or System Failure

If an INSPECT command did not execute to completion, and PRESERVE was specified on the failing job, ICKDSF has saved the data; it will be automatically recovered on the next use of the INSPECT command.

After resolving the reason for the failure, you can submit:

- The same INSPECT job.

  This does a primary surface check of the track being processed at the time of failure, recovers the data, and inspects all the tracks in the specified range, including those tracks that were inspected prior to the previous failure.

- An INSPECT job for a totally different, or partially different range.

  This does a primary surface check of the track being processed at the time of failure, recovers the data, and processes all tracks in the new range.

- An ANALYZE job (for nonemulated nonremovable media devices only) to determine the track that failed.

  ANALYZE will print the cylinder and head of the track that was being processed at the point of failure.

  — Then you can execute a new INSPECT job with a modified range, according to the previous point of failure.

- Execute an INSPECT with neither TRACKS nor any of the range parameters specified.

  This does a primary surface check of the track being processed at the time of failure, and recovers the data.

Notes:

- Recovery is only possible if PRESERVE was specified on the job running at the time of failure. It is not possible if HOLDIT or NOPRESERVE were specified or if you were processing minidisks under CMS
- It is possible that the track that is being processed at the time of failure contained no data. In that case, recovery consists of only doing a primary surface check of the track. This ensures that the track is usable (that is, that the track now contains just a home address and a standard record zero).
- It is possible that INSPECT was "between tracks" when the job failed, and there is no track or data to recover.
- *Caution.* Be careful about using the volume before recovering the data. In addition to the data being unavailable, the track may be unusable to the operating system, and any use of the track can cause I/O errors.
- **If you run INITialize on this volume before the next INSPECT, INITialize erases the data at the backup location.**
- The restore function is not optional. If data exists at the backup location, it is restored.

If there is data on the track being restored (that is, the track being processed at the point of failure), the operator is prompted either to restore the data or erase the data.

If the data cannot be restored, for whatever reason, the operator is prompted for the next action.

To put the restored data in storage, you can run a job that specifies HOLDIT.

- When you are processing minidisks under CMS, data is not saved at the backup location. Therefore, no recovery of data is possible.

Preserving Data during INSPECT

When you specify the PRESERVE parameter of the INSPECT command, a backup copy of the data on a track is stored at the backup location as well as held in storage. This backup location uses two tracks. One track (the "recovery" track) contains the control information, and another track (the "preserve" track) contains the actual data. This information resides on:

- The CE cylinder of the volume, if there is a uniquely addressable CE cylinder on the volume.
- The SA cylinder of the volume, if there is no CE cylinder, but there is an SA cylinder with unused tracks.
- An alternate track, if the device does not have a CE cylinder or an SA cylinder.

If processing ended abnormally, the next use of INSPECT will:

- Detect the data at the backup location, and determine the track to be recovered (the recovery track).
- The home address and record zero of the recovery track are checked for validity.
- The track to be restored is surface checked at the primary level unless a data check is detected. If a data check is detected, skip displacement surface checking is performed.
- The data is restored.

The restoration of the track from the preserved data is automatic, and is independent of the range specified in the current job. INSPECT can be run with no tracks specified and just the restore function is performed.

INSPECT allows you to choose from two levels of data preservation:

- HOLDIT specifies that a copy of the data on the track being processed is to be kept in storage only, with no copy being kept at the backup location.
- PRESERVE or KEEPIT specifies that a copy of the data on the track being processed is to be written at the backup location as well as kept in storage. The copy at the backup location is used only during the restore procedure.

  The backup location and the copy of the data in storage are erased at the completion of the command.

Even when a track that contains no data is processed, the backup location is still accessed. This ensures that the integrity of the volume can be restored if a restart is necessary.

PRESERVE/KEEPIT is the default.

When PRESERVE/KEEPIT is specified, you might notice a performance degradation caused by the writing of the data at the backup location. The performance degradation depends on the total number of tracks being processed by the current INSPECT command.

Some cautions to be used for PRESERVE processing:

- When data is written at the backup location, it is read-back-checked to ensure it was written correctly. However, like any DASD write operation, at the time of recovery when the data is to be restored, **it cannot be guaranteed that the data is always retrievable from the backup location**.
- If the backup location is the CE cylinder, the ANALYZE command (which writes on the CE cylinder) will **NOT** destroy restore data if it exists. ANALYZE indicates the cylinder and head of the track that originally had the problem. **Note, however, that CE microdiagnostics may destroy the data at the backup location**. Therefore, if INSPECT processing does not run to completion, and it is determined to be a problem that will cause the volume to be turned over to the IBM customer engineer, the INSPECT command should be executed to restore the data.

Assigning Alternate Tracks

A track can be flagged as defective, and an alternate assigned, either conditionally or unconditionally. The amount of surface checking done before an alternate is assigned is determined by multiple factors, including:

- The use of the CHECK or NOCHECK parameter
- The availability of skip displacement bytes for the device type
- The current condition of the track

For a detailed description of the type of checking performed, see Appendix G, "Surface Checking" on page 315. In general, you can control the assignment of alternate tracks as follows:

- Issue the INSPECT command and specify the CHECK and ASSIGN parameters to detect defective primary and alternate tracks. If a primary or alternate track is identified as defective by surface checking, the track surface is flagged as defective, and an alternate surface is assigned if the defective track is a primary one.
- Issue the INSPECT command and specify the NOCHECK and ASSIGN parameters to unconditionally assign alternate surfaces for the tracks specified.
- To inspect primary and alternate tracks, but to prevent assignment of alternates, issue the INSPECT command and specify the NOASSIGN and CHECK parameters.

  Note that if PRESERVE is specified and user data cannot be rewritten because of surface defects, an alternate is assigned regardless of the ASSIGN|NOASSIGN parameter value.

This option will indicate, in the map, all defective tracks but alternates will not be assigned. Skip displacement *will* be assigned as required.

- If an inspected primary track is identified as defective, the data is rewritten on the alternate that is assigned. If an alternate track that is associated with a primary surface is identified as defective, the alternate is flagged as defective, and the data is rewritten to a different alternate. If a primary track that was previously flagged as defective is reclaimed, the data is copied from the alternate surface to the reclaimed primary surface, and the alternate surface is made available.

Using the INSPECT Command under CMS

When you run ICKDSF under CMS you can run in two modes:

- Minidisk mode
- ATTACHed mode

Minidisk Support under CMS

To use INSPECT under CMS you must have DEVMAINT authority (as defined in the CP directory) and specify either the USERID or REALADDR parameter. This allows you to INSPECT another user's minidisk (USERID) or INSPECT a volume specifying the real address of the volume (REALADDR) as follows:

USERID: When you have DEVMAINT authority and specify the USERID parameter, you can INSPECT another user's minidisk. The following parameters are valid in this mode:

- ALLTRACKS, ASSIGN/NOASSIGN, CHECK/NOCHECK, CYLRANGE, FROMRANGE, HEADRANGE, HOLDIT, MAP/NOMAP, RECLAIM/NORECLAIM, SKIP/NOSKIP, TORANGE, TRACKS, UNITADDRESS, and USERID

REALADDR: When you have DEVMAINT authority and specify the REALADDR parameter, you can INSPECT a volume using the the real device address: The following parameters are valid when you specify REALADDR:

- ASSIGN/NOASSIGN, CHECK/NOCHECK, CYLRANGE, FROMRANGE, HEADRANGE, HOLDIT, MAP/NOMAP, RECLAIM/NORECLAIM, SKIP/NOSKIP, TORANGE, TRACKS.

Note: The maximum number of cylinders you can INSPECT in this mode is one.

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed to your user ID. When you use ICKDSF in this mode, you can perform any ICKDSF function which can be performed when you are running ICKDSF stand-alone.

Syntax

Command	Required Parameters
INSPECT	**DDNAME(***dname***)|SYSNAME(***sysxxx***) |UNITADDRESS(***ccuu***) |REALADDR(***ccuu***) VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	ALLTRACKS **ASSIGN|NOASSIGN** **CHECK(***n***)|NOCHECK** **CYLRANGE(***cccc,cccc***)** **DEVICETYPE(***devtype***)** **FROMRANGE(***cccc,hhhh***)** **HEADRANGE(***hhhh,hhhh***)** **MAP|NOMAP** **MIMIC(***type***)** **PASSWORDS((***dsname/password***),•••)** **PRESERVE|KEEPIT|NOPRESERVE|HOLDIT** **RECLAIM|NORECLAIM** **SKIP|NOSKIP** **TOLERATE(ENQFAIL)** **TORANGE(***cccc,hhhh***)** **TRACKS((***cylinder,head***)[,•••])** **USERID(***user's ID***)**

Required Parameters

Note that various combinations of parameters can cause INSPECT to function differently. These differences are explained below.

DDNAME|SYSNAME|UNITADDRESS|REALADDR: Identifying the Device

Explanation: **DDNAME|SYSNAME|UNITADDRESS|REALADDR** are required parameters identifying the volume (or minidisk under CMS) that is to be inspected.

DDNAME(*dname*)
Required when inspecting a volume that is mounted online.

For *dname*, specify the MVS JCL DD statement that identifies the device on which the volume is mounted.

SYSNAME(*sysxxx*)
Required when inspecting a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN system control statement.

UNITADDRESS(*ccuu*)
Required when inspecting a volume that is mounted offline or when inspecting a volume in a stand-alone environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual address of a minidisk.

REALADDR(*ccuu*)
Required to specify the real address of a volume. This parameter is only valid when you are running ICKDSF under CMS and you have DEVMAINT authority.

For *ccuu*, specify the real address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Abbreviations:

DDNAME|DNAME
UNITADDRESS|UNIT

Default: None.

Restrictions: The DDNAME parameter cannot be specified when inspecting a volume in a stand-alone, CMS, or the VSE environment or that is mounted offline

The SYSNAME parameter cannot be specified when inspecting a volume in the stand-alone, CMS, or MVS environment.

The UNITADDRESS parameter cannot be specified when inspecting a volume that is mounted online or in the VSE environment.

REALADDR is mutually exclusive with UNITADDRESS, DDNAME, SYSNAME, and USERID.

The maximum number of cylinders that you can INSPECT using REALADDR is one.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(*serial[,owner]*)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before inspecting the volume.

VERIFY
Required when you want to verify the volume serial number and owner identification before inspecting the volume. If the volume serial number or owner identification does not match that found on the volume, the INSPECT command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY
Required when you want to bypass verification of the volume serial number and owner identification.

Abbreviations:

VERIFY|VFY
NOVERIFY|NOVFY|NVFY

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number. Both the volume serial number and owner identification must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, execution of the command terminates.

Optional Parameters

ALLTRACKS: Inspect the Total Volume

Explanation: **ALLTRACKS** is an optional parameter specifying all tracks on the volume (or minidisk inder CMS) are to be processed.

ALLTRACKS is valid only when NOASSIGN and NOCHECK are specified.

NOCHECK NOASSIGN with ALLTRACKS reads all home address records and record zeros on a volume (or minidisk inder CMS) and issues messages if any discrepancies are found. A primary/alternate track assignment map is also produced.

Abbreviations:

ALLTRACKS|ALLTRACK|ALLTRKS|ALLTRK

Default: None.

ASSIGN|NOASSIGN: Specifying Assignment of Alternate Tracks

Explanation: **ASSIGN**|**NOASSIGN** are optional parameters specifying whether alternates are to be assigned if defective primary tracks are detected.

ASSIGN

Indicates that the specified tracks can be flagged defective as required. If the specified track is a primary track, an alternate track is assigned to it. If the specified track is an alternate track that has a previously assigned primary track associated with it, then the primary track will have a new alternate assigned to it.

When used with CHECK(n), any track that is declared defective will undergo assignment. See Appendix G, "Surface Checking" on page 315, for an explanation of how a track is declared defective.

When used with NOCHECK, assignment takes place for all tracks being processed.

NOASSIGN

Indicates that you do not want alternate tracks assigned. When used with NOCHECK, this combination will cause ICKDSF to read the home address and record zero for the specified track(s) and issue a message if anything invalid is detected.

NOCHECK NOASSIGN with ALLTRACKS reads all home address records and record zeros on a volume. A primary/alternate track assignment map is also produced.

Notes:

1. If NOASSIGN is specified with the CHECK parameter, and the track is defective, the inspected track is not marked defective and an alternate is not assigned, regardless of the declared condition of the track. A message is issued for this condition.
2. If CHECK/PRESERVE or CHECK/HOLDIT is specified and user data cannot be rewritten because of surface defects, NOASSIGN is overridden and assignment takes place.

Abbreviations:

```
ASSIGN|ASGN
NOASSIGN|NOASGN|NASGN
```

Default: The default is ASSIGN.

Restrictions: The combination of ASSIGN NOCHECK cannot be specified with the range parameters (FROMRANGE/TORANGE or CYLRANGE/HEADRANGE).

The TRACKS parameter must be used for unconditional alternate track assignment.

CHECK|NOCHECK: Surface Checking of the Volume

Explanation: **CHECK(*n*)|NOCHECK** are optional parameters specifying whether the tracks are to be checked for recording errors. Each track is surface checked by writing and reading specially patterned records. Upon completion of surface checking, the home address and a standard record 0 are rewritten on the track.

CHECK

Indicates that you want each track to be surface checked for recording errors during inspection and declared defective only if the checked track appears defective.

For *n*, substitute a decimal number from 1 through 10 for the number of times you want each track to be checked. See Appendix G, "Surface Checking" on page 315, for information on values of n.

The level of surface checking to be performed is determined by the SKIP/NOSKIP parameters. See "SKIP|NOSKIP: Specify the Level of Surface Checking" on page 148, for more information on these parameters.

NOCHECK

Indicates that you do not want each track to be surface checked for recording errors during inspection. Alternate tracks are unconditionally assigned when NOCHECK and ASSIGN are specified. NOCHECK ASSIGN cannot be specified with ALLTRACKS.

NOCHECK NOASSIGN with ALLTRACKS reads all home addresses and record zeros on a volume (or minidisk under CMS) to ensure validity. A primary/alternate track assignment map is also produced.

The TRACKS parameter must be used for unconditional alternate track assignment.

Abbreviations:

CHECK|CHK
NOCHECK|NOCHK|NCHK

Default: The default is CHECK(1).

Restrictions: The maximum number you can specify with the CHECK(n) parameter is 10.

CYLRANGE: Specify a Range of Cylinders

Explanation: **CYLRANGE(*cccc,cccc*)** is an optional parameter used to specify that part of a volume (or minidisk under CMS) is to be inspected. Use the CYLRANGE parameter to specify the starting and ending cylinder(s) to be processed.

For *cccc,cccc*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the starting and ending cylinders to be examined.

If you specify CYLRANGE and do not specify HEADRANGE, all the heads of the specified cylinders are processed.

For information on specifying the starting and ending heads (HEADRANGE) to go with the starting and ending cylinders, see "HEADRANGE: Specify a Range of Heads" on page 143.

For information on specifying part of a volume (or minidisk under CMS), see "RANGE Parameters: Specifying Discrete Tracks" on page 133.

Abbreviations:

CYLRANGE|CYL

Default: None

Restrictions: Both the starting and ending values are required for CYLRANGE. By specifying an ending value greater than the maximum for the device processing is performed to the maximum value for that device (or minidisk under CMS). However, if you specify REALADDR, the maximum is one cylinder.

CYLRANGE cannot be specified with FROMRANGE/TORANGE or TRACKS.

DEVICETYPE: Identifying the Type of Device

Explanation: **DEVICETYPE(*devtype*)** is required in a stand-alone environment when the MIMIC(MINI) or MIMIC(EMU) parameter is specified.

For *devtype*, substitute one to eight characters for the type of device on which the volume is mounted. (See Figure 2 on page 7 for valid device types.)

Abbreviations:

DEVICETYPE|DEVTYPE|DEVTYP

Default: None.

Restrictions: The DEVICETYPE parameter applies only when you are inspecting a volume in a stand-alone environment and MIMIC(MINI) or MIMIC(EMU) is specified. If DEVICETYPE is specified when it does not apply, it will be ignored.

When using a 3350 Direct Access Storage volume in 3330-compatibility mode, the device type must be specified as 3330-1 or 3330-11, as required. The 3333 Disk Storage and Control Models 1 and 11 must be specified as 3330-1 and 3330-11, respectively.

FROMRANGE: Specify a Specific Starting Location

Explanation: **FROMRANGE(*cccc,hhhh*)** is an optional parameter used to specify that part of a volume or minidisk, under CMS, is to be inspected. Use this parameter to specify the starting track (cylinder and head) to be processed.

For *cccc,hhhh*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the cylinder and head where processing is to start.

If you specify FROMRANGE and do not specify TORANGE, TORANGE is defaulted to the last alternate cylinder (last cylinder of the minidisk under CMS).

For information on specifying the ending track (TORANGE) to go with the starting track, see "TORANGE: Specify a Specific Ending Location" on page 150.

For information on specifying part of a volume to be examined, see "RANGE Parameters: Specifying Discrete Tracks" on page 133.

Abbreviations:

FROMRANGE|FROM

Default: None

Restrictions: FROMRANGE cannot be specified with CYLRANGE/HEADRANGE or TRACKS.

Under CMS if you specify REALADDR, the range must be contained within 1 cylinder.

HEADRANGE: Specify a Range of Heads

Explanation: **HEADRANGE(*hhhh,hhhh*)** is an optional parameter used to specify that part of a volume or minidisk, under CMS, is to be inspected. Use the HEADRANGE parameter to specify the starting and ending head(s) to be processed.

For *hhhh,hhhh*, substitute one to four decimal (n) or hexadecimal (X'n') digits to identify the starting and ending heads to be examined.

If you specify HEADRANGE and do not specify CYLRANGE, the specified heads for all the cylinders on the volume (or minidisk under CMS) are processed.

For information on specifying the starting and ending cylinder(s) (CYLRANGE) to go with the starting and ending heads, see "CYLRANGE: Specify a Range of Cylinders" on page 142.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Discrete Tracks" on page 133.

Abbreviations:

HEADRANGE|HDRANGE|HD

Default: None

Restrictions: Both the starting and ending values are required for HEADRANGE. By specifying an ending value greater than the maximum for the device, processing is performed to the maximum value for that device.

HEADRANGE cannot be specified with FROMRANGE/TORANGE or TRACKS.

MAP|NOMAP: Printing a Volume Map

Explanation: **MAP**|**NOMAP** are optional parameters specifying whether to print a volume (or minidisk under CMS) map during inspection. The map lists tracks that are found to be defective during inspection. Regardless of the tracks being inspected, MAP will list the defective tracks that are currently assigned alternates.

MAP
Indicates that you want to print a map during inspection.

NOMAP
Indicates that you do not want to print a map during inspection.

Abbreviations: None.

Default: The default is MAP.

Note: At the completion of INSPECT processing, the alternate tracks are examined for availability and primary track assignment, regardless of which tracks have been inspected. This information is printed in the map, but no cross check is done to determine if the primary tracks accurately reflect the alternate assignments.

For a complete check of ALL primary and alternate tracks,

```
INSPECT ALLTRACKS NOCHECK NOASSIGN MAP
```

can be run.

MIMIC: Specifying a Special Volume Usage Format

Explanation: **MIMIC(***type***)** is an optional parameter specifying a special usage format for the volume being inspected. You may specify MINI(n), MSS, EMU(n), or EMUALL.

MINI(n)
Indicates that the volume to be inspected is a minidisk of 'n' cylinders. The DEVICETYPE parameter is required when this parameter is specified.

Note that this parameter is valid in the stand-alone environment only. See Appendix B, "VM Support" on page 227, for details of this support.

MSS
Indicates that the volume has been formatted as a Mass Storage System staging pack. This parameter is required when inspecting an MSS staging pack.

Note: Before Device Support Facilities can be run against a staging device, and before the drive can be accessed, the device must be in offline real status. See *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS* for details of the VARY OFFLINE, TEST command that is used to set up the device for nonstaging access.

EMU(n)
Indicates that a partial CKD disk emulated on a fixed block device is to be inspected. The DEVICETYPE parameter is required when this parameter is specified in the stand-alone environment.

This subparameter cannot be specified on an emulated 3330.

EMUALL
Indicates that a full CKD disk is emulated on a fixed block device.

This subparameter cannot be specified on an emulated 3330.

Abbreviations: None.

Default: None.

Restrictions: For the INSPECT command, the only device types valid for MINI(n) are: 2311, 2314, and 2319.

Use of the EMU(n)|EMUALL subparameter is not recommended with the INSPECT command as this causes the device to operate inefficiently. Data checks should be handled using the INSPECT command against the base FBA device (refer to "Emulating a CKD Device on an FBA Device" on page 43).

MIMIC is not valid when processing minidisks under CMS.

Passwords: Providing Passwords for Data Set Security

Explanation: PASSWORDS((*dsname/password*),•••) is an optional parameter specifying passwords for non-VSAM password-protected data sets. The supplied passwords will be used to determine if the user has authority to alter the data sets.

For *dsname*, substitute the fully qualified name of a password-protected data set.

For *password*, substitute the password you want to apply to that data set.

Abbreviations:

PASSWORD | PASSWD | PWD | PD

Default: None.

Restrictions: Up to a maximum of 512 passwords may be specified.

PASSWORD is ignored in the CMS, VSE, and stand-alone environments.

PRESERVE|KEEPIT|NOPRESERVE|HOLDIT: Preventing Destruction of Data on the Tracks Inspected

Explanation: **PRESERVE|KEEPIT|NOPRESERVE|HOLDIT** are optional parameters specifying whether to read and save the data on the inspected tracks. Data read from the specified tracks is held in storage.

Data on a track can also be saved at a backup location as well as held in storage (the storage copy is used to rewrite the data). If processing ends abnormally, the next use of INSPECT detects the data at the backup location and restores the data to the track where processing ended abnormally.

When running online, certain security and integrity precautions are taken to prevent inadvertent destruction of data. When running offline (MVS) or in stand-alone, data destruction is not prevented. See "Volume and Data Set Security" on page 31, and "Data Integrity When Using the INSPECT Command" on page 132 for more information concerning security.

NOPRESERVE *always* destroys the current contents of the track if CHECK is specified, or if unconditional assignment of alternate tracks is taking place (ASSIGN NOCHECK).

When you specify CHECK with either PRESERVE, HOLDIT, or KEEPIT, the data is still at varying degrees of risk.

If a track is part of a data set, INSPECT will enqueue on the data set while processing that track.

PRESERVE or KEEPIT

Indicates that you want to save the data on the inspected tracks. It also indicates that a copy of each track is to be saved at a backup location as well as in storage (the storage copy is used to rewrite the data). If processing ends abnormally, the next use of INSPECT detects the data at the backup location and automatically restores the data to the track on which processing ended abnormally.

PRESERVE and KEEPIT are synonymous.

The recovery backup feature of PRESERVE or KEEPIT is not supported for the IBM 2305, 2311, 2314, and 2319 or for a 3340 emulated on a 3370 FBA device. In this case, a warning message is issued and the job defaults to HOLDIT.

The backup location and the storage copy of the track are erased at the completion of a command.

A performance degradation may be noticed caused by the writing of the data at the backup location. The amount of degradation depends on the total number of tracks being processed for the current INSPECT.

When the data is written at the backup location, it is read-back-checked to ensure it was written correctly. However, when the backup location is used to restore the data at the next use of the INSPECT command, there is no guarantee that the data can be recovered.

For devices that use the CE cylinder as the backup cylinder, if data exists at the backup location, the ANALYZE command, which writes on the CE cylinder, does not destroy the data. ANALYZE does indicate the cylinder and head of the track for which data exists. **Note, however, that the CE microdi-**

agnostics may destroy the data at the backup location. Therefore, if INSPECT processing does not run to completion, and it is determined to be a problem that will cause the volume to be turned over to an IBM Customer Engineer, all attempts should be made to restore the data first, if necessary.

If permanent, uncorrectable data checks are occurring on a track, INSPECT cannot read the data from the track. Therefore, if PRESERVE or KEEPIT is specified, the command will cease processing on that track rather than destroy the data.

If an error occurs when writing the user data back onto the track (or an assigned alternate), INSPECT assigns an alternate to the primary track and writes the data on the alternate. No surface checking takes place at this time, but, if the write to the alternate fails, a new alternate is assigned (up to three times).

NOPRESERVE

Indicates that you do not want to save the data on the inspected tracks. NOPRESERVE *always* destroys the current contents of the track if CHECK is specified, or if unconditional assignment of alternate tracks is taking place (ASSIGN NOCHECK).

The NOPRESERVE option should be used only when the data is known to be nonexistent or unnecessary, or if data errors on a particular track preclude the preservation of the data.

When you specify NOPRESERVE, you cannot inspect tracks inside a VTOC or a VTOCIX data set in an online (MVS) mode. Also, cylinder 0 track 0 cannot be inspected with NOPRESERVE in an online mode.

HOLDIT

Indicates that you want to save the data on the inspected tracks.

Users of the HOLDIT option should make note of the following:

- **HOLDIT** reads the data and keeps it in storage. If processing for the track does not run to completion, the data is lost and record 0 for the track is in a nonstandard format. An INSPECT with NOPRESERVE can be run against the track to fix the record 0 condition. You can then restore the data from a user backup copy.
- You are cautioned against canceling a job that contains an INSPECT HOLDIT. If the job is inspecting multiple tracks, it is indeterminate which, if any, of the tracks have experienced the situation described above. In that case,

  ```
  INSPECT ALLTRACKS NOCHECK NOASSIGN
  ```

  can be run to determine the readability of the home address/record 0's on the volume. Expect message ICK31064I for any track that ICKDSF was processing when the previous job aborted. Recovery can then be done for that track, as described above.
- If permanent, uncorrectable data checks are occurring on a track, INSPECT cannot read the data from the track. Therefore, if HOLDIT is specified, the command will cease processing on that track rather than destroy the data.
- If an error occurs when writing the user data back onto the track (or an assigned alternate), INSPECT assigns an alternate to the primary track and writes the data on the alternate. No surface checking takes place

at this time, but, if the write to the alternate fails, a new alternate is assigned (up to three times).

Abbreviations:

PRESERVE|PRSV
KEEPIT|KEEP
NOPRESERVE|NOPRSV|NPRSV
HOLDIT|HOLD

Default: The default is PRESERVE.

The default for CMS minidisk processing is HOLDIT (even if you specified PRESERVE or KEEPIT).

Restrictions: None.

RECLAIM|NORECLAIM: Reclaiming Tracks Previously Flagged Defective

Explanation: **RECLAIM|NORECLAIM** are optional parameters specifying whether to reclaim tracks that were flagged defective in input but appear to be usable after surface checking.

Reclamation occurs only if surface checking is performed without errors for that track.

If a track is currently flagged defective, the maximum available surface checking procedure is used. If the checking process indicates that the track has no errors, the track is reclaimed.

RECLAIM
Indicates that you want tracks that are identified as usable by surface checking to be reclaimed during inspection.

NORECLAIM
Indicates that you want to suppress track reclamation during inspection.

Abbreviations:

RECLAIM|RCLM
NORECLAIM|NORCLM|NRCLM

Default: The default is NORECLAIM.

Restrictions: The RECLAIM parameter applies only when the CHECK parameter has been specified. If RECLAIM is specified with NOCHECK, it is ignored.

You cannot reclaim tracks on 2305 Fixed Head Storage Models 1 and 2.

SKIP|NOSKIP: Specify the Level of Surface Checking

Explanation: **SKIP|NOSKIP** are optional parameters used to specify the level of surface checking to be performed when CHECK is specified.
Appendix G, "Surface Checking" on page 315, describes surface checking.

SKIP
Specifies that skip displacement checking is to be performed on all tracks within the specified range.

IF SKIP is specified, the total number of tracks, specified by the TRACKS parameter or specified by the range parameters, cannot exceed the number of tracks in one cylinder for the specified device type. Because SKIP does extensive surface checking for all tracks within the range, the run time can be excessive, even with the one cylinder limit. (Extensive surface checking can take up to 5 minutes per track or more, depending upon the device type and system conditions).

When skip displacement processing is being performed for an INSPECT, for every track that is part of a non-VSAM data set, the data set is enqueued upon while processing the track.

When SKIP is specified, the value of *n* in CHECK(n) is ignored.

If the track is part of a data set, INSPECT will enqueue on the data set while processing that track.

For tracks in VSAM data sets, TOLERATE(ENQFAIL) must be specified when processing online.

NOSKIP

Specifies that no skip displacement checking is to be performed. Primary checking only is to be performed on all tracks within the specified range.

If a data check is detected on a track during primary checking, skip displacement checking is performed for that track.

The total number of tracks that can be inspected when NOSKIP is specified is 2500.

Default: If your job specifies TRACKS, the default is SKIP; if your job specifies a range of tracks, the default is NOSKIP.

Restrictions: The SKIP/NOSKIP parameter is ignored if NOCHECK is specified.

TOLERATE: Specify Exclusive Access of a Data Set

Explanation: **TOLERATE(ENQFAIL)** is an optional parameter used to specify that processing is to continue even if exclusive access cannot be obtained for a data set that contains the track being processed.

The TOLERATE(ENQFAIL) parameter is used to specify that in the event exclusive access cannot be obtained for that data set, that processing is to continue for the track. If TOLERATE(ENQFAIL) is not specified, processing continues on the *next* track.

For tracks in VSAM data sets, TOLERATE(ENQFAIL) must be specified when processing online.

Abbreviations:

```
TOLERATE(ENQFAIL|ENQF)
TOL(ENQFAIL|ENQF)
```

Default: TOLERATE(ENQFAIL) off

Restrictions: TOLERATE(ENQFAIL) is ignored if NOSKIP is specified (MVS systems only). ICKDSF assumes TOLERATE(ENQFAIL) is in effect if NOSKIP is specified and a data check requires that an enqueue be performed.

TOLERATE is ignored when processing under CMS.

TORANGE: Specify a Specific Ending Location

Explanation: **TORANGE(*cccc,hhhh*)** is an optional parameter used to specify that part of a volume (or minidisk under CMS) is to be inspected. Use this parameter to specify the ending track (cylinder and head) to be processed.

For *cccc,hhhh*, substitute the 1 to 4 decimal (n) or hexadecimal (X'n') digits to identify the cylinder and head where processing is to end.

If you specify TORANGE and do not specify FROMRANGE, FROMRANGE is defaulted to cylinder 0 track 0.

For information on specifying the starting track of part of a volume (FROMRANGE) see "FROMRANGE: Specify a Specific Starting Location" on page 143.

For information on specifying part of a volume, see "RANGE Parameters: Specifying Discrete Tracks" on page 133.

Abbreviations:

```
TORANGE|TO
```

Default: None

Restrictions: TORANGE cannot be specified with CYLRANGE/HEADRANGE or TRACKS.

TRACKS: Specifying Discrete Tracks

Explanation: **TRACKS(*cylinder,head*)** is an optional parameter used to specify discrete tracks to be inspected.

For *cylinder,head*, specify one to four decimal (n) or hexadecimal (X'n') digits to identify the cylinder, and one to four decimal (n) or hexadecimal (X'n') digits to identify the track to be inspected. To inspect more than one track at a time, specify:

```
TRACKS((cylinder,head) (cylinder,head))
```

Abbreviations:

```
TRACKS|TRACK|TRKS|TRK
```

Default: None.

Restrictions: The maximum number of individual tracks that can be specified with the TRACKS parameter is 20. However, unless NOSKIP is specified, the maximum number of tracks is the number of tracks in a cylinder.

TRACKS is mutually exclusive with the range parameters.

USERID: Specify another User's Minidisk

Explanation: **USERID(*user's ID*)** is an optional parameter used to specify that the INSPECT function is to be performed on another user's minidisk. This parameter is *only* valid when you are running ICKDSF under CMS and you have DEVMAINT authority.

For *user's ID*, substitute the 1 to 8 characters of the user ID of the owner of the minidisk you want to inspect

Abbreviations:

USERID|UID

Default: If USERID is not specified your own minidisk is verified.

Restrictions: USERID can only be specified if you have DEVMAINT authority running ICKDSF under CMS. USERID is ignored in all other environments.

USERID is mutually exclusive with REALADDR

Examples of the INSPECT Command

The following examples show different ways to code the INSPECT command. Most illustrate the use of the command in offline and online mode under MVS. To inspect a volume in a stand-alone environment, make the following changes to MVS examples:

- Eliminate the MVS JCL.
- Substitute the UNITADDRESS parameter for the DDNAME parameter.
- Specify the DEVICETYPE parameter if the MIMIC(EMU) or MIMIC(MINI) parameter is also specified.
- Substitute SYSIN for *dname* in the IPLDD parameter.

To inspect a volume in the VSE environment, do the following:

- Replace the MVS JCL statements with VSE system control statements.
- Substitute the SYSNAME parameter for the DDNAME parameter.
- Substitute *dlblname* for *dname* if the IPLDD parameter is specified and the data set is located on a direct access volume. If the data set is included in the job stream, replace the *dname* with either SYSIN or SYSIPT.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Inspecting a Volume for Conditional Assignment of Alternate Tracks in Online Mode (MVS)

In this example, the assignment of alternate tracks depends on the results of surface checking. The data on each specified track is copied to an alternate track if the track is defective, or is written back to the primary track if the track is not defective. Volume PAY345 was previously mounted as PRIVATE.

```
//EXAMPLE  JOB
//        EXEC  PGM=ICKDSF
//VOLUMEG  DD    UNIT=3330,DISP=OLD,VOL=SER=PAY345
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 INSPECT DDNAME(VOLUMEG) NOVERIFY CHECK(3) -
        TRACKS((1,3),(1,2),(2,0),(2,3),(3,2))
/*
```

Inspecting a Volume for Unconditional Assignment of Alternate Tracks in Offline Mode (MVS)

In this example, the next available alternate track is assigned for the specified primary track. If the primary track is already assigned an alternate track, it will be reassigned a different alternate track. The default parameter PRESERVE causes the contents of the specified primary track to be copied to the assigned alternate.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 INSPECT UNITADDRESS(273) NOVERIFY TRACKS(X'2',3) NOCHECK
/*
```

Inspecting an Entire Volume to Produce a Volume Map in a Stand-alone Environment

In this example, all the tracks on the volume are inspected to produce a volume map that lists all the defective tracks. No alternate tracks are assigned.

```
INSPECT UNITADDRESS(353) NOVERIFY -
        ALLTRACKS NOASSIGN NOCHECK
```

Inspecting a Volume to Reclaim Tracks in Offline Mode (MVS)

In this example, a volume is inspected to reclaim tracks that were previously marked defective. The default parameter PRESERVE causes the contents of the alternate track to be copied to the primary track if the primary track is reclaimed. The alternate track then becomes available for future assignment.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 INSPECT UNITADDRESS(354) NOVERIFY TRACKS((2,3) (3,1) (1,3)) -
        RECLAIM
/*
```

Inspecting a Minidisk in a Stand-alone Environment

In this example, a mini 2314 that has already been initialized is inspected. It provides an unconditional assignment of an alternate track for tracks 5,1 and 5,2. Note that the parameter MIMIC(MINI(20)) is specifying to Device Support Facilities that a minidisk is being inspected. If this parameter was not specified, I/O errors would occur.

Although alternate track assignment is available on these minidisks (because VM does not restrict as many I/O operations as on other devices), I/O errors on the device should be reported to the system administrator so that a permanent alternate track can be assigned to the "real" defective track.

```
INSPECT  UNITADDRESS(150) NOVERIFY TRACKS((5,1),(5,2)) -
     DEVICETYPE(2314) MIMIC(MINI(20)) NOCHECK
```

Inspecting Another User's Minidisk in a CMS Environment

In this example, a minidisk that is owned by another user is inspected. You specify the USERID parameter to inspect the minidisk of a user named BROWN. You must have DEVMAINT authority for this example.

```
INSPECT  UNITADDRESS(391) NOVERIFY USERID(BROWN) TRACKS(3,1)
```

Inspecting a Minidisk under CMS Specifying a Real Address

In this example, part of a volume with a real address of 290 is inspected. Cylinder 3 tracks 0-9 are inspected. If any of the inspected tracks are defective, an alternate is assigned. SKIP specifies that skip displacement checking is to be performed on the specified tracks. You must have DEVMAINT authority for this example.

```
INSPECT  REALADDR(290) NOVERIFY
          FROMRANGE(3,0) TORANGE(3,9) SKIP
```

Inspecting a Volume to Check Tracks for Defects (VSE)

In this example, a volume previously initialized under VSE is inspected. If any of the inspected tracks are defective, an alternate will be assigned. The SKIP parameter specifies that skip displacement surface checking is to be performed on the tracks specified.

```
// JOB      jobname
// ASSGN    SYS010,151
// EXEC     ICKDSF,SIZE=AUTO
   INSPECT SYSNAME(SYS010) NVFY TRKS((2,3) (3,1) (1,3)) -
           SKIP
/*
/&
```

INSTALL Command—CKD

The INSTALL command performs the procedures neccessary for installation, head-disk assembly (HDA) replacement, and physical movement of IBM 3380 DASD. It can be used whenever the validation functions of medial initialization are desirable. The INSTALL command is valid on IBM 3380 volumes only.

The INSTALL command is an enhanced installation procedure which includes the writing of home addresses and record zero on every track of the volume.

This command is provided for use at installation, HDA replacement, and physical movement and can be used in place of medial initialization.

The volume is left in an uninitialized state, and consequently is unusable in an MVS or VSE environment. Alternate tracks are reset and reassigned if neccessary. This command should be followed by a minimal initialization to write a volume label and a VTOC on the volume.

Resuming after a job or system failure

This command should be allowed to run to completion. If processing is interrupted the command should be restarted, otherwise the device may be left in an unusable state. Subsequent invocation of ICKDSF commands on behalf of the device will detect this condition and either suggest or force you to execute an INSTALL command to return the device to its proper condition.

Processing in a Shared Environment

When running the INSTALL command on a volume shared between systems, follow these guidelines:

- Vary the device offline to the other systems.
- Run the INSTALL command.
- Run the INIT command to write the volume label and a VTOC on the volume.
- Vary the device back online to the other systems.

The other systems will then recognize the volume serial number and VTOC location of the volume. Should you choose **not** to follow these guidelines, other systems accessing the volume may experience unpredictable results.

Using the INSTALL Command under CMS

When you are running ICKDSF under CMS, the INSTALL command is valid only with ATTACHed devices (that is, no minidisk support).

Syntax

Command	Required Parameters
INSTALL	SYSNAME(*sysxxx*)|UNITADDRESS(*ccuu*)
	Optional Parameters
	None

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **SYSNAME(*sysxxx*)|UNITADDRESS(*ccuu*)** are required parameters identifying the volume.

SYSNAME

Required for a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN control statement.

UNITADDRESS

Required for processing a volume in OS/VS or in a CMS or stand-alone environment.

For *ccuu*, specify the address, in hexadecimal (3 or 4 digits), of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Abbreviations:

UNITADDRESS|UNIT

Default: None.

Restrictions: The SYSNAME parameter cannot be specified for a volume in the CMS, stand-alone, or OS/VS environment.

The UNITADDRESS parameter cannot be specified for a volume that is running under VSE. The INSTALL command must be run offline.

Optional Parameters

None

Examples of the INSTALL Command

The following example shows how to code the INSTALL command. The example is for offline modes of OS/VS.

To perform INSTALL for a volume in a stand-alone environment, do the following:

- Eliminate the OS/VS JCL.

To perform INSTALL command on a volume in the VSE environment, do the following:

- Replace the OS/VS JCL statements with VSE system control statements.
- Substitute the SYSNAME parameter for the UNITADDRESS parameter.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Using the INSTALL Command

In this example, a volume is to be processed by the INSTALL command. **Note that the volume is not initialized for an MVS or VSE environment after INSTALL processing. It must be followed by initialization at the minimal level specifying the volume identification and NOVERIFY**. A minimal initialization is shown.

```
//EXAMPLE   JOB
//          EXEC  PGM=ICKDSF
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
 INSTALL UNITADDRESS(0353)
   IF LASTCC<8 -
     THEN DO
        INIT UNIT(cuu) NOVERIFY VOLID(ABCD4)
/*
```

REFORMAT Command—CKD

The REFORMAT command updates portions of a previously initialized volume. After a volume has been initialized, it can be reformatted to change the volume serial number, owner identification, and IPL bootstrap and program records. Only volumes that have been initialized can be reformatted.

Using the REFORMAT Command under CMS

When you run ICKDSF under CMS, you can run in two modes:

- Minidisk
- ATTACHed

Minidisk Support under CMS

The minidisk support for the REFORMAT command allows you to change the volume serial number, owner identification, and IPL bootstrap and program records of a minidisk.

The following REFORMAT parameters are supported for minidisks:

- BOOTSTRAP, IPLDD, OWNERID, PURGE, VERIFY, VOLID, and UNITADDRESS

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed to your user ID. When you use ICKDSF in this mode you can perform any ICKDSF function which can be performed when you are running ICKDSF stand-alone.

Syntax

The abbreviation for the REFORMAT command is RFMT.

Command	Required Parameters
REFORMAT	**DDNAME(***dname***)|SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** **VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	BOOTSTRAP|NOBOOTSTRAP **IPLDD({***dname|dlblname***}[,OBJFORMAT|,ABSFORMAT])** **OWNERID(***owner***)** **PURGE|NOPURGE** **VOLID(***serial***)**

Required Parameters

DDNAME|SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume Is Mounted

Explanation: **DDNAME(***dname***)|SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** are required parameters identifying the volume or minidisk that is to be reformatted.

DDNAME

Required when reformatting a volume that is mounted online. The device on which the volume is mounted must be online and the volume mounted as PRIVATE.

For *dname*, specify the MVS JCL DD statement that identifies the device on which the volume is mounted.

SYSNAME

Required when reformatting a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN system control statement.

UNITADDRESS

Required when reformatting a volume that is mounted offline, when reformatting a volume in a stand-alone environment, or when reformatting a minidisk under CMS.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be mounted on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual address of a minidisk.

Abbreviations:

DDNAME|DNAME
UNITADDRESS|UNIT

Default: None.

Restrictions: The UNITADDRESS parameter cannot be specified when reformatting a volume that is mounted online or in the VSE environment.

The DDNAME parameter cannot be specified when reformatting a volume that is mounted offline, or in the CMS, VSE, or stand-alone environments.

The SYSNAME parameter cannot be specified when reformatting a volume in the MVS, CMS, or stand-alone environments.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(***serial[,owner]***)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before reformatting the volume or minidisk.

VERIFY
Required when you want to verify the volume serial number and owner identification before reformatting the volume or minidisk. If the volume serial number or owner identification does not match that found on the volume, the REFORMAT command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY
Required when you want to bypass verification of the volume serial number and owner identification.

Abbreviations:

VERIFY|VFY
NOVERIFY|NOVFY|NVFY

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number; both the volume serial number and the owner identification must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, execution of the command terminates.

Optional Parameters

BOOTSTRAP|NOBOOTSTRAP: Writing IPL Bootstrap Records on the Volume

Explanation: **BOOTSTRAP|NOBOOTSTRAP** are optional parameters specifying whether you are supplying the IPL bootstrap records that are to be written on the volume or minidisk during reformatting.

BOOTSTRAP
Indicates that you are supplying the IPL bootstrap records that are written on the volume or minidisk during reformatting. These records must be the first four records in the IPL program text. The first text record must have IPL1 in the first four columns, followed by 24 bytes of data. The 24 bytes of data have the same format as the first IPL record described above under ABSFORMAT. The next three text records must have IPL2 in their first four columns, followed by 68 bytes of data each in the second and third records and 8 bytes of data in the last record. The 144 bytes of data have the same format as IPL record 2 described above under ABSFORMAT.

NOBOOTSTRAP
Indicates that you are not supplying IPL bootstrap records and require that the necessary IPL bootstrap records be generated by the Device Support Facilities program.

Abbreviations:

```
BOOTSTRAP|BOOT
NOBOOTSTRAP|NOBOOT|NBOOT
```

Default: The default is NOBOOTSTRAP.

Restrictions: The BOOTSTRAP and NOBOOTSTRAP parameters apply only when the IPLDD parameter is specified. This parameter is ignored when the IPL program is supplied in absolute format.

IPLDD: Writing a User-Supplied IPL Program on the Volume

Explanation: **IPLDD(**{*dname*|*dlblname*}**[,OBJFORMAT|,ABSFORMAT])** is an optional parameter that allows you to supply an IPL program to replace an existing IPL program or add one to the volume or minidisk.

This parameter should not be used when reformatting a Mass Storage System staging pack.

The IPLDD and BOOTSTRAP parameters are used to specify one of the following:

- If you do not specify IPLDD, ICKDSF writes special bootstrap records that cause the processing unit to be placed in a WAIT state if the volume is specified during an attempt to load the system.
- If you specify IPLDD but do not specify BOOTSTRAP, ICKDSF supplies an IPL bootstrap record that is written on the volume or minidisk together with the IPL text you supply.
- If you specify IPLDD and BOOTSTRAP, ICKDSF uses the IPL bootstrap and the IPL text you supply. If necessary, ICKDSF updates the bootstrap records to allow for possible user labels.

The maximum size permitted for the IPL program record depends upon the type of volume being reformatted. The maximum sizes allowed are:

Volume Type	Maximum Bytes for IPL Program
2305-1	12180
2305-2	13616
2311	3110
2314	6514
2319	6514
3330-1	12117
3330-11	12117
3340	7286
3344	7286
3350	17902
3375	33984
3380	44948

Also, the number of existing user volume labels can further limit the size allowed for the IPL program if you specify the NOPURGE parameter. If you supply an IPL program that exceeds the size allowed, you receive an error message but the reformatting continues.

For *dname*, substitute 1 to 8 alphameric characters for the DD statement identifying the data set that contains the IPL program you want written on the volume or minidisk being reformatted. The IPL program can be specified within the MVS JCL input stream (SYSIN). If the IPL program is in the MVS JCL input stream, the data must be included immediately after the REFORMAT command. For reformatting in a CMS or stand-alone environment, the *dname* must be specified as SYSIN.

For *dlblname*, substitute 1 to 7 alphameric characters. These represent the file name that appears in the DLBL statement and identify the file that contains the IPL program you want written on the volume or minidisk being reformatted. The IPL program can be included in the VSE JCS input stream (SYSIN). It must then be included immediately after the REFORMAT command.

OBJFORMAT

Specifies that IPL data is being supplied in object deck format, that is, cards will have one of the following strings of EBCDIC characters in columns 2 through 4:

```
TXT
RLD
ESD
END
```

Note that only cards with TXT will be processed. All others will be ignored.

OBJFORMAT may be abbreviated as OBJECT or OBJ.

OBJFORMAT is the default if IPLDD is specified.

ABSFORMAT

Specifies that IPL data is being supplied in the form of variable-length records that contain executable instructions. When IPL data is supplied in this format, it is necessary to provide a minimum of three records. The maximum number of records supplied and their lengths are only limited by the track capacity of the volume on which these records are to be written.

The first two records supplied must be the bootstrap records and are restricted to lengths of 24 and 144 bytes, respectively.

The contents of the IPL records and the contents of the program are not checked by ICKDSF. It is the user's responsibility to ensure that the IPL records can load an executable program. The first IPL record must contain a PSW followed by two CCWs (channel command words). The CCWs should have the following hexadecimal formats:

```
First CCW     06xxxxxx 60000090
Second CCW    08xxxxxx 00000000
```

The first CCW is a command to read in the second IPL record at main storage address xxxxxx. The second CCW is a transfer-in-channel command (a branch) to the CCW that begins the second IPL record.

The second IPL record must be 144 bytes long. Bytes 32 through 42 (starting from byte zero) cannot be used. Bytes 32 through 42 are used by ICKDSF for the seek address (bytes 32 through 37) and the CCHHR (bytes 38 through 42) of the third IPL record.

The 3rd through nth records that are supplied are assumed to be IPL program records and will be written on the volume without any modification after the standard volume label and any existing user volume labels. Note that the IPL program will only be written on the first track of the pack. If the records supplied exceed in length the remaining space on the first track, the function will terminate with an error message.

ABSFORMAT may be abbreviated as ABSOLUTE or ABS.

Abbreviations:

```
IPLDD|IPL
```

Default: The system provides special IPL bootstrap records if you specify the IPLDD parameter without specifying the BOOTSTRAP parameter.

Restrictions: The IPLDD parameter is ignored when the MIMIC(MSS) parameter is specified.

When an IPL program is included in the SYSIN stream, it must immediately follow the REFORMAT command and be terminated by an ENDIPLTEXT card. The ENDIPLTEXT card is optional when the IPL program is in a data set other than the one specified by SYSIN, or when the end-of-file indicator (/*) immediately follows the data for the IPL program.

OWNERID: Specifying a New Volume-Owner Identification

Explanation: **OWNERID(*owner*)** is an optional parameter specifying a new volume-owner identification to be written in the volume label.

For *owner*, substitute 1 to 14 alphameric characters for a new volume-owner identification in the volume label.

Abbreviations:

OWNERID|OWNER

Default: When a volume or minidisk is reformatted and a new owner identification is not specified, the old owner identification remains unchanged.

PURGE|NOPURGE: Writing Over User Volume Labels on a Previously Initialized Volume

Explanation: **PURGE|NOPURGE** are optional parameters specifying whether existing user volume labels are to be written over by the specified IPL program.

PURGE
Indicates that you want to write over existing user volume labels during reformatting.

NOPURGE
Indicates that you do not want to write over any user volume labels. If a volume or minidisk contains user volume labels and the IPLDD parameter is specified, the IPL program record is written following the last user volume label.

Abbreviations:

PURGE|PRG
NOPURGE|NOPRG|NPRG

Default: The default is NOPURGE.

Restrictions: The PURGE parameter applies only when the IPLDD parameter is specified.

VOLID: Specifying a New Volume Serial Number

Explanation: **VOLID(*serial*)** is an optional parameter specifying a new volume serial number to be written in the volume label.

For *serial*, substitute one to six alphameric characters for the new volume serial number. If fewer than six characters are specified, the serial number is left-justified and the remainder of the field is padded with blanks (X'40').

Abbreviations: None.

Default: When you reformat a volume or minidisk and do not specify the VOLID parameter, the old volume serial number remains unchanged.

Restrictions: Any catalog that identifies the volume by its volume serial number must be changed to reflect the new volume serial number for any cataloged data sets that reside on the reformatted volume

No check is made to determine if the volume contains an indexed VTOC. If an indexed VTOC exists on the volume, the data set name of the index remains unmodified. (That is, the index may be named SYS1.IXVTOC.VOL001 but the new volume label is VOL002.) This does not cause any functional problems, but may have an impact on the uniqueness of indexed VTOC data set names.

Examples of the REFORMAT Command

The following examples show different ways to code the REFORMAT command in offline and online mode (MVS) and for VSE. To reformat a volume in a CMS or stand-alone environment, take an example given for MVS and do the following:

- Eliminate the MVS JCL.
- Substitute the UNITADDRESS parameter for the DDNAME parameter.
- Substitute SYSIN for *dname* in the IPLDD parameter.

Any values specified here are examples only and should not be interpreted as the values for your system.

Reformatting a Volume to Change the Volume Serial Number in Offline Mode (MVS)

In this example, the volume serial number is being changed. Any catalogs that identify the volume by its old volume serial number must be modified. The VERIFY parameter is specified to ensure that the correct volume is being accessed before the volume serial number is changed.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 REFORMAT UNITADDRESS(353) VERIFY(OLDVOL) -
          VOLID(NEWVOL)
/*
```

Reformatting a Volume to Add an IPL Program in Online Mode (MVS)

In this example, an IPL program is added to the volume, and the owner identification is changed. Volume VOL123 was previously mounted as PRIVATE. If any user volume labels exist on the volume, the IPL program is written over them because the PURGE parameter is specified. Device Support Facilities will supply the appropriate IPL bootstrap records.

If IPL text already exists on the volume, the operator will be prompted as to whether or not to continue processing.

```
//EXAMPLE  JOB
//         EXEC  PGM=ICKDSF
//IPLDECK  DD    *
.
.
.
//VOLUMEL  DD    UNIT=3330,DISP=OLD,VOL=SER=VOL123
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
 REFORMAT DDNAME(VOLUMEL) IPLDD(IPLDECK) -
          OWNERID(BROWN) PURGE NOVERIFY
/*
```

Reformatting a Volume to Change the Volume Serial Number (VSE)

In this example, the volume serial number is verified and a new volume serial number is assigned.

```
// JOB      jobname
// ASSGN    SYS010,151
// EXEC     ICKDSF,SIZE=AUTO
   REFORMAT SYSNAME(SYS010) VERIFY(OLDVOL) VOLID(NEWVOL)
/*
/&
```

Reformatting a minidisk in a CMS Environment

In this example, the volume serial number of a minidisk at virtual address 291 is being changed from OLDVOL to NEWVOL, and the owner identification from PAYROLL to NEWPAY. The VERIFY parameter is specified to ensure that the correct volume and owner is being accessed before the volume serial/owner ID is changed.

```
REFORMAT UNITADDRESS(291) VERIFY(OLDVOL,PAYROLL) -
         VOLID(NEWVOL) OWNERID(NEWPAY)
```

REVALidate Command—CKD

The REVALidate command combines the track validation functions of medial initialization with the problem determination and data verification functions of the ANALYZE command, and also the INSPECT functions, if required. This command is valid on IBM 3380 volumes only. REVALidate, in one command, performs the following combination of functions:

- A drive test
- Home address and record zero validation
- Data verification of the factory functional verification data patterns (FFVDP)
- Surface checking on tracks if required.

ICKDSF Functions Performed by the REVALidate Command

The REVALidate command can be used as an alternative to the following procedure:.

- ANALYZE with the DRIVETEST parameter
- INIT with the VALIDATE DATA parameters
- ANALYZE with the SCAN parameter
- INSPECT with the TRACKS SKIP parameters if required.

At the completion of this command, all tracks are formatted for use by IBM operating systems. Alternate tracks are reset and reassigned if neccessary. **However, the volume label, VTOC, and all user data has been destroyed**. This command should be followed by a minimal initialization to write a volume label and a VTOC on the volume. See "Initializing a Volume at the Minimal Level" on page 101 for more information on minimal initialization.

This command operates on the full volume and destroys all data. It should not be used as an alternative when media maintenance actions are required for the device. See *ICKDSF: Primer for the User of IBM 3380 Direct Access Storage* for more information on media maintenance.

Summary information regarding the results of ANALYZE SCAN and INSPECT processing is presented.

For more information on each of the above functions see the command description in "Part 2. ICKDSF Commands—Count-Key-Data (CKD) Devices" on page 71.

Resuming after a job or system failure

This command should be allowed to run to completion. If processing is interrupted the command should be restarted, otherwise the device may be left in an unusable state. Subsequent invocation of ICKDSF commands on behalf of the device will detect this condition and either suggest or force you to execute a REVALidate command to return the device to its proper condition.

Processing in a Shared Environment

When running the REVALidate command on a volume shared between systems, follow these guidelines:

- Vary the device offline to the other systems.
- Run the REVALidate command.
- Run the INIT command to write a volume label and a VTOC on the volume.
- Vary the device back online to the other systems.

The other systems will then recognize the volume serial number and VTOC location of the volume. Should you choose **not** to follow these guidelines, other systems accessing the volume may experience unpredictable results.

Using the REVALidate Command under CMS

When you are running ICKDSF under CMS, the REVALidate command is valid only with ATTACHed devices (that is, no minidisk support).

Syntax

Command	Required Parameters
REVAL	**SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** **VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	None

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** are required parameters identifying the volume.

SYSNAME
Required for a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN control statement.

UNITADDRESS
Required for processing a volume in OS/VS or in a CMS or stand-alone environment.

For *ccuu*, specify the address, in hexadecimal (3 or 4 digits), of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Abbreviations:

UNITADDRESS|UNIT

Default: None.

Restrictions: The SYSNAME parameter cannot be specified for a volume in the CMS, stand-alone, or OS/VS environment.

The UNITADDRESS parameter cannot be specified for a volume that is running under VSE. The REVALidate command must be run offline.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(***serial[,owner]***)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before processing the volume.

VERIFY
Required when you want to verify the volume serial number and owner identification before processing the volume. If the volume serial number or owner identification does not match that found on the volume, the REVALidate command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY
Required when you want to bypass verification of the volume serial number and owner identification.

Abbreviations:

VERIFY|VFY
NOVERIFY|NOVFY|NVFY

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number; both the volume serial number and the owner identification must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, the command terminates.

Optional Parameters

None

Examples of the REVALidate Command

The following example shows how to code the REVALidate command. The example is for offline modes of OS/VS.

To perform REVALidate for a volume in a stand-alone environment, do the following:

- Eliminate the OS/VS JCL.

To perform REVALidate command on a volume in the VSE environment, do the following:

- Replace the OS/VS JCL statements with VSE system control statements.
- Substitute the SYSNAME parameter for the UNITADDRESS parameter.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Using the REVALidate Command

In this example, a volume is to be processed by the REVALidate command. **Note that the volume is not initialized for an MVS or VSE environment after REVALidate processing. It must be followed by initialization at the minimal level specifying the volume identification and NOVERIFY**. A minimal initialization is shown.

```
//EXAMPLE   JOB
//          EXEC  PGM=ICKDSF
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
 REVAL UNITADDRESS(0353) VERIFY(VOL123,PAYROLL)
   IF LASTCC<8 -
     THEN DO
        INIT UNIT(cuu) NOVERIFY VOLID(ABCD4)
/*
```

Part 3. ICKDSF Commands—Fixed Block Architecture (FBA) Devices

This part of the book is intended as a reference when you specify the Device Support Facilities commands for fixed block architecture (FBA) devices. Valid FBA devices for Device Support Facilities (ICKDSF) are given in "Devices You Can Use" on page 8.

"Examples of ICKDSF Tasks—FBA Devices" on page 55 and this part of the book are a guide to help you determine why, when, and how to run Device Support Facilities.

This part describes the Device Support Facilities commands for FBA devices. The commands are listed in alphabetic order. "Syntax of the Device Support Facilities Commands" on page 67 describes the syntax of the commands. The presentation of each command includes:

- A general description of the command and its use
- A table summarizing the syntax of the command
- An explanation of required and optional parameters
- Examples

Device Support Facilities Functions

The following Device Support Facilities commands operate on fixed block architecture (FBA) devices: ANALYZE, CONTROL, INIT, INSPECT, MAPALT, and REFORMAT.

- ANALYZE helps to detect and differentiate recording surface and drive-related problems on a volume. It can also scan data to aid in detecting the existence of possible media errors.
- CONTROL allows you to reclaim the primary and alternate blocks associated with the System Reserved Area of IBM 9335 devices.
- INIT prepares a direct-access storage volume so that it can be used in a VSE operating system environment or it reinitializes a volume that was previously prepared. It can also be used to surface check a volume.
- INSPECT provides surface checking functions for defective blocks.
- MAPALT produces a report showing the assignment of primary blocks to alternate blocks on fixed block architecture devices.
- REFORMAT can be used to update or replace information on a previously initialized volume. The volume label and owner ID can be replaced.

Device Support Facilities performs two basic functions: volume formatting and disk surface maintenance.

- Volume formatting makes a volume usable by an operating system and provides a means of updating old volume formats. This is also known as "initializing" a volume.

  Newly installed volumes and replaced head disk assemblies (HDAs) will always require formatting.
- Disk surface maintenance aids in detecting problems with device operation and attempts to correct data errors that can be traced specifically to disk surface defects.

ANALYZE Command—FBA

The ANALYZE command is used to examine the drive and/or the user's data on a volume to help determine if errors exist. The output is intended to aid in distinguishing between drive problems and media problems, in addition to providing assistance in locating and fixing the problem.

The IBM 9313 and 9332 are not supported by the ANALYZE command.

There are two basic functions of the ANALYZE command: the drive test, and the data scan.

ANALYZE Processing

Two basic tests are performed by ANALYZE—the drive test and the data verification test. The drive test is a general exercising of the storage device by issuing LOCATE, READ, and WRITE commands. From the results, ANALYZE determines the drive's ability to perform expected actions.

The data verification test reads (without data transfer across the channel to the processor) every data record on the volume. This determines whether the data can be read correctly—that is, without data check errors.

In addition, a data verification option is provided that allows you to specify a portion of the volume for testing. One or more blocks can be specified. You may also perform data verification for the entire volume.

ANALYZE does not require exclusive control of the drive under test. Most of ANALYZE's CCW chains (that is, its channel programs) do not hold the drive busy for longer than 0.5 second. However, if the operating system executes its error recovery procedures as a result of I/O errors detected during ANALYZE processing, the error recovery procedures may hold the drive busy for a longer period of time.

ANALYZE supports fixed block devices in fixed block mode only. This mode is based on data transfers of 512-byte increments (blocks) for storing and retrieving data.

ANALYZE data verification testing allows you to control the duration of each I/O by specifying either SPEED or NOSPEED.

The Drive Test

ANALYZE determines that the drive is fully operational by performing I/O operations on the CE block that test the drive's functional capabilities. Each I/O operation, called a channel command word (CCW) chain, tests a specific function of the drive and logical volume. Tests are executed in order of increasing complexity.

Drive problems that can be detected during ANALYZE's initial tests are access arm positioning errors and the ability to read and write.

If PRESERVE or CONTINUE data exists on the CE cylinder, ANALYZE does not destroy this data. This data is written by an INSPECT or INIT. If this data exists, ANALYZE bypasses writing over the data, and reports the presence of the data.

The drive test is optional. It is controlled by the DRIVETEST|NODRIVETEST parameter.

The drive test is not supported if you are processing under CMS unless the device is ATTACHed.

The Data Verification Test

The ANALYZE data verification test determines the status of the drive's storage medium by reading (without data transfer to the processor) all data records on the volume (minidisk under CMS). If a data check is detected during the test, both the address and relative block number are noted in a diagnostic message on the printer. When a block is found to be defective, you can bypass the defect by assigning an alternate block to it.

ANALYZE allows you options for the data verification tests:

- You may specify data verification of the whole volume or minidisk.
- You may select a range of data to be verified by blocks (LIMITS or BLOCKRANGE parameters).
- You may also select the SPEED or NOSPEED parameter on the data verification tests.

  NOSPEED specifies:

  — For the 3310, 3370, and 9335, the number of blocks read by each I/O is 32, 248, and 71, respectively.

  SPEED specifies:

  — Perform one I/O for the maximum number of blocks (352, 744, and 426, respectively) within the range specified for the 3310, 3370, and 9335.

ANALYZE Output

The ANALYZE command is designed to produce output that can be used to aid in problem determination for a current volume. Certain errors that might occur are correctable by the device (and/or its storage control) and are normally never seen by software. Other errors can be corrected by software (usually under the operating system error recovery procedures (ERPs)) and are never seen by the user. Some errors are unrecoverable under all procedures.

What ANALYZE Tests

Errors that are persistent and that cannot be corrected by the device's internal error-correction circuitry result in messages to the output device.

When you run ANALYZE, the following conditions are assumed to exist:

- Each block has a properly formatted ID field and 512-byte data area.
- The blocks used by ANALYZE in the CE area are defect-free.

ANALYZE performs two testing procedures to detect errors—the drive test and the data verification test.

Using ANALYZE Output

As part of an installation's operating procedure, ANALYZE output can assist in determining whether or not recovery procedures can or should be executed. However, the use of ANALYZE output by customer engineers or service representatives to isolate and to repair drive malfunctions should not replace the use of more detailed diagnostic tools and procedures available for that purpose. ANALYZE output in such instances should serve only as the initial reference material that shows the final problem symptom but does not specifically isolate the problem source.

Using the ANALYZE Command under CMS—FBA

When you run ICKDSF under CMS you can run in two modes:

- Minidisk mode
- ATTACHed mode

Minidisk Support under CMS

You can use the ANALYZE command to perform the data verification test on your own minidisk.

The following parameters are valid under CMS:

- ALL, LIMITS/BLOCKRANGE, NODRIVETEST, SCAN/NOSCAN, SPEED/NOSPEED, and UNITADDRESS

Notes:

1. The drivetest is not supported under CMS.
2. You must have DEVMAINT authority to issue the ANALYZE command for the IBM 9335.

The following parameters are valid *only* when you are under CMS and have DEVMAINT authority (as defined in the CP directory):

- USERID and REALADDR

You can use these parameters as follows:

USERID: When you are under CMS amd have DEVMAINT authority and specify the USERID parameter you can perform the data verification test on another user's minidisk. The following parameters are valid in this mode:

- ALL, LIMITS/BLOCKRANGE, NODRIVETEST, SCAN/NOSCAN, SPEED/NOSPEED, UNITADDRESS, and USERID

REALADDR: When you are under CMS amd have DEVMAINT authority and specify the REALADDR parameter you can perform the data verification test on a volume when you specify the real device address: The following parameters are valid when you specify REALADDR:

- ALL, LIMITS/BLOCKRANGE, NODRIVETEST, SCAN/NOSCAN, SPEED/NOSPEED,

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed to your user ID. When you use ICKDSF in this mode, you can perform any ICKDSF function which can be performed when you are running ICKDSF stand-alone.

Syntax

Command	Required Parameters
ANALYZE	**SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)|REALADDR(***ccuu***)**
	Optional Parameters
	DRIVETEST|NODRIVETEST **LIMITS(***scanlo,scanhi***)|BLOCKRANGE(***bbbbbb,bbbbbb***)|ALL** **SCAN|NOSCAN** **SPEED|NOSPEED** **USERID(***user's ID***)**

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **SYSNAME(***sysxxx***)\|UNITADDRESS(***ccuu***)\|REALADDR(***ccuu***)** are required parameters identifying the volume or minidisk that is to be analyzed.

SYSNAME

Required when analyzing a volume in the VSE environment. For *sysxxx*, specify the SYSNAME in the ASSGN system control statement.

UNITADDRESS

Required when analyzing a volume in a stand-alone environment or a mini-disk.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program may enter a nonterminating wait state.

Under CMS, use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual address of a minidisk.

REALADDR
Required to specify the real address of a volume. This parameter is only valid when you are running ICKDSF under CMS and have DEVMAINT authority (as defined in the CP directory).

For *ccuu*, specify the real address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Abbreviations:

UNITADDRESS|UNITADDR|UNIT

Default: None

Restrictions: The SYSNAME parameter cannot be specified when analyzing a volume in a CMS or stand-alone environment

The UNITADDRESS parameter cannot be specified when you are analyzing a volume that is running under VSE.

REALADDR is mutually exclusive with UNITADDRESS, DDNAME, SYSNAME, and USERID.

Optional Parameters

DRIVETEST|NODRIVETEST

Explanation: **DRIVETEST**|**NODRIVETEST** are optional parameters specifying whether or not the drive test function of the command should be executed.

DRIVETEST
Specifies that the drive test should be executed.

NODRIVETEST
Specifies that the drive test should be bypassed.

Abbreviations:

DRIVE
NODRIVE

Default: DRIVETEST

Restrictions: NODRIVE NOSCAN performs no functions but will complete without error.

When executing in a VM environment, DRIVETEST is valid only for devices that are ATTACHed to the current userid. I/O errors will occur if the drive test is attempted for LINKed devices or minidisks. DRIVETEST is ignored when you are processing minidisks.

LIMITS|BLOCKRANGE|ALL: Specifying the Area of Data Verification

Explanation: **LIMITS(***scanlo,scanhi***)|BLOCKRANGE(***bbbbbb,bbbbbb***)|ALL** are optional parameters specifying the range of blocks to be read during the data verification tests.

LIMITS(*scanlo,scanhi***)**

scanlo specifies the relative block number for the start of the data verification tests. *scanlo* can be expressed in decimal (n) or hexadecimal (X'n') form.

scanhi specifies the relative block number for the last block of the data verification tests. *scanhi* can be expressed in decimal (n) or hexadecimal (X'n') form.

The numbers *scanlo* and *scanhi* must be equal or in ascending order, and they must fall within the device address limits: low to high *relative* block number. Incorrect specification will result in an error message and cause execution of the command to terminate.

BLOCKRANGE(*bbbbbb,bbbbbb***)**

For *bbbbbb,bbbbbb*, substitute 1 to 6 decimal (n) or hexadecimal (X'n') digits for the starting and ending block numbers to be verified.

BLOCKRANGE is identical in function to LIMITS and is included for consistency with other ICKDSF commands.

ALL

specifies that all blocks of a volume (minidisk under CMS) are to be read during the data verification tests.

Abbreviations:

```
  LIMITS|LIMIT|LIMS|LIM
  BLOCKRANGE|BLKRANGE|BLKR
ALL
```

Default: If you do not specify LIMITS or BLOCKRANGE, the default is ALL.

Restrictions: LIMITS, BLOCKRANGE, or ALL apply only when the SCAN parameter has been specified.

LIMITS, BLOCKRANGE, and ALL are mutually exclusive.

SCAN|NOSCAN: Run Data Verification Tests

Explanation: **SCAN|NOSCAN** are optional parameters specifying whether the data verification tests are to be run. Data verification is executed after the drive tests complete successfully or if NODRIVE is specified.

SCAN

Indicates that you want the data verification tests to be run.

NOSCAN

Indicates that you do not want the data verification tests to be run.

Abbreviations:

SCAN|SCN
NOSCAN|NOSCN|NSCAN|NSCN

Default: NOSCAN

Restrictions: None.

SPEED|NOSPEED: Specify Data Verification Tests

Explanation: **SPEED|NOSPEED** are optional parameters that specify how many blocks are read by each I/O for the data verification tests.

SPEED
For the 3310, 3370, and 9335, this parameter specifies that the maximum number of blocks is read for each I/O (352, 744, and 426, respectively).

NOSPEED
For the 3310, 3370, and 9335, NOSPEED specifies the minimum number of blocks for each I/O is 32, 248, and 71, respectively.

Abbreviations:

NOSPEED|NSPEED.

Default: NOSPEED.

Restrictions: You should not specify SPEED when you are running under VM, as this degrades the performance for other VM users.

USERID: Specify Another User's Minidisk

Explanation: **USERID(*user's ID*)** is an optional parameter used to specify that the ANALYZE data verification test is to be performed on another user's minidisk. This parameter is *only* valid when you are using ICKDSF under CMS and have DEVMAINT authority.

For *user's ID*, substitute the 1 to 8 characters of the user ID of the owner of the minidisk you want to verify.

Abbreviations:

USERID|UID

Default: If USERID is not specified your own minidisk is verified.

Restrictions: USERID can only be specified if you have DEVMAINT authority. USERID is ignored in all other system environments.

Examples of the ANALYZE Command

The following examples show different ways to use the ANALYZE command.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Analyzing a Volume in a Stand-Alone Environment

In this example, the 3310 Direct Access Storage drive tests are performed.

```
ANALYZE UNITADDRESS(141)
```

Using keyword abbreviations, you could have specified:

```
ANALYZE UNIT(141)
```

Analyzing a Volume in a Stand-Alone Environment

In this example, the 3370 Direct Access Storage drive tests are performed and are followed by the data verification tests for the entire 3370 volume. ALL is the default.

```
ANALYZE UNITADDRESS(141) SCAN
```

Analyzing a Volume in a Stand-Alone Environment

In this example, the 3310 Direct Access Storage drive tests are performed, followed by partial data verification tests starting at relative block number 25 and ending at relative block number 50. Abbreviations have been used.

```
ANALYZE UNIT(141) LIMS(X'19',X'32') SCN
```

Analyzing a Volume in a VSE Environment

In this example, drive tests are performed on the volume SYS001. Specification of the NOSCAN parameter indicates that data verification tests are not to be performed.

```
// JOB   jobname
// ASSGN SYS001,150
// EXEC  ICKDSF,SIZE=AUTO
   ANALYZE SYSNAME(SYS001) NOSCAN
/*
/&
```

Analyzing Another User's Minidisk under CMS

In this CMS environment example, the data verification test is performed for another user's minidisk. You specify the USERID parameter to ANALYZE the minidisk which is owned by user SMITH at the user's virtual address 0351. You must be have DEVMAINT authority (as defined in the CP directory) to specify the USERID parameter.

```
ANALYZE UNIT(0351) SCAN USERID(SMITH) NODRIVETEST
```

Analyzing a Volume under CMS Specifying a Real Address

In this CMS environment example, the data verification test is performed for a volume at real address 290 by using the REALADDR parameter. You must have DEVMAINT authority (as defined in the CP directory) to specify the REALADDR parameter.

```
ANALYZE REALADDR(290) SCAN NODRIVETEST
```

Diagnostic Messages

ANALYZE prints diagnostic messages to the output device. They can be used by a programmer (and by the customer engineer or service representative) to aid in determining if and what type of problem might exist on the volume.

The informational and diagnostic messages and their explanations are included in Appendix C, "Device Support Facilities Messages (ICK)" on page 233.

In addition, ANALYZE produces a movable head and/or fixed-head error table that summarizes errors associated with the read/write circuitry and/or data verification. See Figure 12.

Movable Head Error Table

In addition to the diagnostic messages, if errors associated with the read/write circuitry occurred during the drive test, ANALYZE produces a movable head error table to summarize the errors. This table is printed on the system printer after all tests are completed. Figure 12 shows the format of the movable head error table. The physical heads are listed in the first column on the left. A character 'X' is placed in the column corresponding to the type of error detected for a particular head.

MOVABLE HEAD ERROR TABLE

HEAD NUMBER	DATA CHK	SEEK VERIFY CHECK	WRITE CHK	DATA CHK CE CYL	DATA COMP ERROR
00					
01					
02					
03					
04					
05					
06					
07					
08					
09	----UC----				
10	----U-----	------X------	-----X-----	----X-----	-----X----
11	-----C----	------X------	-----X-----	----X-----	-----X----

In the data check column:

"U" indicates that at least one ECC uncorrectable error occurred on this head.

"C" indicates that at least one ECC correctable error occurred on this head.

"UC" indicates that at least one ECC correctable and one ECC uncorrectable error occurred on this head.

"X" indicates that at least one error, as indicated by the column heading, has occurred on this head.

The actual number of errors that occurred for a specific head can be obtained by examining previous messages provided in the ANALYZE output.

Figure 12. The Movable Head Error Table—FBA

CONTROL Command—FBA

The FBA CONTROL command allows you to reclaim primary and alternate blocks which are associated with the System Reserved Area of the IBM 9335 devices. This command is provided for VSE and stand-alone environments only or when executing ICKDSF under CMS with ATTACHed devices.

Warning: As the FBA CONTROL command destroys information in the system reserved area of the IBM 9335, you must be careful when you invoke this command.

This function should only be invoked if system messages, EREP reports, or Service Processor messages indicate that a System Reserved Area reclaim is neccessary.

Using the FBA CONTROL Command

The FBA CONTROL command is designed to allow you to reclaim primary and secondary blocks associated with the system control area of the IBM 9335. You use this function by specifying the CONTROL command and the RECLAIM(SYSAREA) parameter. This function reclaims primary and alternate blocks that are not listed in the IBM 9335 Defective Sector Map. Also, the surface of each reclaimed block is checked, and if it is found defective an alternate is reassigned to the primary.

As the CONTROL command destroys information (microcode and information in the System Reserved Area of IBM 9335 devices is deleted) you receive a warning message requesting permission to proceed. If you give permission to proceed, a microcode alert subcommand diagnostic control CCW is issued to inform the processor that the information in the System Reserved Area of the 9335 device will be destroyed.

Using the CONTROL Command under CMS—FBA

When you are executing ICKDSF under CMS, you can ATTACH devices to your userid. This allows you to process the volume under CMS. ATTACHed mode is the only CONTROL command support under CMS (that is, no minidisk support). When you use the CONTROL command under CMS with ATTACHed devices, the ICKDSF support is the same as provided by the stand-alone support.

Syntax

Command	Required Parameters
CONTROL	SYSNAME(*sysxxx*)|UNITADDRESS(*ccuu*) RECLAIM(SYSAREA)
	Optional Parameters
	None

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **SYSNAME(*sysxxx*)|UNITADDRESS(*ccuu*)** are required parameters identifying the volume that is to be analyzed.

SYSNAME

Required when analyzing a volume in the VSE environment. For *sysxxx*, specify the SYSNAME in the ASSGN system control statement.

UNITADDRESS

Required when analyzing a volume in a stand-alone environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program may enter a nonterminating wait state.

Abbreviations:

UNITADDRESS|UNITADDR|UNIT

Default: None

Restrictions: The SYSNAME parameter cannot be specified when analyzing a volume in a CMS (ATTACHed mode) or stand-alone environment. The UNITADDRESS parameter cannot be specified when you are analyzing a volume that is running under VSE.

RECLAIM(SYSAREA)

Explanation: **RECLAIM(SYSAREA)** is a required parameter that must be specified when you want to reclaim primary or alternate blocks associated with the System Reserved Area of an IBM 9335 device.

Note: Primary and alternate blocks listed in the IBM 9335 Defective sector Map are not reclaimed.

Abbreviations: None

Default: None

Restrictions: The CONTROL command RECLAIM(SYSAREA) function is only applicable to IBM 9335 devices.

This function must not be invoked unless system error messages, Service Processor messages, or EREP reports indicate that a System Reserved Area reclaim is neccessary.

Optional Parameters

None

Examples of the FBA CONTROL Command

The following example illustrates a way to code the FBA CONTROL command.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Reclaiming the System Reserved Area of an IBM 9335 Device in the VSE Environment

In this example, the primary and alternate blocks of the System Reserved Area of an IBM 9335 device are reclaimed.

```
// JOB   jobname
// ASSGN SYS001,150
// EXEC  ICKDSF,SIZE=AUTO
   CONTROL UNITADDRESS(141) RECLAIM(SYSAREA)
/*
/&
```

INIT Command—FBA

The INIT command initializes a volume. A direct-access storage volume must be initialized before an operating system can use it. The INIT command can initialize fixed block architecture (FBA) volumes at minimal and maximal levels by specifying parameters that control the level of initialization. These two levels of initialization are described in the sections that follow.

For INIT support of the VM minidisks see:

- Appendix B, "VM Support" on page 227

For the IBM 9313 and 9332, minimal initialization is the only level which can be performed. Also, the BLOCKRANGE, CHECK, CONTINUE, MAP, and RECLAIM parameters are not valid. They are ignored if specified for these devices.

Initializing a Volume at the Minimal Level

Initialization is performed at the minimal level when the NOCHECK option is specified. Minimal initialization formats the volume labels and VTOC. Note that minimal initialization does not delete any previous IPLTEXT that is on the volume.

The FBAVTOC parameter is used to specify the space for the VTOC for an FBA device. This parameter allows you to specify the starting block number, number of label records, and control interval size of the VTOC. You can default the location, or you can have ICKDSF determine the size and location by specifying FBAVTOC(END). See Appendix F, "The Fixed Block Architecture VTOC (FBAVTOC)" on page 311, for more information about the FBA VTOC.

For the IBM 9313 and 9332, minimal initialization is the only level which can be performed.

When you are processing minidisks, minimal initialization is the only level which can be performed.

Initializing a Volume at the Maximal Level

Initialization is performed at the maximal level when the CHECK parameter is specified and includes the minimal initialization functions. At this level, each block is surface checked for recording errors and any blocks that have errors have new alternates assigned. The volume labels and VTOC are then formatted as in minimal initialization.

If you specify RECLAIM with CHECK, an attempt is made to reclaim the primary surface of blocks that have had alternates assigned by prior INIT or INSPECT command executions. Any blocks that were previously marked defective by an INIT or INSPECT execution but pass the surface checking test in this run, are marked available. However, any primary or alternate blocks that were flagged defective by the factory will not be reclaimed by the INIT command.

If a maximal initialization does not run to completion, the CONTINUE function is provided to allow you to recover in a minimum amount of time, and to ensure the volume is usable. See "Resuming After a Job or System Failure," for more information concerning continuing after a failure.

For the IBM 9313, 9332, and minidisks, maximal initialization is not allowed.

Resuming After a Job or System Failure

If a maximal initialization does not run to completion for any reason, the job does not need to be restarted from the beginning of the volume. ICKDSF checkpoints itself automatically during processing and can determine where it is to resume.

After a job or system failure, you can submit a new job:

- At the minimal or maximal level.
- For the entire volume, or for a different specified range, or for the same specified range (maximal INIT).
- To resume from the point of failure, or to resume from the start of the current specified range (the job which was running when the job or system failed).

Because INIT processing always destroys the volume label, a subsequent invocation must include the VOLID parameter.

After a previous failure, if CONTINUE is specified, processing continues from the point of failure *in correlation with* the current specified range as follows:

- If the new range is totally before the previous point of failure, no new blocks are processed.
- If the new range is totally beyond the point of failure, then the entire new range is processed.
- If the new range specification overlaps the point of failure, then processing begins from the point of failure).
- If NOCONTINUE is specified, processing begins from the current specified range.
- If the previous INITialize specified RECLAIM:
  - If processing failed during the actual reclaim procedure, processing starts from the beginning of the reclaim procedure and executes for the entire volume.
  - If processing failed after the actual reclaim procedure was complete, processing begins from the continue location, but is forced to the end of the volume, to ensure that RECLAIM has processed to completion.
  - Note that this is true even for a minimal initialize.

RANGE Parameters: Specifying Part of a Volume

BLOCKRANGE is an optional parameter that gives you the ability to surface check a range of blocks when you specify the INIT command.

Using the INIT Command under CMS—FBA

When you run ICKDSF under CMS, you can run in two modes:

- Minidisk
- ATTACHed

Minidisk Support under CMS

When you run ICKDSF under CMS the following is supported for minidisks:

- Minimal INIT *only* is supported

When you initialize a minidisk at the minimal level, you can specify the minidisk volume label (VOLID) the VTOC, and the owner identification (OWNERID),

The following parameters are supported when you are running under CMS

- UNITADDRESS, FBAVTOC, LABELS, MAP, OWNERID, VERIFY, and VOLID.,

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed. ICKDSF in this mode can then do any functions which can be performed when you are running ICKDSF stand-alone

Syntax

Command	Required Parameters
INIT	**SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** **VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	BLOCKRANGE(*bbbbbb,bbbbbb***)** **CHECK(***n***)|NOCHECK** **CONTINUE|NOCONTINUE** **FBAVTOC(END|***rbn[,extent[,cisize]]***)** **LABELS(***n***)** **MAP|NOMAP** **OWNERID(***owner***)** **PURGE|NOPURGE** **RECLAIM|NORECLAIM** **VOLID(***serial***)**

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** are required parameters identifying the volume or minidisk that is to be initialized.

SYSNAME

Required when initializing a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN control statement.

UNITADDRESS

Required when initializing a volume in a CMS or stand-alone environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS, use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual address of a minidisk.

Abbreviations:

```
UNITADDRESS|UNIT
```

Default: None.

Restrictions: The SYSNAME parameter cannot be specified when initializing a volume in the stand-alone environment or when executing ICKDSF under CMS

The UNITADDRESS parameter cannot be specified when initializing a volume that is running under VSE.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(***serial[,owner]***)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before initializing the volume or minidisk.

VERIFY

Required when you want to verify the volume serial number and owner identification before initializing the volume or minidisk. If the volume serial number or owner identification does not match, the INIT command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY

Required when you want to bypass verification of the volume serial number and owner identification.

Abbreviations:

```
VERIFY|VFY
NOVERIFY|NOVFY|NVFY
```

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number; both the volume serial number and the owner identification must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, the command terminates.

Optional Parameters

BLOCKRANGE: Specifying Part of the Volume

Explanation: **BLOCKRANGE(*bbbbbb,bbbbbb*)** is an optional parameter specifying that part of a volume is to be surface checked before standard initialization is performed.

For *bbbbbb,bbbbbb* substitute 1 to 6 decimal (n) or hexadecimal (X'n') digits for the starting and ending blocks to be surface checked.

Abbreviations:

BLOCKRANGE|BLKRANGE|BLKR

Restrictions: RECLAIM is not valid with BLOCKRANGE, because RECLAIM must operate on the entire volume.

BLOCKRANGE is not valid for the IBM 9313 or 9332.

BLOCKRANGE is ignored for minidisks.

CHECK|NOCHECK: Surface Checking of the Volume

Explanation: **CHECK(*n*)|NOCHECK** are optional parameters specifying whether the blocks are to be checked for recording errors. Each block is surface checked by writing and reading specially patterned records. If surface checking identifies a defective block, the block is flagged and an alternate is assigned.

CHECK

Indicates that you want each block to be checked for recording errors during (maximal) initialization. This parameter destroys the contents of each block that it processes.

For *n*, substitute a decimal number from 1 through 10 for the number of times you want each block to be checked. See Appendix G, "Surface Checking" on page 315, for information about n values.

NOCHECK

Indicates that you do not want the blocks to be surface checked for recording errors during (minimal) initialization.

Abbreviations:

CHECK|CHK
NOCHECK|NOCHK|NCHK

Default: The default is NOCHECK.

Restrictions: The maximum number you can specify with the CHECK parameter is 10.

CHECK(n) is not valid for VM minidisks.

CHECK(n) is not valid for the IBM 9313 or 9332.

CONTINUE|NOCONTINUE: RESUME FROM A CHECKPOINT

Explanation **CONTINUE|NOCONTINUE** are optional parameters specifying whether processing is to start from the last checkpointed location (CONTINUE) or to ignore the checkpointed information (NOCONTINUE) and to start processing from the start of the specified range or full volume.

These parameters are interrogated only if processing was previously interrupted during a maximal initialization.

CONTINUE
Indicates that processing is to start from the last checkpointed location.

A message is issued to indicate where processing is resumed.

NOCONTINUE
Indicates that you do not want to resume from the last checkpointed location, but want to resume processing at the start of the range or volume.

Default: The default is CONTINUE

Restrictions: CONTINUE is not valid for the IBM 9313 or 9332. CONTINUE is ignored for minidisks.

FBAVTOC: Specifying the Volume Table of Contents for a Fixed Block Architecture (FBA) Device

Explanation: **FBAVTOC(END|***rbn[,extent[,cisize]]***)** is an optional parameter specifying the starting location, number of label records, and control interval size of the volume table of contents (VTOC). See Appendix F, "The Fixed Block Architecture VTOC (FBAVTOC)" on page 311, for more information on the FBA VTOC.

When you specify FBAVTOC(END), ICKDSF automatically determines the size and location of the VTOC. The VTOC is put on the last blocks of the volume or minidisk; it is 99 VTOC entries in length and has a control interval size of 1024.

For *rbn*, specify 1 to 6 decimal (n) or hexadecimal (X'n') characters for the relative block number of the start of the VTOC.

Note that blocks 0 and 1 are reserved for the IPL block and the volume label blocks (VLB) respectively.

For *extent*, specify one to three decimal (n) or hexadecimal (X'n') characters for the number of label records in the VTOC. The valid range is from 3 to 999; however, the number you specify is rounded up to an integral multiple of the number of labels per control interval.

For *cisize*, specify one to four decimal (n) or hexadecimal (X'n') characters for the size of the control interval in bytes. The value specified must be an integral multiple of the device's physical block size, and must not exceed a value of 8192.

See Appendix F, "The Fixed Block Architecture VTOC (FBAVTOC)" on page 311, for FBAVTOC format and size calculations.

Abbreviations: None

Default: FBAVTOC(2,56,1024)

Restrictions: A VTOC can not be placed at blocks 0 and 1, which are reserved for the IPL block and the volume label blocks (VLB) respectively.

LABELS: Reserving Space for Additional Volume Labels

Explanation: **LABELS(*n*)** is an optional parameter specifying that space on block 1 is to be reserved for additional volume labels, also known as user volume labels. In addition to the volume 1 label that must be written on the volume or minidisk, a maximum of five additional volume labels can be specified.

To specify *n*, substitute a decimal number from 1 through 5 for the number of additional user volume labels to be placed on the volume or minidisk, during initialization.

Abbreviations:

LABELS|LABEL|LBL

Default: None.

Restrictions: If LABELS is not specified, user volume labels that existed from a previous initialization are destroyed.

If a value greater than 5 is specified, 5 is used, a message is printed, and the command continues.

MAP|NOMAP: Printing an Alternate Block Map

Explanation: **MAP|NOMAP** are optional parameters specifying whether to print an alternate block map during initialization. The alternate block map lists the primary blocks on the volume that have alternates assigned.

MAP
Indicates that you want to print an alternate block map during initialization. The format of this map is the same as that printed by the MAPALT command when DETAIL is specified.

NOMAP
Indicates that you do not want to print a block map during initialization.

Abbreviations: None.

Default: The default is MAP.

Restrictions: MAP is not valid for the IBM 9313, 9332, or minidisks.

OWNERID: Specifying the Volume-Owner Identification

Explanation: **OWNERID(*owner*)** is an optional parameter specifying the volume-owner identification to be written in the volume label.

For *owner*, substitute 1 to 14 alphameric characters for the volume-owner identification to be written in the volume label.

Abbreviations:

OWNERID|OWNER

Default: The default for *owner* during a first-time initialization is 14 blanks.

If you do not specify OWNERID when reinitializing a previously initialized volume (or minidisk under CMS), the owner identification remains unchanged.

PURGE|NOPURGE: Writing Over Data on a Previously Initialized Volume

Explanation: **PURGE|NOPURGE** are optional parameters that specify whether you want to write over the data in the following types of data sets during initialization:

- Unexpired
- VSAM
- Data secured

PURGE specifies that you want to write over the data. NOPURGE specifies that you do not want to write over the data. For more information on execution of the PURGE and NOPURGE parameters, see "Volume and Data Set Security" on page 31.

A volume that appears to contain real data cannot be initialized unless the PURGE parameter is specified.

PURGE
Indicates that you want to write over existing data during initialization. For more information about the actions taken when data is found on the volume see "Volume and Data Set Security" on page 31.

NOPURGE
Indicates that you do not want to write over existing data. If a volume appears to contain real data, it cannot be initialized unless the PURGE parameter is specified.

Abbreviations:

PURGE|PRG
NOPURGE|NOPRG|NPRG

Default: The default is NOPURGE.

Restrictions: The NOPURGE parameter does not apply in a CMS or stand-alone environment. If you initialize a volume in a in a CMS or stand-alone environment, all existing data on the volume is purged, regardless of the data set security attributes.

RECLAIM|NORECLAIM: Reclaiming Blocks Previously Flagged Defective

Explanation: **RECLAIM|NORECLAIM** are optional parameters specifying whether to reclaim primary or alternate blocks that were previously flagged defective but appear to be usable after surface checking. Fixed block architecture devices' blocks that were flagged at the factory are not reclaimed.

Should INIT RECLAIM remain incomplete, defective primary blocks might remain that do not have valid alternate pointers. Any subsequent INIT command forces the reclaim process to completion.

RECLAIM
Indicates that you want blocks that are identified as usable by surface checking to be reclaimed during initialization.

NORECLAIM
Indicates that you want to suppress block reclamation during initialization.

Abbreviations:

```
RECLAIM|RCLM
NORECLAIM|NORCLM|NRCLM
```

Default: The default is NORECLAIM.

Restrictions: The RECLAIM parameter applies only when the CHECK parameter has been specified. If RECLAIM is specified with NOCHECK, it is ignored.

RECLAIM is invalid for a minidisk.

RECLAIM is not valid with BLOCKRANGE, because RECLAIM must operate on the entire volume.

RECLAIM is not valid for the IBM 9313 or 9332.

VOLID: Specifying the Volume Serial Number

Explanation: **VOLID(*serial*)** is an optional parameter specifying the volume serial number to be written in the volume label.

For *serial*, substitute one to six alphameric characters for the volume serial number. If fewer than six characters are specified, the serial is left-justified, and the remainder of the field is padded with blanks (X'40').

Abbreviations: None.

Default: When you initialize a volume that was previously initialized and do not specify the VOLID parameter, the old volume serial number remains unchanged.

Restrictions: When you initialize a volume for the first time, you must specify the VOLID parameter, or the INIT command terminates.

If you change the volume serial number of a volume that is mounted online, thereafter you must refer to the volume by the new volume serial number, even though the reference may occur in the same step.

Examples of the INIT Command

The following examples show different ways to code the INIT command.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Initializing a Fixed Block Architecture Volume in a Stand-Alone Environment

In this example, a fixed block architecture device is initialized at a minimal level. The volume is labeled 331022. The owner ID of SMITH is written in the volume label. Space is reserved for 3 additional user volume labels starting at block 1, and a fixed block architecture VTOC starting at block 2 for 100 label records with a VTOC control interval size of 512. NOMAP suppresses the list of alternates.

```
INIT   UNITADDRESS(2E4) NOVERIFY NOCHECK VOLID(331022) -
     OWNERID(SMITH) LABELS(3) FBAVTOC(2,100,512) NOMAP
```

Initializing a Fixed Block Architecture Volume in a VSE Environment

In this example, a fixed block architecture device is initialized at a maximal level by specifying CHECK and VERIFY. The volume serial and owner ID are verified before initialization is continued. Surfaces are checked and, if any blocks are found to be defective, alternate blocks are assigned. An FBA VTOC is located starting at block 2 for 100 label records with a VTOC control interval size of 1024. MAP specifies a detail map is to be printed. Any blocks that were previously flagged defective but now appear to be usable after surface checking are reclaimed.

```
// JOB      jobname
// ASSGN    SYS005,353
// EXEC     ICKDSF,SIZE=AUTO
     INIT  SYSNAME(SYS005) VERIFY(337011,JONES) CHECK(3) -
           FBAVTOC(2,100,1024) MAP RECLAIM
/*
/&
```

Initializing a Fixed Block Architecture Minidisk in a Stand-Alone Environment

In this example, a fixed block architecture 3370 minidisk is initialized in a stand-alone environment. The volume ID and owner ID are checked. Space is reserved for a fixed block architecture VTOC starting at block 2, for 100 label records with a VTOC control interval size of 512. The volume is labeled 337019. Space is reserved for 3 user volume labels. The size of the minidisk is obtained from VM.

```
INIT  UNITADDRESS(151) VERIFY(337022,PSMITH) -
   FBAVTOC(X'2',X'64',X'200') VOLID(337019) LABELS(3)
```

Initializing a Minidisk in a CMS Environment

In this example, a minidisk is initialized at the minimal level. It establishes a VOLID, OWNERID, and the VTOC on a minidisk with a virtual address of 391. The VTOC is written at the default location of cylinder 0, track 1 of the minidisk for a length of one track. The minidisk is labeled 338012 and the owner ID is PAYROLL. The VTOC is written in MVS format.

After performing the INIT, the minidisk is ready for future use in an MVS operating system environment.

```
INIT  UNITADDRESS(391) NOVERIFY VOLID(338012) OWNERID(PAYROLL)
```

INSPECT Command—FBA

You can use the INSPECT command to surface check a volume for defective blocks and to assign alternate blocks without necessarily destroying existing data on the volume. Several options are available for inspection:

- The BLOCKS or BLOCKRANGE parameters are provided to identify which blocks are to be inspected.
- The CHECK parameter is provided to check each block surface and to permit optional specification of the number of checking cycles that are to be performed on each block being inspected. Normally, an alternate is assigned to a primary block only after an inspection of the surface reveals defects. The NOCHECK parameter suppresses such checking and, unless alternate block assignment is suppressed, an alternate will be unconditionally assigned to all specified blocks.
- The PRESERVE, KEEPIT, and HOLDIT parameters are provided to save data that would normally be destroyed during block inspection. Before a surface is inspected, its contents are read and saved. After a surface is inspected and found to be nondefective, the saved data contents are written back to the same area. If a block is found to be defective, the saved data contents are written to an assigned alternate.

  If you specify HOLDIT, the data is kept only in storage. If processing terminates before the data is rewritten, the data is lost.

  If you specify PRESERVE, or KEEPIT, a copy of the data in storage is written at a backup location as well as kept in storage. If processing ends abnormally, the next use of INSPECT detects the data at the backup location and automatically restores the data to the block on which processing ended abnormally.

  The NOPRESERVE parameter allows inspection to destroy the data content of a block. This allows inspection of blocks where data cannot be read.
- The ASSIGN parameter automatically assigns alternates for primary or alternate blocks that are found to be defective. Automatic assignment of alternates is suppressed by the NOASSIGN parameter.
- The MAP parameter provides a printed list of the inspected blocks that have alternates assigned. The NOMAP parameter suppresses the printed output.

For the IBM 9313 and 9332 *only* the following is allowed:

- Only unconditional assignment of alternate blocks (ASSIGN NOCHECK BLOCKS...) is allowed.
- The PRESERVE, KEEPIT, and HOLDIT parameters are all processed as HOLDIT.
- The CHECK or BLOCKRANGE parameters are invalid and cause command termination.
- MAP is ignored.

Inspecting a Volume to Assign Alternate Blocks

A block can be flagged as defective, and an alternate assigned, either conditionally or unconditionally. You can control the assignment of alternate blocks as follows:

- Issue the INSPECT command and specify the CHECK and ASSIGN parameters to detect defective blocks. If a surface is identified as defective by surface checking, the surface is flagged as defective, and an alternate surface is assigned to the defective block.
- Issue the INSPECT command and specify the NOCHECK and ASSIGN parameters to unconditionally assign alternate surfaces for the blocks inspected.
- To inspect primary and alternate blocks, but to prevent assignment of alternates, issue the INSPECT command and specify the NOASSIGN parameter. Note that if PRESERVE is specified and user data cannot be rewritten because of surface defects, an alternate is assigned regardless of the ASSIGN or NOASSIGN parameter specification.

RANGE Parameters: Specifying Part of a Volume

BLOCKRANGE is an optional parameter that allows you to inspect part of a volume. The BLOCKS parameter allows you to specify discrete blocks. BLOCKS and BLOCKRANGE are mutually exclusive.

Preserving Data during INSPECT

When you specify the PRESERVE or KEEPIT parameter of the INSPECT command, a backup copy of the data is stored on the same volume as well as held in storage. The backup location uses two blocks on the CE cylinder. One block (the "backup" block) contains the control information, and another block (the "preserve" block) contains the actual data.

If processing ends abnormally, the next use of INSPECT will:

- Detect the data at the backup location, and determine the block to be recovered (the recovery block)
- Surface check the block to be restored. If a data check is detected, skip displacement surface checking is performed.
- Restore the data from the preserve block.

The restoration of the block from the preserved data is automatic, and is independent of the range specified in the current job. INSPECT can be run with no blocks specified and only the restore function is performed.

INSPECT allows you to choose from two levels of data preservation:

- HOLDIT specifies that a copy of the data from the block being processed is to be kept in storage only, with no copy being kept at the backup location.
- PRESERVE or KEEPIT specifies that a copy of the data from the block being processed is to be written at the backup location as well as kept in storage. The copy at the backup location is used only during the restore procedure.

The backup location and the copy of the data in storage are erased at the completion of the command.

When PRESERVE/KEEPIT is specified, you might notice a performance degradation caused by the writing of the data at the backup location. The performance degradation depends on the total number of blocks being processed by the current INSPECT command.

- For the IBM 9313 and 9332 the PRESERVE, KEEPIT, and HOLDIT parameters are processed as HOLDIT.

Some cautions to be used for PRESERVE processing:

- When data is written at the backup location, it is read-back-checked to ensure it was written correctly. However, at the time of recovery, when the data is to be restored, **it cannot be guaranteed that the data is always retrievable from the backup location**.
- Although the backup block and the preserve block are located on the CE cylinder, the ANALYZE command (which writes on the CE cylinder), does **NOT** destroy the restore data if it exists. ANALYZE will indicate the block number for which data exists. **Note, however, that CE microdiagnostics may destroy the data at the backup location**. Therefore, if INSPECT processing does not run to completion, and it is determined to be a problem that will cause the volume to be turned over to the IBM Customer Engineer, the INSPECT command should be executed to try to restore the data.
- When you are processing minidisks under CMS, data is not saved at the backup location. Therefore, no recovery of data is possible.

Recovering Data After a Job or System Failure

If an INSPECT command did not execute to completion, and PRESERVE was specified on the failing job, ICKDSF has saved the data, and it will be automatically recovered on the next use of the INSPECT command.

After resolving the reason for the failure, you can submit:

- The same INSPECT job.

  This will do a primary surface check of the block being processed at the time of failure, recover the data, and then inspect all the blocks in the specified range, including those blocks that were inspected prior to the previous failure.

- An inspect job for a totally different, or partially different, range.

  This will do a primary surface check of the block being processed at the time of failure, recover the data, and then process all blocks in the new range.

- An ANALYZE job (for nonemulated nonremovable media devices only) to determine the block that failed.

  ANALYZE will print the cylinder and head of the block that is being processed at the point of failure.

  - Then execute a new INSPECT job with a modified range according to the previous point of failure.

This will do a primary surface check of the block being processed at the time of failure, recover the data, and then process all blocks in the new range.

- An INSPECT with neither BLOCKS nor BLOCKRANGE specified.

  This will do a primary surface check of the block being processed at the time of failure, and recover the data.

Notes:

- Recovery is only possible if PRESERVE was specified on the job running at the time of failure. It is not possible if HOLDIT or NOPRESERVE were specified.
- It is possible that INSPECT was "between blocks" when the job failed, and there is no block or data to recover.
- The data at the recovery block may be invalid. In this case, in addition to the data being unavailable, the block may be unusable to the operating system and any use of the block can cause I/O errors.
- If you run INIT on this volume before the next INSPECT, INIT will erase the recovery data.
- The restore function is not optional. If data exists at the backup location it is restored.

  If the data on the preserve block is different than the data on the block being restored (that is, the block being processed at the point of failure), the operator is prompted either to restore the data or erase the data.
- If the data cannot be restored, for whatever reason, the operator is prompted for the next action.

  To put the restore data in storage, you can run a job that specifies HOLDIT.
- When you are processing minidisks under CMS, data is not saved at the backup location. Therefore, no recovery of data is possible.

Using the INSPECT Command under CMS—FBA

When you run ICKDSF under CMS you can run in two modes:

- Minidisk mode
- ATTACHed mode

Minidisk Support under CMS

To use INSPECT under CMS you must have DEVMAINT authority (as defined in the CP directory) and specify either the USERID or REALADDR parameter. This allows you to INSPECT another user's minidisk (USERID) or INSPECT a volume specifying the real address of the volume (REALADDR) as follows:

USERID: When you have DEVMAINT authority and specify the USERID parameter, you can INSPECT another user's minidisk. The following parameters are valid in this mode:

- ASSIGN/NOASSIGN, BLOCKRANGE/BLOCKS, CHECK/NOCHECK, HOLDIT, MAP/NOMAP, PRESERVE/NOPRESERVE, VERIFY,NOVERIFY, USERID, and UNITADDRESS

REALADDR: When you have DEVMAINT authority and specify the REALADDR parameter, you can INSPECT a volume using the real device address.

Note: The maximum number of blocks you can INSPECT in this mode is one.: The following parameters are valid in this mode:

- ASSIGN/NOASSIGN, BLOCKRANGE/BLOCKS, CHECK/NOCHECK, HOLDIT, MAP/NOMAP, PRESERVE/NOPRESERVE, VERIFY,NOVERIFY, and UNITADDRESS

ATTACHed Devices under CMS

Under CMS, volumes can be ATTACHed to your user ID. When you use ICKDSF in this mode, you can perform any ICKDSF function which can be performed when you are running ICKDSF stand-alone.

Syntax

Command	Required Parameters
INSPECT	**SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)|REALADDR(***ccuu***)** **VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	ASSIGN|NOASSIGN **BLOCKRANGE(***bbbbbb,bbbbbb***) |BLOCKS(***rbn[,•••]***)** **CHECK(***n***)|NOCHECK** **MAP|NOMAP** **PRESERVE|KEEPIT|HOLDIT|NOPRESERVE** **USERID(***user's ID***)**

Required Parameters

SYSNAME|UNITADDRESS|REALADDR: Identifying the Device

Explanation: **SYSNAME(***sysxxx***)\|UNITADDRESS(***ccuu***)\|REALADDR(***ccuu***)** are required parameters identifying the volume that is to be inspected.

SYSNAME

Required when inspecting a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN system control statement.

UNITADDRESS

Required when inspecting a volume in a stand-alone environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS, use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual address of a minidisk.

REALADDR(*ccuu*)
Required to specify the real address of a volume. This parameter is only valid when you are running ICKDSF under CMS with DEVMAINT authority.

For *ccuu*, specify the real address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Abbreviations:

UNITADDRESS|UNIT
REALADDR|RADDR

Default: None.

Restrictions: The UNITADDRESS parameter cannot be specified when inspecting a volume that is running under VSE.

The SYSNAME parameter cannot be specified when inspecting a volume in the stand-alone, or CMS environment.

REALADDR is mutually exclusive with UNITADDRESS SYSNAME, and USERID.

The maximum number of blocks that you can INSPECT using REALADDR is one.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(*serial[,owner]*)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before inspecting the volume or minidisk.

VERIFY
Required when you want to verify the volume serial number and owner identification before inspecting the volume or minidisk. If the volume serial number or owner identification does not match that found on the volume or minidisk the INSPECT command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY
Required when you want to bypass verification of the volume serial number and owner identification.

Abbreviations:

VERIFY|VFY
NOVERIFY|NOVFY|NVFY

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number. Both the volume serial number and owner identification must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, execution of the command terminates.

When you specify VERIFY, the volume label must exist for the volume.

Optional Parameters

ASSIGN|NOASSIGN: Specifying Assignment of Alternate Blocks

Explanation: **ASSIGN|NOASSIGN** are optional parameters specifying whether alternates are to be assigned when defective primary and assigned alternate blocks are detected.

ASSIGN
Indicates that you want alternates to be assigned when defective primary and assigned alternate blocks are detected.

NOASSIGN
Indicates that you do not want alternate blocks assigned. If PRESERVE is specified and user data cannot be rewritten because of surface defects, an alternate is assigned.

Abbreviations:

```
ASSIGN|ASGN
NOASSIGN|NOASGN|NASGN
```

Default: The default is ASSIGN.

Restrictions: If NOASSIGN is specified with the CHECK parameter, an inspected block is not marked defective and an alternate is not assigned.

The combination of ASSIGN NOCHECK cannot be specified with the BLOCKRANGE parameter.

BLOCKRANGE|BLOCKS: Specifying Which Blocks to Inspect

Explanation: BLOCKRANGE(*bbbbbb,bbbbbb***) |BLOCKS(***rbn[,•••]***)** are provided to identify which blocks are to be inspected. The BLOCKS parameter allows you to inspect discrete blocks and the BLOCKRANGE parameter allows you to inspect a range of blocks.

BLOCKRANGE
indicates that part of a volume is to be surface checked before the volume is processed.

For *bbbbbb,bbbbbb*, substitute 1 to 6 decimal (n) or hexadecimal (X'n') digits for the starting and ending blocks to be surface checked.

BLOCKS
specifies each block to be inspected.

For *rbn*, substitute the relative block number of each block to be inspected.

Specify one to six decimal (n) or hexadecimal (X'n') characters. The number of *rbn* parameters specified cannot exceed 20.

Abbreviations:

BLOCKRANGE|BLKRANGE|BLKR
BLOCKS|BLOCK

Default: None.

Restrictions: BLOCKS and BLOCKRANGE are mutually exclusive.

BLOCKRANGE is not valid with ASSIGN NOCHECK.

BLOCKRANGE is not valid for the IBM 9313 or 9332.

When you are processing minidisks under CMS, if you specify the USERID parameter you can INSPECT the whole minidisk. If you specify the REALADDR parameter, the maximum you can INSPECT is one block.

The number of blocks specified by the *rbn* of the BLOCKS parameter cannot exceed 20.

The maximum range of blocks that can be specified in the BLOCKRANGE parameter is 80,000.

Maximum block numbers for fixed block architecture devices are as follows:

- For the IBM 3310: 126015
- For the IBM 3370-1: 557999
- For the IBM 3370-2: 712751
- For the IBM 9313: 277055
- For the IBM 9332: 360035
- For the IBM 9335: 805139

CHECK|NOCHECK: Surface Checking of the Volume

Explanation: CHECK(*n*)|**NOCHECK** are optional parameters specifying whether the blocks are to be checked for recording errors. Each block is surface checked by writing and reading specially patterned records. If surface checking identifies a defective block, the block is flagged, and an available alternate block is assigned.

CHECK

Indicates that you want each block to be surface checked for recording errors during inspection and assigned an alternate only if the checked block appears defective.

For *n*, substitute a decimal number from 1 through 10 for the number of times you want each block to be checked. See Appendix G, "Surface Checking" on page 315, for information on values of n.

NOCHECK

Indicates that you do not want each block to be surface checked for recording errors during inspection. Alternate blocks are unconditionally assigned as a result of inspection when NOCHECK and ASSIGN are specified.

Abbreviations:

CHECK|CHK
NOCHECK|NOCHK|NCHK

Default: The default is CHECK(1).

Restrictions: The maximum number you can specify with the CHECK(n) parameter is 10.

CHECK is not valid for the IBM 9313 or 9332.

MAP|NOMAP: Printing an Alternate Block Map

Explanation: **MAP|NOMAP** are optional parameters specifying whether to print an alternate block map during inspection. The alternate block map lists inspected blocks that have alternates assigned.

If you specify MAP, the format of the printed list is the same as the alternate block map printed by the MAPALT command when DETAIL is specified.

MAP
Indicates that you want to print an alternate block map during inspection.

NOMAP
Indicates that you do not want to print an alternate block map during inspection.

Abbreviations: None.

Default: The default is MAP.

Restrictions: MAP is not valid for the IBM 9313 or 9332.

PRESERVE|KEEPIT|HOLDIT|NOPRESERVE: Preventing Destruction of Data on the Blocks Inspected

Explanation: **PRESERVE|KEEPIT|HOLDIT|NOPRESERVE** are optional parameters specifying whether to read and save the data on the inspected blocks. Data read from the specified blocks is held in storage. Data on a block can also be saved at a backup location as well as held in storage. If processing ends abnormally, the next use of INSPECT detects the data at the backup location and restores the data to the block at which processing ended abnormally.

PRESERVE or KEEPIT
Indicates that you want to save the data on the inspected blocks. It also indicates that a copy of the block is to be saved at the backup location as well as in storage (the storage copy is used to rewrite the data). If processing ends abnormally, the next use of INSPECT detects the data at the backup location and automatically restores the data to the block on which processing ended abnormally.

PRESERVE and KEEPIT are synonymous.

The backup location and the storage copy of the block are erased at the completion of a command.

The backup location consists of two blocks on the CE cylinder. One of the blocks contains control information (backup block) and the other block contains the data that is being saved (preserve block).

A performance degradation may be noticed due to the writing of the data at the backup location. The amount of degradation depends on the total number of blocks being processed for the current INSPECT.

When the data is written at the backup location, it is read-back-checked to ensure it was written correctly. However, when the preserve block containing the data is to be restored at the next use of the INSPECT command, there is no guarantee that the data can be recovered.

When data exists at the backup location on the CE cylinder, the ANALYZE command (which writes on the CE cylinder) does not destroy the data. ANALYZE does indicate the cylinder and head of the block for which data exists. **Note however, that the CE microdiagnostics may destroy the data at the backup location**. Therefore, if INSPECT processing does not run to completion, and it is determined to be a problem that will cause the volume to be turned over to an IBM Customer Engineer, all attempts should be made to restore the data first, if necessary.

HOLDIT
Indicates that you want to save the data on the inspected blocks. HOLDIT reads the data and keeps a copy in storage only. If processing for the block does not run to completion, the data is lost and must be restored from a backup copy.

NOPRESERVE
Indicates that you do not want to save the data on the inspected blocks.

Caution: If you specify the NOPRESERVE parameter with either the CHECK or ASSIGN parameter, data on the volume will be destroyed.

Note that the VTOC is no exception to this rule. Special care must be taken with the volume table of contents in a VSE environment.

The NOPRESERVE option should be used only when the data is known to be unwanted, or if data errors on a particular block preclude the preservation of the data.

Abbreviations:

PRESERVE|PRSV
KEEPIT|KEEP
HOLDIT|HOLD
NOPRESERVE|NOPRSV|NPRSV

Default: The default is PRESERVE.

Restrictions: For the IBM 9313, 9332, and CMS minidisks the PRESERVE, KEEPIT, and HOLDIT parameters are processed as HOLDIT.

USERID: Specify Another User's Minidisk

Explanation: **USERID(*user's ID*)** is an optional parameter used to specify that the INSPECT command is to be performed on another user's minidisk. This parameter is valid when you are running ICKDSF under CMS with DEVMAINT authority.

For *user's ID*, substitute the 1 to 8 characters of the user ID of the owner of the minidisk you want to verify.

Abbreviations:

USERID|UID

Default: If USERID is not specified when you are running ICKDSF under the VM operating system, your own minidisk is verified.

Restrictions: USERID can only be specified if you are using ICKDSF under CMS and have DEVMAINT authority.

When you specify the USERID parameter you can INSPECT the whole minidisk.

Examples of the INSPECT Command

The following examples show different ways to code the INSPECT command.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Inspecting a Fixed Block Architecture Device in a Stand-alone Environment

In this example, a fixed block architecture device is inspected for defective blocks. It provides surface checking of the blocks specified in the BLOCKS parameter and conditional assignment of alternate blocks for any blocks that are found defective. The data on the inspected blocks is preserved. A detail map of the inspected blocks is printed.

```
INSPECT  UNITADDRESS(150) BLOCKS(50,55,57,58) CHECK(1) -
         ASSIGN PRESERVE MAP NOVERIFY
```

In this example, a fixed block architecture device that has already been initialized is inspected for defective blocks. The blocks specified in the BLOCKS parameter are surface checked for errors, but an inspected block is not marked defective and an alternate is not assigned.

```
INSPECT  UNITADDRESS(2E4) BLOCKS(2000,2020,2021,2022) -
         CHECK(1) NOASSIGN NOMAP NOVERIFY
```

Unconditional assignment of alternate blocks for a fixed block architecture device in a stand-alone environment

In this example, an inspection is performed on a fixed block architecture device that has already been initialized. The blocks specified in the BLOCKS parameter are unconditionally assigned the next available alternate blocks.

```
INSPECT  UNITADDRESS(380) BLOCKS(X'FA',X'10E',X'118') -
         NOVERIFY NOCHECK ASSIGN
```

Inspecting Another User's Minidisk in a CMS Environment

In this example, a minidisk that is owned by a another user is inspected. You specify the USERID parameter to inspect the minidisk of a user named BROWN at virtual address 391.

Blocks 50 and 55 of the minidisk are inspected. If the inspected block is defective, an alternate is assigned.

```
INSPECT  UNITADDRESS(391) NOVERIFY USERID(BROWN) BLOCKS(50,55)
```

Inspecting a Minidisk under CMS Specifying a Real Address

In this example, part of a volume with a real address of 290 is inspected. You must have DEVMAINT authority to specify REALADDR.

Block 1000 is inspected. If the inspected block is defective, an alternate is assigned.

```
INSPECT  REALADDR(290) NOVERIFY BLOCKS(1000)
```

MAPALT Command—FBA

The MAPALT command maps alternate block assignments.

Note: The MAPALT command is valid only in the VSE and stand-alone environments.

The MAPALT command, for exclusive use on fixed block architecture (FBA) direct access storage devices, produces a printed report, or map, of the primary blocks that have been assigned to alternate blocks on the device. MAPALT parameters allow you to specify a limited range of primary blocks to be mapped or to specify that the entire volume be mapped.

Execution of the MAPALT command is restricted to reading the identification fields of the primary blocks only. It is not necessary to create backup copies of the data on the volume before the command is invoked. Thus, the command may be executed against either a newly installed device or a device that already contains user data.

The MAPALT command can only be executed against the device address that defines the entire device in fixed block mode. **Emulated devices on fixed block architecture devices are not supported.**

The IBM 9313 and 9332 are not supported by the MAPALT command.

Using the MAPALT Command under CMS—FBA

When you are running ICKDSF under CMS, the MAPALT command is valid only with ATTACHed devices (that is, no minidisk support). With ATTACHed devices, the MAPALT support is the same as the stand-alone support for this command.

VM users will be able to IPL the stand-alone version of Device Support Facilities to execute the MAPALT command against ATTACHed volumes. Users who IPL a VSE system under VM can execute the command against ATTACHed volumes by submitting batch jobs to their virtual machines.

MAPALT Report Output

A two-part report is produced. The first part, which is optional, is as shown below:

PRIMARY BLOCK NUMBER	BLOCK ADDRESS	ALTERNATE ADDRESS
nnnnnn	C=nnn H=nn	C=nnn H=nn
nnnnnn	C=nnn H=nn	C=nnn H=nn
nnnnnn	C=nnn H=nn	C=nnn H=nn
nnnnnn	C=nnn H=nn	C=nnn H=nn
nnnnnn	C=nnn H=nn	C=nnn H=nn

This is the detailed report that shows the relative block number of the defective primary block, its location on the volume (cylinder and head number), and the location of the alternate block that has been assigned to the primary (also in cylinder and head number). Depending on the device type, the primary and alternate block addresses can be the same on this report.

The second portion of the report, which is always printed, is a summary message showing the number of blocks (within the specified range) found to have been assigned to alternates.

MAPALT Diagnostic Output

I/O errors that cause interruption of program execution are reported by the program. The following information appears on the program output when such errors occur:

```
CSW = xxxxxx xxxxxx CCW = xxxxxxxx xxxxxxxx
SENSE = xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
        xxxxxxxx xxxxxxxx
```

This information should be used, in accordance with installation procedures, as an aid in isolating and correcting the cause of the problem.

If a data check is encountered, an error message is printed giving the relative block number of the block causing the error, and processing then continues with the next sequential block.

Syntax

Command	Required Parameters
MAPALT	**SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)**
	Optional Parameters
	DETAIL|NODETAIL **LIMITS(***lorbn,hirbn***)|ALL**

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume is Mounted

Explanation: **SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** are required parameters identifying the volume that is to be mapped.

SYSNAME

Required when mapping a volume in the VSE environment. For *sysxxx*, specify the SYSNAME in the ASSGN control statement.

UNITADDRESS
Required when mapping a volume in a stand-alone environment.

For *ccuu*, specify the hexadecimal address (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be on a channel that is operational. If the device is on a channel that is not operational, the program may enter a nonterminating wait state.

Abbreviations:

SYSNAME|SNAME
UNITADDRESS|UNITADDR|UNIT

Default: None.

Restrictions: The SYSNAME parameter cannot be specified in a stand-alone environment.

The UNITADDRESS parameter cannot be specified in the VSE environment.

Optional Parameters

DETAIL|NODETAIL: Controlling the Program Output

Explanation: **DETAIL**|**NODETAIL** is an optional parameter that controls the level of detail in the program output report.

DETAIL
Causes printing of the detailed report that lists each defective block assigned to an alternate, gives its location on the disk, and gives the location of the assigned alternate block.

NODETAIL
Causes printing of the detailed report to be suppressed.

Abbreviations:

DETAIL|DTL
NODETAIL|NODTL

Default: NODETAIL

Restrictions: This parameter only controls printing of the detailed report and does not affect other output from the program. The summary message will be printed at all times upon termination of the program.

LIMITS|ALL: Specifying the Area to Be Mapped

Explanation: **LIMITS**(*lorbn,hirbn*)|**ALL** are optional parameters specifying the area of the disk where alternate block mapping is to be performed.

LIMITS(lorbn,hirbn)
The value *lorbn* specifies the relative block number for the start of mapping. This value must be lower than, or equal to, the value specified by *hirbn*. *lorbn* can be expressed as one to six decimal (n) or hexadecimal (X'n') characters.

The value *hirbn* specifies the relative block number for the end of mapping. This value must be lower than, or equal to, the highest relative block number on the device. *hirbn* can be expressed as one to six decimal (n) or hexadecimal (X'n') characters.

ALL
Indicates that the entire volume is to be mapped.

Abbreviations:

LIMITS|LIMIT|LIMS|LIM

Default: ALL

Examples of the MAPALT Command

The following examples illustrate ways you might code the MAPALT command in various situations.

Example 1: Mapping a Volume in the Stand-Alone Environment

In this example, a full volume scan of a 3310 direct access storage volume is performed and a summary report is made of the total number of blocks found that have alternates assigned.

```
MAPALT  UNITADDRESS(141) ALL NODETAIL
```

Using abbreviations and defaults, you could also have specified:

```
MAPALT  UNIT(141)
```

Example 2: Mapping a Volume in the Stand-Alone Environment

In this example, the range of blocks from 200,000 to 300,000 on a 3370 direct access storage volume is scanned. A detailed report is made of each block within that range that has an alternate assigned, and a summary report is made giving the total number of assigned alternates found.

```
MAPALT  UNITADDRESS(142) LIMITS(200000,300000) DETAIL
```

Using abbreviations, you could have specified:

```
MAPALT  UNITADDR(142) LIM(200000,300000) DTL
```

Example 3: Mapping a Volume in the VSE Environment

In this example, a full volume scan is performed on the fixed block architecture device assigned to SYS001. A detailed report is made of each block on the volume that has an alternate assigned, and a summary report is made giving the total number of alternates found.

```
// JOB   jobname
// ASSGN SYS001,140
// EXEC  ICKDSF,SIZE=AUTO
  MAPALT  SYSNAME(SYS001) ALL DETAIL
/*
/&
```

REFORMAT Command—FBA

The REFORMAT command updates a portion of a previously initialized volume. For fixed block architecture devices, a volume can be reformatted to change the volume serial number and owner identification. Only volumes that have been initialized can be reformatted.

Using the REFORMAT Command under CMS—FBA

When you run ICKDSF under CMS, you can run in two modes:

- Minidisk
- ATTACHed

Minidisk Support under CMS

The minidisk support for the REFORMAT allows you to change the volume serial number, and owner identification of a minidisk. Only previously initialized volumes can be reformatted.

The following REFORMAT parameters are supported for minidisks:

- OWNERID, VERIFY/NOVERIFY, VOLID, and UNITADDRESS

ATTACHed Devices under CMS

Under CMS volumes can be ATTACHed. ICKDSF in this mode can then do any functions which can be performed when you are running ICKDSF stand-alone

Syntax

The abbreviation for the REFORMAT command is RFMT.

Command	Required Parameters
REFORMAT	**SYSNAME(***sysxxx***)|UNITADDRESS(***ccuu***)** **VERIFY(***serial[,owner]***)|NOVERIFY**
	Optional Parameters
	OWNERID(*owner***)** **VOLID(***serial***)**

Required Parameters

SYSNAME|UNITADDRESS: Identifying the Device on Which the Volume Is Mounted

Explanation: **SYSNAME(*sysxxx*)|UNITADDRESS(*ccuu*)** are required parameters identifying the volume or minidisk that is to be reformatted.

SYSNAME

Required when reformatting a volume in the VSE environment. For *sysxxx*, specify the SYSNAME specified in the ASSGN system control statement.

UNITADDRESS

Required when reformatting a volume in a stand-alone or CMS environment.

For *ccuu*, specify the address in hexadecimal (3 or 4 digits) of the channel and unit on which the volume is mounted.

Note: The device must be mounted on a channel that is online. If the device is on a channel that is offline, the program might enter a nonterminating wait state.

Under CMS use the UNITADDRESS parameter to specify the virtual address of an ATTACHed volume or the virtual address of a minidisk.

Abbreviations:

UNITADDRESS|UNIT

Default: None.

Restrictions: The UNITADDRESS parameter cannot be specified when reformatting a volume that is running under VSE.

The SYSNAME parameter cannot be specified when reformatting a volume in the stand-alone environment or under CMS.

VERIFY|NOVERIFY: Verifying the Volume Serial Number and Owner Identification

Explanation: **VERIFY(*serial[,owner]*)|NOVERIFY** are required parameters specifying whether to verify the volume serial number and owner identification before reformatting the volume or minidisk.

VERIFY

Required when you want to verify the volume serial number and owner identification before reformatting the volume or minidisk. If the volume serial number or owner identification does not match that found on the volume, the REFORMAT command terminates.

For *serial*, substitute 1 to 6 alphameric characters for the volume serial number.

For *owner*, substitute 1 to 14 alphameric characters for the owner identification.

NOVERIFY

Required when you want to bypass verification of the volume serial number and owner identification.

Abbreviations:

VERIFY|VFY
NOVERIFY|NOVFY|NVFY

Default: None.

Restrictions: You cannot verify the owner identification separately from the volume serial number; both must be verified when you want to verify the owner identification.

When you specify the VERIFY parameter and verification fails, execution of the command terminates.

Optional Parameters

OWNERID: Specifying a New Volume-Owner Identification

Explanation: **OWNERID(*owner*)** is an optional parameter specifying a new volume-owner identification to be written in the volume label.

For *owner*, substitute one to fourteen alphameric characters for a new volume-owner identification in the volume label.

Abbreviations:

OWNERID|OWNER

Default: When a volume is reformatted and a new owner identification is not specified, the old owner identification remains unchanged.

VOLID: Specifying a New Volume Serial Number

Explanation: **VOLID(*serial*)** is an optional parameter specifying a new volume serial number to be written in the volume label.

For *serial*, substitute one to six alphameric characters for the new volume serial number. If fewer than six characters are specified, the serial number is left-justified and the remainder of the field is padded with blanks (X'40').

Abbreviations: None.

Default: When you reformat a volume or minidisk and do not specify the VOLID parameter, the old volume serial number remains unchanged.

Restrictions: The catalog must be changed to reflect the new volume serial number for any cataloged data sets that reside on the reformatted volume.

Examples of the REFORMAT Command

The following examples show different ways to code the REFORMAT command.

Any values specified here are examples only and should not necessarily be interpreted as the values for your system.

Example 1: Reformatting a Fixed Block Architecture Volume to change the Volume Serial Number (VSE)

In this example, the volume serial number and owner ID are verified and a new volume serial number and owner ID are assigned.

```
// JOB      jobname
// ASSGN    SYS010,380
// EXEC     ICKDSF,SIZE=AUTO
   REFORMAT SYSNAME(SYS010) VERIFY(OLDVOL,SMITH) -
            VOLID(NEWVOL) OWNERID(JONES)
/*
/&
```

Example 2: Reformatting a Fixed Block Architecture Volume to Change the Volume Serial and Blank Out the Owner-ID (VSE)

In this example, the volume serial number and owner ID are verified, a new volume serial number is assigned, and the owner ID is blanked.

```
// JOB      jobname
// ASSGN    SYS011,381
// EXEC     ICKDSF,SIZE=AUTO
   REFORMAT SYSNAME(SYS011) VERIFY(OLDVOL,JONES) -
            VOLID(NEWVOL) OWNERID('b')
/*
/&
```

where the b in 'b' is a blank character.

Reformatting a minidisk in a CMS Environment

In this example, the volume serial number of a minidisk at virtual address 291 is being changed from OLDVOL to NEWVOL, and the owner identification from PAYROLL to NEWPAY. The VERIFY parameter is specified to ensure that the correct volume and owner is being accessed before the volume serial/owner ID is changed.

```
REFORMAT UNITADDRESS(291) VERIFY(OLDVOL,PAYROLL) -
         VOLID(NEWVOL) OWNERID(NEWPAY)
```

Appendix A. Volume Layout and Record Formats on Count-Key-Data (CKD) Devices

Figure 13 shows that, immediately after its index point, each track on a count-key-data device begins with a home address that describes the track type and condition. The home address is followed by a record 0 that links defective primary tracks with their associated alternate tracks. After the home address and record 0, there are zero or more keyed or nonkeyed user-written records. (The IBM 2305 Fixed Head Storage Models 1 and 2 do not have home address records.)

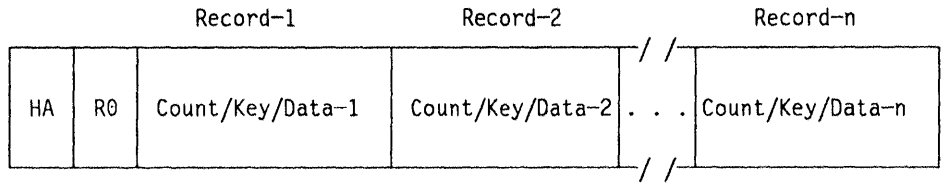


Figure 13. Standard Track Layout (Count-Key-Data Devices)

Figure 14 on page 222 is an example of the volume map that can be received after processing a 3330 volume with Device Support Facilities with either the INIT or the INSPECT command.

```
ICKDSF xxx DEVICE SUPPORT FACILITIES yyy  TIME: 09:14:01    11/20/86    PAGE 1

  INIT DDNAME(INPUT) PURGE CHECK(1) VTOC(0,10,10) VOLID(NEWVOL) NOVERIFY

•
•
•

ICK01307I DEFECTIVE TRACK LIST FOR VOLUME NEWVOL

ICK01308I THE FOLLOWING PRIMARY TRACKS WERE FOUND DEFECTIVE:

  CCHH OF TRACK -- CCHH OF ALTERNATE -- FLAGGED DEFECTIVE -- TRACK CONDITION
    00D4 0008        0328 0000               YES              DEFECTIVE
    00DB 0008        0328 0001               YES              DEFECTIVE
    0189 0000        0328 0002               YES              DEFECTIVE
    018C 0011        0328 0003               YES              DEFECTIVE
    020F 0005        0328 0004               YES              DEFECTIVE
    0224 000D        0328 0005               YES            NOT-DEFECTIVE
    027B 000E        0328 0006               YES              DEFECTIVE
    02CE 0003        0328 0007               YES            NOT-DEFECTIVE
    0307 000A        0328 0008               YES              DEFECTIVE

ICK01309I THE FOLLOWING ALTERNATE TRACKS WERE FOUND DEFECTIVE:

  CCHH OF TRACK -- CCHH OF PRIMARY ---- FLAGGED DEFECTIVE -- TRACK CONDITION
    032B 000D        032B 000D               YES              DEFECTIVE

ICK01313I VOLUME CONTAINS   133 ALTERNATE TRACKS --   123 AVAILABLE.

ICK01314I VTOC IS LOCATED AT CCHH=X'0000 000A' AND IS     10 TRACKS.

ICK00001I FUNCTION COMPLETED HIGHEST CONDITION CODE WAS 0 09:14:01  11/20/86
```

Figure 14. Volume Map Output (Count-Key-Data Devices)

Notes to Figure 14:

1. **xxx** = Operating System (for example, VS2, VSE, XA (MVS/XA), or SA (stand-alone)). **yyy** = Release (for example, 10).
2. Message ICK01307I is the first line of the volume map; it identifies the volume serial number for which the map is produced.
3. Message ICK01308I is printed as the second line of the volume map when one or more defective primary tracks were found.

   Note: Only as a result of medial and maximal initialization will all defective tracks be listed in the volume map. Minimal initialization and the INSPECT command result in a list of only those tracks that were checked during execution of the respective command, namely, the alternate tracks and the individual tracks inspected, respectively. Some of the listed tracks may not be defective, because the list may include tracks that are marked defective because of unconditional assignment of alternate tracks.

4. Four columns of information are provided:

   CCHH OF TRACK: The hexadecimal track address of the defective primary track.

   CCHH OF ALTERNATE: The hexadecimal track address of the alternate track assigned to the primary track. If a primary track is marked defective but does not have an associated alternate track, this column contains the primary track address.

   FLAGGED DEFECTIVE: A YES indicates that the home address flag byte has its defective-track flag on; a NO indicates that the defective-track flag is off.

   TRACK CONDITION: This column is printed only if the CHECK parameter was specified with the command or if an unrecoverable track was found. DEFECTIVE indicates that the track could not be read without a DATA CHECK occurring. NOT-DEFECTIVE indicates that the track could be read and written without difficulty. The NOT-DEFECTIVE condition appears only when the RECLAIM parameter is not specified, and indicates that the defective-track flag was on and, although the track did not appear to have defects, it was not reclaimed. UNRECOVERABLE indicates that surface checking could not be performed because I/O errors occurred, that prevented either the home address or the record 0 from being written on the track.

5. Message ICK01309I appears when one or more alternate tracks are found defective. The same information is provided for alternate tracks as is provided for primary tracks, but the second column identifies the primary track associated with the alternate track.

   Note: In some cases, the track address for the primary track is the same track address as that for the alternate track, which indicates that the alternate track is not associated with a primary track. When an alternate track is found unrecoverable, the "FLAGGED DEFECTIVE" column is blank, and the primary track address is set to NONE.

6. Message ICK01313I indicates the number of alternate tracks defined for the volume and the number of alternate tracks that are neither defective nor assigned.

7. Message ICK01314I identifies the track address where the volume table of contents begins and the number of tracks reserved for it.

Figure 15 shows the contents of cylinder 0, track 0 after the volume has been initialized. The first track of the first cylinder is reserved for system-defined records:

- Home address and record 0
- IPL bootstrap records
- Volume label
- Optional user-volume labels
- Optional IPL program

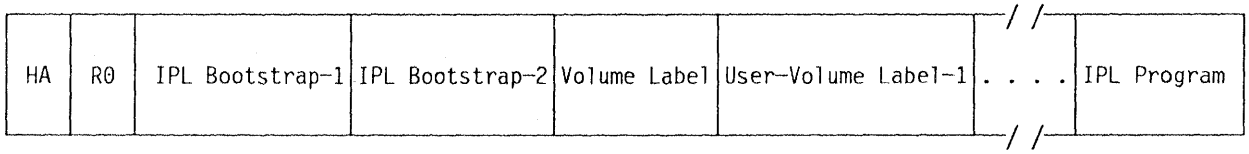


Figure 15. Cylinder 0, Track 0 Layout (Count-Key-Data Devices)

Figure 16 shows the contents of the IPL bootstrap records as the system supplies them. Each bootstrap record has a count area and a 4-byte key area. The key of the first record is IPL1, and the count area indicates that its data area is 24 bytes long. The key of the second record is IPL2, and the count area indicates that its data area is 144 bytes long. Bootstrap record 1 causes bootstrap record 2 to be read, and bootstrap record 2 causes the IPL program to be read. The IPL program contains user-defined values.

IPL Bootstrap Record 1:

00000000 00000000	PSW
06003A98 60000060	CCW: Read
08003A98 00000000	CCW: TIC

IPL Bootstrap Record 2:

07003AB8 40000006	CCW: Seek
31003ABE 40000005	CCW: Search Id Equal
08003AA0 00000000	CCW: TIC
06000000 20000000	CCW: Read
000000000000	Seek Address—0:0
0000000004	Search Address—R 4
0000000000000000	Padding
0000000000000000	(101 Bytes)

Figure 16. IPL Bootstrap Records

Figure 17 shows how the primary and alternate track associations are recorded on the volume.

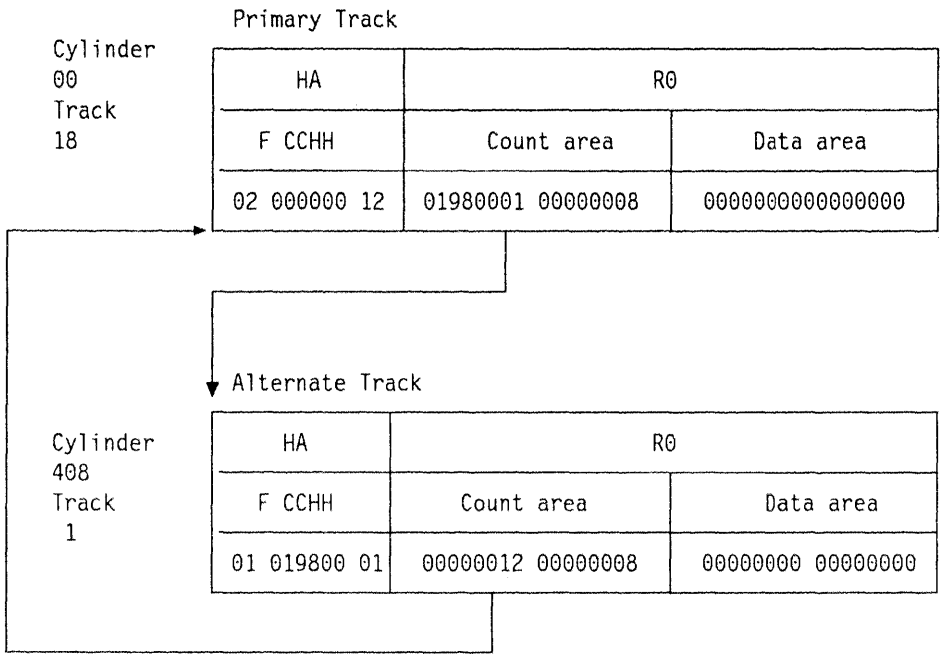


Figure 17. Primary/Alternate Track Association (Count-Key-Data Devices)

Appendix B. VM Support

ICKDSF divides minidisks into two categories:

- Minidisks that are used by CMS
- Minidisks that are used by guests, specifically MVS or VSE operating systems

Minidisks are supported in both the CMS version of ICKDSF and in the stand-alone version.

In the stand-alone environment, ICKDSF operates only on minidisks that are to be used by guests, and support is limited to the functions that are required to initialize the minidisk for operating system usage. This support is provided using the MIMIC(MINI) parameter, but it is recommended that you process minidisks under CMS. "Minidisk Support in the Stand-alone Environment" on page 230 contains more information on the stand-alone support.

In the CMS environment, ICKDSF support is provided for all minidisks, and includes the ability to perform media maintenance as well as device initialization functions.

Support for ATTACHed devices is provided in both environments, and is also discussed here.

Executing ICKDSF under CMS

ICKDSF runs under the CMS operating system of VM/SP, VM/SP HPO, and VM/XA 370 mode. Support is provided in two modes:

- Minidisk
- ATTACHed devices

Minidisk Support under CMS

ICKDSF divides minidisks into two categories: those that are used by CMS, and those that are to be used by guests, specifically MVS or VSE operating systems.

Minidisk support is provided for users who have DEVMAINT authority (as defined in the CP directory) and for users who do not have DEVMAINT authority

- **Users who do *not* have DEVMAINT authority**

  In this mode, support is provided for those functions needed for minidisks that are to be used by guests for MVS or VSE operating systems. This includes INIT and REFORMAT functions, plus the data scanning function of the ANALYZE command. You operate only on your own minidisks.

- **Users who do have DEVMAINT authority**

  In this mode you can perform all the functions provided above, and in addition you can perform the media maintenance functions of the INSPECT and ANALYZE command for either your own or another user's minidisks. In this mode, you can also specify the real address of a device for the ANALYZE and INSPECT commands.

Figure 18 lists the Device Support Facilities supported commands and indicates the devices each command supports when you process minidisks under CMS. A hyphen (-) denotes nonsupport, a D denotes DEVMAINT authority is required, and N denotes DEVMAINT authority is not required to issue the command.

	ANALYZE	INIT	INSPECT	REFORMAT
CKD Devices				
3330-1,2,11	N,D	N	D	N
3333-1,2,11	N,D	N	D	N
3340	N,D	N	D	N
3344	N,D	N	D	N
3350	N,D	N	D	N
3375	N,D	N	D	N
3380 A04, B04, AA4, AD4, BD4, AE4, BE4, AJ4, AK4, BJ4, BK4, CJ2	N,D	N	D	N
FBA Devices				
3310	N,D	N	D	N
3370-1,2	N,D	N	D	N
9313	-	N	D	N
9332	-	N	D	N
9335	D	N	D	N

- not supported
N does not require DEVMAINT authority
D requires DEVMAINT authority

Figure 18. Devices Supported by ICKDSF Commands (Minidisks under CMS)

Minidisk Support without DEVMAINT authority

This support allows you to use the ANALYZE, INIT, and REFORMAT commands against your own minidisks as follows:

- ANALYZE command

  You can use the ANALYZE command to verify that the data on the media surface of your minidisks is readable.

Notes:

1. The ANALYZE drive test is not allowed for a minidisk.
2. The IBM 9335 is not supported for ANALYZE

For more information on using the ANALYZE command under CMS see "Using the ANALYZE Command under CMS" on page 76 or "Using the ANALYZE Command under CMS—FBA" on page 177.

- INIT command

  You can initialize minidisks at the minimal initialization level. This allows you to write the IPL bootstrap records, the volume label and the IPL text, the new owner identification, and reserve space for the VTOC (and index data set).

  For more information on using the INIT command under CMS see "Using the INIT Command under CMS" on page 105 or "Using the INIT Command under CMS—FBA" on page 191.

- REFORMAT command

  Under CMS, you can change the volume serial number, owner identification, and IPL bootstrap records on a minidisk.

  For more information on using the REFORMAT command under CMS see "Using the REFORMAT Command under CMS" on page 159 or "Using the REFORMAT Command under CMS—FBA" on page 217.

Minidisk Support with DEVMAINT authority

With DEVMAINT authority you can execute ICKDSF commands that perform problem determination and media maintenance functions.

To use these functions requires the following:

- DEVMAINT authority (as defined in the CP directory) for your userid
- Use of either the ANALYZE or INSPECT commands
- Use of the USERID or REALADDR parameters of the ANALYZE or INSPECT commands

DEVMAINT authority gives you permission to perform problem determination and media maintenance functions to another user's minidisks. For information on how to obtain DEVMAINT authority see *VM/SP System Facility for Programming*.

With DEVMAINT authority you can:

- Use the ANALYZE command with the USERID parameter to specify another user's ID to verify if the data is readable.

  For more information on using the ANALYZE command under CMS, see "Using the ANALYZE Command under CMS" on page 76 or "Using the ANALYZE Command under CMS—FBA" on page 177.

- Use the INSPECT command with the USERID parameter to perform surface analysis of another user's minidisks

  For more information on using the INSPECT command under CMS, see "Using the INSPECT Command under CMS" on page 137 or "Using the INSPECT Command under CMS—FBA" on page 204.

- Specify a track or block using the REALADDR parameter for a minidisk, when the real cylinder and head (or block) are known but the virtual cylinder and head (or block) are not known. REALADDR is valid with the ANALYZE and INSPECT commands.

ATTACHed Devices Support under CMS

When you are executing ICKDSF under CMS, if you ATTACH devices to your VM user identification you then have control over the volume. You can then perform any ICKDSF function that can be performed in stand-alone mode.

The following ICKDSF commands are valid when you are processing ATTACHed devices:

- ANALYZE
- CONTROL
- INIT
- INSPECT
- INSTALL
- MAPALT
- REFORMAT
- REVALidate

Figure 2 on page 7 lists the Device Support Facilities commands and indicates the devices each command supports when you process ATTACHed devices under CMS

Minidisk Support in the Stand-alone Environment

In the stand-alone environment, ICKDSF supports certain functions to VM minidisks that enable future use in an operating system environment (MVS, VSE).

The diagnostic and media maintenance functions are inoperative, unless otherwise noted.

If diagnostic and/or media maintenance are required on a device in the stand-alone environment, **the device must be ATTACHed to the USERID** that is executing ICKDSF and the MIMIC parameter should not be used. A device that is LINKed to a user id operates as if it were a full volume minidisk.

Commands Supported under Stand-alone

The commands supported are as follows:

- INIT

  The INIT (initialize) command is supported at the minimal level for all devices that are supported as VM minidisks (see Notes 1 and 2). This includes all current DASD devices. Also, if your minidisk is on an MSS, such that VM considers the device type to be a 3330V, ICKDSF INIT will operate as it would for any other 3330.

  Executing minimal INIT:

  - Count-key-data devices

    Minidisk processing for the INIT command for count-key-data devices is activated by

MIMIC(MINI(n)) DEVTYPE(devicetype)

where n = number of cylinders.

The device type (DEVTYPE) parameter is required.

If the MIMIC parameter is omitted and a minimal INIT is attempted, I/O errors will appear to occur.

- FBA Devices

  Minidisk processing for the INIT command for FBA devices is "automatic." ICKDSF can dynamically determine the number of blocks allocated to this minidisk, and can initialize the device accordingly. MIMIC(MINI(n)) should **not** be specified.

  FBA minidisks are supported in the stand-alone environment only.

  MAP is not supported.

- INSPECT

  INSPECT is supported for 2311, and 2314/2319 minidisks only (see note 1).

- REFORMAT

  The REFORMAT command executes on a minidisk exactly as it executes on a real volume.

- BUILDIX

  BUILDIX is an MVS environment command. It is therefore unaware of the existence of the minidisk and executes as it would on a real volume.

- ANALYZE

  The SCAN option of the ANALYZE command can be executed on minidisks in any operating environment, with the following restrictions:

  - NODRIVETEST must *always* be specified.
  - LIMITS (or the equivalent range parameters) must always be specified, and must be within the bounds of the minidisk.

  If you do not adhere to these restrictions, apparent I/O errors will occur.

Notes:

1. The last cylinder of the minidisk for 2311, 2314, and 2319 is always reserved for use as an alternate cylinder; therefore, these minidisks must have a minimum of two cylinders. Medial and maximal initialization and all functions of the INSPECT command are also provided for these count-key-data DASD devices.

   Although alternate track assignment is available on these minidisks (because VM does not restrict as many I/O operations as on other devices), I/O errors on the device should be reported to the system administrator so that a permanent alternate track can be assigned to the "real" defective track.

2. The functions performed by ICKDSF are not always supported by the operating system under which it runs. For example, ICKDSF will initialize an FBA minidisk in the stand-alone environment. You are cautioned to ensure that the VSE operating system will operate correctly against this device.

Appendix C. Device Support Facilities Messages (ICK)

This appendix contains all Device Support Facilities messages with the ICK prefix.

For ease of reference, you might want to remove this appendix from the publication and insert it in one of the following publications:

- *MVS/370 Message Library: System Messages*, if you use Device Support Facilities in an MVS/370 environment
- *MVS/XA Message Library: System Messages*, if you use Device Support Facilities in an MVS/Extended Architecture environment

Introduction

Messages are detailed as fully as possible to permit you to understand a condition and to take necessary action. Information for each message includes, when appropriate:

- *Explanation*: What the message means, why it appears, what caused it, what its variable fields are.
- *System Action*: What is happening as a result of the condition causing the message. For example, is the system waiting for responses.
- *Operator and/or Programmer Response*: If a response is necessary, who performs it, what the pertinent responses are, and their effect on the system or program.
- *Problem Determination*: If the problem recurs or is persistent to the point that system efficiency suffers, what actions can be performed to obtain adequate data for trained hardware or programming support personnel to diagnose the problem.

The last item, Problem Determination, will often refer to the tabular listing of problem determination actions in Appendix D, "Problem Determination Tables for MVS" on page 299. These common action tables do not teach diagnostic techniques, but instruct the operator or programmer what to do when problems recur. The problem determination actions are aids to identify hardware or programming problems and they ensure that support personnel have the essential programming information available to diagnose and correct the difficulty.

Message Routing and Descriptor Codes

Routing codes provide the ability to route system operator messages to selected functional areas.

Device Support Facilities messages will be routed to default functional areas as defined by a specific installation or, in the absence of such definition, they will be routed to the master console.

The descriptor codes will always indicate system status.

Component Name ICK

Program Producing Message Device Support Facilities

Audience/Where Produced For operator: the operator's console.
For system programmer: printed output.

Message Format ICK s nnnn t text or ICK nnn t text

s

Condition code indicating severity of message received at the output printer. Messages received at the console do not have a condition code indicating severity.

0 Information: no effect upon execution
1 Warning: function might fail
2 Error: function probably failed
3 Serious Error: function failed
4 Terminal error: Device Support Facilities processor terminated

It is possible to determine which of several messages issued during execution of a command results in a specific condition code value: The highest condition code multiplied by four yields the command's setting of the LASTCC.

nnnn or nnn

Message serial number: 4 digits for printed messages, and 3 digits for messages that appear at the operator's console.

t

Type code:
I Informational, no action required
E Eventual action required
D Decision required, processing waiting
A Operator action required

text

Message text

** Indicates a 'second-level' message that further explains a preceding message. The track addresses (cccc:hhhh) in message text appear in hexadecimal format.

Associated Publications None

Problem Determination If the problem recurs, perform the Problem Determination actions to ensure that qualified support personnel will have the essential programming information available to diagnose and correct the difficulty. Refer to Appendix D, "Problem Determination Tables for MVS" on page 299, for problem determination instructions.

Messages Received at the Console

ICK001D **CONFIRM PURGING OF UNEXPIRED DATA SETS, REPLY U TO PURGE, ELSE T**

Explanation: When the INIT command is specified, if one or more data sets have been found on the volume, permission is being requested to **possibly** destroy the contents of one or more tracks of this data set.

When the INSPECT command is specified, if one or more unexpired data sets have been found on the volume, permission is being requested to **possibly** destroy the contents of one or more tracks of this data set. No check is made to determine if the tracks being inspected are actually within the data set. If the tracks specified in the INSPECT command are not within the unexpired data set, the data set is not affected by the INSPECT command. The data set is **not** purged from the VTOC. The message is requesting permission only to purge the tracks or blocks specified, unless recovery is in effect.

System Action: Device Support Facilities waits for your response.

Operator Response: For the INIT command, reply 'U' to permit unconditional purging of the data set or reply 'T' to stop purging and to terminate the function.

For the INSPECT command, reply 'U' to permit INSPECT to proceed on the tracks specified in the command, or reply 'T&sq. to stop possible destruction of the tracks specified and to terminate the function.

Programmer Response: None.

Problem Determination: Not applicable.

ICK002D **ERROR IN REPLY, REPLY 'U' OR 'T'**

Explanation: When replying to messages whose only acceptable response is U or T, the operator responded with another character. Any other character will cause this request for reentry to be made.

System Action: Device Support Facilities waits for the operator's response.

Operator Response: Respond with U for unconditional purging, or T to terminate.

Programmer Response: None.

Problem Determination: Not applicable.

ICK003D **REPLY 'U' TO ALTER VOLUME ccuu CONTENTS, ELSE 'T'**

Explanation: When processing the volume ccuu offline, the operator is requested to confirm the processing because the offline volume is not checked by the operating system, and data on the volume might be lost or modified. Depending on the command and parameter used, this data might be as little as the volume serial (REFORMAT command with VOLID parameter), a track (INSPECT with NOPRESERVE), or the whole volume (INIT with CHECK).

System Action: Device Support Facilities waits for operator's reply.

Operator Response: Respond U to proceed with command processing; respond T to terminate the command.

Programmer Response: None.

Problem Determination: Not applicable.

ICK004D **READY DEVICE ccuu AND REPLY 'U', ELSE 'T'**

Explanation: The device ccuu is not in the READY state.

System Action: Device Support Facilities waits for the operator's response.

Operator Response: Ensure that the device is in the READY state, and respond U to continue processing the command or respond T to terminate the command.

Programmer Response: None.

Problem Determination: Not applicable.

ICK005E **DEFINE INPUT DEVICE: 'dddd,cuu' or 'CONSOLE'**

Explanation: *dddd* is the device type and *cuu* is the channel and unit address. This message appears only in a stand-alone environment. The operator must specify the device type and location of the command input stream. You can specify the console by entering a null line.

System Action: Device Support Facilities waits for the operator's response.

Operator Response: Specify the input device type and its channel and unit address. You can specify the console by entering a null line. *cuu* can be specified as 3 or 4 digits.

Programmer Response: None.

Problem Determination: Not applicable.

ICK006E **DEFINE OUTPUT DEVICE: 'dddd,cuu' or 'CONSOLE'**

Explanation: *dddd* is the device type and *cuu* is the channel and unit address. This message appears only in a stand-alone environment. The operator must specify the device type and the location of the printed output. You can specify the console by entering a null line.

System Action: Device Support Facilities waits for a reply.

Operator Response: Specify the output device type and its channel and unit address. You can specify the console by entering a null line. *cuu* can be specified as 3 or 4 digits.

Programmer Response: None.

Problem Determination: Not applicable.

ICK007E **INVALID INPUT DEVICE SPECIFIED**

Explanation: This message appears only in a stand-alone environment. The operator specified an invalid device type to message ICK005E.

System Action: Device Support Facilities repeats message ICK005E.

Operator Response: Specify a correct input device type. See "Identifying the Input Device" on page 16.

Programmer Response: None.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK008E INVALID OUTPUT DEVICE SPECIFIED

Explanation: This message appears only in a stand-alone environment. The operator specified an invalid device type to message ICK006E.

System Action: Device Support Facilities repeats message ICK006E.

Operator Response: Specify a correct output device type. See "Identifying the Output Device" on page 16.

Programmer Response: None.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK009I SVC INTERRUPT OCCURRED

Explanation: This message appears only in a stand-alone environment. Probable program error. The SVC instruction cannot be used in a stand-alone environment.

System Action: Device Support Facilities terminates.

Operator Response: Notify the system programmer.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK010I PROGRAM INTERRUPT OCCURRED

Explanation: This message appears only in a stand-alone environment. Probable program error. An instruction executed incorrectly.

System Action: Device Support Facilities terminates.

Operator Response: Notify the system programmer.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK011E I/O ERROR -- error-type, ccuu, command, csw, sense

Explanation: This message appears only in a stand-alone environment. An I/O error of the type indicated occurred on the device at address ccuu. The command in error is indicated as are the contents of the channel status word (CSW) and the results of a sense operation against the device.

System Action: Device Support Facilities terminates.

Operator Response: None.

Programmer Response: Correct the cause of the error, and restart Device Support Facilities.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK012E INTERVENTION REQUIRED, ccuu

Explanation: This message appears only in a stand-alone environment. The indicated device is not in the READY state.

System Action: Device Support Facilities waits for correction of the problem.

Operator Response: Ensure that the device is in a READY state.

Programmer Response: None.

Problem Determination: Not applicable.

ICK013E CONSOLE READ FAILED, REENTER LAST LINE

Explanation: This message appears only in a stand-alone environment. An I/O error occurred when attempting to read a line that was entered at the operator's console.

System Action: Device Support Facilities waits for the line to be reentered.

Operator Response: Reenter the line.

Programmer Response: None.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK014D SET DEVICE ccuu IN WRITE-MODE AND REPLY 'U', ELSE 'T'

Explanation: The device at address *ccuu* is set in read-only mode.

System Action: Device Support Facilities waits for operator's response.

Operator Response: Set the device to allow write operations and reply U. If write operations cannot be allowed, reply T, and the command terminates.

Programmer Response: None.

Problem Determination: Not applicable.

ICK015E TODAY'S DATE, REPLY 'mm/dd/yy'

Explanation: This message appears only in a stand-alone environment. If you want the date to appear as part of the output title line, respond with the month, day, and year as indicated; otherwise, press the ENTER key.

System Action: The date appears in the title line of the output listing, or blanks are substituted if a date was not supplied.

Operator Response: Respond with the month, day, and year if you want the date to appear in the title line of the output listing. If you do not want the date, respond by pressing the ENTER key.

Programmer Response: None.

Problem Determination: Not applicable.

ICK016E SUPPLY TIME OF DAY, REPLY 'hh:mm:ss'

Explanation: This message appears only in a stand-alone environment. If you want the time of day to appear in the title line of the output listing, respond with the hour, minute, and second as indicated; otherwise, respond by pressing the ENTER key.

System Action: The time of day appears in the title line of the output listing, or blanks are substituted if a time of day was not specified.

Operator Response: Respond by specifying the hour, minute, and second if you want the time of day to appear in the title line of the output listing. If you do not want it to appear, respond by pressing the ENTER key.

Programmer Response: None.

Problem Determination: Not applicable.

ICK017D **filename IS A SECURED FILE**

Explanation: The specified 'filename' is a data-secured file and Device Support Facilities requests permission to purge this file in an associated message, ICK018D.

System Action: Processing continues, as the message is informational only.

Operator Response: None.

ICK018D **CONFIRM PURGING OF SPECIFIED FILE NAME, REPLY U TO PURGE, ELSE T**

Explanation: The filename is displayed in message ICK017D. If the INIT command is being executed, permission is being requested to purge the data set.

If the INSPECT command is executing, permission is being requested to **possibly** destroy the contents of one or more tracks of this data set. No check is made to determine if the tracks being inspected are actually contained in this data set. The data set is **not** purged from the VTOC.

System Action: Processing continues after the correct reply is obtained from the operator. If the operator replies T, the command terminates and Device Support Facilities continues processing with the next command. If the operator replies U, processing of the command continues.

Operator Response: The operator must reply to this message with either U or T.

Programmer Response: None.

ICK019D **CONFIRM PURGING OF ALL VSAM FILES, REPLY U TO PURGE, ELSE T**

Explanation: The volume is known to contain one or more VSAM data sets.

When the INIT command is specified, permission is being requested to purge all the VSAM data sets on the volume.

When the INSPECT command is specified, if one or more VSAM data sets have been found on the volume, permission is being requested to **possibly** destroy the contents of one or more tracks of VSAM data set(s). No check is made to determine if the tracks being inspected are actually within the VSAM data set(s). If the tracks specified in the INSPECT command are not within the VSAM data set, the VSAM data set is not affected by the INSPECT command. The data set(s) is **not** purged from the VTOC. The message is requesting permission only to purge the tracks or blocks specified, unless recovery is in effect.

System Action: Processing continues after a correct reply is received. If you reply 'T', the command terminates and ICKDSF continues with the next command. If you reply 'U', command processing continues.

Operator Response: You must reply either 'U' or 'T' to this message.

Programmer Response: None.

ICK020D **DEVICE ccuu IS SHARED. REPLY U OR T**

Explanation: The volume addressed by the "ccuu" is being shared, and Device Support Facilities requests permission to operate on the subject volume.

System Action: Processing continues after the correct reply is obtained from the operator. If the operator replies T, the command terminates and Device Support Facilities continues processing with the next command. If the operator replies U , processing of the command continues.

Operator Response: The operator must reply to this message with either U or T.

Programmer Response: None.

ICK024I **UNABLE TO OPEN VOLUME**

Explanation: The volume that was specified cannot be opened. This could be caused by:

- The DD statement is missing or invalid.
- The channel/unit address is invalid.
- When processing in a shared environment, the device is not in an offline status on the system executing Device Support Facilities.
- I/O errors associated with the volume.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: The status of the volume should be checked. Previous messages should be examined to aid in determining the volume status.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK030E **DEFINE INPUT DEVICE: FN FT FM, OR "CONSOLE", OR "READER"**

Explanation: The system requires you to specify the input device.

System Action: ICKDSF waits for your response.

Programmer Response: None.

Operator Response: If the console is to be used as the ICKDSF input command stream, enter CONSOLE or press the ENTER key. If the ICKDSF input command stream is contained in a CMS file, enter fn ft fm. If the ICKDSF input command stream is contained in your virtual reader file, enter READER (the file must be the first file in the reader). Type "?" to invoke the online help panels.

ICK031E **DEFINE OUTPUT DEVICE: FN FT FM, OR "CONSOLE", OR "PRINTER"**

Explanation: The system requires you to specify the output device.

System Action: ICKDSF waits for your response.

Programmer Response: None.

Operator Response: If the console is to be used as the ICKDSF output device, enter CONSOLE or press the ENTER key. If the output device is to be a CMS file, enter fn ft fm. If ICKDSF output device is a printer, enter PRINTER. Type "?" to invoke the online help panels.

ICK032E **FUNCTION NOT SUPPORTED FOR VIRTUAL MACHINE IN XA MODE**

Explanation: This function is not supported in VM/XA.

System Action: The command terminates.

Programmer Response: As the function you requested in not available under MVS/XA, it must performed in MVS/XA 370 mode.

Operator Response: None.

ICK033E **ENTER CYL OR BLOCK SIZE FOR THE SPECIFIED MINIDISK**

Explanation: The system requires you to specify the size of the minidisk in cylinders for CKD devices or blocks for FBA devices. No check is made to determine the accuracy of the size of the minidisk. This message follows ICK12316I.

System Action: ICKDSF waits for your response.

Programmer Response: None.

Operator Response: Specify the minidisk size (1-8 decimal digits) in cylinders for CKD devices or blocks for FBA devices.

ICK057I **INVALID DEVICE TYPE: VTOC INDEX NOT SUPPORTED ON THIS DEVICE**

Explanation: The device type is not a valid device type for initialization with a VTOC index.

System Action: The creation of the VTOC index is not attempted. INIT command processing continues.

Operator Response: None.

Programmer Response: None.

ICK058I **ccuu I/O ERROR DETECTED DURING INDEX CREATION: ERROR CODE= ***

Explanation: An I/O error occurred during VTOC index creation processing. Error codes:

```
1 - Error occurred reading volume
    label
2 - Error occurred reading VTOC
3 - Error occurred writing VTOC
4 - Error occurred writing index
    records
5 - Index extent violated; increase
    index size
```

System Action: VTOC index creation processing is terminated. The VSE volume indicator is turned on in the VTOC. INIT command processing continues.

Operator Response: Refer the error indication to the system coordinator for problem determination.

Programmer Response: None.

ICK059I **INDEX STARTING LOCATION INVALID AS SPECIFIED**

Explanation: The INDEX parameter on the INIT command statement is invalid. The starting location, as specified, is not valid for one of the following reasons:

1. It caused an overlap with the VTOC.
2. It defined cylinder zero, track zero as the starting location of the index data set.
3. It was outside the physical limits of the device.
4. The relative track specified is invalid for the device.

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: Check the values specified for the INDEX parameter and correct the value in error. Resubmit the job.

ICK061I **ccuu VTOC INDEX CREATION SUCCESSFUL: VOLUME NOW IN INDEX FORMAT**

Explanation: The VTOC index was successfully created on the volume.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK062I **ccuu VTOC INDEX CREATION FAILED: RETURN CODE= 8**

Explanation: Index creation was unsuccessful.

System Action: None.

Operator Response: None.

Programmer Response: Check the job output for additional information on the reason for job failure.

ICK126D **DATA ALREADY EXISTS FOR TRACK X'cccc hhhh' REPLY 'R' TO RECOVER, 'E' TO ERASE THE RECOVERY DATA, OR 'T' TO TERMINATE**

Explanation: This message is issued if recovery data exists for a track (ICK2115I), but there already exists data on that track.

This situation might occur if the original data on the track had not yet been erased at the point of failure.

This situation can occur if an INSPECT did not run to completion, but the volume was available for user use before this invocation of the INSPECT command.

System Action: The operator is prompted for a reply to this message.

If 'r' is specified, the data will be recovered from the recovery information. (The new data on the track is erased).

If 'e' is specified, the recovery data is destroyed. The current data on the track remains.

If 't' is specified, processing terminates. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK130D **DATA CANNOT BE RECOVERED FOR TRACK X'cccc hhhh' REPLY 'C' TO ERASE AND CONTINUE, 'I' TO IGNORE, T TO TERMINATE.**

Explanation: The previous INSPECT command did not complete execution during PRESERVE backup processing. **Note**: The data cannot be recovered from the backup location.

System Action: The operator is prompted for a reply to this message. The ignore option is provided if HOLDIT is specified on the current invocation of the INSPECT command.

If 'c' is specified, the data is erased at the backup location, and processing begins from the start of the specified range. This should be specified if the user has no need to recover data from the backup location.

If 'i' is specified, the recovery data is ignored, and processing continues for the current invocation. Note that this response is allowed only if HOLDIT was specified. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

If 't' is specified, processing terminates. The recovery data remains intact. This should be specified if the user wishes to retry the recovery process, or examine the situation before allowing the data to be erased. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK158D **DATA CANNOT BE RECOVERED FOR BLOCK xxxxxx REPLY 'C' TO ERASE AND CONTINUE, 'I' TO IGNORE, T TO TERMINATE.**

Explanation: The previous invocation of the INSPECT command did not execute to completion during PRESERVE backup processing. ***Note***: The data cannot be recovered from the backup location.

System Action: The operator is prompted for a reply to this message. The ignore option is provided if HOLDIT is specified on the current invocation of the INSPECT command.

If 'c' is specified, the data is erased from the backup location, and processing begins from the start of the specified range. This should be specified if the user has no need to recover data from the backup location.

If 'i' is specified, the recovery data is ignored, and processing continues for the current invocation. Note that this response is allowed only if HOLDIT was specified. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

If 't' is specified, processing terminates. The recovery data remains intact. This should be specified if the user wishes to retry the recovery process, or examine the situation before allowing the data to be erased. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK159D **DATA ALREADY EXISTS FOR BLOCK xxxxx REPLY 'R' TO RECOVER, 'E' TO ERASE THE RECOVERY DATA, OR 'T' TO TERMINATE**

Explanation: This message is issued if recovery data exists for a block (ICK12157), but there already exists data on that block that is not ICKDSF data and is not the original user data.

This situation can occur if an INSPECT did not run to completion, but the volume was available for user use before this invocation of the INSPECT command.

System Action: The operator is prompted for a reply to this message.

If 'r' is specified, the data will be recovered from the recovery information. (The new data on the block is erased).

If 'e' is specified, the recovery data is destroyed. The current data on the block remains.

If 't' is specified, processing terminates. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK177D **REPLY U TO ERASE CONTENTS OF SYSTEM RESERVED AREA ON ccuu, ELSE T**

Explanation: When a CONTROL command with the RECLAIM(SYSAREA) parameter is issued, the operator is requested to confirm the processing before Device Support Facilities starts the reclaim process which will destroy the contents of the System Reserved Area. ccuu is the address of the device that will be modified if the command continues.

System Action: Device Support Facilities waits for the operator's response.

Programmer Response: None.

Operator Response: Respond U to permit the command to start system reserved area reclaim processing or respond T to terminate and prevent destruction of system reserved area contents.

Problem Determination: Not applicable.

ICK00215I **MINIDISK INFORMATION FOR DEVICE ccuu**
RELOCATION FACTOR = nnnnn
SIZE = mmmmm
OWNER = owner-id
FPO LINK ADDRESS = xxxx

Explanation: This message indicates where the user minidisk is located on the real volume. (i.e., for CKD devices, cylinder zero of the mini-disk located at cylinder nnnn of the real volume). Relocation factor and minidisk size are in decimal. The user's minidisk is full-pack overlay LINKed to the invoker as virtual address xxxx by the ICKDSF.

System Action: The command continues processing.

Programmer Response: None.

Operator Response: None.

ICK00216I **REAL DEVICE ccuu**
FPO LINK ADDRESS = xxxx

Explanation: This message indicates that the specified real device ccuu is full pack overlay LINKed as virtual address xxxx to the invoker.

System Action: The command continues processing.

Programmer Response: None.

Operator Response: None.

ICK401I **ccuu SUSPECTED DRIVE PROBLEM**

Explanation: An error has been detected during the drive tests.

System Action: Command processing terminates.

Programmer Response: If you are running under VM (either stand-alone, VSE, MVS, or MVS/XA running under VM), this message can be an indication that a diagnostic function was attempted against a device that was a minidisk and/or LINKed to the userid. Diagnostic functions must be done on devices that are ATTACHed to the userid.

If this is determined to be the case, and if a scan was to be performed as the primary purpose of this ANALYZE command invocation, NODRIVE can be specified. Otherwise, the device must be ATTACHed to the user. See Appendix B, "VM Support" on page 227 for more information on VM support.

Operator Response: Depending on your installation's procedures, you might do the following:

- Restore the entire volume to another drive from a backup volume.
- Discontinue using the drive.

Save the output, which contains further information about the hardware problems.

Take the action appropriate to your installation's procedures for handling suspected equipment problems. If the problem recurs, contact your service representative.

Problem Determination: Table I, items 1, 2, 3, 4, 30.

ICK404I ccuu VOLUME HAS UNFORMATTED DATA BLOCKS

Explanation: An attempt was made to read a data block which was not initialized (data field not written).

System Action: After 504 data checks, the ANALYZE command is terminated.

Operator Response: Perform appropriate procedures to recover all desired data from the volume and then run the VSE utilities INTDK or INTDSK with the 'IQ' option to initialize all data blocks.

ICK407I ccuu NO DRIVE PROBLEMS FOUND

Explanation: Drive test successfully completed executing and did not detect any problems.

System Action: Processing continues.

Operator Response: None. (If a data problem prompted you to run this command, and if the problem persists, run the data verification test. If a drive problem prompted you to run this command, and if the problem persists, further analysis is needed according to your installation's procedures.)

ICK411I FUNCTIONAL VERIFICATION DATA WRITE/READ TEST STARTED

Explanation: Full volume HA/R0 validation and functional data verification test for the REVALidate command process has started.

System Action: Processing Continues.

Programmer Response: None.

Operator Response: None.

ICK412I FUNCTIONAL VERIFICATION DATA WRITE/READ TEST ENDED

Explanation: Full volume HA/R0 validation and functional data verification test for the REVALidate command process has ended.

System Action: Processing Continues.

Programmer Response: None.

Operator Response: None.

ICK501I INVALID DEVICE TYPE SPECIFIED FOR BUILDIX COMMAND

Explanation: A request was made to change the format of a VTOC on a volume whose device type is not supported by the BUILDIX command (for example, a 2314 DASD).

System Action: Execution of the command is terminated.

Operator Response: None.

ICK502I BUILDIX FUNCTION STARTED

Explanation: Execution of the BUILDIX command has begun.

System Action: None.

Operator Response: None.

ICK503I ccuu REQUEST RECEIVED TO CONVERT VTOC TO **FORMAT

Explanation: This message verifies the request for a change of VTOC format. '**' specifies the format requested—either OS or IX.

System Action: Execution of the command continues.

Operator Response: None.

ICK504I ccuu VTOC FORMAT IS CURRENTLY **FORMAT, REQUEST ACCEPTED

Explanation: This message verifies that the BUILDIX function specified on the command statement is valid for the current format of the VTOC on the volume. '**' identifies the format of the VTOC, either OS or IX.

System Action: Execution of the command continues.

Operator Response: None.

ICK505I ccuu VTOC FORMAT IS CURRENTLY **FORMAT, REQUEST REJECTED

Explanation: The format of the VTOC is currently the same as the format requested in the BUILDIX command statement. '**' identifies the format of the VTOC, either OS or IX.

System Action: Execution of the command is terminated.

Operator Response: None.

ICK508A ccuu SHOULD VTOC CONVERSION PROCEED? REPLY 'U' TO CONTINUE, ELSE 'T'

Explanation: This message permits the operator to verify that the user is authorized to request the BUILDIX function before execution of the command begins.

System Action: The system waits for the operator's reply.

Operator Response: Verify that the user is authorized to issue the command, and reply 'U' to continue or 'T' to terminate. Any reply other than 'U' or 'T' causes this message to be issued again.

ICK509I ccuu DIRF FLAG SET IN VTOC, BUILDIX CANNOT PROCEED

Explanation: An error occurred during VTOC processing on a previous job, causing the DADSM interrupt flag to be set in the VTOC. The VTOC is not accurate.

System Action: Execution of the command is terminated.

Operator Response: Prepare a job that will cause reconstruction of the VTOC and run it against the volume. (For example, run an IEFBR14 job to allocate a temporary data set to the volume.) On conclusion of this job, resubmit the BUILDIX command.

ICK510I **ccuu BUILDIX REQUEST CANCELLED DUE TO OPERATOR ACTION**

Explanation: The operator replied 'CANCEL' to a message requiring a response.

System Action: Execution of the command is terminated.

Operator Response: None.

ICK511I **ccuu CVAF ERROR: RETURN CODE=** ERROR CONDITION= ***

Explanation: The common VTOC access facility returned to Device Support Facilities with a return code indicating either a logical error or a physical error. Return codes have the following meanings:

4, 12
Indicate logical errors

8
Indicates an invalid index structure

16
Indicates an I/O error

The error condition will be printed only for a return code of 4 or 8. See Appendix B of *Data Facility Device Support: User's Guide and Reference* for more information on these error conditions.

For MVS/XA, see *MVS/Extended Architecture System-Data Administration.*

For MVS/370, see *MVS/370 System Programming Library: Data Management.*

System Action: Execution of the command terminates with the following conditions existing, depending on the return code:

4, 8, 12
The volume is left in OSVTOC format.

16
The volume is left as it was before the command was issued.

Operator Response: None.

ICK512I **ccuu ERROR: SYS1.VTOCIX. IS A VIO DATASET. BUILDIX TERMINATED.**

Explanation: The index data set was allocated as a VIO data set and is not supported by the BUILDIX command.

System Action: Execution of the command terminates with a return code of 12. The VTOC is left unchanged.

Operator Response: None.

ICK513I **ccuu BUILDIX PROCESSING COMPLETED: VTOC IS NOW IN **FORMAT.**

Explanation: The BUILDIX command completed successfully. '***' shows the new VTOC format, either OS or IX.

System Action: Device Support Facilities is terminated normally.

Operator Response: None.

ICK514I **INDEXED VTOC FACILITY NOT ON SYSTEM. BUILDIX TERMINATED.**

Explanation: A request was made to change a VTOC to IXVTOC format, but the system does not contain indexed VTOC programming support.

System Action: Execution of the command is terminated.

Operator Response: None.

ICK515I **ccuu BUILDIX COMMAND FAILED.**

Explanation: An error occurred that caused execution of the command to be terminated.

System Action: Execution of the command is terminated.

Operator Response: None.

ICK516I **ccuu I/O ERROR DETECTED DURING VTOC CONVERSION: ERROR CODE= ***

Explanation: An I/O error occurred during BUILDIX processing. Error codes:

```
1 - Error occurred in reading the
    volume label
2 - Error occurred in reading the
    VTOC
3 - Error occurred in writing the
    VTOC
4 - Error occurred in writing
    index records
5 - Index extent was violated;
    increase index size
```

System Action: Execution of the BUILDIX command is terminated.

Operator Response: Refer the error indication to the system coordinator for problem determination.

ICK517I **ccuu ERROR: VOLUME IS A DOS STACKED PACK**

Explanation: The volume being processed has a VTOC on the first track of the volume. VTOC conversion is not possible on such a volume.

System Action: Execution of the command is terminated.

Operator Response: None.

ICK518I **ccuu ERROR: VOLUME CONTAINS SPLIT CYLINDER EXTENTS**

Explanation: It was found that the volume contains one or more shared extent data sets. These are not supported by the BUILDIX command.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

ICK519I **ccuu ERROR: SYS1.VTOCIX. DATASET NOT FOUND ON VOLUME**

Explanation: The index data set could not be found on the volume.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

ICK520I **ccuu ERROR: DUPLICATE INDEX DATASET NAME FOUND ON VOLUME**

Explanation: Two data sets were found on the volume that had names beginning with SYS1.VTOCIX. Only one per volume is allowed.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

ICK521I **ccuu ERROR: INDEX DATASET EXTENT NOT CONTIGUOUS**

Explanation: The index data set was found to occupy more than one extent. This data set must occupy one, and only one, extent.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

ICK522I **INVALID DDNAME SPECIFIED**

Explanation: The DDname is incorrect in the command statement.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: Correct the job control statement and resubmit the job.

ICK524I **UNABLE TO OPEN VOLUME**

Explanation: The volume specified cannot be opened. The DD statement could be missing or invalid.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: Correct the error, and resubmit the job.

ICK526I **CONVERSION TO OSFORMAT COULD NOT COMPLETE SUCCESSFULLY**

Explanation: At the conclusion of BUILDIX processing to convert to OSFORMAT, the VTOC indicates an error condition.

System Action: BUILDIX terminates abnormally.

Operator Response: None.

Programmer Response: This message is usually an indication that the OS VTOC on the volume contains an error. The VTOC should be examined to determine the nature of the error.

Problem Determination: Not applicable.

ICK528I **INDEX DATA SET CANNOT START AT CYLINDER 0, TRACK 0**

Explanation: You specified cylinder zero track zero for the index data set location. This is an invalid location.

System Action: BUILDIX processsing terminates. None

Operator Response: None.

Programmer Response: Respecify the command using a valid location.

ICK600I **MAPALT STARTED**

Explanation: Execution has started for the MAPALT command.

System Action: None.

Operator Response: None.

ICK602I **INCORRECT DEVICETYPE, MAPALT TERMINATED**

Explanation: The device type specified is not supported by the MAPALT command.

System Action: Execution of the MAPALT command is terminated.

Operator Response: MAPALT can only be executed on fixed block devices in fixed block mode.

Programmer Response: None.

ICK603I **UNABLE TO OPEN VOLUME**

Explanation: The volume that was specified cannot be opened. The ASSGN statement could be missing or invalid, or the channel and unit address could be invalid.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Correct the error and resubmit the job.

Programmer Response: None.

ICK604I **ccuu LIMITS PARAMETER INVALID AS SPECIFIED, MAPALT TERMINATED**

Explanation: The relative block numbers(s) specified by the LIMITS parameter are invalid.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Correct the value(s) in the LIMITS parameter, and resubmit the job.

Programmer Response: None.

ICK605I **ccuu UNRECOVERABLE I/O ERROR DETECTED, MAPALT TERMINATED**

Explanation: An unrecoverable I/O error (other than a data check) was encountered.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Check the job output, which will contain diagnostic information to aid in analyzing the error. Execute the Device Support Facilities ANALYZE command as a problem determination aid, and follow installation procedures for recovery from this type of error.

Programmer Response: None

ICK606I **nnnnn BLOCKS ASSIGNED TO ALTERNATES IN LIMITS SPECIFIED**

Explanation: This summary message shows the number of blocks that were assigned to alternates within the limits specified on the command statement.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK607I **ccuu MAPALT ABNORMALLY ENDED, REPORT MAY BE INCOMPLETE**

Explanation: This message is printed after an error has been encountered that caused execution of the command to be terminated. The report may be incomplete.

System Action: None.

Operator Response: Check previous messages on job output, and follow installation procedures.

Programmer Response: None.

ICK608I **ccuu MAPALT ENDED NORMALLY, RETURN CODE= n**

Explanation: Execution of the command ended without any errors detected that would have caused termination of execution. A return code of 0 or 4 will be indicated in the message. A return code of 0 means no errors were encountered. A return code of 4 means one or more recoverable errors were encountered.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK609I **PERMANENT DATA CHECK FOUND READING ID FOR BLOCK nnnnnn**

Explanation: A permanent data check was encountered while attempting to read the ID field of the primary block shown in the message.

System Action: Diagnostic information is printed on the output device, and execution of the command continues.

Operator Response: None.

Programmer Response: Save the job output for the system coordinator, and follow installation procedures for data recovery and/or assignment of an alternate block.

ICK610I **TEMPORARY DATA CHECK FOUND READING ID FOR BLOCK nnnnnn**

Explanation: A temporary data check while attempting to read the ID field of the primary block shown in the message.

System Action: Diagnostic information is printed on the output device and execution of the command continues.

Operator Response: None.

Programmer Response: Save the job output for the system coordinator and follow installation procedures for logging and tracking temporary error conditions.

ICK611I **MAIN STORAGE NOT AVAILABLE, MAPALT TERMINATED**

Explanation: Dynamic acquisition of storage for control blocks and work areas failed.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Increase main storage size, and resubmit the job.

Programmer Response: None.

ICK705I **VOLUME SERIAL NUMBER FOR DEVICE ccuu IS xxxxxx**

Explanation: Informational message concerning the volume serial and/or VTOC of the volume at ccuu. If the volume serial was changed, you also receive one or both of the following:

CHANGED FROM xxxxxx - The volume serial number of the ccuu was changed.

The following message could also be received:

VOLUME SERIAL DUPLICATE FOR DEVICE ccuu. VOLUME MADE UNAVAILABLE - The new volume serial on the ccuu is a duplicate of one already known to the operating system. The device has been unloaded.

If the VTOC location was changed, you also receive the following:

VTOC LOCATION MOVED - The VTOC location of the volume at ccuu has moved.

System Action: Device Support Facilities continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

Messages Received at the Output Printer

ICK00001I **FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS nn hh:mm:ss mm/dd/yy**

Explanation: A command has been executed; the value *nn* is the last condition code (LASTCC) generated during execution of the command. *hh:mm:ss* and *mm/dd/yy* are the hours, minutes, seconds and month, day, year respectively of the date and time of the message.

Note: The LASTCC value is the highest condition code found in the messages printed during execution of the command.

System Action: LASTCC is set to *nn*; MAXCC is set to *nn* if *nn* is greater than the current value of MAXCC.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK00002I **ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS nn**

Explanation: This message is issued upon completion of a Device Support Facilities job step. The highest condition code (MAXCC) set during the job step is printed (see message ICK00001I) and returned to the Device Support Facilities invoker in register 15.

System Action: None.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK00204I **PRECEDING COMMAND BYPASSED DUE TO CONDITION CODES**

Explanation: The IF-THEN-ELSE command sequence that was specified caused the command to be bypassed. When specifying an IF-THEN-ELSE command sequence, either the THEN or the ELSE clause is executed, and the one not executed is bypassed.

System Action: The bypassed portion of the command sequence is checked for syntax errors but is not executed. Device Support Facilities processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK00206I **IMPROPERLY PLACED COMMA HAS BEEN FOUND AND IGNORED**

Explanation: A redundant comma was encountered when scanning the command. It is ignored. The omission of positional parameters cannot be indicated by commas; the omission of leading positional parameters is not permitted.

System Action: The command is executed.

Operator Response: None.

Programmer Response: Correct the syntax error to prevent the message from occurring.

Problem Determination: Table I, items 2, 4, and 29.

ICK00222I **WARNING: COMMAND-END DELIMITER APPEARS WITHIN APOSTROPHES**

Explanation: The optional command delimiter, a semicolon, was found within a quoted string. A closing single quotation mark may have been omitted.

System Action: The usage is accepted, and the semicolon is treated as a valid character instead of as a delimiter.

Operator Response: None.

Programmer Response: Check the usage of the semicolon, and correct if necessary.

Problem Determination: Table I, items 2, 4, and 29.

ICK00233I **TOO MANY RIGHT PARENTHESES FOUND. EXCESS IGNORED**

Explanation: Too many closing parentheses were found at the end of the command or following a first-level parameter.

System Action: The excess is ignored, and the command continues processing.

Operator Response: None.

Programmer Response: Remove the excess parentheses.

Problem Determination: Table I, items 2, 4, and 29.

ICK00700I **DEVICE INFORMATION FOR ccuu IS CURRENTLY AS FOLLOWS:**
PHYSICAL DEVICE=yyyy
LOGICAL DEVICE=yyyy
STORAGE CONTROLLER=aaaa
STORAGE CONTROL DESCRIPTOR=bb
DEVICE DESCRIPTOR=cc

Explanation: where

PHYSICAL DEVICE=yyyy identifies the physical device type for the unit being processed. This message is always issued.

LOGICAL DEVICE=yyyy identifies the logical device (for example a 3330 emulated on a 3370). This line is displayed only if the logical device is different from the physical device.

STORAGE CONTROLLER=aaaa identifies the storage control type for the unit being processed. This is issued if the sense-id CCW is supported.

STORAGE CONTROL DESCRIPTOR=bb describes the features associated with the storage controller. It contains the information present in sense-id byte 3. See the storage control document for the device for more information concerning the specific meaning of this byte. This message is issued if the sense-id CCW is supported.

DEVICE DESCRIPTOR=cc describes the features associated with the device. It contains the information present in sense-id byte 6. See the storage control document for the device for more information concerning the specific meaning of this byte. This message is issued if the sense-id CCW is supported.

System Action: Device Support Facilities processing of the command continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK00701I **EQUIPMENT CHECK RETRY SUCCESSFUL**

Explanation: The system has successfully recovered from an equipment check.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None. The CCW, CSW, and sense information are provided to help determine the cause of the error if the assistance of a service representative is required.

ICK01016I **ALTERNATE TRACK CCHH=X'cccc hhhh' ASSIGNED TO PRIMARY TRACK CCHH=X'cccc hhhh'**

Explanation: An alternate track was assigned to a primary track because the primary track was marked defective, or was found to be defective by surface checking, or an unconditional alternate track assignment was requested for the primary track.

System Action: The command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01018I **ALTERNATE CCHH=X'cccc hhhh' RE-ASSIGNED FOR PRIMARY CCHH=X'cccc hhhh'**

Explanation: The alternate track originally assigned to the indicated primary track was either marked defective or was found defective through surface checking. Another alternate track was assigned in place of the original alternate track. If the PRESERVE option was specified, the data found on the original alternate track was copied to the new alternate track.

System Action: The command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01020I **ALTERNATE TRACK CCHH=X'cccc hhhh' WAS RECLAIMED**

Explanation: An alternate track that was previously marked defective appeared to be nondefective when surface checking was performed. The track is added to the set of available alternate tracks.

System Action: The command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01021I **PRIMARY TRACK CCHH=X'cccc hhhh' WAS RECLAIMED**

Explanation: A primary track that was previously marked defective appeared to be nondefective when surface checking was performed. The primary track is marked available, and the assigned alternate track is added back into the set of available alternate tracks. If the PRESERVE parameter was specified, any data written on the alternate track is copied back onto the reclaimed primary track.

System Action: The command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01022I **DEFECTIVE PRIMARY TRACK STATUS WILL BE RESET FOR TRACK X'cccc hhhh'**

Explanation: During the INSTALL/REVALidate command process, the defective primary track status for the specified track will be disassociated from the alternates. Surface checking function will be performed to the specified primary track later to determine if the track is defective.

System Action: Processing continues

Programmer Response: None.

Operator Response: None.

ICK01135I **PRESERVE INFORMATION EXISTS ON CE CYLINDER FOR TRACK X'cccc hhhh'**

Explanation: During ANALYZE DRIVETEST processing, it was determined that the PRESERVE function of the INSPECT command did not execute to completion for the specified track. The track might be unusable. Data may be saved for this track.

System Action: This information is not erased. ANALYZE processing continues with the remainder of the drive test.

Programmer Response: Execute the INSPECT command for this device to ensure the usability of the specified track, and to recover the data if necessary.

ICK01136I **CONTINUE INFORMATION EXISTS ON CE CYLINDER FOR TRACK X'cccc hhhh'**

Explanation: During ANALYZE DRIVETEST processing, it was determined that a previous invocation of the INIT command did not execute to completion. X'cccc hhhh' represents the last track for which a checkpoint was taken. It is likely a track beyond this checkpoint is unusable.

System Action: This information is not erased. ANALYZE processing continues with the remainder of the drive test.

Programmer Response: Execute the INIT command for this device to ensure the usability of the tracks since the previous checkpoint.

ICK01305I **ccuu ALL DATA 'MACHINE READABLE'**

Explanation: This is an information message indicating that all data on the volume can be read successfully.

System Action: Processing continues.

Operator Response: None

Programmer Response: None

ICK01306I **SKIP DISPLACEMENTS ASSIGNED**

Explanation: This is an information message indicating that the skip displacement process occurred and skip displacement(s) have been successfully assigned.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

ICK01307I **DEFECTIVE TRACK LIST IN HEXADECIMAL FOR VOLUME volid**

Explanation: This message is the first line of the volume map. Following this line is a list (in hexadecimal) of any tracks that were found defective during execution of the command.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01308I **THE FOLLOWING PRIMARY TRACKS WERE FOUND DEFECTIVE:**

Explanation: This message is a line in the volume map that identifies (in hexadecimal) any primary tracks that were found defective during execution of the command.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01309I **THE FOLLOWING ALTERNATE TRACKS WERE FOUND DEFECTIVE:**

Explanation: This message is a line in the volume map that identifies (in hexadecimal) any alternate tracks that were found defective during execution of the command.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01310I **NO DEFECTIVE TRACKS WERE FOUND**

Explanation: This message is a line in the volume map that indicates that no defective tracks were found during execution of the command. If the command did not specify that all the tracks were to be checked, tracks that were not checked could be defective.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01311I **NO DEFECTIVE PRIMARY TRACKS WERE FOUND**

Explanation: This message is a line in the volume map that indicates that no defective primary tracks were found during execution of the command. This does not mean that no defective primary tracks exist if the command did not specify that all the primary tracks are to be checked.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01312I **NO DEFECTIVE ALTERNATE TRACKS WERE FOUND**

Explanation: This message is a line in the volume map that indicates that no defective alternate tracks were found during execution of the command. This message does not mean that no defective alternate tracks exist on the volume if the command did not specify that all of the alternate tracks are to be checked.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01313I **VOLUME CONTAINS nnnn ALTERNATE TRACKS -- mmmm AVAILABLE**

Explanation: This message is a line in the volume map. The volume is defined as having *nnnn* alternate tracks reserved; *mmmm* tracks have not been assigned to defective primary tracks nor have been found to be defective and unavailable for assignment.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01314I **VTOC IS LOCATED AT CCHH=X'cccc hhhh' AND IS xxxx TRACKS**

Explanation: This message is a line in the volume map that indicates the cylinder and track where the volume table of contents begins and the number of tracks it occupies.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01315I VOLUME IS IN UN-INITIALIZED STATE

Explanation: Volume label and VTOC information were erased during the INSTALL/REVALidate command process. The volume has no volume label or VTOC.

System Action: None.

Programmer Response: If the INSTALL/REVALidate command process was successful, use the INIT command to initialize the volume.

Operator Response: None

ICK01316I INSTALL FUNCTION COMPLETED SUCCESSFULLY

Explanation: The INSTALL command processing completed successfully.

System Action: Device Support Facilities terminated normally. Processing continues with your next command.

Programmer Response: None.

Operator Response: None.

ICK01318I REVALidate FUNCTION COMPLETED SUCCESSFULLY

Explanation: The REVALidate command processing completed successfully.

System Action: Device Support Facilities terminated normally. Processing continues with your next command.

Programmer Response: None.

Operator Response: None.

ICK01331I TRACK X'ccchhhh' MAY REQUIRE INSPECTION

Explanation: INIT or INSPECT, during CHECK NOSKIP processing for 3340/3344/ 3350 detected a data check on track x'ccchhhh'. Further processing, however, did not detect a defect. This message indicates that if an INSPECT TRACKS (x'cccc', x'hhhh') CHECK(n) is done subsequently for this track, a defect might be detected (and skipped).

System Action: INIT or INSPECT processing continues with the next track.

Operator Response: None.

Programmer Response: After completion of the INIT or INSPECT command, the programmer may want to run an INSPECT CHECK(n) TRACKS(...) for any track indicated in the message.

Problem Determination: Not applicable.

ICK01332I SURFACE ANALYSIS CYLINDER COULD NOT BE UPDATED

Explanation: After assigning a skip displacement to a track, Device Support Facilities could not update the delta map on the surface analysis cylinder with the current skip displacement data for this track.

The CCW, CSW, and sense information associated with the message is printed.

System Action: Processing for the track that contains the defect has completed successfully. Subsequent invocations for this track may have to reanalyze the entire track to locate the defect if the home address becomes unreadable. This message is for information only. There is no user consequence resulting from this condition.

Operator Response: None.

Programmer Response: None.

Problem Determination: Since the usability of the track is not affected, there is no need to examine this information any further. If there is concern, the failing CCW, CSW, and sense information can be examined in detail.

ICK01334I CURRENTLY PROCESSING TRACK CCHH=X'cccc hhhh'

Explanation: Information message given when Device Support Facilities is performing full volume processing. The message is issued to indicate Device Support Facilities progress.

System Action: Processing continues with the next track.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01336I TRACK 'ccchhhh' EXPERIENCED NON-RECURRING DATA CHECK

Explanation: Skip displacement analysis processing for this track detected more than one discrete data check that was not repeatable.

System Action: Processing continues with the next track.

Operator Response: None.

Programmer Response: This condition might arise from too much random noise on a track. If this message is issued for multiple tracks, or many tracks under the same head, this could indicate a potential hardware problem.

Problem Determination: Table I, items 1, 3, 4, 30.

ICK01380I THE FOLLOWING TRACKS ARE UNRECOVERABLE:

Explanation: This message is a line in the volume map that identifies tracks that were found unrecoverable during execution of the command. An unrecoverable track is a track where the home address and/or record zero could not be read successfully after all recovery attempts were made. This message is followed by one or more indications of a track in error as follows: CCHH of track = x'cccc hhhh'.

System Action: If the cause of the unrecoverability of the track is a data error, the command runs to completion with an error return code. If the cause of the unrecoverability is anything other than a data error, command processing stops after detection of the error.

Operator Response: None.

Programmer Response: Prior messages should be examined (including the CCW, CSW, and sense that caused the I/O) to determine the cause of the failure. This message may be caused if the read/write mode switch is set to READ mode, or some other condition is causing the inability of ICKDSF to write on the volume.

Assistance may be required from the customer service representative.

Problem Determination: Table I, items 1, 2, 3, 4, 29, 30.

ICK01381I RECURRING CORRECTABLE ERRORS ON TRACK

Explanation: This message is a line in the volume map that identifies tracks where Device Support Facilities detected recurring correctable data checks. This message is issued only for those count-key-data devices that do not have skip displacement areas. The message is followed by one or more instances of: CCHH of track = x'cccc hhhh'.

System Action: The command completed successfully.

Operator Response: None.

Programmer Response: If the existence of a correctable data check on the indicated track(s) is determined to be detrimental to the installation, an alternate track can be unconditionally assigned to the indicated track using the INSPECT command.

Problem Determination: Table I, items 1, 2, 3, 4.

ICK01400I ccuu ANALYZE STARTED

Explanation: ANALYZE has started to execute its tests on the logical volume identified in the message (that is, on the volume whose unit address is ccuu).

System Action: Processing continues.

Operator Response: None

ICK01405I ccuu ALL DATA 'MACHINE READABLE' WITHOUT ERRORS

Explanation: The volume was successfully read during the data verification test (SCAN option).

System Action: Processing terminates normally.

Operator Response: None. (If a data problem prompted you to run ANALYZE, and if the problem persists, further analysis is needed according to your installation's procedures.)

ICK01406I ccuu ANALYZE ENDED

Explanation: ANALYZE command has completed processing.

System Action: Processing terminates.

Operator Response: None.

ICK01407I ccuu NO DRIVE PROBLEMS FOUND

Explanation: Drive tests successfully completed executing and did not detect any problems.

System Action: Processing continues.

Operator Response: None.

ICK01408I ccuu DATA VERIFICATION TEST STARTED

Explanation: ANALYZE has started to execute its data verification tests.

System Action: Processing continues.

Operator Response: None.

ICK01411I FUNCTIONAL VERIFICATION DATA WRITE/READ TEST STARTED

Explanation: Full volume HA/R0 validation and functional data verification test for the REVALidate command process has started.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

ICK01412I FUNCTIONAL VERIFICATION DATA WRITE/READ TEST ENDED

Explanation: Full volume HA/R0 validation and functional data verification test for the REVALidate command process has ended.

System Action: Processing continues.

Programmer Response: None.

Operator Response: None.

ICK01414I THE FOLLOWING ERRORS WERE ALSO DETECTED

Explanation: This message provides sense data for use by the service representative when a drivetest failure occurs. This message follows messages ICK21407 and/or ICK21409. Please refer to these messages.

System Action: Processing continues.

Programmer Response: Contact your service representative.

Operator Response: None.

ICK01415I CORRECTABLE DATA CHECK OCCURRED ON CE CYLINDER, HEAD = X'hhhh', RECORD = X'rr'

Explanation: An I/O error occurred on the customer engineer (CE) cylinder on the head and record indicated. The message is followed by CCW, CSW, and sense bytes describing the error on the failing record.

System Action: ICKDSF command processing continues. Eventual termination may occur.

Operator Response: None.

Programmer Response: The CCW, CSW, and sense information are provided for use by the service representative if termination occurs.

ICK01501I INVALID DEVICE TYPE SPECIFIED FOR BUILDIX COMMAND

Explanation: A request was made to change the format of a VTOC on a volume whose device type is not supported by the BUILDIX command (for example, a 2314 DASD).

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: Verify that the volume is on a device type supported by the BUILDIX command, and change the JCL or command statement.

ICK01502I BUILDIX FUNCTION STARTED

Explanation: Execution of the BUILDIX command has begun.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK01503I **ccuu REQUEST RECEIVED TO CONVERT VTOC TO **FORMAT**

Explanation: This message verifies the request for a change of VTOC format. '**' specifies the format requested—either OS or IX.

System Action: Execution of the command continues.

Operator Response: None.

Programmer Response: None.

ICK01504I **ccuu VTOC FORMAT IS CURRENTLY **FORMAT, REQUEST ACCEPTED**

Explanation: This message verifies that the BUILDIX function specified on the command statement is valid for the current format of the VTOC on the volume. '**' identifies the format of the VTOC, either OS or IX.

System Action: Execution of the command continues.

Operator Response: None.

Programmer Response: None.

ICK01508A **ccuu SHOULD VTOC CONVERSION PROCEED? REPLY 'U' TO CONTINUE, ELSE 'T'**

Explanation: This message permits the operator to verify that the user is authorized to request the BUILDIX function before execution of the command begins.

System Action: The system waits for the operator's reply.

Operator Response: Verify that the user is authorized to issue the command, and reply 'U' to continue or 'T' to terminate. Any reply other than 'U' or 'T' causes this message to be issued again.

Programmer Response: None.

ICK01513I **ccuu BUILDIX PROCESSING COMPLETED: VTOC IS NOW IN **FORMAT.**

Explanation: The BUILDIX command completed successfully. '**' shows the new VTOC format, either OS or IX.

System Action: Device Support Facilities is terminated normally.

Operator Response: None.

Programmer Response: None.

ICK01600I **MAPALT STARTED**

Explanation: Execution has started for the MAPALT command.

System Action: None.

Operator Response: None.

ICK01606I **nnnnn BLOCKS ASSIGNED TO ALTERNATES IN LIMITS SPECIFIED**

Explanation: This summary message shows the number of blocks assigned alternates, on this or previous runs, within the limits specified on the command statement.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK01608I **ccuu MAPALT ENDED NORMALLY, RETURN CODE= n**

Explanation: Execution of the command ended without any errors detected that would have caused termination of execution. A return code of 0 or 4 will be indicated in the message. A return code of 0 means no errors were encountered. A return code of 4 means one or more recoverable errors were encountered.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK01609I **PERMANENT DATA CHECK FOUND READING ID FOR BLOCK nnnnnn**

Explanation: A permanent data check was encountered while attempting to read the ID field of the primary block shown in the message.

System Action: Diagnostic information is printed on the output device, and execution of the command continues.

Operator Response: None.

Programmer Response: Save the job output for the system coordinator, and follow your installation,s procedures for data recovery. If this message is received for up to 3 blocks, the Device Support Facilities INSPECT command can be used to assign an alternate block for the block(s) experiencing the data checks. If this message is received for more than 3 blocks, contact your IBM Customer Engineer or Service Representative.

Problem Determination: Table I, items 1, 2, 3, 4, 30.

ICK01701I **ONLY 5 USER LABELS ALLOWED**

Explanation: Only five user volume labels are allowed for fixed block architecture devices. In the LABEL parameter, you have asked for space for more than five labels.

System Action: Space is reserved for six labels, and processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01704I **ALTERNATE ASSIGNED FOR BLOCK xxxxxx**

Explanation: Block xxxxxx was found defective during surface analysis. The block has been assigned an alternate.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01708I **RECLAIM SUCCESSFUL FOR BLOCK xxxxxx**

Explanation: BLOCK xxxxxx has been successfully reclaimed.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01714I RECLAIM UNSUCCESSFUL FOR BLOCK xxxxxx

Explanation: The defective block xxxxxx failed surface analysis and could not be reclaimed.

System Action: An alternate is assigned to the block. Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01715I TOTAL NUMBER OF ALTERNATES ASSIGNED = number

Explanation: Gives the total number of alternate blocks assigned in this run.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01725I EXCESSIVE NUMBER OF ALTERNATES ASSIGNED FOR A CYLINDER

Explanation: During initialization of a 3370 volume, more than 24 alternates were assigned on one cylinder. This is more than the number of alternates on a cylinder.

System Action: Processing continues.

Operator Response: None.

Programmer Response: This could cause performance degradation on the volume. Hardware support may be called for.

Problem Determination: Table I, items 1, 2, 3 and 4.

ICK01726I nnnnnn FACTORY FLAGGED BLOCKS.

Explanation: Final message of INIT reclaim processing. nnnnnn is the number of factory-flagged blocks assigned an alternate.

System Action: INIT processing surface analysis phase is started

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01727I RECLAIM SPECIFIED WITH NOCHECK. NO RECLAIM DONE

Explanation: RECLAIM function was not performed, because NOCHECK was specified or defaulted. CHECK is required for RECLAIM.

System Action: Processing continues without RECLAIM.

Operator Response: None.

Programmer Response: Specify CHECK and rerun your job if RECLAIM is wanted.

Problem Determination: Not applicable.

ICK01729I SPECIFIED CISIZE ADJUSTED TO NEXT HIGHER MULTIPLE OF DEVICE BLOCKSIZE

Explanation: When you specify a CISIZE which is not a multiple of 512, Device Support Facilities rounds the value up to the next higher multiple.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01730I RECLAIM AND/OR CHECK NOT SUPPORTED FOR FBA MINI-DISKS

Explanation: The CHECK and/or RECLAIM function requested is not supported for fixed block architecture (FBA) minidisks.

System Action: Processing continues without CHECK or RECLAIM.

Operator Response: None.

Programmer Response: None. If CHECK or RECLAIM is needed, the full volume must be initialized.

Problem Determination: Not applicable.

ICK01731I MAP FUNCTION NOT SUPPORTED FOR MINI-DISKS

Explanation: The MAP parameter was specified on the command, but MAP is not supported for fixed block architecture (FBA) minidisks.

System Action: Processing continues without MAP function.

Operator Response: None.

Programmer Response: None

Problem Determination: Not applicable.

ICK01732I MAP FUNCTION NOT SUPPORTED FOR THIS DEVICE TYPE

Explanation: The MAP parameter was specified or defaulted on a device where MAP is not supported.

System Action: Processing continues bypassing the MAP function.

Operator Response: None.

Programmer Response: None

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK01754I xx OF THE INSPECTED BLOCKS HAVE ALTERNATES ASSIGNED

Explanation: This is the summary message printed when MAP is specified. xx is the number of blocks inspected that have alternates assigned.

System Action: None.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01759I **USER DATA ON BLOCK xxxxxx RESTORED**

Explanation: Processing of the command failed with a permanent error, and Device Support Facilities was able to restore the data on block xxxxxx before terminating the command.

System Action: The command is terminated because of a previously reported error. Processing continues with the next command.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01760I **SURFACE OF BLOCK xxxxxx DEFECTIVE**

Explanation: The check function found the surface of block xxxxxx defective.

System Action: If ASSIGN is specified, an alternate is assigned.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01761I **SURFACE OF BLOCK xxxxxx NOT DEFECTIVE**

Explanation: The check function found the surface of block xxxxxx not defective.

System Action: Processing continues with the next block.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01765I **NEW ALTERNATE ASSIGNED TO BLOCK xxxxxx**

Explanation: Block xxxxxx has been assigned a new alternate.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01781I **dataset IS A PASSWORD PROTECTED DATA SET BUT USER EXIT DIRECTS BYPASS CHECK**

Explanation: The user security exit module has directed Device Support Facilities to bypass password verification of the named password-protected data set on the volume.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01795I **TRACK x'ccchhhh' IS CONTAINED IN DATA SET** dataset

Explanation: The track specified in the INSPECT command is contained in the named data set. This message may be followed by others such as ICK31780I.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01832I **PATH xxxx,y HAS BEEN WRITE ALLOWED**

Explanation: This in an informational message indicating to the operator and the programmer which path ids have been write-allowed by the CONTROL command. 'xxxx' is the unit address, 'y' is the path ID (CHPID).

System Action: Processing continues.

Operator Response: All devices on that path have been cleared. However, the operator must vary the required paths back online before they are accessible.

Programmer Response: None.

Problem Determination: Not applicable.

ICK01833I **DEVICE ccuu FENCE STATUS CLEARED**

Explanation: This is an information message indicating that the fence status for the device *ccuu* has been cleared by the CONTROL command.

System Action: Processing continues

Operator Response: None

Programmer Response: None

Problem Determination: Not applicable

ICK02100I **LAST INVOCATION CHECKPOINTED AT X'cccc hhhh', RECOVERY IN PROCESS**

Explanation: A previous use of the INIT command did not complete. The track specified in this message is the last checkpointed location.

System Action: Depending on the device type and the nature of the failure of the previous use of the INIT command, appropriate action is taken to ensure that the track that was being processed at the time of the failure is returned to its proper condition.

After the recovery process is complete, processing begins for this use of the INIT command.

Programmer Response: None

Operator Response: None

ICK02101I **INITIALIZE IS CONTINUING FROM TRACK X'cccc hhhh'.**

Explanation: CONTINUE data existed when INITialize was invoked. X'cccc hhhh' indicates the last checkpointed track.

System Action: Processing begins at track x'cccc hhhh' for the remainder of the CURRENT specified range.

ICK02103I **INITIALIZE IS CONTINUING FROM BLOCK xxxxxx.**

Explanation: CONTINUE data existed when INITialize was invoked. xxxxxx indicates the last checkpointed block.

System Action: Processing begins with block xxxxxx for the remainder of the CURRENT specified range.

ICK02104I **NO ADDITIONAL TRACKS REMAIN IN THE SPECIFIED RANGE**

Explanation: CONTINUE processing was previously activated. (ICK02101I) After verification of the tracks necessary to ensure the validity of the volume, there were no additional tracks to process in the specified range.

System Action: Processing continues with the minimal initialization functions.

ICK02105I **PROCESSING IS CONTINUING FROM LAST CHECKPOINT**

Explanation: The previous invocation of this command did not complete processing. The current command has detected that situation and will continue processing from the last checkpoint.

System Action: The command process continues from the last checkpoint.

Programmer Response: None.

Operator Response: None.

ICK02129I **RECOVERY COMPLETE FOR TRACK X'cccc hhhh'.**

Explanation: The backup recovery function is complete for the specified track.

System Action: Processing continues with the current invocation parameters.

Programmer Response: None.

ICK02150I **INITIALIZE IS CONTINUING WITH RECLAIM PROCESSING**

Explanation: A previous use of the INIT command failed during reclaim processing and the reclaim process is being restarted.

System Action: The reclaim process is restarted.

Operator Response: None.

Programmer Response: None.

ICK02156I **INITIALIZE PREVIOUSLY INTERRUPTED AT BLOCK xxxxxx**

Explanation: A previous use of the INIT command did not complete. The block specified in the message is the last checkpointed location.

System Action: The CONTINUE data and specified parameters are examined to determine what processing should be done.

Operator Response: None.

Programmer Response: None.

ICK02163I **CONTINUE INFORMATION EXISTS ON CE CYLINDER FOR BLOCK xxxxxx**

Explanation: During ANALYZE DRIVETEST processing, it was determined that a previous invocation of the INIT command did not execute to completion. xxxxxx represents the last block for which a checkpoint was taken.

System Action: This information is not erased. ANALYZE processing continues with the remainder of the drive test.

Programmer Response: This message is presented for informational purposes. The INIT command can be restarted if necessary, with CONTINUE or NOCONTINUE specified, depending upon the circumstances surrounding the previous INIT interruption.

ICK02164I **PRESERVE INFORMATION EXISTS ON CE CYLINDER FOR BLOCK xxxxxx**

Explanation: During ANALYZE DRIVETEST processing, it was determined that the PRESERVE function of the INSPECT command did not execute to completion for the specified block. Data has been saved for this block.

System Action: This information is not erased. ANALYZE processing continues with the remainder of the drive test.

Programmer Response: Execute the INSPECT command for this device to ensure the usability of the specified block, and to recover the data.

ICK02166I **RECOVERY COMPLETE FOR BLOCK xxxxxx**

Explanation: The backup recovery function is complete for the specified block.

System Action: Processing continues with the current invocation parameters.

Programmer Response: None.

ICK02174I **VOLUME CONTAINS xxxxx ALTERNATE TRACKS -- AVAILABLE ALTERNATES UNDETERMINED**

Explanation: The VTOC for this volume cannot be read, or does not exist. Therefore, the current number of available alternate tracks cannot be determined. Previous message(s) indicate why the VTOC is inaccessible. xxxxx is the total number of alternate tracks on the volume.

System Action: Processing continues

Programmer Response: Previous messages indicate why the VTOC is inaccessible.

Operator Response: None

ICK10705I **VOLUME SERIAL NUMBER FOR DEVICE ccuu IS xxxxxx**

Explanation: Informational message concerning the volume serial and/or VTOC of the volume at ccuu. If the volume serial was changed, you also receive one or both of the following:

CHANGED FROM xxxxxx - The volume serial number of the ccuu was changed.

The following message could also be received:

VOLUME SERIAL DUPLICATE FOR DEVICE ccuu. VOLUME MADE UNAVAILABLE - The new volume serial on the ccuu is a duplicate of one already known to the operating system. The device has been unloaded.

If the VTOC location was changed, you also receive the following:

VTOC LOCATION MOVED - The VTOC location of the volume at *ccuu* has moved.

System Action: Device Support Facilities continues processing.

Operator Response: If the device is being shared by other systems, this volume may need to be remounted at the sharing systems.

Programmer Response: None.

Problem Determination: Not applicable.

ICK10710I I/O ERROR OCURRED ON DEVICE ccuu

Explanation: An I/O error occurred on the device whose address is specified by ccuu. The nature of the I/O error is described on the next two lines, where the failing channel command word (CCW), the channel status word (CSW), and the sense bytes are printed.

System Action: The command continues processing. Eventual termination of the command may occur, but such a termination is generally followed with a terminating message.

Operator Response: None.

Programmer Response: Make sure that the problem is, indeed, with the device. Correct the device problem, and retry the command.

ICK11005I VTOC DOES NOT EXIST

Explanation: A volume table of contents was not found on the volume during execution of the command.

System Action: A volume table of contents was created, and the command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK11009I IPL PROGRAM SUPPLIED FOR MSS -- IGNORED

Explanation: A volume being formatted as an MSS staging pack cannot contain IPL records.

System Action: The IPL program specified is ignored, and the command continues processing.

Operator Response: None.

Programmer Response: Remove the IPLDD parameter from the command.

Problem Determination: Not applicable.

ICK11010I UNABLE TO CLOSE VOLUME

Explanation: A system error occurred that prevents the volume from being closed properly.

System Action: The command probably did not complete successfully because the volume did not close when the command completed. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK11019I NEXT-AVAILABLE-ALTERNATE POINTER OR COUNTER IN VTOC IS INVALID

Explanation: The value of the pointer to the next available alternate track or the count of the number of alternate tracks that are available is in error.

System Action: Processing continues. If it is possible, the value in the VTOC is updated when processing completes.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK11038I NO VALID TRACKS WERE SPECIFIED

Explanation: If the TRACKS parameter specifies only invalid track addresses, this message indicates that the command was not processed.

System Action: The command terminates. However, PRESERVE data is processed before termination. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command, and specify one or more valid track addresses.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK11050I VTOC LOCATION SPECIFIED FOR MSS STAGING PACK -- IGNORED

Explanation: You cannot specify a VTOC location for a Mass Storage System staging pack.

System Action: The VTOC parameter is ignored, and the command continues processing.

Operator Response: None.

Programmer Response: Remove the VTOC parameter from the command.

Problem Determination: Not applicable.

ICK11051I LABELS PARAMETER SPECIFIED FOR MSS STAGING PACK -- IGNORED

Explanation: You cannot place user volume labels on a Mass Storage System staging pack.

System Action: The LABELS parameter is ignored, and the command continues processing.

Operator Response: None.

Programmer Response: Remove the LABELS parameter from the command.

Problem Determination: Not applicable.

ICK11065I **DATA PRESERVED FOR TRACK CCHH=X'cccc hhhh' ON ALTERNATE TRACK CCHH=X'cccc hhhh'**

Explanation: This message follows message ICK21047I if the preserved data is successfully written to an alternate track.

System Action: Processing continues with the next track. The return code is set to 4.

Programmer Response: Further inspection of the failing track might be required to determine the reason for the failure.

Problem Determination: Not applicable.

ICK11095I **UNABLE TO READ VOLUME LABEL - PROCESS CONTINUING**

Explanation: An inspect has been requested on a volume that does not contain a volume label.

System Action: Processing of the command continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK11130I **CAN NOT DETACH LINKED DEVICE ccuu RC= nnnn**

Explanation: The DIAGNOSE 08 allows the user program to issue a CP command. The 'DETACH' command is used to detach the specified address that was full-pack overlay linked by ICKDSF during the media maintenance processing. The operation failed with CP return code nnnn.

System Action: The command terminates. ICKDSF continues with the next command.

Programmer Response: Refer to *CP Command Reference for General Users* for more information on DETACH.

Operator Response: None.

ICK11306I **NO STORAGE AVAILABLE FOR PACK MAP**

Explanation: During initialization or inspection of a volume, storage is dynamically acquired for data elements that are collected later to format the pack map. A data element was being allocated, but storage was not available.

System Action: The command is completed, but without a pack map.

Operator Response: None.

Programmer Response: Specify a larger region size, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK11315I **UNABLE TO READ VOLUME LABEL**

Explanation: An I/O error occurred when attempting to read the volume label. Because the number of user volume labels cannot be determined, the IPL program record could be written over existing records.

System Action: The command continues processing, but does not write the IPL program records.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK11328I **SKIP DISPLACEMENTS CHANGED FOR TRACK X'cccc hhhh'**

Explanation: Skip displacement surface checking detected at least one area on the track that is potentially defective. The defective area(s) have been skipped. The surface of the track is defect free if no subsequent messages are issued.

System Action: Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK11329I **SURFACE ANALYSIS CYLINDER COULD NOT BE READ FOR TRACK X'cccc hhhh'**

Explanation: The surface analysis cylinder is accessed for skip displacement devices when there is a need to restore the skip displacement information on a track to the factory level.

This message can occur if there is no factory map information on the surface analysis cylinder, or if an uncorrectable I/O error occurred.

The CCW, CSW, and sense information that caused the error is printed.

System Action: Depending upon the input track condition and the device type being processed, the home address may be rewritten without any factory skip displacement information before surface checking of the track occurs. If the factory skip displacement information might be overridden, subsequent messages will be issued.

Problem Determination: Because the usability of this track is not affected, there is no need to examine the information any further. If there is concern, the failing CCW, CSW, and sense information can be examined in detail.

ICK11390I **INVALID DATA ON SA CYLINDER**

Explanation: The data on the surface analysis cylinder for this volume (which contains skip displacement information for this volume) is not in the expected format.

System Action: This is an informational message only. ICKDSF will provide other messages for any other actions that it might take as a result of this situation. Processing continues.

Operator Response: None.

Programmer Response: None.

Problem Determination: Not applicable.

ICK11392I **SKIP DISPLACEMENT DATA DEFAULTED FOR TRACK X'cccc hhhh'**

Explanation: The skip displacement data for the specified track could not be read.

System Action: The track is processed as though there were no skip displacements assigned to it, and a complete surface check is done on this track.

New skip displacements are assigned as required.

Note that if there had been factory assigned skip displacements for this track, they have been reset.

Operator Response: None.

Programmer Response: None. When ICKDSF processing completes for this track, the specified condition of the track (that is, not defective or defective) is valid.

Problem Determination: Not applicable.

ICK11411I **ccuu INCORRECT DEVICE TYPE FOR DRIVE TEST, DRIVE TEST BYPASSED**

Explanation: The device type for the volume indicated by device 'ccuu' is not for a fixed media device, but drive test was specified (or defaulted).

System Action: ANALYZE processing continues with the data verification test. The ANALYZE return code is set to 4.

Operator Response: None.

Programmer Response: See Figure 2 for valid device types for the drive test (ANALYZE command).

Problem Determination: See "Programmer Response."

ICK11412I **DRIVE TEST NOT SUPPORTED FOR MINIDISK, DRIVE TEST BYPASSED**

Explanation: You specified the ANALYZE DRIVETEST function which is not supported for minidisks.

System Action: The drive test is bypassed.

Programmer Response: none

Operator Response: none

ICK11413I **MAIN STORAGE NOT AVAILABLE FOR WRITE/READ TESTS ON CE CYLINDER**

Explanation: Storage is not available for CCW and work areas.

System Action: The write and read test portion of the drive test is terminated. If the data scan was requested, the command continues with the scan processing.

Operator Response: Increase storage size and resubmit the job.

Programmer Response: None.

ICK11414I **WRITE INHIBIT SWITCH ON, WRITE TESTS BYPASSED**

Explanation: An error was detected when the ANALYZE drive test attempted to write on the CE cylinder and the device was in a "write inhibited" condition.

System Action: All tests that attempt to write on the CE cylinder are bypassed, but processing continues.

Operator Response: If the device has an "R/W" or "READ" switch, check that the switch is in the "READ/WRITE" position, and/or is functioning properly.

This condition can also exist if the storage control has been 'write inhibited' by the operating system. See the explanation of the CONTROL command to determine further action.

ICK11417I **HOME ADDRESS READ FROM C.E. CYLINDER IS INCORRECT: SHOULD BE X'cccc hhhh', IS X'cccc hhhh'.**

Explanation: The cylinder and head information in the home address for the specific track on the C.E. cylinder is incorrect.

System Action: ANALYZE attempts to rewrite the home address with the correct cylinder and head information. If the rewrite is successful, processing continues with the write tests. If the rewrite is unsuccessful, this message is followed by the failing CCW, CSW, and sense, and an indication that the write tests will be bypassed. Processing continues, bypassing the write tests.

Operator Response: None.

Programmer Response: If ANALYZE was being run because of a suspected write problem, and the rewrite of the home address on the C.E. cylinder is unsuccessful, further investigation of the situation is required.

Problem Determination: See "Programmer Response."

ICK11741I **DATA CHECK ON ID OR DATA FIELD ORIENTATION ADDR = nnnnn.**

Explanation: This message is printed for each block that fails the data verification test phase of ANALYZE with a data check that indicates that data synchronization on the block's ID or data field was unsuccessful. nnnnn is the relative block number in error. This message is followed by a printout of the failing CCW, CSW and sense information.

System Action: ANALYZE processing continues.

Operator Response: None.

Programmer Response: Normally, this error is an indication of a media surface problem that can be resolved either by assigning an alternate to the failing block with the INSPECT command, or by using the INIT command with the CHECK parameter to reinitialize the FBA device.

Note: A large number of these messages occurring for a newly installed device may be an indication of an incorrectly formatted device. In this case, hardware assistance will probably be necessary.

Problem Determination: See "Programmer Response."

ICK11743I **DATAVER UNCORRECTABLE ERROR ADDR BLOCK nnnnn**

Explanation: This message is printed for each block that fails the data verification test phase of ANALYZE with an ECC uncorrectable data check. This message is followed by a printout of the failing CCW, CSW, and sense information. nnnnn is the relative block number of the failing block.

System Action: ANALYZE processing continues. The return code is set to a 4.

Operator Response: None.

Programmer Response: Normally, this error is an indication of a media surface problem that can be resolved either by assigning an alternate to the failing block with the INSPECT command, or by using the INIT command with the CHECK parameter to reinitialize the FBA device. If repeated attempts to resolve the problem using the INSPECT or INIT command fail, hardware support should be sought to aid in resolving the problem.

Problem Determination: See "Programmer Response."

ICK11745I **DATAVER CORRECTABLE ERROR ADDR BLOCK = nnnnn**

Explanation: This message is printed for each block that fails the data verification test phase of ANALYZE with an ECC correctable data check. This message is followed by a printout of the failing CCW, CSW, and sense information. nnnnn is the relative block number of the failing block.

System Action: ANALYZE processing continues. The return code is set to a 4.

Operator Response: None.

Programmer Response: Normally, this error is an indication of a minor media surface defect and should not be considered a problem if the device is being used with systems such as VSE and VM, which have error recovery procedures that do ECC correction. The performance impact of fixing the data in storage is less than that of the device reading the data from an alternate block. If it is deemed necessary to eliminate this type of error, the INSPECT command can be used to unconditionally assign an alternate to the failing block.

Note: If an excessive number of these errors occur, hardware support should be sought to aid in determining if a hardware problem exists.

Problem Determination: See "Programmer Response."

ICK11752I BLOCK xxxxxx OUTSIDE DEVICE LIMITS

Explanation: Block number xxxxxx specified by the BLOCKS parameters is invalid.

System Action: The command continues. The invalid block number is ignored.

Operator Response: None.

Programmer Response: Resubmit the job, specifying the correct block number in the BLOCKS parameter. See "BLOCKRANGE|BLOCKS: Specifying Which Blocks to Inspect" on page 207.

Problem Determination: Table I, items 1, 3, and 4.

ICK11782I DATAVER UNCORRECTABLE ERROR ADDR CCHH X'cccc hhhh'

Explanation: This message is printed for each track that fails the data verification test with an apparent ECC uncorrectable data check. If the error is of the type that could have been corrected by storage control retry, or if the record that experienced the error contained two non adjacent ECC correctable data checks, this error would appear as ECC uncorrectable (including correctable errors in the key or data fields subsequent to record 1). In either of the above cases, subsequent I/O against the same data might perceive the error as correctable. This message is followed by a printout of the failing CCW, CSW, and sense information. cccc hhhh is the hexadecimal cylinder and head address of the failing track.

System Action: Processing continues. The return code is set to a 4.

Operator Response: None.

Programmer Response: Normally, this error is an indication of a media surface problem that can be resolved by either performing a skip displacement to skip the defect, or (if neccessary) assigning an alternate track. You can use the INSPECT command to perform skip displacement or assign an alternate track, or you can use the INIT command with the CHECK parameter to reinitialize the CKD device.

Note: If repeated attempts to resolve the problem using the INSPECT or INIT command fail, hardware support should be sought to aid in resolving the problem.

Problem Determination: See "Programmer Response."

ICK11784I DATAVER CORRECTABLE ERROR ADDR CCHH ='cccc hhhh'

Explanation: This message is printed for each track that fails the data verification test phase with an ECC correctable data check. This message is followed by a printout of the failing CCW, CSW, and sense information. cccc hhhh is the hexadecimal cylinder and head address of the failing track.

System Action: Processing continues. The return code is set to 4.

Operator Response: None.

Programmer Response: Normally, this error is an indication of a media surface problem that can be resolved by skip displacing over the error with either the INSPECT command, or by using the INIT command with the CHECK parameter to reinitialize the CKD device. This error is an indication of a minor media surface defect and need not be skip displaced if the device is being used with any IBM operating system (which have error recovery procedures that do ECC correction). After skip displacement of the error is performed, no performance impact is experienced, and all ECC correctable errors are gone from the track.

Note: If repeated attempts to resolve the problem using the INSPECT or INIT command fail, hardware support should be sought to aid in resolving the problem.

Problem Determination: See "Programmer Response."

ICK11826I DATAVER INVALID TRACK FORMAT ON CCHH X'cccc hhhh'

Explanation: An invalid track format condition was detected when scanning data on the specified track. This is usually an indication that data was written to the track beyond the track capacity. It is generally a user error.

System Action: Processing continues. At the completion of the command, the return code is set to 4.

Operator Response: None.

Programmer Response: Determine the data set containing the track specified. Determine how, where, and when the data on the specified track was written, in order to determine how the invalid track format condition occurred.

Problem Determination: See "Programmer Response."

ICK12105I NO BLOCKS REMAIN IN THE SPECIFIED RANGE

Explanation: CONTINUE processing was previously activated. (ICK02103I) There are no blocks to process in the specified range subsequent to the last checkpointed location.

System Action: Processing continues with the minimal initialization functions.

ICK12107I CONTINUE FUNCTION CANCELLED

Explanation: The continue function of the INITialize command has been cancelled. The reason for the cancellation has been presented in previous message(s).

System Action: Processing continues for the specified range. Checkpointing no longer is taking place.

Programmer Response: If processing completes normally, this message can be disregarded.

If processing does not run to completion for FBA devices, this message can be disregarded unless RECLAIM is specified. For FBA devices where RECLAIM is specified, the RECLAIM job should be restarted.

For CKD devices that do not run to completion, there may exist a track on the device that contains an invalid format.

After the reason for the abnormal termination has been resolved, at least a medial INIT should be run for the entire volume.

Note: Subsequent invocations of the INIT command may or may not experience the same error, and may or may not attempt to CONTINUE from a previously recorded location. This will depend on the reason that the function was cancelled, as well as the nature of the failure. If multiple initializes are necessary, NOCONTINUE can be specified to ensure that processing never resumes from an unwanted location.

As long as processing successfully completes for each invocation, the persistence of this message can be ignored.

ICK12108I CONTINUE TRACK CANNOT BE USED

Explanation: The track normally used to contain checkpoint information is in use by this device. This message is followed by ICK12107I.

System Action: See ICK12107I.

Programmer Response: See ICK12107I

ICK12109I CONTINUE DATA COULD NOT BE RESET

Explanation: An error occurred while checkpoint data was being written. This message is followed by ICK12107I.

System Action: See ICK12107I.

Programmer Response: See ICK12107I

ICK12115I DATA BEING RECOVERED FOR TRACK X'cccc hhhh'

Explanation: The previous INSPECT command did not complete execution during PRESERVE backup processing.

System Action: Primary surface checking functions are to be done for the specified track, and data is recovered if data exists.

ICK12116I PRESERVE DATA CANNOT BE BACKED UP FOR THIS DEVICE TYPE

Explanation: The PRESERVE recovery function is not supported for this device type.

System Action: Processing continues as for HOLDIT.

ICK12117I PRESERVE RECOVERY TRACK CANNOT BE USED. USE 'HOLDIT'

Explanation: The track normally used to contain recovery information is in use by this device and PRESERVE was specified.

System Action: Processing terminates.

Programmer Response: Resubmit the job, specifying HOLDIT instead of PRESERVE or KEEPIT.

ICK12118I ERROR READING BACKUP TRACK

Explanation: An I/O error occurred while INSPECT was determining the existence of recovery data for this volume.

System Action: The CCW, CSW and sense information are printed. This message is followed by another message describing the system action.

ICK12119I PRESERVE BACKUP DATA IGNORED

Explanation: This message follows ICK12118I if HOLDIT is specified. Any recovery data that might exist is left intact, and processing continues.

System Action: Processing continues.

ICK12123I PRESERVE BACKUP DATA CANNOT BE ESTABLISHED OR RESET

Explanation: An I/O error was encountered while performing the PRESERVE backup function.

System Action: If the data was being reset, this message is issued as a warning. Processing continues.

ICK12124I ERROR TRYING TO ACCESS RECOVERY TRACK X'cccc hhhh' --RECOVERY IN PROGRESS

Explanation: An I/O error was encountered trying to determine the current disposition of the track we are attempting to recover.

System Action: Standard ICKDSF track recovery operations take place for this track. If recovery is successful, the backup recovery process continues.

Programmer Response: If track recovery is unsuccessful, subsequent messages are issued.

ICK12125I NO PRESERVE BACKUP DATA EXISTS FOR TRACK X'cccc hhhh'

Explanation: The track backup recovery is attempting to recover is unrecoverable. Standard ICKDSF track recovery operations failed for this track. However, no data exists for that track.

System Action: Processing continues as if backup recovery is complete.

Programmer Response: Standard installation procedures should be followed regarding unrecoverable tracks.

ICK12126D DATA ALREADY EXISTS FOR TRACK X'cccc hhhh' REPLY 'R' TO RECOVER, 'E' TO ERASE THE RECOVERY DATA, OR 'T' TO TERMINATE

Explanation: This message is issued if recovery data exists for a track (ICK2115I), but there already exists data on that track.

This situation might occur if the original data on the track had not yet been erased at the point of failure.

This situation can occur if an INSPECT did not run to completion, but the volume was available for user use before this invocation of the INSPECT command.

System Action: The operator is prompted for a reply to this message.

If 'r' is specified, the data will be recovered from the recovery information. (The new data on the track is erased).

If 'e' is specified, the recovery data is destroyed. The current data on the track remains.

If 't' is specified, processing terminates. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK12128I RECOVERY DATA HAS BEEN ERASED

Explanation: This message is issued if you reply 'c' to ICK22158D or ICK22130D, or repiy 'e' to ICK12126D or ICK12159D.

ICK12130I UNABLE TO RESET RECOVERY DATA

Explanation: An I/O error occurred while the command checkpoint data is being updated to indicate that command processing is complete.

System Action: Processing continues. All required functions have been completed.

Programmer Response: The volume should be used as normal. The I/O error occurred on a track that is not a user track, and this message can be ignored. Subsequent invocations of the INIT command will issue a warning message. If desired, take action appropriate to your installation's procedures for handling the I/O error problems. If the problem cannot be resolved call your service representative.

Operator Response: None.

ICK12151I RECLAIM PROCESSING PREVIOUSLY INTERRUPTED, RECLAIM FORCED

Explanation: A previous use of the INIT command failed during reclaim processing and the RECLAIM parameter has not been specified with the CONTINUE parameter. Reclaim processing is being forced to ensure that factory defects are properly flagged and that all primary and alternate pairs are properly connected.

System Action: The reclaim processing is restarted.

Operator Response: None.

Programmer Response: None.

ICK12153I BLOCKRANGE IGNORED

Explanation: The BLOCKRANGE specification is being ignored because a previous use of the INIT command with the RECLAIM parameter failed. Full volume processing is forced to insure the data integrity of the device.

System Action: Surface analysis will continue from the last checkpointed block to the end of the volume.

Operator Response: None.

Programmer Response: None.

ICK12157I DATA BEING RECOVERED FOR BLOCK xxxxxx

Explanation: The previous invocation of the INSPECT command did not execute to completion during PRESERVE processing.

System Action: Surface checking functions will be completed for the specified block, and data will be be recovered.

ICK12159D DATA ALREADY EXISTS FOR BLOCK xxxxx REPLY 'R' TO RECOVER, 'E' TO ERASE THE RECOVERY DATA, OR 'T' TO TERMINATE

Explanation: This message is issued if recovery data exists for a block (ICK12157), but there already exists data on that block that is not ICKDSF data and is not the original user data.

This situation can occur if an INSPECT did not run to completion, but the volume was available for user use before this invocation of the INSPECT command.

System Action: The operator is prompted for a reply to this message.

If 'r' is specified, the data will be recovered from the recovery information. (The new data on the block is erased).

If 'e' is specified, the recovery data is destroyed. The current data on the block remains.

If 't' is specified, processing terminates. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK12161I ERROR READING RECOVERY BLOCK - DATA IGNORED

Explanation: The block which backup recovery is attempting to recover currently cannot be read successfully.

System Action: Whatever data currently exists on the block is ignored, and normal surface checking procedures are done for this block.

Programmer Response: None

ICK12162I ERROR READING BACKUP BLOCK

Explanation: An I/O error occurred while INSPECT was determining the existence of recovery data for this volume.

System Action: The CCW, CSW and sense information are printed. This message is followed by another message describing the system action.

ICK12168I SPEED IGNORED BECAUSE HEADRANGE SPECIFIED

Explanation: Because SPEED operates on a cylinder at a time, it is not valid with HEADRANGE, and is ignored if both HEADRANGE and SPEED is specified..

System Action: Processing continues with NOSPEED.

Programmer Response: None.

Operator Response: None.

ICK12171I PRESERVE DATA BEING ERASED FOR TRACK 'cccc hhhh'

Explanation: During execution of the INIT command, it has been determined that a previous use of the INSPECT command did not run to completion. Because this is an initialize, it is assumed that the data saved from that track (cccc hhhh) is no longer needed.

System Action: The preserved data is erased, and processing continues.

Programmer Response: None.

Operator Response: None.

ICK12172I PRESERVE DATA BEING ERASED FOR BLOCK xxxxxx

Explanation: During execution of the INIT command, it has been determined that a previous use of the INSPECT command did not run to completion. Because this is an initialize, it is assumed that the data saved by block xxxxxx is no longer needed.

System Action: The preserved data is erased, and processing continues.

Programmer Response: None.

Operator Response: None.

ICK12173I WARNING PREVIOUS PROCESS DID NOT COMPLETE

Explanation: Checkpoint data has been detected that indicates that an INSTALL or REVALidate command did not complete processing. This could be an indication that the volume is in an unusable state.

System Action: This message is followed by ICK12107. The CONTINUE function is cancelled. Processing continues.

Programmer Response: If this message is issued during a medial initialization the condition will be corrected and this message can be ignored. Otherwise, it is recommended that a command is run to ensure the device is returned to a usable condition (for example, medial initialization or REVALidate). If it can be determined that message ICK12130 was issued during by a previous command, this message can be ignored.

Operator Response: None

ICK12200I ENQUEUE FAILED ON TRACK X'cccc hhhh'. PROCESS CONTINUING

Explanation: The enqueue for the data set which contains track x'cccc hhhh' was unsuccessful. Because TOLERATE(ENQFAIL) was specified, processing continues.

System Action: INSPECT continues on this track.

Programmer Response: None.

ICK12202I INSUFFICIENT STORAGE AVAILABLE FOR DATA SET ENQUEUE -- TOLERATE(ENQFAIL) IS IN EFFECT FOR ALL TRACKS

Explanation: GETMAIN failed while obtaining the storage necessary to process the data set enqueue procedures. Because TOLERATE(ENQFAIL) was specified, processing continues as if the enqueue failed.

System Action: INSPECT continues for all tracks without the enqueue function.

Programmer Response: None.

ICK12204I UNABLE TO ENQUEUE VSAM DATA SET FOR TRACK X'cccc hhhh', PROCESS CONTINUING

Explanation: The specified track is part of a VSAM data set. ICKDSF does not support data set enqueue for VSAM data sets. Because TOLERATE(ENQFAIL) is specified or assumed, processing continues for this track.

System Action: Processing continues for this track.

Programmer Response: None.

Operator Response: None.

ICK12316I UNABLE TO DETERMINE MINIDISK SIZE

Explanation: You specified the INIT command for a minidisk under CMS.The system support code for the INIT command does not exist in VM/SP, VM/HPO, or VM/XA or the minidisk is a T-disk. This message is followed by ICK033E.

System Action: The system requests you to specify the size of the minidisk.

Programmer Response: none

Operator Response: none

ICK20011I FUNCTION CANNOT BE EXECUTED. INSUFFICIENT MAIN STORAGE

Explanation: There is not enough virtual storage available to execute a command.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Increase the amount of virtual storage available to Device Support Facilities, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, and 29.

ICK20800I UNABLE TO SCRATCH dsname

Explanation: An error occurred while attempting to scratch the identified data set. A subsequent second-level message identifies the error.

System Action: The data set is not scratched. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: See the associated message.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK20801I ** NO VOLUME MOUNTED

Explanation: Volume not mounted.

System Action: The data set is not scratched. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command and make sure the correct volume is mounted.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK20802I ** PASSWORD VERIFICATION FAILED

Explanation: The data set to be scratched is password protected, and the operator did not supply the proper password.

System Action: The data set is not scratched. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command, and specify the correct password when prompted.

Problem Determination: Table I, items 1, 2, 3, 4, 25a, and 29.

ICK20803I ** DATA SET HAS NOT EXPIRED ON VOLUME volser

Explanation: The PURGE parameter was not specified, and the data set's retention period has not expired.

System Action: The data set is not scratched. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Specify the PURGE parameter if you want to destroy the data set, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, 25a, and 29.

ICK20804I **** PERMANENT I/O ERROR ON VOLUME**

Explanation: Because of an I/O error condition, the volume table of contents cannot be read.

System Action: The data set is not scratched. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Either restore the volume or initialize the volume offline.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK20805I **** UNABLE TO MOUNT VOLUME volser**

Explanation: An appropriate device was not available for mounting the volume.

System Action: The data set was not scratched. Device Support Facilities processing continues with the next command.

Operator Response: Ensure that a device is available for mounting, and reissue the command.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK20806I **** DATA SET WAS IN USE**

Explanation: The data set to be scratched was being used by another program.

System Action: The data set is not scratched, and the command continues processing.

Operator Response: Ensure that no other job is using the data set (do not specify DISP=SHR), and reissue the command.

Programmer Response: None.

Problem Determination: Table I, items 1, 2 3, 4, 25a, 25c, and 29.

ICK20810I **** INVALID RACF AUTHORIZATION**

Explanation: A RACF-protected data set resides on the volume. The volume cannot be purged, because the user does not have the correct authorization to scratch the data set.

System Action: The data set is not scratched, and the command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK20950I **INVALID FORMAT STRUCTURE**

Explanation: An element of one of the static text structures is incorrect. Probable program error.

System Action: The request to print a line is ignored. The command continues processing, but no output is printed.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK20951I **OUTPUT COLUMN SPECIFIED OUT OF RANGE**

Explanation: An output column specification is outside the allowed print line width, for example, beyond column 120. Probable program error.

System Action: This field and subsequent fields for the same line are ignored. The command continues processing, but no output is printed.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK20952I **EXCESSIVE FIELD LENGTH FOR BD OR PU CONV**

Explanation: A binary-to-decimal or packed-to-unpacked conversion length was specified greater than 15. Probable program error.

System Action: The default of 15 was used, and the command continues processing.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK20953I **A REDO SUB-STRUCTURE IS NESTED**

Explanation: A redo structure cannot be defined within a set of structures to be redone. Probable program error.

System Action: The current redo operation is terminated. All structures are treated only once.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 3, 4, and 29.

ICK20954I **STATIC TEXT ENTRY REQUESTED NOT IN MODULE**

Explanation: A request for a specific static text entry in a specified static text module could not be resolved. The static text index is incorrect, or the programmer has neglected to enter a message into the static text module. Probable program error.

System Action: The request is ignored, and the command continues processing.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 2, 3, 4, 26c, and 29.

ICK20955I **INVALID PACKED DECIMAL FIELD**

Explanation: A conversion request for packed-to-unpacked decimal encountered a digit that was not in the range of 0 to 9. Probable program error.

System Action: Conversion stops for the current request. The command continues processing without the packed-to-unpacked conversion.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21000I **ALTERNATE TRACK CCHH=X'cccc hhhh' FOUND UNRECOVERABLE**

Explanation: While checking the track surface, error conditions were encountered that prevented the alternate track's home address or record 0 from being rewritten to indicate that it is a defective track. Since the track cannot be marked defective, it cannot be used by the operating system.

System Action: If the cause of the unrecoverability of the track is a data error, the command runs to completion with an error return code. If the cause of the unrecoverability is anything other than a data error, command processing stops after detection of the error.

Operator Response: None.

Programmer Response: Prior messages should be examined (including the CCW, CSW, and sense that caused the I/O) to determine the cause of the failure. This message may be caused if the the read/write mode switch is set to READ mode, or some other condition is causing the inability of ICKDSF to write on the volume.

If running under VM (either stand-alone or an operating system version running under VM), this message can be an indication that a diagnostic or media maintenance function was attempted against a device that was a minidisk and/or LINKED to the userid. Diagnostic and media maintenance functions must be done on devices that are ATTACHed to the userid. See Appendix B, "VM Support" on page 227 for more information on VM support.

Assistance may be required from the IBM Customer Service Representative.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK21001I **PRIMARY TRACK CCHH=X'cccc hhhh' FOUND UNRECOVERABLE**

Explanation: While checking the track surface, error conditions were encountered which prevented the primary track's home address or record 0 from being rewritten, to indicate that it is a defective track. Since the track cannot be marked defective, it cannot be used by the operating system. This message may appear if, during execution of a command, the read/write mode switch is inadvertently set to READ mode.

System Action: Device Support Facilities processing continues, unless the unrecoverable track threshold is met (see message ICK31013I).

Operator Response: None.

Programmer Response: Prior messages should be examined (including the CCW, CSW, and sense that caused the I/O) to determine the cause of the failure. This message may be caused if the the read/write mode switch is set to READ mode, or some other condition is causing the inability of ICKDSF to write on the volume.

If running under VM, (either stand-alone or an operating system version running under VM), this message can be an indication that a diagnostic or media maintenance function was attempted against a device that was a minidisk and/or LINKED to the userid. Diagnostic and media maintenance functions must be done on devices that are ATTACHed to the userid. See Appendix B, "VM Support" on page 227 for more information on VM support.

Assistance may be required from the IBM Customer Service Representative.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK21002I **INVALID VTOC ORIGIN SPECIFICATION FOR TRACK CCHH=X'cccc hhhh'**

Explanation: The VTOC parameter specifies an invalid track location for the volume table of contents. The VTOC cannot begin on cylinder 0, track 0 or encroach on the alternate track area.

System Action: The command is terminated. Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Either accept the default location, or reissue the command with a correct location specified.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21003I **INVALID VTOC EXTENT SPECIFICATION: xxxx**

Explanation: The VTOC parameter specifies an invalid extent (number of tracks) for the VTOC area. The VTOC cannot have an extent of 0 or encroach on the alternate track area.

System Action: The command is terminated. Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Either accept the default extent, or reissue the command with a correct extent specified.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21007I **R-ZERO INCORRECT -- a CCHH=X'cccc hhhh' (X'cccc hhhh') b CCHH=X'cccc hhhh'**

Explanation: This message indicates the existence of an invalid primary/alternate track association on a volume. See Figure 17 on page 225 for an example of a proper association.

"a CCHH=cccc hhhh" indicates the cylinder/head of the primary track in error. If that track is flagged defective, "a" is set to "D"; otherwise, "a" is set to "N". The "cccc hhhh" in parentheses is the cylinder/head of the track pointed to in record zero of the primary track in error.

"b CCHH=cccc hhhh" indicates the cylinder/head for the alternate track in error. If that track is flagged defective, the "b" is set to "D"; otherwise, "b" is set to "N." See Figure 17 on page 225 for an example of a proper association.

If a primary track is detected that contains an unknown alternate track pointer (the cylinder/head in record zero does not indicate a valid track address), the "b CCHH=cccc hhhh" portion of the message is printed "N CCHH=FFFF FFFF".

System Action: If you are performing an INITialize command, processing stops. If you are performing an INSPECT command, processing continues with the next track.

Operator Response: None.

Programmer Response: For INITialize processing, this message is only issued for a minimal INIT. INITialize the volume at the medial level or INSPECT the primary or alternate track to to correct the problem.

For INSPECT processing, this message is issued when an inconsistency exists for a track, and ICKDSF is not permitted to fix the problem. Retry the problem with ASSIGN specified. Sometimes, specifying RECLAIM can solve the problem.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21008I UNABLE TO WRITE IPL RECORDS

Explanation: The IPL program records could not be written. Messages printed just prior to this message indicate the kind of the error.

System Action: The IPL program records are not written, but the command continues processing.

Operator Response: None.

Programmer Response: Correct the error, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21011I CRITICAL TRACK CCHH=X'cccc hhhh' DESCRIPTOR INCORRECT

Explanation: The critical tracks are those tracks that are reserved for the volume table of contents and cylinder 0, track 0. The contents of the record 0 on the identified track are in error.

System Action: None.

Operator Response: None.

Programmer Response: Reinitialize the volume at the medial level or INSPECT the track at 'cccc hhhh'.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21014I SURFACE CHECK FUNCTION FAILS ON TRACK CCHH=X'cccc hhhh'

Explanation: While writing or reading the indicated track, a channel program failure other than DATA CHECK occurred.

System Action: The command terminates. Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: The CCW, CSW, and sense information associated with the message should be examined to determine the kind of I/O error. If operating under VM (either stand-alone or an operating system version running under VM), this message can be an indication that a diagnostic or media maintenance function was attempted against a device that was a minidisk and/or LINK to the userid. Diagnostic and media maintenance functions must be done on devices that are ATTACHed to the userid. See Appendix B, "VM Support" on page 227 for more information on VM support.

Problem Determination: Table I, items 1, 2, 3, 4, and 29 (or 30).

ICK21017I NO MORE ALTERNATE TRACKS LEFT ON PACK

Explanation: All nondefective alternate tracks have been assigned to primary tracks.

System Action: Assignment of alternate tracks ends; execution of the command continues.

Operator Response: None.

Programmer Response: Use the RECLAIM function of the INSPECT command to attempt to reclaim primary and alternate tracks. For the IBM 3375 and 3380, only the INSPECT command can be used.

The assistance of an IBM service representative might be neccessary to determine the cause of the alternate track assignments.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21028I NOT ALLOWED TO CHANGE VOLUME LABEL

Explanation: Only an authorized user can change the volume label on a volume in a system that has Resource Access Control Facility (RACF) installed.

System Action: The contents of the volume label are not changed, but the command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21030I NOT ALLOWED TO WRITE IPL RECORDS

Explanation: Only an authorized user can change the IPL records on a volume in a system that has Resource Access Control Facility (RACF) installed.

System Action: The IPL records are not written, but the command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21031I UNABLE TO RECLAIM ALTERNATE TRACK CCHH=X'cccc hhhh'

Explanation: An I/O error occurred when attempting to rewrite the home address and record 0 on the indicated alternate track.

System Action: The alternate track is not reclaimed. The command continues processing.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21032I UNABLE TO RECLAIM PRIMARY TRACK CCHH=X'cccc hhhh'

Explanation: An I/O error occurred when attempting to rewrite the home address and record 0 on the indicated primary track.

System Action: The primary track is not reclaimed. The command continues processing.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21033I **TRACK DISSOCIATION FAILED: CCHH=X'cccc hhhh'**

Explanation: When reclaiming a primary or alternate track, the track addresses associating the primary and alternate tracks could not be removed because an error occurred that prevented correct rewriting of the record 0.

System Action: The alternate track is marked defective, and the command continues processing.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21037I **INVALID TRACK ADDRESS SPECIFIED: CCHH=X'cccc hhhh'**

Explanation: Only tracks within the defined primary or alternate track areas of a volume can be inspected. The error is caused either by specifying a track beyond these areas or by a syntax error when specifying the track address.

System Action: The requested action on the invalid track is ignored, but the remainder of the command continues processing for any tracks that were correctly specified.

Operator Response: None.

Programmer Response: Correct the invalid specification, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21039I **PRIMARY TRACK CCHH=X'cccc hhhh' DEFECTIVE, NO ALTERNATE -- DATA SAVED**

Explanation: Surface checking found the indicated primary track to be defective, but no alternate track is available to be assigned to it.

System Action: The contents of the primary track are saved and rewritten on the primary track, if possible. The command continues processing.

Operator Response: None.

Programmer Response: Reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21040I **PRIMARY TRACK CCHH=X'cccc hhhh' DEFECTIVE, NO ALTERNATE -- DATA LOST**

Explanation: Surface checking found the indicated track to be defective, but no alternate track is available to be assigned to it.

System Action: The contents of the primary track are lost and not rewritten on the primary track. The command continues processing.

Operator Response: None.

Programmer Response: Reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21041I **ALTERNATE TRACK CCHH=X'cccc hhhh' DEFECTIVE -- DATA SAVED**

Explanation: Surface checking found the indicated alternate track to be defective, but no alternate track is available to be reassigned to it.

System Action: The contents of the alternate track are saved and rewritten to the alternate track, if possible. The command continues processing.

Operator Response: None.

Programmer Response: Reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21042I **ALTERNATE TRACK CCHH=X'cccc hhhh' DEFECTIVE -- DATA LOST**

Explanation: Surface checking found the indicated alternate track to be defective, but no alternate track is available to be reassigned to it.

System Action: The contents of the alternate track are lost and not rewritten on the alternate track. The command continues processing.

Operator Response: None.

Programmer Response: Reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21044I **UNABLE TO PRESERVE -- UNABLE TO READ TRACK CCHH=X'cccc hhhh'**

Explanation: A record on the indicated cylinder and track could not be read because of I/O errors.

If the sense information in the subsequent message is all zeros and the CSW only indicates Channel End, Device End, and Incorrect Length (0C40 in the first two bytes of the second word), it is usually an indication that previous ICKDSF processing has caused a nonstandard record zero to remain on the volume. An INSPECT of the track, with NOPRESERVE CHECK(1) should be specified to correct the situation. Be aware that there is no user data on the track.

System Action: The record on the indicated track is not lost, and the track is not marked defective. The specified track is not surface-checked. The command continues processing with the next track.

Operator Response: None.

Programmer Response: Start error recovery procedures for the data set containing the track in error, and rerun the command for the track in error with NOPRESERVE specified.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21047I **PRESERVED DATA CANNOT BE REWRITTEN TO TRACK CCHH=X'cccc hhhh'**

Explanation: An I/O error occurred that prevented successful write operations while attempting to write preserved records back to the original track or its assigned alternate track.

System Action: Regardless of the current value of the ASSIGN command, a (new) alternate track is assigned and the data is rewritten. This is tried up to three times in an attempt to write the data correctly.

Operator Response: None.

Programmer Response: Subsequent messages will indicate the final status of the data.

Problem Determination: Not applicable.

ICK21048I CYL-0 TRK-0 DEFECTIVE, PACK UNUSABLE FOR IPL

Explanation: Cylinder 0, track 0 has been found defective. The track has been assigned an alternate track, but the volume cannot be used as an IPL volume.

System Action: The command continues processing.

Operator Response: None.

Programmer Response: Call the customer engineer if you must use this volume as an initial-program-load volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21055I VTOC IS LOCATED ON CYLINDER 0 HEAD 0

Explanation: The volume table of contents exists on the first track of the volume. This prevents the writing of IPL program records on the first track of the volume.

System Action: The IPL program records are not written on the first track of the volume as requested by the user. Command processing continues.

Operator Response: None

Programmer Response: If it is necessary to have IPL records on the volume, reinitialize the volume with the VTOC on any other track other than the first track of the volume.

ICK21080I ALTERNATE TRACK INSPECTION NOT VALID FOR DEVICE TYPE

Explanation: The user requested to inspect an alternate track on a 2305-1 or 2305-2. The alternate track cannot be inspected directly for these devices. Inspection of the primary track that is associated with the alternate track automatically inspects the alternate track.

System Action: Processing continues with the next track.

Operator Response: None.

Programmer Response: If inspection of the alternate track is required, examine the map to determine the address of the primary track associated with the alternate track and rerun the job using the primary track address.

Problem Determination: Not applicable.

ICK21316I UNABLE TO OPEN IPL INPUT DATA SET

Explanation: The data set specified by the IPLDD parameter cannot be opened. An associated message identifies the reason for the failure.

System Action: The command continues processing, but the IPL program record is not written.

Operator Response: None.

Programmer Response: Correct the error that was identified in the associated message, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21317I IPL INPUT RECORD-1 FORMAT IMPROPER

Explanation: The first user-specified IPL bootstrap record did not contain IPL1 as the first four characters of the record.

System Action: The system-defined IPL bootstrap records are written on the volume and processing continues. The remainder of the user-specified IPL bootstrap records and IPL program records are ignored.

Operator Response: None.

Programmer Response: Correct the IPL bootstrap record format and reissue the command, or accept the system-defined IPL bootstrap records.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21318I IPL INPUT RECORD-2 FORMAT IMPROPER

Explanation: The second user-specified IPL bootstrap record did not contain IPL2 as the first four characters of the record.

System Action: The system-defined IPL bootstrap records are written on the volume and processing continues. The user-specified IPL bootstrap records and IPL program records are ignored.

Operator Response: None.

Programmer Response: Correct the IPL bootstrap record format and reissue the command, or accept the system-defined IPL bootstrap records.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21319I UNABLE TO WRITE BOOTSTRAP-1 RECORD ON VOLUME

Explanation: An I/O error occurred when attempting to write the first IPL bootstrap record on cylinder 0, track 0. The volume is unusable for IPL.

System Action: The command continues processing, but the IPL bootstrap records are not written on the volume.

Operator Response: None.

Programmer Response: Issue the INSPECT command to attempt to reclaim the track, or reinitialize the volume.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK21320I UNABLE TO WRITE BOOTSTRAP-2 RECORD ON VOLUME

Explanation: An I/O error occurred when attempting to write the second IPL bootstrap record on cylinder 0, track 0. The volume is unusable for IPL.

System Action: The command continues processing, but the IPL bootstrap records are not written on the volume.

Operator Response: None.

Programmer Response: Issue the INSPECT command to attempt to reclaim the track, or reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21321I UNABLE TO WRITE IPL PROGRAM ON VOLUME

Explanation: An I/O error occurred when attempting to write the IPL program record on the volume. The volume is unusable for IPL.

System Action: The command continues processing, but the IPL program record is not written on the volume.

Operator Response: None.

Programmer Response: Issue the INSPECT command to attempt to reclaim the track, or reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21322I CALCULATED IPL PROGRAM LENGTH IS EXCESSIVE

Explanation: The total number of bytes required by the user-specified IPL program exceeds the maximum permitted for the volume being initialized.

System Action: The command continues processing, but the IPL program is not written on the volume.

Operator Response: None.

Programmer Response: Determine the cause of the excessive size, recompile, or reassemble the IPL program; then reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21323I IPL PROGRAM INPUT CONTAINS NO TXT CARDS

Explanation: The input records specified for the IPL program source were read, but contained no records that were identified as standard text (TXT) records.

System Action: The IPLDD parameter is ignored, and the command continues processing.

Operator Response: None.

Programmer Response: Be sure the correct input is supplied in the input stream or in the specified data set.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21394I EXCESSIVE DATA CHECKS OCCURRING FOR TRACK X'cccc hhhh'.

Explanation: During skip displacement processing for the specified track the number of data checks exceeded the device threshold for this device type.

System Action: Processing has completed for this track. Processing continues with the next track.

Operator Response: None.

Programmer Response: This message is an indication that an abnormal amount of data checks have occurred for this track. Although this message is intended as a warning, the system programmer might want to seek the aid of a customer engineer, especially if this message is issued for multiple tracks.

Problem Determination: Not applicable.

ICK21398I RECORD ZERO MAY BE INVALID ON TRACK X'cccc hhhh'

Explanation: At the completion (successful or otherwise), of surface checking for the specified track, the rewriting of a standard record zero could not be verified.

System Action: If this message is issued during an INIT or an INSPECT NOPRESERVE, processing continues with the next track.

Operator Response: None.

Programmer Response: This message is often accompanied by an indication that surface checking failed on the specified track. The accompanying messages, including the I/O error messages, should be examined to aid in determining the cause of the failure. System representative support might be required.

When the source of the error is established and removed, an INSPECT of the track should be run to fix the bad record zero situation. If the accompanying messages indicate a data loss during INSPECT PRESERVE processing, data recovery actions may be required.

Problem Determination: Not applicable.

ICK21401I ccuu SUSPECTED DRIVE PROBLEM

Explanation: Error detected during the drive test.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: Previous messages should aid in giving an indication of the problem. Save printer output. Take the action appropriate to your installation's procedures for handling suspected equipment problems. If the problem cannot be explained, contact your IBM customer engineer or service representative.

In the interim, depending on your installation's procedures, you might do the following:

- Restore the entire volume to another drive from a backup volume.
- Discontinue using the drive.

Problem Determination: See "Programmer Response."

ICK21402I ccuu ERROR READING DATA

Explanation: This message is issued if any data checks are detected during the data scan portion of the command.

System Action: Processing continues.

Operator Response: None.

Programmer Response: Previous messages should be examined to determine the nature and location of the data checks, and appropriate action should be taken based on the previous messages.

Problem Determination: See "Programmer Response."

ICK21403I TEMPORARY EQUIPMENT CHECK LIMIT EXCEEDED, ERROR INFORMATION FOLLOWS

Explanation: During the write and read test portion of the drive test, more temporary equipment checks were detected than are considered reasonable for this device type. The message will be followed by the failing channel status word (CSW), channel command word (CCW), and sense bytes for each temporary equipment check that occurred.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: The failing CCW, CSW, and sense information should be examined to determine the cause of the equipment checks. Assistance of hardware support is required to interpret the sense information and/or to aid in correcting the cause of the error.

ICK21404I **WRT TST: FORMAT WRITE OF CE CYLINDER FAILED ON HEAD = X'hhhh'**

Explanation: An error was detected during the format write to the customer engineer (CE) cylinder.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: Examine previous messages to determine the cause of the error.

ICK21405I **WRT TST: UPDATE WRITE OF CE CYLINDER FAILED ON HEAD = X'hhhh'**

Explanation: An error was detected during the update write to the customer engineer (CE) cylinder.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: Examine previous messages to determine the cause of the error.

ICK21406I **CLEANUP TEST: CLEANUP OF CE CYLINDER FAILED ON HEAD = X'hhhh'**

Explanation: An error was detected during the attempt to cleanup the tracks on the customer engineer (CE) cylinder after the drive test has completed its testing.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: Examine previous messages to determine the cause of the error.

ICK21407I **UNCORRECTABLE DATA CHECK OCCURRED ON CE CYLINDER, HEAD = X'hhhh', RECORD = X'rr'**

Explanation: An uncorrectable data check was detected during the read test portion of the drive test. The message is followed by the failing CSW, CCW, and sense bytes.

System Action: Processing continues. Eventual termination may occur.

Operator Response: None.

Programmer Response: Contact hardware support to aid in resolving the problem.

ICK21408I **CE CYLINDER DATA CHECK THRESHHOLD EXCEEDED**

Explanation: During read testing from the customer engineer (CE) cylinder, more data checks were detected than are considered resonable for this device type.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: Examine the error information contained in message(s) ICK01415 and/or ICK21407 to determine the failing head(s) and record(s). Contact hardware support to aid in resolving the problem.

ICK21409I **CORRECTABLE DATA CHECK LIMIT EXCEEDED ON CE CYLINDER, HEAD = X'hhhh'**

Explanation: During the read test portion of the drive test, this head exceeded the error criterion. The message will be preceded by ICK01415 for each record on this head that experienced a correctable data check.

System Action: Processing terminates.

Operator Response: None.

Programmer Response: Examine the error information contained in message ICK01415 to determine the failing records. Contact hardware support to aid in resolving the problem.

ICK21410I **READ TEST: READ TEST ON CE CYLINDER FAILED ON HEAD = X'hhhh'**

Explanation: An error other than a data check occurred during the read test portion of the drive test.

System Action: Command processing terminates.

Operator Response: None.

Programmer Response: Examine previous messages to determine the cause of the problem.

ICK21721I **FIXED BLOCK WRITE C.E. AREA TEST: BWRCE**

Explanation: An error was detected during the write to the CE cylinder.

System Action: Processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21723I **FIXED BLOCK CALIBRATION SEEK TEST: BCALSK**

Explanation: An error was detected during an attempt to move the access arm from block 244 to a specified location and back again to read.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21725I **MULTIPLE BLOCK WRITE TEST: BHDSK**

Explanation: An error was detected during an attempt to write multiple blocks and seek to the next sequential track.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21727I FIXED BLOCK INCREMENTAL SEEK TEST: BINCSK

Explanation: An error was detected during an attempt to move the access arm incrementally to read.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21729I FIXED BLOCK SERVO SEEK TEST: BSRVOSK

Explanation: An error was detected during an attempt to write and read back block 244 of the CE area.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: See "Programmer Response."

ICK21731I FIXED BLOCK RANDOM SEEK TEST: BRANSK

Explanation: An error was detected during an attempt to switch heads randomly to read.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21733I FIXED BLOCK READ PREFORMATTED BLOCK TEST: BWRNFPS

Explanation: An error was detected during an attempt to read block 296 of the CE area.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21735I FIXED BLOCK WRITE PRESELECTED BLOCKS TEST: BWRNFP

Explanation: An error was detected during an attempt to write multiple blocks in the CE area.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21736I FIXED BLOCK WRITE/READ PRESELECTED BLOCKS TEST: BWRRDRV

Explanation: An error was detected during the write/read tests on preselected blocks on a track in the CE cylinder.

System Action: ANALYZE drive test continues processing by testing blocks on the next track of the CE cylinder.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21737I FIXED BLOCK ERROR READING ID FIELD, TESTING TERMINATED

Explanation: An error was detected reading the ID field on the CE cylinder.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21738I FIXED BLOCK DATA COMPARE FAILED: BWRRDRV

Explanation: An error was detected where the data in the data record read did not compare equally with the data previously written to the same record.

System Action: ANALYZE drive test continues processing by testing blocks on the next track of the CE cylinder.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21739I FIXED BLOCK DATA COMPARE FAILED: BWRCE

Explanation: An error is detected when the record read did not compare equally with the same record previously written.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21750I SNS TEST: ERROR IN SENSE CCW

Explanation: An error was detected when ANALYZE attempts to obtain sense information and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21752I **RECAL TEST: RECALIBRATE TO CYL 0 HEAD 0**

Explanation: An error was detected when ANALYZE attempts to recalibrate the access arm to cylinder 0, head 0. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21754I **RHAMH TEST: READ HOME ADDRESS UNDER MOVABLE HEADS FAILED.**

Explanation: An error was detected when ANALYZE attempts to read home addresses under all the movable heads, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21755I **READ FOR PRESERVE OF BLOCK xxxxxx FAILED: NON-CORRECTABLE DATA CHECKS PROCESSING OF BLOCK TERMINATED**

Explanation: The user data on block xxxxxx could not be read because of permanent data checks.

System Action: Diagnostic information is printed on the output device. Processing of the block is terminated to prevent the loss of the user data on the block. Processing continues with the next valid block specified in the BLOCKS parameter.

Operator Response: None.

Programmer Response: Start error recovery procedures for the data set containing the block in error, and rerun the command for the block in error with NOPRESERVE specified.

Problem Determination: Table I, items 1, 2, 3, and 4.

ICK21756I **RHAFH TEST: READ HOME ADDRESS UNDER FIXED HEADS FAILED.**

Explanation: An error was detected when ANALYZE attempts to read home addresses under all the fixed heads, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21758I **SKINCR TEST: INCREMENTAL MOVEMENT OF ACCESS ARM FAILED.**

Explanation: An error was detected when ANALYZE attempts to move the access arm incrementally and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21760I **SKRAN TEST: RANDOM CYLINDER ACCESS FAILED.**

Explanation: An error was detected when ANALYZE attempts to move the access arm randomly from one cylinder address to another, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21762I **SKMAX TEST: SEEK FROM CYLINDER 0 TO MAXIMUM CYLINDER AND FAILED.**

Explanation: An error was detected when ANALYZE attempts to move the access arm from cylinder 0 to the maximum cylinder address, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21764I **RHA TEST: READ HOME ADDRESS ON C.E. CYLINDER FAILED. WRITE TEST BYPASSED.**

Explanation: An error was detected when ANALYZE attempts to read the CE cylinder's home address, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing continues, but does not execute the write tests on the CE cylinder. The return code is set to 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21766I **WRT TEST: WRITE RECORDS R0 AND R1 ON C.E. CYLINDER FAILED.**

Explanation: An error was detected when ANALYZE attempts to write records 0 and 1 on the CE cylinder, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21768I **RDMT TEST: MULTI-TRACK READ ON C.E. CYLINDER FAILED**

Explanation: An error was detected when ANALYZE attempts to read records on the CE cylinder, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21770I **RMDT DATA COMPARE TEST: WRONG RECORD READ**

Explanation: An error was detected when a drive error caused the wrong record to be read. This message is followed by a printout of the failing CCW, CSW, and sense information, although the sense information will be zero if no unit check is indicated in the CSW.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21772I **WRTPAD READ WRONG RECORD--EXP=xxxxxxx, REC=yyyyyyy**

Explanation: An error was detected when ANALYZE attempts to write a record using the write count, key, data CCW, and read it back. The record read back was not as expected. xxxxxxx is the description of the record that was expected (that is, the data that was written). yyyyyyy is the description of the record that was received.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21774I **WRTPAD TEST: WRITE COUNT, KEY, DATA FAILED**

Explanation: An error was detected when ANALYZE attempts to write a record using the write count, key, data CCW, and fails. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21776I **DATAVER TEST: ERROR DURING DATA VERIFICATION**

Explanation: A drive error was detected during the data verification test. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: Processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Seek hardware support for the drive error.

Problem Determination: See "Programmer Response."

ICK21778I **CLEANUP TEST: CLEANUP OF C.E. CYLINDER TRACK 0 FAILED**

Explanation: An error was detected when ANALYZE has completed its test and an error occurs during the cleanup of track 1 on the CE cylinder. This message is followed by a printout of the failing CCW, CSW, and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK21786I **RPS ERROR: READ SECTOR OR SET SECTOR FAILED**

Explanation: An error was detected when ANALYZE attempts to read sector or to set sector, and fails. This message is followed by a printout of the failing CCW, CSW and sense information.

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: See "Programmer Response."

ICK21798I **INVALID TRACK FORMAT ON CYLINDER = xxxx HEAD x**

Explanation: A track was encountered during reclaim processing with a format which is not consistent with the information in the factory defect map for the track.

System Action: Reclaim processing of the indicated track is terminated. Reclaim processing continues with the next sequential track.

Operator Response: None.

Programmer Response: Seek hardware support

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK21826I **INVALID TRACK FORMAT ON TRACK CCHH = X'cccc hhhh'**

Explanation: An invalid track format condition has been detected while attempting to preserve data on the specified track. This is usually an indication that data has been written to the track beyond the track capacity. This is generally a user error.

System Action: The record on the specified track is not lost, and the track is not marked defective. The specified track is not surface-checked. Processing continues with the next track. The return code is set to 8.

Operator Response: None.

Programmer Response: Determine the data set containing the specified track. Determine how, when, and where the data on the specified track was written, in order to determine how the invalid track format condition occurred. Start to recover the data on the track using your installation's procedures. Inspection of the failing track can then take place with NOPRESERVE specified.

Problem Determination: See "Programmer Response."

ICK21831I **NO PATHS HAVE BEEN WRITE ALLOWED**

Explanation: During execution of the CONTROL command to allow write access to the paths to a device, the write-allow was unsuccessful for all paths.

System Action: The return code is set to 8.

Operator Response: None.

Programmer Response: Rerun the job for a device that is known to go through the storage control that has been write-prevented.

This condition can also arise if the required path has been reserved too long and the operator requested that retry be terminated. If this condition exists, action must be taken to ensure that the path is released, and the job can be rerun.

Problem Determination: Not applicable.

ICK21832I **FENCE STATUS INDETERMINATE**

Explanation: A CONTROL command (with the CLEARFENCE parameter) to reset a fence status is unsuccessful. The device fence status cannot be determined at this time.

System Action: The return code is set to 8 and processing continues.

Operator Response: None.

Programmer Response: This condition can arise if the required path has been reserved too long and the operator requests that retry is terminated. If this condition exists, action must be taken to ensure that the path is released and the job can be rerun

If it is established that a fence condition exists for this device, other methods may have to be used to unfence the device. Contact your IBM service representative.

Problem Determination: Not applicable.

ICK21834I **PATH xxxx,y RESERVED-REPLY U TO RETRY WRITE ALLOW, T TO IGNORE**

Explanation: This message is issued to the system operator if the CONTROL command encounters a path that remains reserved for all of its retries. 'xxxx' is the unit address, 'y' is the path ID (CHPID). If the reply is U, the write-allow is reissued to the device. If the reply is T, this path is ignored and CONTROL continues with the next path.

System Action: See Explanation.

Operator Response: See Explanation.

Programmer Response: None.

Problem Determination: Not applicable.

ICK21835I **DEVICE ccuu RESERVED-REPLY U TO RETRY CLEAR FENCE, ELSE T**

Explanation: This message is issued if a device or path remains reserved for the duration of CONTROL command (with the CLEARFENCE parameter) processing. If you reply "U", the CONTROL command is re-issued. If you reply "T", this path is ignored and the CONTROL command processing continues with the next path.

System Action: The system waits for your response

Operator Response: Reply either "U" or "T".

Programmer Response: None.

Problem Determination: Not applicable.

ICK21836I **IPL TEXT EXISTS ON volid. REPLY U TO OVERLAY, ELSE T.**

Explanation: An attempt is made to replace the IPL text on a volume that already contains IPL text.

System Action: If the reply is U, the new IPL text will be placed on the volume. If the reply is T, the command terminates.

Operator Response: See Explanation.

Programmer Response: None.

Problem Determination: Not applicable.

ICK21850I **EXTENDED CKD FUNCTIONS CANNOT BE ACTIVATED - MINIMAL INIT FORCED**

Explanation: The device has never been online to the operating system and the operating system cache logic has not been activated. Device Support Facilities surface checking functions cannot be performed.

System Action: Processing continues as if a minimal initialization has been requested. This enables the volume label and the VTOC to be written to the volume.

Operator Response: The device must be varied online and varied offline (or varied online and the job can be run online) in order to activate the operating system cache logic.

Programmer Response: If the device cannot be varied online in its current state (that is, the volume label is the same as a volume label that is already online), the INIT or the REFORMAT command can be run to change the volume label.

After the device has been varied online (and offline again if necessary), the job can be reexecuted.

Problem Determination: Not applicable.

ICK22130D **DATA CANNOT BE RECOVERED FOR TRACK X'cccc hhhh' REPLY 'C' TO ERASE AND CONTINUE, 'I' TO IGNORE, T TO TERMINATE.**

Explanation: The previous INSPECT command did not complete execution during PRESERVE backup processing. **Note**: The data cannot be recovered from the backup location.

System Action: The operator is prompted for a reply to this message. The ignore option is provided if HOLDIT is specified on the current invocation of the INSPECT command.

If 'c' is specified, the data is erased at the backup location, and processing begins from the start of the specified range. This should be specified if the user has no need to recover data from the backup location.

If 'i' is specified, the recovery data is ignored, and processing continues for the current invocation. Note that this response is allowed only if HOLDIT was specified. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

If 't' is specified, processing terminates. The recovery data remains intact. This should be specified if the user wishes to retry the recovery process, or examine the situation before allowing the data to be erased. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK22131I **ALTERNATE TRACK CANNOT BE ASSIGNED FOR RECOVERY TRACK**

Explanation: The track for which backup recovery is processing is defective, but there are no available alternate tracks.

System Action: An attempt is made to write the backup recovery data to the recovery track without an alternate track assigned. If that procedure fails, subsequent messages will indicate the system action.

Programmer Response: It is recommended that INSPECT RECLAIM be executed for tracks that have alternates assigned to attempt to make alternate tracks available for future use.

ICK22158D **DATA CANNOT BE RECOVERED FOR BLOCK xxxxxx REPLY 'C' TO ERASE AND CONTINUE, 'I' TO IGNORE, T TO TERMINATE.**

Explanation: The previous invocation of the INSPECT command did not execute to completion during PRESERVE backup processing. **Note**: The data cannot be recovered from the backup location.

System Action: The operator is prompted for a reply to this message. The ignore option is provided if HOLDIT is specified on the current invocation of the INSPECT command.

If 'c' is specified, the data is erased from the backup location, and processing begins from the start of the specified range. This should be specified if the user has no need to recover data from the backup location.

If 'i' is specified, the recovery data is ignored, and processing continues for the current invocation. Note that this response is allowed only if HOLDIT was specified. The recovery data remains intact. The next invocation of the INSPECT command will again attempt to process the recovery data.

If 't' is specified, processing terminates. The recovery data remains intact. This should be specified if the user wishes to retry the recovery process, or examine the situation before allowing the data to be erased. The next invocation of the INSPECT command will again attempt to process the recovery data.

ICK22176I **DATA CHECK THRESHOLD EXCEEDED ON ONE OR MORE HEADS**

Explanation: The ANALYZE SCAN function found that the data error rate on one or more heads exceeded the data check error rate threshold criterion for the device.

System Action: A moveable head error table is printed on the ICKDSF output device with a "T" in the "DATA CHK" column indicating which head or heads exceeded the data check error rate threshold.

Operator Response: None.

Programmer Response: The moveable head error table and previous messages will aid in giving an indication of the problem. Save the printer output from the ANALYZE run and take action appropriate to your installation's procedures for handling suspected equipment problems. If your problem cannot be determined contact your IBM service representative.

Problem Determination: See Programmer Response.

ICK22201I **TRACK X'cccc hhhh' WAS NOT INSPECTED DUE TO ENQUEUE FAILURE ON -- xxxxxx**

Explanation: The data set enqueue for track x'cccc hhh' failed. xxxxxx is the data set name. TOLERATE(ENQFAIL) was not specified. The track was not INSPECTed.

System Action: INSPECT continues on the next track.

Programmer Response: The job should be rerun for this track at a time when the data set may be available. If you know that the data set will never be available for exclusive control and the track must be inspected, the job can be rerun for this track with TOLERATE(ENQFAIL) specified.

ICK22205I **TRACK X'cccc hhhh' WAS NOT INSPECTED -- VSAM DATA SET CANNOT BE ENQUEUED**

Explanation: The specified track is part of a VSAM data set. ICKDSF does not support data set enqueue for VSAM data sets. Because TOLERATE(ENQFAIL) was *not* specified or assumed, the track was not inspected.

System Action: Processing continues on the next track.

Programmer Response: An INSPECT for this track should be done with the device mounted offline, or specifying TOLERATE(ENQFAIL).

ICK30003I **FUNCTION TERMINATED. CONDITION CODE IS nn hh:mm:ss mm/dd/yy**

Explanation: A command has encountered a terminating error condition during execution. The value *nn* is the last condition code (LASTCC) generated during execution of the command. Messages printed just prior to this message indicate the nature of the error.

Note that the LASTCC value is the highest condition code found in the messages printed during execution of the command.

hh:mm:ss and *mm/dd/yy* are the hours, minutes and seconds, and month, day, and year, respectively, of the date and time of the message.

System Action: Device Support Facilities processing continues with the next command. LASTCC is set to *nn*; MAXCC is set to *nn* if *nn* is greater than the current value of MAXCC.

Operator Response: None.

Programmer Response: Correct the cause of the error, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30004I FUNCTION TERMINATED. INSUFFICIENT MAIN STORAGE

Explanation: The size of allocated virtual storage is too small.

System Action: Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Increase the allocated size of virtual storage, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, and 29.

ICK30008I FUNCTION NOT SUPPORTED IN THIS ENVIRONMENT

Explanation: The system adapter was entered with a request to perform a command not supported in the current operating system environment. Probable program error.

System Action: Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: The version of Device Support Facilities is not appropriate for the operating system environment. Ensure that the appropriate version is installed.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30200I TOO MANY POSITIONAL PARAMETERS AFTER xxxx

Explanation: A parameter list has too many positional parameters specified, specifically those following the characters *xxxx*.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Remove the excess parameters from the command, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30201I CONSTANT xxxx EXCEEDS LENGTH LIMIT

Explanation: The constant *xxxx* contains more characters than the maximum permitted by the command syntax.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the constant, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30202I ABOVE TEXT BYPASSED UNTIL NEXT COMMAND. CONDITION CODE IS 12.

Explanation: A syntax error has been encountered; the remainder of the command is ignored. Messages printed just prior to this message indicate the nature of the error.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the syntax error, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30203I ITEM xxxx DOES NOT ADHERE TO RESTRICTIONS

Explanation: An indicated parameter does not conform to the naming conventions required. For example, *dname* may be misspelled.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the parameter error, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30205I DELIMITER xxxx IS NOT PROPERLY PRECEDED BY A CONSTANT OR KEYWORD

Explanation: A delimiter was found where a subparameter list or data was expected. The delimiter is improperly used: Parentheses may be improperly positioned, or a positional parameter may be missing.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the syntax error, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30207I REMAINDER OF COMMAND INPUT STREAM IGNORED

Explanation: An error has occurred that prohibits further scanning of the command stream. Messages printed just prior to this message indicate the nature of the error.

Note: The condition code (MAXCC) is always set to 16 when this situation is encountered.

System Action: Device Support Facilities processing terminates.

Operator Response: None.

Programmer Response: Correct the error, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30208I **LEFT PARENTHESIS MISSING FOLLOWING KEYWORD xxxx**

Explanation: The keyword *xxxx* is not followed by an opening parenthesis, which should begin the required subparameter list or value associated with the command keyword.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the requirements of the keyword, correct the syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30209I **RIGHT PARENTHESIS MISSING AFTER xxxx**

Explanation: A closing parenthesis was not found where expected, or a subparameter list was not properly delimited.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30210I **INVALID PARENTHESES FOR SPECIFYING REPEATED SUBPARAMETER LIST**

Explanation: Parentheses that are used for delimiting repeated subparameter lists are missing or not matched.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30211I **KEYWORD xxxx IS IMPROPER**

Explanation: A misspelled, improperly specified, or inapplicable keyword was encountered in the command.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30212I **INVALID LEFT PARENTHESIS AFTER xxxx**

Explanation: An opening parenthesis was found which appeared to delimit the positional parameter *xxxx*, but the positional parameter was not a constant or a list of constants.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30213I **KEYWORD xxxx APPEARS TOO OFTEN**

Explanation: The keyword *xxxx* has appeared too often in the command. A parameter list may be incorrectly specified.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax by removing the keyword, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30214I **HEX OR BINARY CONSTANT SPECIFIED IMPROPERLY**

Explanation: A hexadecimal or binary constant is not specified in the correct format: X'*n*' or B'*n*', respectively.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30216I **ABOVE TEXT BYPASSED UNTIL NEXT COMMAND**

Explanation: An error was encountered during the syntax check of this command, and syntax checking was terminated. Messages printed just prior to this message indicate the nature of the error.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30217I **PASSWORD IMPROPER AFTER xxxx**

Explanation: A password, denoted by a slash (/), was encountered where a password is not allowed.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Remove the password from the parameter, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30218I **TOO MANY REPEATED SUBPARAMETER LISTS APPEAR**

Explanation: Too many repeated subparameter lists appear in the command.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the command syntax, correct the error, then reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30219I VERB NAME xxxx UNKNOWN

Explanation: The verb *xxxx* is not recognized as a command.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command with the correct command name.

Problem Determination: Table I, items 2, 4, and 29.

ICK30220I IMPROPER NUMERIC DIGIT FOUND IN xxxx

Explanation: An invalid character was encountered in the constant *xxxx*. A decimal number can only be specified with the symbols 0 through 9; a hexadecimal number can only be specified with the symbols 0 through 9 and A through F; and a binary number can only be specified with the symbols 0 and 1.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax error, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30221I CONSTANT xxxx NOT WITHIN VALUE RANGE

Explanation: The value of the constant *xxxx* is outside the range of values allowed for the associated parameter.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the command syntax for allowed values, correct the error, then reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30223I TOO MANY CONSTANTS IN LIST BEGINNING AT xxxx

Explanation: Too many constants were specified in the command beginning at the characters *xxxx*.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30225I REQUIRED (SUB)PARAMETER OF xxxx IS MISSING

Explanation: A required parameter or subparameter, identified by *xxxx*, is missing.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Add the missing parameter, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30226I INCONSISTENT PARAMETERS INVOLVING xxxx

Explanation: Some commands contain parameters that are defined as mutually exclusive. If one parameter is specified, the other parameter is not allowed.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK30234I WARNING: TOO FEW RIGHT PARENTHESES FOUND AT END OF COMMAND

Explanation: Too few closing parentheses were found at the end of the command to properly close the subparameter lists.

System Action: Command processing is terminated.

Operator Response: None.

Programmer Response: Correct the command syntax.

Problem Determination: Table I, items 2, 4, and 29.

ICK30300I ERROR OPENING dsname | ddname

Explanation: An error occurred when attempting to open the indicated data set or volume. See the associated message for the cause of the error.

System Action: See the associated message.

Operator Response: None.

Programmer Response: See the associated message.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30301I ERROR CLOSING dsname

Explanation: An error was encountered while attempting to close the data set. See the associated message for the cause of the error.

System Action: See the associated message.

Operator Response: None.

Programmer Response: See the associated message.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30302I ACTION ERROR ON dsname

Explanation: This message identifies the name of the data set that was processing when the error occurred. See the associated message for an explanation of the error.

System Action: None.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30304I ****JCL STATEMENT MISSING**

Explanation: The DD statement that was identified in the DDNAME parameter is missing (MVS/370).

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Be sure the DD statement is in the job stream as specified in the command. Check the DDNAME for correct spelling.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30309I ****RECORD xxxxx NOT WRITTEN. LENGTH INVALID**

Explanation: The record *xxxxx* was not written for one of the following reasons:

- The record length was greater than LRECL of the output data set.
- The record length was less than the LRECL of the output data set and RECFM was F (fixed).

Note: *xxxxx* is the first five bytes of the record in hexadecimal format.

System Action: Device Support Facilities continues processing until four such errors occur. After four errors, the command continues to execute, but no further records are written to the output data set.

Operator Response: None.

Programmer Response: Redefine the data set with the correct LRECL value.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30312I ****SYSTEM UNABLE TO OPEN**

Explanation: The DCBOFLG was not set ON following an OPEN SVC (MVS), or the OPEN system macro failed (VSE). See the associated messages for the cause of the error.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Perform the Problem Determination actions.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30313I ****synad.text**

Explanation: An I/O error occurred. The contents of the MVS Data Management Services SYNADAF error message are written.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the explanation of the SYNADAF message. Correct the error, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30315I ****RECORD SIZE GREATER THAN 32767 NOT SUPPORTED**

Explanation: The system cannot process a logical record whose size is greater than 32,767 bytes.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Change the DD statement to specify a LRECL length that is less than 32,767 bytes.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30317I ****PERMANENT I/O ERROR**

Explanation: An error was detected while performing an I/O operation. The data set name is given in the associated message.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the MVS JCL statements to be sure the data set was correctly defined. If the data set was correctly defined, a hardware error was encountered and the data set must be re-created on another I/O device.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK30318I ****INVALID DATA SET SPECIFICATION**

Explanation: In the VSE environment, the I/O adapter open/close routine was unable to open a data set because of an invalid specification of the data set in the DLBL statement.

System Action: Processing for the specific data set is terminated. Command processing will continue if the opening of the specific data set is deemed noncritical to the main function performed by the command.

Operator Response: None.

Programmer Response: Probable user error. Check the DLBL statement. If there are no errors, contact your system programmer.

ICK30320I ****INVALID DEVICETYPE**

Explanation: In the VSE environment, the I/O adapter open/close routine was unable to open a data set, because it exists on a device that is not supported for sequential processing.

System Action: Processing for the specific data set is terminated, and the command processing will continue if the opening of the specific data set is deemed noncritical to the main function performed by the command.

Operator Response: None.

Programmer Response: Probable user error. The data set does not exist on a valid device that is supported for sequential processing (SAM access method). Correct the situation, and resubmit the job. If the error persists, contact your system programmer.

ICK30321I ****OPEN/CLOSE ABEND EXIT TAKEN**

Explanation: The OPEN/CLOSE system services SVC detected an error when opening or closing a data set. The ABEND message was written to the JOBJCL data set.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the ABEND message, correct the error, then reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30500I **MAIN STORAGE NOT AVAILABLE, COMMAND TERMINATED**

Explanation: Dynamic acquisition of storage for control blocks and work areas failed.

System Action: The command is terminated.

Operator Response: Increase main storage size, and resubmit the job.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30700I **EQUIPMENT CHECK ON CCUU**

Explanation: An equipment check I/O error occurred.

System Action: Command processing terminates.

Operator Response: None.

Programmer Response: Examine the accompanying failing CCW, CSW, and sense information to determine the cause of the equipment checks. Assistance of hardware support is required to interpret the sense information and/or to aid in correcting the cause of the error.

ICK30701I **EQUIPMENT CHECK ON ccuu**

Explanation: An EQUIPMENT CHECK I/O error occurred.

System Action: If you do not receive message ICK00701I "EQUIPMENT CHECK RETRY SUCCESSFUL", execution of the command is terminated.

Operator Response: None.

Programmer Response: The failing CCW, CSW, and sense information should be examined to determine the cause of the equipment check. Assistance of the customer engineer is required to interpret the sense information and/or to aid in correcting the cause of the error.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK30702I **OPERATOR DID NOT READY DEVICE**

Explanation: When the operator was prompted by message ICK004D to make a device ready, the response was to terminate the command.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30703I **DEVICE ALREADY IN USE**

Explanation: The device referred to in the Device Support Facilities command by the UNITADDRESS or DDNAME parameter is already being used by another job.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30704I **INVALID DEVICE-TYPE FOR FUNCTION**

Explanation: The type of device specified is invalid for the command issued. For instance, a Mass Storage System staging pack cannot be a 2314 device.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command, and specify an appropriate device.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30705I **OPERATOR DID NOT SET DEVICE TO WRITE-MODE**

Explanation: During I/O operations, the device was found set for read-only mode. The operator elected to terminate the command in response to message ICK014D instead of setting the device to write mode.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Ensure that the device is in write mode, and resubmit the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK30706I **DEVICETYPE PARAMETER MISSING OR ERRONEOUS**

Explanation: This message appears only in a stand-alone environment. The DEVICETYPE parameter must specify one of the valid device types. See Figure 2 on page 7 for valid device types.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the error, and reissue the command.

Problem Determination: Table I, items 2, 3, 4, and 29.

ICK30707I **MIMIC MINI/EMU SPECIFICATION ERRONEOUS**

Explanation: The number of cylinders specified, either for an MVS minidisk for the VM environment or for an emulated count-key-data device on a fixed block architecture device, is greater than the total number of primary cylinders that exist on the volume.

System Action: The command terminates and Device Support Facilities continues processing with the next command.

Operator Response: None.

Programmer Response: Correct the invalid number of cylinders in the MINI or EMU specification, and resubmit the job.

ICK30708I DEVICETYPE PARAMETER REQUIRED WITH MIMIC MINI SPECIFICATION.

Explanation: When you specify the MIMIC MINI parameter, you must also specify the DEVICETYPE parameter.

System Action: The command terminates and Device Support Facilities continues processing with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and resubmit the job.

ICK30709I ACCESS DENIED TO SHARED DEVICE ccuu BY OPERATOR

Explanation: If the operator replies T to message ICK008D, this message appears before termination of the command.

System Action: Command processing terminates, and Device Support Facilities continues processing with the next command.

Operator Response: None.

Programmer Response: Resubmit the job when the volume is accessible.

ICK30711I CHANNEL ERROR: CSW=xxxxxx xxxxxxxx

Explanation: An unrecoverable channel error occurred. xx . . . xx is a hexadecimal display of the last seven bytes of the CSW at the time of the error.

System Action: Execution of the command is terminated. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table 1, items 1, 2, 3, 4, 30.

ICK30712I xxx DEVICE TYPE VERIFICATION FAILED

Explanation: During OPEN, Device Support Facilities was unable to determine the device type at address xxx, or the device type is not supported. If the sense ID CCW X'E4' is supported by the device at address xxx, then the following message is issued:

```
PHYSICAL DEVICE=UNKNOWN
SENSE ID=xxxxxxxxxxxxxx
```

See the appropriate device manual for the sense ID information. Basically, the format is as follows:

```
byte 0    = X'FF'
bytes 1-3  = storage control id
bytes 4-6 = device id
```

If the sense ID CCW is not supported by the device, the following message is issued:

```
PHYSICAL DEVICE=UNKNOWN
LOGICAL DEVICE=UNKNOWN
```

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: The direct access devices supported by Device Support Facilities are listed in the Introduction. Check that the device specified by the UNITADDRESS, DDNAME, or SYSNAME parameter is supported by Device Support Facilities. If the device is supported by Device Support Facilities, follow the problem determination shown.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 29.

ICK30714I DDNAME NOT FOUND

Explanation: The ddname specified on the DDNAME parameter of input command could not be found for this job step.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Add a DD card to the JCL for the job step to include the ddname specified in the command.

Problem Determination: Table I, items 1, 2, 3, 4.

ICK31004I VTOC CREATION FAILED

Explanation: An I/O error occurred when attempting to write a volume table of contents on the volume.

System Action: The volume table of contents is not written on the volume, and the command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31006I VOLUME LABEL CREATION FAILED

Explanation: An I/O error occurred when attempting to write a volume label.

System Action: A volume label was not written, and the command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: The failing CCW, CSW, and sense information should be examined to aid in determining the cause of the failure.

During minimal initialization, writing of the volume label is the first write operation that occurs on the volume. This message can be an indication that the volume is in read only mode.

If the error is caused by a data check an INSPECT of cylinder 0, track 0 may be all that is required.

Other failure conditions may require the aid of a customer service representative.

Problem Determination: Table 1, 2, 3, 4, and 29.

ICK31007I ALLTRACKS NOT SUPPORTED FOR THIS FUNCTION

Explanation: The ALLTRACKS parameter is only valid for NOCHECK NOASSIGN processing

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Limit the amount of track specified by using the range parameters.

Problem Determination: Not applicable.

ICK31012I VOLID NOT SPECIFIED AND NO VOLUME LABEL EXISTS

Explanation: A volume being initialized must either contain a volume label from which the old volume serial can be used, or the VOLID parameter must specify a new volume serial number.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Be sure that the correct volume is mounted, or specify the VOLID parameter, and reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31013I UNRECOVERABLE TRACK THRESHOLD MET

Explanation: Multiple failures have occurred while writing the home address/record zero on multiple tracks. This message follows many ICK21000 and/or ICK21001 messages.

System Action: The command terminates.

Operator Response: None.

Programmer Response: Prior messages should be examined (including the CCW, CSW, and sense that caused the I/O) to determine the cause of the failure. This message may be caused if the read/write mode switch is set to READ mode, or some other condition is causing the inability of ICKDSF to write on the volume. Inability to write on the volume is characterized by the previous ICK21000 messages being for the first 10 successive alternate tracks on the volume.

If running under VM (either stand-alone or an operating system version running under VM), this message can be an indication that a diagnostic or media maintenance function was attempted against a device that was a minidisk and/or LINK to the userid. Diagnostic and media maintenance functions must be done on devices that are ATTACHed to the userid. See Appendix B, "VM Support" on page 227 for more information on VM support.

Assistance may be required from the customer service representative.

Problem Determination: Not applicable.

ICK31015I UNABLE TO READ VOLUME LABEL

Explanation: An I/O error occurred when attempting to read the volume label to verify the volume serial number.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command and specify the NOVERIFY parameter, or reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31019I NEXT-AVAILABLE-ALTERNATE POINTER OR COUNTER IN VTOC IS INVALID

Explanation: The value of the pointer to the next available alternate track or the count of the number of alternate tracks that are available is in error.

System Action: Processing continues. If it is possible, the value in the VTOC is updated when processing completes.

Operator Response: None.

Programmer Response: Reinitialize the volume at either the minimal or the maximal level.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31022I UNABLE TO CHANGE VOLUME LABEL

Explanation: An I/O error occurred while attempting to change the volume serial number, the owner identification, or the address of the volume table of contents in the volume label.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31023I INVALID UNITADDRESS, SYSNAME OR DDNAME SPECIFIED

Explanation: The channel and unit addresses specified do not exist in the system I/O configuration, or the DDNAME is incorrect, or the SYSNAME is incorrect.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the value substituted for *ccuu*, or for *dname*, or for *sysxxx*, then reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31024I UNABLE TO OPEN VOLUME

Explanation: The volume that was specified cannot be opened. This could be caused by:

- The DD statement is missing or invalid.
- The channel/unit address is invalid.
- When processing in a shared environment, the device is not in an offline status on the system executing Device Support Facilities.
- I/O errors associated with the volume.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: The status of the volume should be checked. Previous messages should be examined to aid in determining the volume status.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31025I VOLUME NOT MOUNTED PRIVATE

Explanation: When online initializing, inspecting, or reformatting is requested, the specified volume must be mounted PRIVATE to prevent interference by other jobs.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Be sure the volume is mounted PRIVATE, or perform the job offline.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31026I COMMAND INVALID FOR UNINITIALIZED VOLUME

Explanation: The INSPECT and REFORMAT commands can only be specified for a previously initialized volume.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Initialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31027I UNABLE TO WRITE VOLUME LABEL DURING VTOC CREATION

Explanation: An I/O error occurred when attempting to rewrite the volume label field that locates the volume table of contents.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31034I 'ASSIGN NOCHECK' PROHIBITED WITH RANGE PARAMETERS

Explanation: The combination of ASSIGN and NOCHECK indicates that the user has determined that an alternate track or block is necessary, independent of the surface analysis processes performed by ICKDSF. This combination of parameters, if used with the range parameters, can easily exhaust the alternate tracks or blocks on a volume, because unconditional alternate assignment takes place.

System Action: Processing terminates.

Programmer Response: If it is determined that an unconditional assignment of alternate tracks or blocks is necessary, use ASSIGN and NOCHECK with the TRACKS or BLOCKS parameter.

ICK31035I UNABLE TO READ FORMAT-4 DSCB

Explanation: An I/O error occurred when attempting to read the format-4 data set control block (DSCB) for the volume table of contents. This record is read, then rewritten, to indicate the location.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31036I UNABLE TO INITIALIZE 2314 MINI DISK

Explanation: For the 2314 minidisk specification, initialization of less than 2 cylinders was requested. This is not acceptable because one cylinder is always used as a cylinder of alternate tracks.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the MIMIC(MINI(n)) parameter, and reissue the command.

ICK31043I NO STORAGE AVAILABLE TO HOLD RECORD BEING PRESERVED, TRACK CCHH=X'cccc hhhh'

Explanation: To preserve the contents of a track during surface checking, the complete track is read into a dynamically allocated buffer in accordance with the track capacity of the volume. There is insufficient space in which to allocate this buffer.

System Action: The track under inspection is not surface checked to avoid loss of data, and the command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Increase the available storage space, or specify the NOPRESERVE parameter if the data may be destroyed, then reissue the command.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31046I UNABLE TO WRITE FORMAT-4 DSCB AT CCHH=X'cccc hhhh'

Explanation: An I/O error occurred when attempting to update the volume table of contents. The address of the first available alternate track and the number of alternate tracks could not be written in the VTOC.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume and specify a different cylinder and track location for the volume table of contents, or issue the INSPECT command to assign an alternate track.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31049I UNITADDRESS SPECIFIED FOR VOLUME MOUNTED ON-LINE

Explanation: The UNITADDRESS parameter is valid for offline processing only, but this device was mounted online.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Mount the volume offline, or specify the DDNAME parameter.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31052I MINIMAL INITIALIZATION NOT PERMITTED FOR MSS STAGING PACK

Explanation: To initialize a pack as a Mass Storage System staging pack, the VALIDATE or the CHECK parameter must be specified to initialize the pack at medial or maximal level, respectively.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reissue the command with the VALIDATE parameter specified.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31054I **DEVICE NOT SUPPORTED FOR THE SPECIFIED FUNCTION**

Explanation: The volume that was specified for the command is not supported for the function as defined by the combination of parameters used. For example, ANALYZE is not valid for a 2314 device. See Figure 2 on page 7 for other incompatible functions.

System Action: Processing of the command is terminated.

Operator Response: None.

Programmer Response: Correct the control statement, and reissue the command.

ICK31056I **VTOC LOCATION IN VOLUME LABEL INCORRECT**

Explanation: The VTOC location specified in the standard volume label is incorrect. Therefore, addressability to the volume table of contents is lost.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume, and reestablish addressability to the volume table of contents.

ICK31057I **INVALID DEVICE TYPE: VTOC INDEX NOT SUPPORTED ON THIS DEVICE**

Explanation: The device type is not a valid device type for initialization with a VTOC index.

System Action: The creation of the VTOC index is not attempted. INIT command processing continues.

Operator Response: None.

Programmer Response: None.

ICK31058I **ccuu I/O ERROR DETECTED DURING INDEX CREATION: ERROR CODE= ***

Explanation: An I/O error occurred during VTOC index creation processing. Error codes:

```
1 - Error occurred in reading the
    volume label
2 - Error occurred in reading the
    VTOC
3 - Error occurred in writing the
    VTOC
4 - Error occurred in writing
    index records
5 - Index extent was violated;
    increase index size
```

System Action: VTOC index creation processing is terminated. The VSE volume indicator is turned on in the VTOC. INIT command processing continues.

Operator Response: Refer the error indication to the system coordinator for Problem Determination.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

ICK31059I **INDEX STARTING LOCATION INVALID AS SPECIFIED**

Explanation: The INDEX parameter on the INIT command statement is invalid. The starting location, as specified, is not valid for one of the following reasons:

- It caused an overlap with the VTOC.
- It defined cylinder 0, track 0 as the starting location of the index data set.
- It was outside the physical limits of the device.
- The relative track specified is invalid for the device.

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: Check the values specified for the INDEX parameter, and correct the value in error. Resubmit the job.

ICK31061I **ccuu VTOC INDEX CREATION SUCCESSFUL: VOLUME NOW IN INDEX FORMAT**

Explanation: The VTOC index was successfully created on the volume.

System Action: None.

Operator Response: None.

Programmer Response: None.

ICK31062I **ccuu VTOC INDEX CREATION FAILED: RETURN CODE= 12**

Explanation: Index creation was unsuccessful.

System Action: None.

Operator Response: None.

Programmer Response: Check the job output for additional information on the reason for job failure.

ICK31063I **UNABLE TO READ HOME ADDRESS BEFORE DATA SAVED FROM CCHH=X'cccc hhhh'**

Explanation: Before the data on a track is read, the home address is read from the specified track. This read operation failed.

System Action: Processing continues with the next track. The return code is set to 8.

Operator Response: None.

Programmer Response: An attempt should be made to recover the data on the track if necessary. (This may not be possible because the home address cannot be read.) Inspection of the failing track can then take place with NOPRESERVE specified, to cause all possible recovery actions to take place.

Note: No further inspection of this track takes place; no information regarding the condition of this track is retained or included in the map when processing completes.

Problem Determination: Not applicable.

ICK31064I **UNABLE TO READ RECORD ZERO BEFORE DATA SAVED FROM CCHH=X'cccc hhhh'**

Explanation: Before the data on a track is read, record zero is read from the specified track. This read operation failed.

If the sense information is all zeros, and the CSW indicates only channel end/device end/incorrect length, it could be an indication that ICKDSF processing was previously aborted while executing on this track.

System Action: Processing continues with the next track. The return code is set to 8.

Operator Response: None.

Programmer Response: An attempt should be made to recover the data on the track if necessary. (This may not be possible because record zero cannot be read.) Inspection of the failing track can then take place with NOPRESERVE specified, to cause all possible recovery actions to take place.

Note: No further inspection of this track takes place; no information regarding the condition of this track is retained or included in the map when processing completes.

Problem Determination: Not applicable.

ICK31066I **PRESERVE RECOVERY FAILED-DATA MAY BE LOST FOR TRACK CCHH=X'cccc hhhh'**

Explanation: This message follows message ICK21047I if the preserved data cannot be successfully written to an alternate track.

System Action: Processing terminates for this command.

Operator Response: None.

Programmer Response: Analysis of the failing CCW, CSW, and sense byte(s) (which accompany the messages) is required to determine the cause of the failure.

When the failing situation has been determined and corrected, an INSPECT of the failing track is required to ensure that the track is in a proper format. Data recovery procedures may be required for the previous data on the track.

Problem Determination: Not applicable.

ICK31082I **DEVICE TYPE OF THE SPECIFIED UNIT ADDRESS IS INVALID FOR THIS COMMAND**

Explanation: The device type is not acceptable for this command.

System Action: The command terminates.

Operator Response: None.

Programmer Response: A previous message should indicate the device type of the specified unit. The description for the input command should be examined to determine the valid device types for the specified command.

Problem Determination: Not applicable.

ICK31096I **UNABLE TO READ VOLUME LABEL - CANNOT VERIFY VOLUME**

Explanation: An INSPECT has been requested on a volume that does not contain a volume label and verify is specified, or the volume is online to an MVS operating system.

System Action: Because verification cannot take place without a volume label (and/or the operating system cannot function with an online volume that contains an unreadable volume label), the command terminates.

Operator Response: None.

Programmer Response: Either use the INIT command to write a volume label, rerun the INSPECT command with the NOVERIFY parameter, and/or vary the device offline.

Problem Determination: Not applicable.

ICK31300I **VERIFICATION FAILED: VOLUME-SERIAL WRONG**

Explanation: The VERIFY parameter was specified and the specified volume serial number does not match the volume serial number found in the volume label.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Ensure that the correct volume is mounted, correct the volume serial number specified in the command, or specify the NOVERIFY parameter.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31301I **VERIFICATION FAILED: OWNER-ID WRONG**

Explanation: The VERIFY parameter was specified, and the specified owner identification does not match the owner identification found in the volume label.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Ensure that the correct volume is mounted, correct the owner identification specified in the command, or specify the NOVERIFY parameter.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31303I **UNABLE TO VERIFY AUTHORIZATION FOR PROTECTED DATA SET**

Explanation: At attempt was made to scratch a data set, but the user was not authorized to do so.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Obtain the proper authorization, or reissue the command in the offline mode.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31304I **SYSTEM OPERATOR DID NOT CONFIRM DATA SET PURGING**

Explanation: The operator replied T to message ICK001D.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Determine if the data may be purged and, if so, reissue the command and reply U to message ICK001D.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31305I UNEXPIRED OR PASSWORD PROTECTED DATA SET FOUND ON VOLUME

Explanation: When attempting to process an online volume using INIT with NOPURGE or INSPECT with NOPRESERVE, unexpired or password-protected data sets were found that prevent initialization or inspection of the volume.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: If the unexpired or password protected data sets can be purged, reissue the INIT command with the PURGE parameter or the INSPECT command with the PRESERVE parameter, or process the volume offline.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31324I VOLUME CONTAINS VSAM DATA SET(S)

Explanation: The volume being processed online contains one or more VSAM data sets.

System Action: The data sets are not destroyed, and the command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Use Access Method Services to delete the VSAM data sets, or initialize the volume offline.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31325I OPERATOR REFUSED PURGE PERMISSION

Explanation: The operator responded T to message ICK003D.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Determine why the request to process was refused, and reissue the command.

Problem Determination: Not applicable.

ICK31326I IMPROPER RACF ACCESS AUTHORITY

Explanation: A volume that is RACF protected cannot be processed unless the user has the proper level of authorization.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Obtain the proper level of authorization, or process the volume in offline mode.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31327I NO STORAGE AVAILABLE FOR MAXIMUM TRACK CAPACITY RECORD

Explanation: No storage was available to build the bit pattern in a main storage buffer that would eventually be written on the volume during surface check.

System Action: The command terminates. Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Specify a larger region size for the job step, and resubmit the job.

ICK31330I UNIT PARAMETER INVALID

Explanation: The UNIT parameter has been specified and is invalid in this operating system environment.

System Action: Execution is terminated.

Operator Response: None.

Programmer Response: Check the accepted parameters for the specified command to determine what parameters are valid for this operating system.

Problem Determination: Not applicable.

ICK31331I REALADDR PARAMETER INVALID

Explanation: You specified the REALADDR parameter which is invalid in this operating system environment.

System Action: Execution terminates.

Programmer Response: Check the accepted parameters for the specified command to determine the valid parameters for this operating system.

Operator Response: none.

ICK31335I SPECIFIED RANGE IS INVALID WHEN USED WITH REALADDR PARAMETER

Explanation: When you use the INSPECT REALADDR, the specified addresses cannot exceed one cylinder for CKD devices, or one block for FBA devices

System Action: Execution terminates.

Programmer Response: Check the accepted parameters for the specified command to determine the valid parameters.

Operator Response: none.

ICK31396I TOO MANY DATA CHECKS ON VOLUME

Explanation: During processing for the specified volume, more data checks were detected than are considered reasonable for this device type.

System Action: The command terminates.

Operator Response: None.

Programmer Response: This message is an indication that an abnormal amount of data checks have occurred for this volume. If the volume was expected to have a large number of data checks, the command can be reexecuted. Otherwise, the system programmer might want to seek the aid of a customer engineer. It might to useful to run an ANALYZE SCAN of the volume and provide that output, in addition to the current ICKDSF output, to the service representative.

Problem Determination: Not applicable.

ICK31403I ccuu DATA SCAN VALUES INVALID, testing terminated

Explanation: The relative block number(s), tracks, or cylinders specified by the LIMITS parameter are invalid.

System Action: Data verification is not done.

Operator Response: Specify valid relative block number(s), tracks, or cylinders in the LIMITS parameter, and resubmit the request.

ICK31404I ccuu VOLUME HAS UNFORMATTED DATA BLOCKS

Explanation: An attempt was made to read a data block which was not initialized (data field not written).

System Action: After 504 data checks, the ANALYZE command is terminated.

Operator Response: Perform appropriate procedures to recover all desired data from the volume and then reinitialize the volume, using the ICKDSF INIT command specifying the CHECK parameter.

ICK31410I MAIN STORAGE NOT AVAILABLE, TESTING TERMINATED

Explanation: Dynamic acquisition of storage for work areas and control blocks failed.

System Action: ANALYZE command processing is terminated.

Operator Response: Increase main storage size.

ICK31412I DEVICE NOT READY, TESTING TERMINATED

Explanation: A condition occurred that caused "Intervention Required" to be posted in the failing CSW sense byte.

System Action: The ANALYZE command is terminated.

Operator Response: Ready the drive, and reissue the command.

ICK31413I DATAVER DATA FORMAT UNACCEPTABLE ON CYLINDER cccc.

Explanation: One of the following conditions might have occurred at hexadecimal cylinder cccc on the device:

- The data on the identified cylinder is written in a format unreadable by the standard IBM CCWs. This condition can be further examined and/or corrected using the INIT and/or INSPECT commands.
- A drive error (such as dropping ready) occurred during the test. The condition that caused the drive error is usually intermittent or marginal, and was not detected during the drive tests. Hardware support is required.
- The volume under test is also accessed by another program, and the other program erased a record that ANALYZE might be trying to reread. This condition is not probable, but if it occurs, rerun the ANALYZE job

System Action: ANALYZE processing terminates with a return code of 8.

Operator Response: None.

Programmer Response: See "Explanation." Check subsequent messages for status of drive.

Problem Determination: See "Programmer Response."

ICK31505I ccuu VTOC FORMAT IS CURRENTLY **FORMAT, REQUEST REJECTED

Explanation: The format of the VTOC is currently the same as the format requested in the BUILDIX command statement. '**' identifies the format of the VTOC, either OS or IX.

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: Check the command statement. The format requested must be opposite from the current VTOC format as identified in this message. Be certain that the JCL DD statement correctly identifies the volume. Change the statement(s) where required, and resubmit the job.

ICK31509I ccuu DIRF FLAG SET IN VTOC, BUILDIX CANNOT PROCEED

Explanation: An error occurred during VTOC processing on a previous job, causing the DADSM interrupt flag to be set in the VTOC. The VTOC is not accurate.

System Action: Execution of the command is terminated.

Operator Response: Prepare a job that will cause reconstruction of the VTOC, and run it against the volume. (For example, run an IEFBR14 job to allocate a temporary data set to the volume.) On conclusion of this job, resubmit the BUILDIX command.

Note: The VTOC DIRF bit and the indexed bit (DS4VTOC=5) should never be set on together. If they are, the IEFBR14 job, mentioned in the previous paragraph, cannot correct the VTOC. To correct this condition, you can use the SUPERZAP program, available with your system, to turn the DIRF bit off.

Programmer Response: None.

ICK31510I ccuu BUILDIX REQUEST CANCELLED DUE TO OPERATOR ACTION

Explanation: The operator replied 'CANCEL' to a message requiring a response.

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: Check with the operator for the reason for cancellation of the job.

ICK31511I ccuu CVAF ERROR: RETURN CODE= ERROR CONDITION= ***

Explanation: The common VTOC access facility returned to Device Support Facilities with a return code indicating either a logical error or a physical error. Return codes have the following meanings:

4, 12
Indicate logical errors

8
Indicates an invalid index structure

16
Indicates an I/O error

The error condition will be printed only for a return code of 4 or 8. See Appendix B of *Data Facility Device Support: User's Guide and Reference* for more information on these error conditions.

For MVS/XA, see *MVS/Extended Architecture System-Data Administration.*

For MVS/370, see *MVS/370 System Programming Library: Data Management.*

System Action: Execution of the command terminates with the following conditions existing, depending on the return code:

4, 8, 12
The volume is left in OS VTOC format.

16
The volume is left as it was before the command was issued.

Operator Response: None.

Programmer Response: Refer to the DADSM program logic manual for a full explation of CVAF return codes and error condition codes.

ICK31512I **ccuu ERROR: SYS1.VTOCIX. IS A VIO DATASET. BUILDIX TERMINATED.**

Explanation: The index data set was allocated as a VIO data set and is not supported by the BUILDIX command.

System Action: Execution of the command terminates with a return code of 12. The VTOC is left unchanged.

Operator Response: None.

Programmer Response: Ensure that allocation of the index data set specifies a permanent data set. Check the JCL statement that defines the index data set and change the parameter. Resubmit the job.

ICK31514I **INDEXED VTOC FACILITY NOT ON SYSTEM. BUILDIX TERMINATED.**

Explanation: A request was made to change a VTOC to IXVTOC format, but the system does not contain indexed VTOC programming support.

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: None.

ICK31515I **ccuu BUILDIX COMMAND FAILED.**

Explanation: An error occurred that caused execution of the command to be terminated.

System Action: Execution of the command is terminated.

Operator Response: None.

Programmer Response: Check the job output for messages describing the type of failure.

ICK31516I **ccuu I/O ERROR DETECTED DURING VTOC CONVERSION: ERROR CODE= ***

Explanation: An I/O error occurred during BUILDIX processing. Error codes:

```
1 - Error occurred in reading the
    volume label
2 - Error occurred in reading the
    VTOC
3 - Error occurred in writing the
    VTOC
4 - Error occurred in writing
    index records
5 - Index extent was violated;
    increase index size
```

System Action: Execution of the BUILDIX command is terminated.

Operator Response: Refer the error indication to the system coordinator for Problem Determination.

Programmer Response: None.

ICK31517I **ccuu ERROR: VOLUME IS A DOS STACKED PACK**

Explanation: The volume being processed has a VTOC on the first track of the volume. VTOC conversion is not possible on such a volume.

System Action: Execution of the command is terminated.

Operator Response: None

Programmer Response: Correct the problem on the volume, possibly by initializing the volume with an INIT command. Note, however, that this will purge all existing data on the volume.

ICK31518I **ccuu ERROR: VOLUME CONTAINS SPLIT CYLINDER EXTENTS**

Explanation: It was found that the volume contains one or more shared extent data sets. These are not supported by the BUILDIX command.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

Programmer Response: None.

ICK31519I **ccuu ERROR: SYS1.VTOCIX. DATASET NOT FOUND ON VOLUME**

Explanation: The index data set could not be found on the volume.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

Programmer Response: Check the job control statements to see if a DD statement exists that will cause allocation of the index data set. If the statement exists, check if the name is specified correctly. Correct the error and resubmit the job.

ICK31520I **ccuu ERROR: DUPLICATE INDEX DATASET NAME FOUND ON VOLUME**

Explanation: Two data sets were found on the volume that had names beginning with SYS1.VTOCIX. Only one is allowed per volume.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

Programmer Response: Submit a job to scratch one of the data sets, or scratch both data sets if new index allocation is desired.

ICK31521I **ccuu ERROR: INDEX DATASET EXTENT NOT CONTIGUOUS**

Explanation: This data set was found to occupy more than one extent. The index data set must occupy one, and only one, extent.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: None.

Programmer Response: Scratch the invalid index data set, and resubmit the job with a job control statement that will ensure that only one extent is allocated to the index.

ICK31522I INVALID DDNAME SPECIFIED

Explanation: The DDname is incorrect in the command statement.

System Action: Execution of the BUILDIX command is terminated.

Operator Response: Correct the job control statement, and resubmit the job.

Programmer Response: None.

ICK31526I CONVERSION TO OSFORMAT COULD NOT COMPLETE SUCCESSFULLY

Explanation: At the conclusion of BUILDIX processing to convert to OSFORMAT, the VTOC indicates an error condition.

System Action: BUILDIX terminates abnormally.

Operator Response: None.

Programmer Response: This message is usually an indication that the OS VTOC on the volume contains an error. The VTOC should be examined to determine the nature of the error.

Problem Determination: Not applicable.

ICK31528I INDEX DATA SET CANNOT START AT CYLINDER 0, TRACK 0

Explanation: You specified cylinder zero track zero for the index data set location. This is an invalid location.

System Action: BUILDIX processsing terminates. None

Operator Response: None.

Programmer Response: Respecify the command using a valid location.

ICK31602I INCORRECT DEVICETYPE, MAPALT TERMINATED

Explanation: The device type specified is not supported by the MAPALT command.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Specify the correct device type in the DEVICETYPE keyword of the command statement, and resubmit the job. MAPALT can only be executed on fixed block devices in fixed block mode.

Programmer Response: None.

ICK31603I UNABLE TO OPEN VOLUME

Explanation: The volume that was specified cannot be opened. The ASSGN statement could be missing or invalid, or the channel and unit address could be invalid.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Correct the error, and resubmit the job.

Programmer Response: None.

ICK31604I ccuu LIMITS PARAMETER INVALID AS SPECIFIED, MAPALT TERMINATED

Explanation: The relative block number(s) specified by the LIMITS parameter are invalid.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Correct the value(s) in the LIMITS parameter, and resubmit the job.

ICK31605I ccuu UNRECOVERABLE I/O ERROR DETECTED, MAPALT TERMINATED

Explanation: An unrecoverable I/O error (other than a data check) was encountered.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Check the job output, which will contain diagnostic information to aid in analyzing the error. Execute the Device Support Facilities ANALYZE command as a Problem Determination aid, and follow installation procedures for recovery from this type of error.

Programmer Response: None

ICK31607I ccuu MAPALT ABNORMALLY ENDED, REPORT MAY BE INCOMPLETE

Explanation: This message is printed after an error has been encountered that caused execution of the command to be terminated. The report may be incomplete.

System Action: None.

Operator Response: Check previous messages on job output, and follow installation procedures.

Programmer Response: None.

ICK31611I MAIN STORAGE NOT AVAILABLE, MAPALT TERMINATED

Explanation: Dynamic acquisition of storage for control blocks and work areas failed.

System Action: Execution of the MAPALT command is terminated.

Operator Response: Increase main storage size, and resubmit the job.

Programmer Response: None.

ICK31700I VTOC EXTENT INVALID

Explanation: Either the starting or ending block number as calculated from the FBAVTOC parameter is invalid.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the FBAVTOC parameter so the ending parameter is within the device limits, and rerun the job.

Problem Determination: Not applicable.

ICK31702I AN UNRECOVERABLE I/O ERROR OCCURRED DURING RECLAIM PROCESSING

Explanation: An unrecoverable i/o error occurred during INIT reclaim processing.

System Action: Command processing is terminated.

Operator Response: None.

Programmer Response: Prior messages should be examined (including the CCW, CSW, and sense that caused the I/O) to determine the cause of the failure. The INIT command should be re-run after the problem has been resolved.

Assistance may be required from the IBM Customer Service Representative.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31705I ALTERNATE BLOCKS EXHAUSTED

Explanation: A prime block needed an alternate but all available alternate blocks have already been assigned.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume with RECLAIM specified to free up some alternate blocks. If this is not successful, seek hardware support.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31706I AN UNRECOVERABLE I/O ERROR OCCURRED DURING SURFACE ANALYSIS

Explanation: An unrecoverable i/o error occurred during INIT surface analysis.

System Action: Command processing is terminated.

Operator Response: None.

Programmer Response: Prior messages should be examined (including the CCW, CSW, and sense that caused the I/O) to determine the cause of the failure. The INIT command should be re-run after the problem has been resolved.

Assistance may be required from the IBM Customer Service Representative.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31709I I/O ERROR FORMATTING BLOCK IDENTIFIER

Explanation: An uncorrectable error occurred while formatting the block identifier field of a block on a 3370.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31710I I/O ERROR FORMATTING BAD BLOCK XXXXXX

Explanation: An uncorrectable error occurred while formatting the defective block xxxxxx.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31711I CANNOT READ SA INFORMATION

Explanation: Because of an I/O error, the surface analysis information (factory-flagged list) cannot be read.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31712I INVALID SA INFORMATION FORMAT

Explanation: The data format of SA information (factory-flagged list) is in error. Typically, the end of data is not found.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31713I CANNOT READ PRIME CYLINDER IDS

Explanation: An uncorrectable I/O error occurred while reading the ID fields of prime blocks.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Seek hardware support.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31716I INVALID PARAMETER(S) FOR DEVICE TYPE: list of parameters

Explanation: Parameter(s) in the parameter list are invalid for device type being processed. For example, the VTOC parameter is specified for an INIT of an FBA device, or the CHECK parameter is specified for the IBM 3375 or 3380.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct parameter(s), and rerun the job.

Problem Determination: Not applicable.

ICK31717I VERIFICATION FAILED: VOL1 LABEL DOES NOT EXIST

Explanation: Verification was requested, but failed because the volume being initialized does not have a VOL1 label to use for the verify. This may indicate this is the wrong volume.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Resolve the conflict, and rerun the job.

Problem Determination: Not applicable.

ICK31718I **INIT FAILED: VOLID NOT SPECIFIED AND I/O ERROR READING VOLUME LABEL**

Explanation: The VOLID parameter is not specified, and because of an I/O error, reading the volume label, a volume serial is not available.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Try specifying VOLID. If this is not successful, investigate the cause of the I/O error reading the volume label.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31722I **VERIFICATION FAILED: I/O ERROR READING VOL1 LABEL**

Explanation: An uncorrectable I/O error occurred reading the VOL1 label. Verification cannot continue

System Action: Diagnostic information is printed on the output device. The command is terminated. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume, specifying NOVERIFY and a VOLID. Start recovery procedures for data on the volume.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31728I **FBA MINIDISK NOT SUPPORTED IN THIS ENVIRONMENT**

Explanation: Fixed block architecture minidisks are not supported in VSE, VS1, or MVS environments.

System Action: The command terminates. Processing continues with the next command.

Operator Response: None.

Programmer Response: Execute the command in the stand-alone environment.

Problem Determination: Not applicable.

ICK31748I **COMMAND CANNOT BE EXECUTED - DEVICE ACCESS LIMITED**

Explanation: The current command cannot be executed because certain types of access are prohibited for the device you have specified. For example, a diagnostic control CCW cannot be issued for a VM minidisk.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check that the command you issued is compatible with the specified device.

Problem Determination: Not applicable.

ICK31749I **DEVICE TYPE NOT SUPPORTED IN THIS ENVIRONMENT**

Explanation: Fixed block architecture devices are not supported in the VS1 or MVS environment.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Execute the command in the stand-alone or VSE environment.

Problem Determination: Table I, items 1, 3, and 4.

ICK31750I **TRACKS OR ALLTRACKS PARAMETER REQUIRED FOR CKD DEVICES**

Explanation: BLOCKS was specified for a count-key-data device. You must specify either TRACKS or ALLTRACKS for these devices.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Resubmit the job, specifying either TRACKS or ALLTRACKS.

Problem Determination: Table I, items 1, 3, and 4.

ICK31751I **BLOCKS PARAMETER REQUIRED FOR FBA DEVICE**

Explanation: TRACKS OR ALLTRACKS was specified for a fixed block architecture device. You must specify BLOCKS for these devices.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Resubmit the job, specifying BLOCKS.

Problem Determination: Table I, items 1, 3, and 4.

ICK31753I **NO VALID BLOCK NUMBERS SPECIFIED**

Explanation: All the block numbers specified by the BLOCKS parameters were invalid. This message is preceded by one or more ICK11752I messages.

System Action: The command terminates. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Resubmit the job, specifying the correct block number(s) in the BLOCKS parameter.

Problem Determination: Table I, items 1, 3, and 4.

ICK31756I **READ FOR PRESERVE OF BLOCK xxxxxx FAILED: UNRECOVERABLE I/O ERROR**

Explanation: The user data on block xxxxxx could not be read because of unrecoverable I/O errors.

System Action: Diagnostic information is printed on the output device and processing of the block is terminated to prevent loss of the user data on the block. Command processing continues with the next valid block specified in the BLOCKS parameter.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31757I **READ FOR PRESERVE OF BLOCK xxxxxx FAILED: INSPECT ECC CORRECTION LOGIC FAILED**

Explanation: The read for the user data on block xxxxxx failed with an error correction code (ECC) correctable error, and the ECC correction information in the sense did not correspond to the expected information for the executed CCW chain.

System Action: Diagnostic information is printed on the output device, and processing of the command is terminated. Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4 and 29.

ICK31758I **USER DATA ON BLOCK xxxxxx LOST**

Explanation: Command processing failed with a permanent error. Device Support Facilities was unable to restore the user data on block xxxxxx.

System Action: The command is terminated. Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Start recovery procedures for the data set containing block xxxxxx.

Problem Determination: Table I, items 1, 2, 3, 4 and 29.

ICK31762I **CHECK FUNCTION FOR BLOCK xxxxxx FAILED: UNRECOVERABLE I/O ERROR**

Explanation: An unrecoverable I/O error occurred during surface analysis of block xxxxxx.

System Action: Diagnostic information is printed on the output device. The command is terminated, and Device Support Facilities continues with the next command. If you specified PRESERVE, you receive either message ICK31758I or message ICK01759I on the output device to indicate the status of the user data on the block.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4 and 30.

ICK31763I **ASSIGN FUNCTION FOR BLOCK xxxxxx FAILED: NO ALTERNATE BLOCKS AVAILABLE**

Explanation: An attempt was made to assign an alternate to block xxxxxx, and there are no more alternates available.

System Action: The command is terminated. Device Support Facilities continues with the next command. If you specified PRESERVE, you receive either message ICK31758I or message ICK01759I on the output device to indicate the status of the user data on the block.

Operator Response: None.

Programmer Response: Reinitialize the device.

Problem Determination: Table I, items 1, 2, 3, and 4

ICK31764I **ASSIGN FUNCTION FOR BLOCK xxxxxx FAILED: FORMAT DEFECTIVE BLOCK ERROR**

Explanation: An unrecoverable error occurred while using the format defective block CCW chain to assign a new alternate to block xxxxxx.

System Action: Diagnostic information is printed on the output device. The command is terminated, and Device Support Facilities continues with the next command. If you specified PRESERVE, you receive either message ICK31758I or message ICK01759I on the output device to indicate the status of the user data on the block.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31766I **ASSIGN FUNCTION FOR BLOCK xxxxxx FAILED: NO NON-DEFECTIVE ALTERNATES FOUND**

Explanation: The format defective block CCW chain was retried 10 times for block xxxxxx and all ten assigned alternates failed surface analysis.

System Action: The command is terminated, and Device Support Facilities continues with the next command. If you specified PRESERVE, you receive either message ICK31758I or message ICK01759I on the output device to indicate the status of the user data on the block.

Operator Response: None.

Programmer Response: See Problem Determination

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31767I **ASSIGN FUNCTION FOR BLOCK xxxxxx FAILED: UNRECOVERABLE I/O ERROR OCCURRED DURING SURFACE ANALYSIS**

Explanation: An unrecoverable I/O error occurred during surface analysis of the alternate assigned to block xxxxxx.

System Action: Diagnostic information is printed on the output device. The command is terminated and Device Support Facilities continues with the next command. If you specified PRESERVE, you receive either message ICK31758I or message ICK01759I on the output device to indicate the status of the user data on the block.

Operator Response: None.

Programmer Response: See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31768I **INVALID LOGICAL AND PHYSICAL DEVICE COMBINATION FOR REQUESTED FUNCTION**

Explanation: The logical and physical device types were not equal. This was determined from the information returned from the sense ID (Sense I/O Type) CCW. Emulation of fixed block architecture devices is not supported by Device Support Facilities.

System Action: The command is terminated, and Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Verify the device configuration is supported by Device Support Facilities. See Problem Determination.

Problem Determination: Table I, items 1, 2, 3, 4, and 29.

ICK31769I DEVICE BLOCK SIZE OF nnnn NOT SUPPORTED

Explanation: The block size (nnnn) information returned by the device from a read device characteristics CCW does not equal 512.

System Action: The command is terminated, and Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: See Problem Determination

Problem Determination: Table I, items 1, 3, 4, and 30.

ICK31772I INSUFFICIENT STORAGE AVAILABLE TO READ VTOC

Explanation: Dynamic acquisition of storage for an input buffer failed.

System Action: The command is terminated and Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Increase the amount of storage available for GETMAIN/GETVIS.

Problem Determination: Table I, items 1, 3, and 4.

ICK31773I VOL1 LABEL NOT FOUND

Explanation: An online volume does not have a VOL1 label.

System Action: The command is terminated, and Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Initialize the volume, or rerun the command in the stand-alone environment.

Problem Determination: Not applicable.

ICK31774I I/O ERROR READING VOL1 LABEL

Explanation: A permanent I/O error occurred reading the VOL1 label.

System Action: Diagnostic information is printed on the output device. The command is terminated, and Device Support Facilities continues with the next command.

Operator Response: None.

Programmer Response: Reinitialize the volume, and start recovery procedures for the data on the volume. The INIT command for reinitialization must specify the NOVERIFY and VOLID parameters.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31775 DATAVER: DATA CHECK EXCEEDED THRESHOLD

Explanation: The number of data checks exceeded the limit set for the specified device. For FBA devices, this limit is 504. For CKD devices, this limit is equivalent to the number of alternate tracks for this device, or 50, whichever is larger.

System Action: Processing terminates with a return code of 12.

Operator Response: None.

Programmer Response: Save the console output and the printer output. Take the action appropriate to your installation procedures for determining if an equipment problem is suspected. If the problem is determined to be equipment-related, contact your IBM customer engineer or service representative.

Problem Determination: See "Programmer Response."

ICK31776I dataset IS A RACF-PROTECTED DATA SET, BUT RACF IS INACTIVE

Explanation: The named data set is marked as RACF protected, but RACF is not active on the system. This precludes Device Support Facilities from checking on the user's authority to alter the data set.

System Action: Processing for this command is terminated after the security function is complete.

Operator Response: None.

Programmer Response: Activate RACF on the system or run in offline mode.

Problem Determination: Not applicable.

ICK31777I UNRECOVERABLE I/O ERROR READING THE VTOC

Explanation: The Device Support Facilities security function tried to read the VTOC and an unrecoverable I/O error occurred.

System Action: Processing for this command is halted and the next command, if any, is processed.

Operator Response: None.

Programmer Response: Investigate the cause of the I/O error. If required, run the command in an offline mode or stand-alone environment.

Problem Determination: Table I, items 1, 2, 3, 4, and 30.

ICK31779I dataset IS A RACF-PROTECTED DATA SET, BUT NO RACF PROFILE EXISTS

Explanation: The named data set is marked as RACF protected but the RACF data set does not contain a profile for the named data set.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: Contact the system RACF administrator. If required, run the command in an offline mode or a stand-alone environment.

Problem Determination: Not applicable.

ICK31780I dataset IS A RACF-PROTECTED DATA SET, USER IS NOT AUTHORIZED

Explanation: The named data set is RACF protected and the user does not have authority to alter the data set.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: Contact the system RACF administrator.

Problem Determination: Not applicable.

ICK31782I **dataset IS A PASSWORD PROTECTED DATA SET, NO PASSWORD GIVEN**

Explanation: The named data set is password protected but the user gave no password for it.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: Supply a password for the named data set, using the PASSWORDS parameter, and rerun the job.

Problem Determination: Not applicable.

ICK31783I **dataset IS A PASSWORD PROTECTED DATA SET, USER HAS NO ALTER AUTHORITY**

Explanation: The named data set is password protected and the password supplied by the user does not give authority to alter the data set.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: Supply the correct password and rerun the job.

Problem Determination: Not applicable.

ICK31784I **dataset IS A PASSWORD PROTECTED DATA SET, WRONG PASSWORD GIVEN**

Explanation: The named data set is password protected and the password supplied by the user for this data set is wrong.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: Supply the correct password and rerun.

Problem Determination: Not applicable.

ICK31785I **USER SECURITY EXIT ROUTINE RETURNED AN INVALID CODE = X'code'**

Explanation: The user security exit returned a code other than the codes documented in Appendix H, "User Security Exit Module" on page 319, of this manual.

System Action: Processing for this command is terminated.

Operator Response: None.

Programmer Response: Contact the owner of your installation's user security exit routine for Device Support Facilities.

ICK31786I **dataset IS A PASSWORD PROTECTED DATA SET, USER EXIT REJECTS DATA SET**

Explanation: The named data set is password protected and the user security exit module indicated that the user should not be allowed to alter this data set.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: Consult the owner of your installation's user security exit module.

Problem Determination: Not applicable.

ICK31787I **TRACK x'00000000' CANNOT BE INSPECTED WITH "NOPRESERVE"**

Explanation: Device Support Facilities does not allow INSPECTing cylinder 0, track 0 in an online mode with the NOPRESERVE option.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: If required, run in an offline mode or in a stand-alone environment.

Problem Determination: Not applicable.

ICK31788I **TRACK x'ccchhhh' IS CONTAINED IN THE VTOC**

Explanation: The track identified in the message is contained in the volume table of contents and cannot be INSPECTed with NOPRESERVE.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: If required, run the job in an offline mode or in a stand-alone environment.

Problem Determination: Not applicable.

ICK31789I **VOLUME volser IS RACF-PROTECTED AND THE USER HAS INSUFFICIENT ACCESS AUTHORITY**

Explanation: The named volume is RACF-protected and the user does not have authority to alter its contents.

System Action: Processing for this command is halted and the next command, if any, is processed.

Operator Response: None.

Programmer Response: Consult the RACF administrator for your installation.

Problem Determination: Not applicable.

ICK31790I **dataset IS A PASSWORD PROTECTED DATA SET, UNRECOVERABLE I/O ERROR IN PASSWORD DATA SET**

Explanation: An unrecoverable I/O error occurred in the system password data set while verifying the user's authority to modify the named data set.

System Action: Processing for this command is halted and the next command, if any, is processed.

Operator Response: None.

Programmer Response: Consult the system programmer at your installation.

Problem Determination: Not applicable.

ICK31791I **dataset IS AN UNEXPIRED DATA SET, USER NOT AUTHORIZED**

Explanation: The expiration date for the named data set has not yet passed, and either the PURGE parameter was not specified on the INIT command, or the command is INSPECT with NOPRESERVE.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: If INIT is used, specify the PURGE parameter and rerun. If INSPECT is used, run with PRESERVE.

Problem Determination: Not applicable.

ICK31792I **dataset IS A VSAM DATA SET, USER NOT AUTHORIZED**

Explanation: The named data set is a VSAM data set and either the PURGE parameter was not specified on the INIT command, or NOPRESERVE was specified on the INSPECT command.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: If INIT is used, specify PURGE and rerun the job. If INSPECT is used, run with PRESERVE.

Problem Determination: Not applicable.

ICK31793I **dataset IS A PASSWORD/VSAM/UNEXPIRED DATA SET, USER EXIT TERMINATES COMMAND**

Explanation: The named data set is a password protected, VSAM or unexpired data set and the user security exit module has indicated that Device Support Facilities should terminate processing for this command.

System Action: Processing for this command is terminated and the next command, if any, is processed.

Operator Response: None.

Programmer Response: Consult the owner of the user security exit module at your installation.

Problem Determination: Not applicable.

ICK31794I **dataset: UNRECOVERABLE I/O ERROR READING FORMAT 2/ FORMAT 3 DSCB**

Explanation: An unrecoverable I/O error occurred while reading a Format 2/Format 3 data set label for the named data set.

System Action: Processing for this command is halted and the next command, if any, is processed.

Operator Response: None.

Programmer Response: Consult the system programmer at your installation and, if required, run in an offline mode or in a stand-alone environment.

Problem Determination: Not applicable.

ICK31796I **TRACK INSIDE VTOC INDEX DATASET CANNOT BE INSPECTED WITH "NOPRESERVE"**

Explanation: One or more tracks to be inspected are contained in the INDEX VTOC data set and cannot be inspected with NOPRESERVE. If the TRACKS parameter is specified, the preceding ICK01795I messages identify the tracks you specified that are inside the INDEX VTOC data set.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: If required, run in an offline mode or a stand-alone environment.

Problem Determination: Not applicable.

ICK31797I **TRACKS INSIDE VTOC CANNOT BE INSPECTED WITH "NOPRESERVE"**

Explanation: One or more tracks to be inspected are contained in the VTOC and cannot be inspected with NOPRESERVE.

System Action: Processing for this command is halted after the security function is complete.

Operator Response: None.

Programmer Response: If required, run in an offline mode or a stand-alone environment.

Problem Determination: Not applicable.

ICK31799I **OPERATOR REFUSED PERMISSION TO RECLAIM SYSTEM RESERVED AREA ON** *ccuu*

Explanation: The operator replied T to message ICK177D

System Action: The command terminates. ICKDSF continues with the next command.

Operator Response: None.

Programmer Response: Determine why the requested processing was refused and resubmit the command.

Problem Determination: Not applicable.

ICK31837I **IPL TEXT EXISTS ON VOLUME. OERATOR REFUSED PERMISSION TO OVERLAY**

Explanation: An attempt was made to replace IPL text on a volume that already contains IPL text. The operator refused permission to replace the IPL text.

System Action: The command terminates.

Operator Response: None.

Programmer Response: Either specify a volume that does not contain IPL text, or instruct the system operator to allow permission.

Problem Determination: Not applicable.

ICK31851I **EXTENDED CKD FUNCTIONS CANNOT BE ACTIVATED - COMMAND TERMINATED**

Explanation: The device has never been online to the operating system, and the extended CKD functions (for example, cache logic) have not been activated. Device Support Facilities surface checking functions cannot be performed.

System Action: Execution of the command is terminated.

Operator Response: The device must be varied online and varied offline (or varied online and the job can be run online) in order to activate the extended CKD functions.

Programmer Response: When the device cannot be varied online (that is, no volume label or duplicate volume label) this procedure will have gone far enough to activate the extended CKD logic, and the job can be rerun offline successfully.

Problem Determination: Not applicable.

ICK32106I RANGE SPECIFICATIONS INVALID

Explanation: The combination and/or values of the parameters specifying the current range are invalid.

Programmer Response: Correct the parameters, and reexecute the job.

ICK32110I TOO MANY TRACKS IN SPECIFIED RANGE FOR SKIP DISPLACEMEMT FUNCTION

Explanation: The total number of tracks to be processed is greater than the maximum allowed for SKIP processing.

Programmer Response: Reexecute the job with a smaller range, or execute the job with the NOSKIP parameter.

ICK32111I TOO MANY TRACKS IN SPECIFIED RANGE

Explanation: The total number of tracks in the specified range is greater than the maximum allowed for INSPECT processing.

Programmer Response: Reexecute the job with a smaller range, or save the data if necessary and use the INITialize command.

ICK32120I PRESERVE BACKUP FUNCTION CANNOT BE ACTIVATED

Explanation: The PRESERVE backup function of the INSPECT command cannot be activated. This message follows ICK12118I.

System Action: Processing terminates.

Programmer Response: The job can be rerun, using HOLDIT.

If the error is temporary, the job can be rerun.

ICK32121I CANNOT WRITE DATA ON THE PRESERVE TRACK

Explanation: An I/O error was encountered while backing up user data on the preserve track.

System Action: Processing terminates.

Programmer Response: The job can be rerun, using HOLDIT.

If the error is temporary, the job can be rerun.

ICK32122I PRESERVE BACKUP FUNCTION CANCELLED

Explanation: An I/O error was encountered while performing the PRESERVE backup function. This message is preceded by a message indicating the reason for cancellation.

System Action: Processing terminates.

Programmer Response: The job can be rerun, using HOLDIT.

If the error is temporary, the job can be rerun.

ICK32127I DATA CANNOT BE RECOVERED, INSPECT TERMINATED

Explanation: This message is issued if you reply 't' to ICK22158I, ICK22130I, ICK12126I or ICK12159I.

ICK32160I CANNOT WRITE DATA ON THE PRESERVE BLOCK

Explanation: An I/O error was encountered while backing up user data on the preserve block.

System Action: Processing terminates.

Programmer Response: The job can be rerun using HOLDIT.

If the error is temporary, the job can be rerun.

ICK32165I TOO MANY BLOCKS IN SPECIFIED RANGE

Explanation: The total number of blocks in the specified range is greater than the maximum allowed for INSPECT processing.

Programmer Response: Reexecute the job with a smaller range, or save the data if necessary and use the INITialize command.

ICK32166I ERROR NOT A DATA CHECK-PROCESSING TERMINATED

Explanation: An I/O error occurred that was not a data check. See the previous messages, CCW, CSW, and sense information to determine the cause of the error.

System Action: The command terminates. Processing continues with the next command.

Programmer Response: Examine the previous messages, CCW, CSW, and sense information to determine the cause of the I/O error.

Operator Response: None.

Problem Determination: See "Programmer Response".

ICK32167I ALTERNATE TRACK CANNOT BE ASSIGNED FOR TRACK CCHH=X'cccc hhhh'

Explanation: During execution of the INSPECT command to unconditionally assign alternate tracks, there was no alternate available for X'cccc hhhh'. No assignment for this track has taken place.

System Action: Processing terminates.

Programmer Response: The alternate tracks are exhausted. If this or subsequent tracks require unconditional alternate track assignment, then INSPECT with RECLAIM should be run for tracks that already have alternates assigned to them. If any of these previous tracks can be reclaimed, then the job to assign this track an alternate unconditionally can be run. Note that the tracks that currently have alternate tracks assigned can be found in the map that is produced at the end of INSPECT processing when MAP is specified. If MAP is not specified, run INSPECT NOCHECK NOASSIGN MAP TRACKS(xxxx,xxxx) for any track, and a map is produced.

Operator Response: None.

ICK32170I - CONTINUE DATA EXISTS FOR THIS VOLUME - RUN INITIALIZE

Explanation: For this INSPECT job, it has been determined that a previous INIT job did not complete. For CKD devices, it is very likely that there exists a track on the volume that is unusable by the operating system. The volume label and the VTOC have not been written on the volume.

System Action: Processing terminates.

Programmer Response: Run an INIT (at least at the minimal level) before rerunning this INSPECT job.

Operator Response: None.

Problem Determination: Not applicable.

ICK32171I PREVIOUS COMMAND DID NOT COMPLETE

Explanation: An INSTALL/REVALidate command did not complete and the volume has been left in an unusable state. This command will be inhibited until the condition is corrected.

System Action: Processing terminates.

Programmer Response: Execute a command that ensures that the device is returned to a usable condition (for example, medial initialization or REVALidate).

Operator Response: None.

ICK32176I DATA CHECK THRESHOLD EXCEEDED ON FOLLOWING HEADS:

Explanation: The data error rate on one or more heads exceeded the data check error rate threshold criterion for the device. Heads in error were printed with the message.

System Action: The command terminates. ICKDSF processing continues with the next command.

Programmer Response: Take action appropriate to your installation's procedures for handling the suspected equipment problems. If the problem cannot be determined, contact your IBM service representative.

Operator Response: None.

ICK32177I DATA SETS EXIST ON SMS-MANAGED VOLUME

Explanation: You are performing an online INIT and there are data sets on the volume you are attempting to initialize as a Storage Management Subsystem (SMS) managed volume.

System Action: Processing terminates

Operator Response: None

Programmer Response: Delete the data sets from the volume or perform the INIT offline.

ICK32203I INSUFFICIENT STORAGE AVAILABLE FOR DATA SET ENQUEUE -- FUNCTION TERMINATED

Explanation: GETMAIN failed while obtaining the storage neccessary to process the data set enqueue procedures.

Because TOLERATE(ENQFAIL) was not specified, processing terminates.

System Action: INSPECT processing terminates.

Programmer Response: Either execute the job, providing more storage, rerun the job specifying TOLERATE(ENQFAIL), or execute the job specifying fewer total tracks.

ICK32310I FUNCTION NOT SUPPORTED FOR READ ONLY MINIDISK

Explanation: The ICKDSF command function you requested cannot be performed on a read only minidisk.

System Action: The command terminates. ICKDSF continues with the next command.

Programmer Response: none.

Operator Response: none.

ICK32315I DEVICE ADDRESS COULD NOT BE FOUND TO ESTABLISH A LINK

Explanation: For the minidisk media maintenance function, ICKDSF is to obtain an unused virtual address to establish a full-pack overlay link. Address range 05FF to 0000 in decending order was checked and and none was found available.

System Action: The command terminates. ICKDSF continues with the next command.

Programmer Response: use CP DETACH to detach an unused virtual address, then re-execute the command.

Operator Response: None.

ICK32316I SYSTEM SUPPORT IS UNAVAILABLE FOR SPECIFIED FUNCTION

Explanation: The required system support code from VM/SP, VM/HPO or VM/XA for the CMS version of ICKDSF does not exist.

System Action: The command terminates.

Programmer Response: Check with your installation to determine if the support is provided for your environment. If the support is available, ensure it is installed before running the job.

Operator Response: None.

ICK32317I ccuu DEVICE DOES NOT EXIST

Explanation: The device address you specified does not exist.

System Action: The command terminates.

Programmer Response: Provide the correct address and re-run the job.

Operator Response: None

ICK32345I CYLINDER OR BLOCK INFORMATION IS REQUIRED FOR THE SPECIFIED FUNCTION

Explanation: When you specify REALADDR, you must also specify the cylinder or block address.

System Action: The command terminates.

Programmer Response: Provide a range, or specific cylinder/track/block for the command you specified.

Operator Response: None

ICK32364I CAN NOT OBTAIN ACCESS TO DEVICE ccuu RC= nnnn 'failing reason'

Explanation: The DIAGNOSE E4 FCN 00, 01, 02 or 03 enables ICKDSF to obtain minidisk information or to establish a full-pack overlay link to the specified user's minidisk or real device defined in the system. The operation failed with the CP return code RC=nnnn. The 'failing reason' line of the message gives some common reasons for the failure.

- RC=0005 USER DOES NOT HAVE DIRECTORY OR RACF AUTHORIZATION
- RC=0101 - VIRTUAL DEVICE ADDRESS INVALID -

  The device number does not identify a device in the current I/O configuration of the virtual machine specified by the input userid.
- RC=0102 VIRTUAL DEVICE ADDRESS IS NOT ACCEPTED. IT DOES NOT REPRESENT A MINIDISK
- RC=0103 REAL VOLUME WHICH CONTAINS THE MINI DISK IS NOT MOUNTED
- RC=0200 USERID NOT DEFINED TO THE SYSTEM
- RC=0201 DEVICE DOES NOT IDENTIFY A VIRTUAL DEVICE IN THE DIRECTORY FOR THE INPUT USERID
- RC=0202 VIRTUAL DEVICE ADDRESS IS NOT ACCEPTED IT DOES NOT REPRESENT A MINIDISK
- RC=0203 REAL VOLUME WHICH CONTAINS THE MINI DISK IS NOT MOUNTED
- RC=0300 USERID NOT DEFINED TO THE SYSTEM
- RC=0301 DEVICE DOES NOT IDENTIFY A VIRTUAL DEVICE IN THE DIRECTOR FOR THE INPUT USERID
- RC=0302 VIRTUAL DEVICE ADDRESS IS NOT ACCEPTED IT DOES NOT REPRSENT A MINIDISK
- RC=0305 A FULL PACK OVERLAY ALREADY EXISTS

  Only one full pack overlay is allowed on a volume at any time.
- RC=0306 THE FULL PACK OVERLAY REQUEST INCLUDES CP PAGING, SPOOLING, DIRECTORY, T-DISK OR DUMP AREAS
- RC=0307 THE FULL PACK OVERLAY REQUEST FAILED BECAUSE THE MINIDISK HAS EXISTING LINKS TO IT USERID = NNNNNN

  The target minidisk has a write link to it. Have the user with the LINK remove it and rerun the job. USERID = NNNNNN specifies the user who has the existing link to the volume.
- RC=0402 - INPUT REAL DEVICE IS NOT ACCEPTED BECAUSE IT IDENTIFIES A NON DASD DEVICE
- RC=0403 DEVICE OFFLINE OR NOT MOUNTED
- RC=0405 A FULL PACK OVERLAY ALREADY EXISTS

  Only one full pack overlay is allowed on a volume at any time.
- RC=0406 THE FULL PACK OVERLAY REQUEST INCLUDES CP PAGING, SPOOLING, DIRECTORY, T-DISK OR DUMP AREAS
- RC=0407 THE FULL PACK OVERLAY REQUEST FAILED BECAUSE THE MINIDISK HAS EXISTING LINKS TO IT USERID = NNNNNN

  The target minidisk has a write link to it. Have the user who is LINKed, remove it and rerun the job. USERID = NNNNNN specifies the user who has the existing link to the volume.
- RC=0410 CYLINDER OR BLOCK NUMBER INVALID

  CP has determined the cylinder or block number is invalid. If the cylinder or block is valid for the device type, check the CP directory to make sure this device type is correctly properly defined.

System Action: The command terminates. ICKDSF continues with the next command.

Programmer Response: Correct the problem and rerun the job.

Operator Response: Correct the problem and rerun the job.

ICK40227I AN 'ELSE' COMMAND APPEARS IMPROPERLY

Explanation: An ELSE clause appeared in the command without a corresponding IF statement.

System Action: The remainder of the command is ignored. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the command syntax, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40228I AN 'END' COMMAND IS INVALID

Explanation: An END statement was encountered without a matching DO statement.

System Action: The remainder of the command is ignored. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the DO-END sequence, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40229I 'IF' COMMAND HAS INVALID RELATIONAL EXPRESSION

Explanation: The IF-THEN-ELSE statement sequence is incorrect. Only the system variables LASTCC and MAXCC can be specified, and the values must be decimal numbers from 0 through 99999.

System Action: The remainder of the command stream is ignored. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the requirements of the IF-THEN-ELSE statement sequence, correct the error, then reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40230I 'SET' COMMAND HAS INVALID ASSIGNMENT EXPRESSION

Explanation: A SET statement has invalid syntax. Only the system variables LASTCC and MAXCC can be assigned values, and the values must be decimal numbers from 0 through 99999.

System Action: The remainder of the command stream is ignored. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Check the syntax requirements of the SET statement, correct the error, then reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40232I IMPROPER OR MISSING 'THEN' KEYWORD

Explanation: The THEN clause of the IF-THEN-ELSE command sequence is missing or is misspelled.

System Action: The remainder of the command stream is ignored. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Correct the error, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40236I INPUT STREAM END-OF-FILE FOUND BEFORE END OF COMMAND

Explanation: An end-of-file condition was encountered while scanning the command. This could be caused by incorrect command-continuation syntax or missing records in the input stream.

System Action: The command terminates.

Operator Response: None.

Programmer Response: Correct the command syntax, or add the missing records, then reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40237I TOO MANY LEVELS OF 'IF' COMMAND NESTING

Explanation: More than ten IF statements have been nested.

System Action: The remainder of the command stream is ignored. Device Support Facilities processing continues with the next command.

Operator Response: None.

Programmer Response: Restructure the command stream to avoid the excessive nesting, and reissue the command.

Problem Determination: Table I, items 2, 4, and 29.

ICK40300I ERROR READING INPUT DATA FROM 'fn ft fm' RC= nnnn

Explanation: An error condition was detected with RC=nnnn when using CMS macro FSREAD to read a record from the specified CMS file.

System Action: The command terminates.

Programmer Response: For an explanation of the return code, see the FSREAD macro in *CMS Macros and Function Reference*

Operator Response: None.

ICK40305I ERROR READING INPUT DATA FROM VIRTUAL READER RC= nnnn

Explanation: An error condition has been detected (RC=nnnn) when the CMS macro RDCARD to read a record from a virtual reader file.

System Action: The command terminates.

Programmer Response: For an explanation of the return code, see the RDCARD macro in *CMS Macros and Function Reference*

Operator Response: None.

ICK40310I ERROR WRITING OUTPUT DATA TO 'fn ft fm' RC= nnnn

Explanation: An error condition has been detected (RC=nnnn) when the CMS macro FSWRITE is issued to write a record to the specified CMS file.

System Action: The command terminates.

Programmer Response: For an explanation of the return code, see the FSWRITE macro in *CMS Macros and Function Reference*

Operator Response: none.

ICK40315I ERROR WRITING OUTPUT DATA TO VIRTUAL PRINTER RC= nnnn

Explanation: An error condition has been detected (RC=nnnn) when the CMS macro PRINTL is issued to print a line to a virtual printer.

System Action: The command terminates.

Programmer Response: For an explanation of the return code, see the PRINTL macro in *CMS Macros and Function Reference*

Operator Response: none.

ICK40320I I/O OPERATION INCOMPLETE OR FAILED RC= nnnn

Explanation: An error condition has been detected (RC=nnnn) when the CMS macro HNDINT is issued to handle the I/O interrupt for the DASD I/O operation.

System Action: The command terminates.

Programmer Response: For an explanation of the return code, see HNDINT macro in *CMS Macros and Function Reference*

Operator Response: none.

ICK40999I UABORT CODE nn

Explanation: This message indicates a termination error causing the Device Support Facilities processor to abort. This situation is usually caused by program or system errors. The message appears in the output listing via the write-to-programmer SVC. Code *nn* indicates the nature of the error.

Code	Meaning
24	Text processor's print control table not addressed by the GDT
28	No virtual storage available for: • Page header line • Argument lists • Main title line • Footing lines • Print control table (PCT) • Initialization of the I/O adapter historical data and message area • Automatic (dynamic) storage for a module which uses preallocated automatic storage • A GETMAIN/GETVIS request • Space for STAE control block • Open control-block allocation • Device Information Table (DIT)

Note: The message that sufficient storage was not available could not be issued because the SYSPRINT data set is not open.

32 Request made to process unopened data set

33 Unable to cancel STAE control

36 Processor unable to open SYSPRINT (or whichever DD name is used to denote the processor's standard listing output data set). The SYSIN DD card may be misspelled.

40 Invalid U-macro argument list:

UOPEN, UCLOSE, UPUT,
UGSPACE, UGPOOL, UFPOOL
UTIME
UVOLCHK, UWTO
USTAE

44 Processor is unable to produce a dump (the SNAP system service was not successful: either the DSFDUMP data set could not be opened, or not enough storage was available)

52 Module not found for ULOAD

72 Unsupported device type

80 Invalid ACTIONKEY value

88 Operator replied "T" to message ICK014D

92 Dynamic device information table (DIT) does not exist.

96 Equipment check occurred on device that is being processed

System Action: Device Support Facilities processing terminates.

Operator Response: Notify the system programmer.

Programmer Response: Correct the cause of the error, and restart Device Support Facilities.

Problem Determination: Table I, items 1, 2, 3, 4, 29, and 30.

Appendix D. Problem Determination Tables for MVS

Problem determination is the activity required to identify a failing hardware unit or program and to determine who is responsible for support.

Problem determination is accomplished by using procedures specified by IBM. In some cases, these procedures may be initiated by a message or code that requires operator or programmer response. The response may include the requirement for additional problem-related data to be collected and will attempt, where possible, to indicate "probable" failure responsibility.

Problem determination information is included (for MVS) for applicable messages and codes under the heading "Problem Determination." Standard problem determination actions are identified as items of Tables I and II. Unique actions are identified following the list of standard actions to be taken. In any case, it is intended that the specified actions be taken before calling IBM for support.

TABLE I

If the problem occurs again, follow the problem determination aids specified by the associated message or code before calling IBM for support:

1. If MSGLEVEL=(1,1) was not specified in the JOB statement, specify it and rerun the job.
2. Save the console sheet from the primary console. In systems with remote consoles, save the remote console sheet. In systems with Multiple Console Support (MCS), save a copy of the hard copy log.
3. Save the job stream associated with the job.
4. Save the system output (SYSOUT) associated with the job.
5. Make sure that the failing job step includes one of the following:
   a. SYSABEND DD statement
   b. SYSUDUMP DD statement
   c. PLIDUMP DD statement
   d. SYSMDUMP DD statement
6. Make sure that the PARM parameter of the EXEC statement specifies the following:
   a. MAP
   b. LIST
   c. DIAG
   d. MSG=AP
   e. CORE, if applicable
   f. XREF
   g. DUMP

7. If SMP is used to make all changes to the system, run the LIST CDS and LIST PTFBY functions of SMP to obtain a list of the current maintenance from the SMP control data set (CDS). If any changes are made to the system without using SMP, run the LISTIDR function of the AMBLIST service aid program to obtain a list of all members with a PTF or local fix, and save the output. Run the program against the:
   a. SYS1.LINKLIB data set
   b. SYS1.SVCLIB data set
   c. Library containing the program that issued the message
   d. SYS1.LPALIB data set
8. Run the IMCJOBQD (stand-alone) or IMCOSJQD (system-assisted) service aid program to obtain a formatted copy of the contents of the SYS1.SYSJOBQE or SYS1.SYSWADS data sets, SWADS or the resident job list.
9. Run the AMBLIST service aid program to obtain:
   a. An object module listing, specifying the LISTOBJ function
   b. A load module map and cross-reference listing, specifying the OUTPUT=BOTH option of the LISTLOAD function
10. Have a copy of the Message Control Program (MCP) available.
11. Run the AMDSADMP service aid program to dump the contents of real storage and page data sets on magnetic tape.

    After restarting the system, process the appropriate function of the AMDPRDMP service aid program to print the required portion of the dump tape produced by that program.

    Save both the tape from the AMDSADMP service aid program (should further information from the tape be required) and the listing from the AMDPRDMP service aid program.
12. Run the SEREP program, and save the resulting output.
13. Save all the associated output.
14. The normal response to this message requests the programmer and/or operator to run a specific program. Save all output from that program.
15. Save the program listing associated with the job.
16. Save the dump.
17. Have the system-generation (SYSGEN) output available from:
    a. Stage I
    b. Stage II
18. Run the IFCEREP1 service aid program to dump the SYS1.LOGREC data set and save the resulting output. For MSS, run the following program to dump the SYS1.LOGREC data set.
    a. Service aid IFCISDA0
    b. Program ISDASDA0 with the DETAIL(ALL) parameter

19. Save the assembly listing associated with the job.
20. Save the control cards associated with the job.
21. Save the compiler output associated with the job.
22. Save the source input associated with the job.
23. Save the source program listing associated with the job.
24. Run OLTEP diagnostics for the problem device and save the output.
25. Run the IEHLIST system utility program to obtain a list of the:
    a. Volume table of contents of the associated volume, specifying the FORMAT option
    b. Volume table of contents of the associated volume, specifying the DUMP option
    c. Directory of the associated data set
    d. System catalog (if it is not a VSAM catalog)
26. Run the IEBPTPCH data set utility to print the:
    a. Directory of the applicable data set
    b. Applicable data set
    c. Applicable member
    d. Applicable procedure
27. Have the linkage editor/loader map available.
28. Save the associated volume.
29. Contact IBM for programming support.
30. Contact IBM for hardware support.
31. Save the trace output data set.
32. Print the GTF trace data set with the AMDPRDMP service aid program using the EDIT statement.
33. Print the associated SVC dump data set, with the AMDPRDMP service aid program using the GO statement.
34. Issue the Access Method Services LISTCAT command to list:
    a. The contents of the applicable catalog
    b. The catalog entries for the applicable objects and any related objects
35. Issue the following Access Method Services command:
    a. The MSS LISTMSF command for mountable volumes
    b. The MSS LISTMSF command with the CARTRIDGES parameter
    c. The PRINT command to list the contents of the mass storage volume control inventory data set
    d. The LISTMSVI command
    e. The LISTMSF command with the ALL parameter
36. Issue the Access Methods Services PRINT command to print the repair workfile.

37. Run the AMASPZAP service aid program using the ABSDUMP statement to print the contents of the applicable:
    a. Data set
    b. Track, and save all associated output
38. Issue the Access Method Services AUDITMSS command with the following:
    a. The CHECK parameter
    b. The MAP parameter
    c. The READLABEL parameter
39. Issue the Access Method Services CHECKMSS command.
40. Issue the Access Method Services COMPARET command.
41. Issue the Access Method Services DUMPMSS command to dump the following:
    a. Formatted mass storage control storage
    b. Mass storage control main storage
    c. Mass storage control extended storage
    d. Formatted Staging Adapter storage
    e. Staging Adapter main storage
    f. Staging Adapter extended storage
    g. Mass storage control tables
42. Save the latest output from the Mass Storage Control Table Create program.
43. Display the units for devices associated with the problem area. If a specific unit or units are not known, display the range of all virtual units. See your path chart for address ranges.
44. Obtain the RACF profile of the associated data set where applicable.
45. Stop the processing unit and use the hardware ALTER/DISPLAY facility to display:
    a. All general purpose registers
    b. The page status word
    c. Main storage locations 0—200 (hexadecimal) and 7000—7080 (hexadecimal)
46. If the AMDSADMP program resides on tape, save the tape. If it resides on disk, dump the SYS1.PAGEDUMP data set and cylinder 0, track 0 of this residence disk.
47. Save the output (listings) of the stage 1 and stage 2 AMDSADMP initialization jobs.
48. Follow the procedures for item 9b of this table for load modules AMDSAPGE, AMDSAPRO, and AMDSALDR of SYS1.LINKLIB. Use IEBUPDTE or IEBPTPCH to print the AMDSADMP and AMDSADM2 macros from SYS1.MACLIB.
49. Save the AMDSADMP dump output (tape or listing).

50. If the program seems to be looping, use the display PSW feature of the hardware ALTER/DISPLAY facility along with the hardware Instruction Step facility to trace the loop, instruction by instruction.
51. If there is an error in the contents of a page data set dump, restart the system using a different page data set, then dump the original page data set.
52. Use IEBCOPY to unload SYS1.IMAGELIB to tape.
53. Have a list of RACF-defined entities available.

TABLE II—GTF for Problem Determination

Format 1: Tracing Without Prompting for Event Keywords

Before reproducing the problem, have the system operator issue the START GTF command specifying the tape output, MODE=EXT, and TIME=YES. In response to the AHL100A message, the operator should type TRACE=opt, where opt is the trace option indicated for the particular message or code, within the text of the reply.

When data for the problem has been recorded, run the AMDPRDMP service aid program using the EDIT statement to format the trace output, specifying DDNAME= (ddname of the trace data set).

Format 2: Tracing With Prompting for Event Keywords

Before reproducing the problem, have the system operator issue the START GTF command specifying tape output, MODE=EXT, and TIME=YES. In response to the AHL100A message, the system operator should specify the trace options indicated for the associated message or code within the text of the reply. Then, in response to the AHL100A message, the system operator should specify the event keywords also indicated with the associated message or code.

When data for the problem has been recorded, run the AMDPRDMP service aid program using the EDIT statement to format the trace output, specifying DDNAME= (ddname of the trace data set).

Format 3: Specialized Tracing Action

Before reproducing the problem, have the system operator issue the START GTF command specifying tape output, MODE=EXT and TIME=YES. In response to the AHL100A message, the system operator should type 'TRACE=SYS,USR'. The DD statement for a data set in error should specify DCB=DIAGNS=TRACE.

When data for the problem has been recorded, run the EDIT function of AMDPRDMP service aid program, specifying the options SYS and USR=FFF.

TABLE II—GTF for Problem Determination

Format 4: Specialized Tracing Action for VSAM

Before reproducing the problem, have the system operator issue the START GTF command specifying tape output, MODE=EXT and TIME=YES. In response to the AHL100A message, the system operator should type 'TRACE=SYS,USR'. The DD statement for a data set in error should specify AMP=TRACE.

When data for the problem has been recorded, run the EDIT function of the AMDPRDMP service aid program, specifying the options SYS and USR=(FFF,FF5).

Appendix E. The VTOC Index

The VTOC index is a separate, sequential data set containing four record types that give information on volume and VTOC status. Systems that contain and use indexed VTOC programming support can realize performance improvements in the areas of volume space management and I/O operations to the VTOC.

The INIT and BUILDIX commands will build the VTOC index on eligible count-key-data devices. The INIT command creates space for the index during volume initialization in both operating system and stand-alone environments. The BUILDIX command, which requires that the host operating system contain indexed VTOC programming support, builds VTOC indexes on volumes currently in use on the system. Both commands prepare the VTOC on the target volume to indexed VTOC (IXVTOC) format.

The four record types in the index provide the system with a comprehensive means of space management compared to nonindexed methods. The four record types are:

1. The VTOC index map (VIXM) is the record that maps the index itself. It contains a one-for-one bit map of the records in the index, showing which records are in use (bit set to one) and which records are free to be assigned (bit set to zero). The programs reserve index space for a maximum of three VIXMs.
2. The VTOC pack space map (VPSM) contains separate bit maps for management of space on the volume at both cylinder and track level. Free and allocated space is represented in these maps. A maximum of five VPSMs are reserved by the programs.

   Note: The VPSM is built using the information in the format-4 DSCB which gives the number of cylinders on the volume as the sum of the primary cylinders plus the number of alternate cylinders. This value is shown in the VPSM. However, the alternate cylinders as well as all other unused bits in the bit maps are marked as allocated.
3. The VTOC map of DSCBs (VMDS) contains a bit map showing free (bit set to zero) and allocated (bit set to one) data set control blocks in the VTOC. A maximum of three VMDSs are reserved.
4. The VTOC index entry record (VIER) contains pointer entries to each format-1 DSCB in the VTOC. Entries are built in the VIER using the DSCB key (data set name) and the disk address (cylinder, head, record number) of the format-1 DSCB. The number of VIERs is variable to the maximum of total index records reserved minus the records used by the other three record types.

Calculating the Size of the VTOC

To calculate VTOC size, you generally determine the maximum number of data sets that will reside on the volume. The number of VTOC tracks to reserve can be easily calculated by dividing this number by the number of DSCBs per track, rounding to the next higher track.

The number of DSCBs in the VTOC determines the number of data sets or VSAM data spaces that can reside on a volume and is therefore essential information for the DADSM routines that allocate and release space.

The number of DSCBs that will fit on a single track of each type can be found in Figure 19 on page 307.

Calculating the Size of the VTOC Index

The size of the index data set depends upon:

1. The size of the volume (total number of cylinders and tracks)
2. The track length (in bytes)
3. The number of tracks occupied by the VTOC

For the INIT command, the user may calculate the number of tracks to reserve for the index and place that value as the third, optional subparameter of the INDEX parameter; or the third subparameter may be omitted and the program allowed to calculate the size of the index. For the BUILDIX command, the user must calculate the size of the index.

In calculating the size of the index, the INIT command determines the number of VIXM, VPSM, and VMDS records needed (not exceeding the established maximum for each type), and assumes that all VTOC DSCBs are (or could be) format-1 DSCBs and that each VIER manages a maximum of 48 format-1 DSCBs. The number 48 assumes the VIERs are half full and the average data set name length is approximately 12. If your data set name lengths are greater than 12, you should compute the size of your VTOC index using the calculations shown following Figure 19 on page 307.

For efficient use of DASD space, it is recommended that you calculate the number of tracks to reserve for both the VTOC and the index data set. Figure 19 on page 307 shows the maximum values for VTOC and VTOC index size for the devices that can support the VTOC index. It is not likely that the assumptions used to generate the chart would be valid in an actual situation. However, the chart places the upper limits for calculating maximum usable space. Space reserved above these limits is wasted.

DEVICE	CYLS		TRKS PER CYL	DSCBs/ TRK	INDEX RECS/ TRK [1]	MAX. VTOC [2] (TRKS)	MAX. INDEX [3] (TRKS)	# OF RECORD TYPES [4]		
	PRI	ALT						VPSM [5]	VMDS [6]	VIER [7]
2305-1	48	1	8	18	5	21	3	1	1	8
2305-2	96	1	8	34	6	23	4	1	1	17
3330	404	7	19	39	6	192	27	1	1	156
3330-11	808	7	19	39	6	384	53	2	1	312
3340 (35MB)	348	1	12	22	3	181	29	1	1	82
3344 3340 (70MB)	696	2	12	22	3	363	56	1	1	163
3350 (native)	555	5	30	47	8	347	42	2	2	328
3375	959	1	12	51	14	222	20	1	1	272
3380 A04, B04 AA4, AD4, BD4 AJ4, BJ4, CJ2	885	1	15	53	18	246	16	1	1	285
3380 AE4, BE4	1770	1	15	53	18	492	31	2	2	553
3380 AK4, BK4	2655	1	15	53	18	738	46	3	3	821

[1] Index record length of 2048 bytes.
[2] This maximum value assumes that all data sets on the volume are single track data sets
[3] This maximum value assumes that the VTOC is the maximum size. See the previously given assumptions which are used by the INIT command to calculate the index size.
[4] The first VIXM map will manage up to 7840 index records. (One VIXM is more than sufficient for current devices.) The second VIXM will manage an additional 16032 records.
[5] The number of VPSMs required for a device is dependent upon its size. 16032 bits are available for cylinder and track mapping in each VPSM (a maximum of 5 VPSMs is currently reserved).
[6] The first VMDS will manage 16032 DSCBs. The second and third VMDS will manage an additional 16032 DSCBs.
[7] Each VIER is assumed to contain 48 entries (pointers to each Format-1 DSCB). See the previously given assumptions which are used by the INIT command to calculate the index size.

Figure 19. Table of Maximum VTOC and VTOC Index Sizes

To calculate the number of tracks to reserve for the index, first determine the average data set name length of the format-1 DSCBs which will reside on the volume. If the length chosen is too small, the data set names may not fit in the index. If the length chosen is too large, some space is unused in the index. Next determine the maximum number of data sets that will reside on the volume. Then calculate the number of VIERs needed to contain that many data set names of the chosen average data set name length, using the calculations shown below. For the number of VIXM, VPSM, and VMDS records, refer to the chart for actual numbers. (Note that the 3330-11 and 3350 will always require two VPSMs, and a 3350 with a maximum-size VTOC will require two VMDSs.) Add together the totals of all four record types and divide by the number of index records per track, rounding to the next higher track, to get the number of tracks to reserve for the index.

The value calculated for the index size is to be placed as the third subparameter of the INDEX parameter on the INIT command statement; or it is to be placed as the number of tracks to be allocated in the SPACE or ABSTR parameter in the DD statement required to execute the BUILDIX command.

The following shows the calculations to determine the number of VIERs needed to contain N data set names, each of which has an average data set name length of D. The calculations assume that all VIERs except one at each level are half filled, that a new level of the index is created when four lower level VIERs are created, and that n+1 unused VIERs are required for an n-level index.

Calculate the number of VIERs as follows where:

$\lfloor\ \rfloor$ = round quotient down to integer value

$\lceil\ \rceil$ = round quotient up to integer value

N = number of data set names in the index

D = average data set name length

E^1 = number of data set names that fit in the half-filled level 1 VIER

E^2 = number of data set names that fit in a half-filled level 2 or greater VIER

L(n) = number of VIERs at level n

$\lfloor 244/(D+8) \rfloor * 4 = E^1$ number of data set names that fit in a half-filled level 1 VIER

$\lfloor 244/(D+7) \rfloor * 4 = E^2$ number of data set names that fit in a half-filled level 2 or greater VIER

Calculate L(n) (where n is 1,2,3,...). and L(n) is the number of VIERs at level n

$\lceil N/E^1 \rceil - 1 = L(1)$ number of level 1 VIERs. Use L(n)=1 if calculation yields zero.

For n=2,3,... L(n) is calculated from L(n-1)

If L(n-1) ≥ 4 then

$\lceil L(n-1)/E^2 \rceil - 1 = L(n)$ Use L(n)=1 if calculation yields zero.

If L(n-1) < 4, then there are no VIERs at level n or higher.

The total number of VIERs is

$L(1) + L(2) + \ldots + L(n)$

The assumptions used for the above calculations represent the state of the index immediately after building it using the BUILDIX command. The state of the index is such that the largest number of VIERs possible are used to contain the data set names. However, since the data set names on a volume will not all have the same length, the calculations should be used only as an approximation. You should choose N (maximum number of data set names) and D conservatively to account for the varying number of data set names that will fit in a VIER.

Note: Fragmentation (because of scratch and reallocation activity) will cause the index to expand significantly (by up to 50%).

Appendix F. The Fixed Block Architecture VTOC (FBAVTOC)

FBAVTOC Format Characteristics

VTOCs on FBA devices are formatted in a similar manner to VSAM relative record data sets in that the 140-byte label records are stored in control intervals. For example, three label records fit into a 512-byte control interval. The control interval size is always an integral multiple of the device's physical block size. The upper limit is 8192 bytes.

The FBAVTOC extent is expressed in number of 'slots', which is equivalent to the number of file label records. It can range from a minimum of 3 slots to a maximum of 999 slots, fitting into as many control intervals as required. The program rounds up the extent values to that of the next full control interval.

The FBAVTOC's starting location can be any physical block on the volume, except for blocks 0 and 1, which are reserved for the IPL and volume label blocks (VLB).

A default (non-SYSRES) FBAVTOC has the following default characteristics:

- Starting location—block 2, immediately after the VLB
- Extent—56 file label records
- Control interval size—1024 bytes (or FBA block size, if larger)

If you specify FBAVTOC(END), ICKDSF creates a VTOC that is 99 entries long with a control interval size of 1024, and places it on the last blocks of the volume.

Label Record Format

Each record of the FBAVTOC contains 140 bytes, written as binary zeros. The first two records of the FBAVTOC are reserved for specific records:

- The data set control block (format 4) of the FBAVTOC
- The space management label (format 5)

The first 4-byte field of the space management label is written as hexadecimal 5s (X'05'). The 45th byte of the space management label is written as hexadecimal X'F5'.

Each control interval of the FBAVTOC contains a fixed number of slots and has the following format (Figure 20).

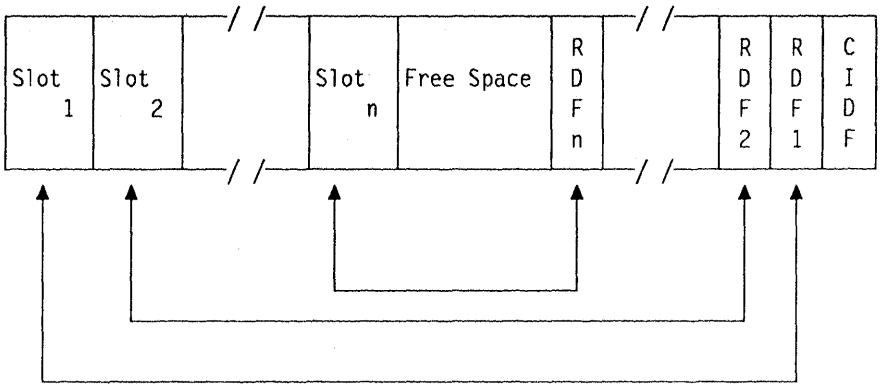


Figure 20. Format of a FBAVTOC Control Interval

There is one RDF (record definition field) associated with each slot in a control interval. The length field in the RDF (bytes 1 and 2) contains the slot length, which is equal to the label record length. Bit 5 of byte 0 of the RDF indicates whether the associated slot contains a record/label (bit 5=0) or not (bit 5=1). The RDF has one of the following contents:

- X'00008C' if the slot contains a label.
- X'04008C' if the slot does not contain a label, that is, it is empty.

The number of labels per control interval (LBPCI) is determined by the formula

```
LBPCI = FLOOR of (CISIZE-4)/(RECSIZE+3)
```

where:

FLOOR takes the integral part of the argument
CISIZE = control interval size
RECSIZE = label record size, which is equal to 140

LBPCI is recorded in the standard volume label (VOL1) in bytes 29 through 32

The control interval definition field (CIDF) has the following format:

```
Bytes 0,1   free-space offset = RECSIZE * LBPCI
Bytes 2,3   free-space length = CISIZE-4-LBPCI * (RECSIZE+3)
```

FBAVTOC Space Requirements

To calculate the space required for a fixed block architecture VTOC:

The number of labels per control interval (LBPCI) is determined by the formula

```
LBPCI = FLOOR of (CISIZE-4)/(RECSIZE+3)
```

where:

FLOOR takes the integral part of the argument
CISIZE = control interval size
RECSIZE = label record size, which is equal to 140

The number of control intervals required for the FBAVTOC is

CEILING of (number of labels specified/LBPCI)

where:

CEILING takes the next highest integer if there is a remainder

The number of physical blocks reserved for the FBAVTOC is

CI * (CISIZE/physical block size)

where:

CI = number of control intervals required for the VTOC

Appendix G. Surface Checking

Surface checking for a track is performed when:

- You specify the INIT command with the CHECK(n) parameter for devices other than the IBM 3375 and 3380.
- You specify the INSPECT command with the CHECK(n) parameter for all devices including the IBM 3375 and 3380

The purpose of surface checking is to detect possible defects that might exist on the surface of a track. These defects might be correctable or uncorrectable. Surface defects exist with different degrees of visibility (that is, some defects can be detected by any I/O to the defective area of the track, while other defects might only be detected once out of every 100 I/Os, and only with a certain bit pattern). The I/O that Device Support Facilities does to any one track is limited by performance considerations, the practicality of finding a defect, and what can be done if a defect is detected for the device type being processed.

The surface checking functions performed by Device Support Facilities are not equivalent to the surface checking performed on a volume at the factory. Marginal defects that exist on the surface of a track cannot be detected consistently by Device Support Facilities because of their intermittent nature. Therefore, different results can be obtained for the same track and/or volume from multiple runs of Device Support Facilities. Also, Device Support Facilities output can differ from other surface checking products because of different checking algorithms.

The IBM 3340, 3344, 3350, 3375, and 3380 device types contain extra space that can be used if a defective area is detected on a track. By assigning a skip displacement to a given track, the hardware knows to skip over the defective area and to use the extra area for this track. No alternative track assignment is needed. There is no performance degradation, and the track functions as if no defect exists.

Device Support Facilities performs two types of surface checking: Primary checking is supported for all device types; skip displacement checking can be done for 3340, 3444, 3350, 3375, and 3380 devices only.

Primary Checking

Primary checking is performed when CHECK(n) is specified as follows:

- For all devices (IBM 2305, 2311, 2314, 3330, 3340, 3344, 3350, 3375, and 3380) when you use the INSPECT command
- For the IBM 2305, 2311, 2314, 3330, 3344, and 3350 when you use the INIT command
- It is the only level supported for the IBM 2305, 2311, 2314, and 3330 devices

It consists of writing and reading back n bit patterns for the specific device (see "CHECK(n) Values" on page 317 for the values for n by device).

Primary checking for count-key-data devices consists of the following:

1. A bit pattern for the specified device is written and read back once.
2. If a data check is detected (correctable or uncorrectable), the following is performed:
   - For devices with skip displacement support, the I/O is repeated to determine if the data check persists. If it does, skip displacement checking is performed on the track. See "Skip Displacement Checking" for more information.
   - For devices without skip displacement support, the bit pattern is reread up to 10 times to determine if the error is repeatable. If the error is not repeatable, processing continues.

     If the error is repeatable:

     — For correctable errors, an entry is made in the map output if MAP was specified. Processing continues to the next track. If you want to assign an alternate to that track, use INSPECT NOCHECK ASSIGN TRACKS.

     — For uncorrectable data checks, the track is declared defective.
3. This is repeated n times (from CHECK(n)) unless a data check is encountered. Each pass through uses a different device-dependent bit pattern. If all the patterns for the device are used, the sequence of patterns is repeated starting with the first pattern used. See "CHECK(n) Values" on page 317 for the values of n that write each bit pattern at least once.

Primary checking for fixed block architecture devices consists of the following:

1. A bit pattern for the specified device is written and read back once.
2. If a correctable data check is detected, the block is reread 10 times to determine if the error is repeatable.
3. If an uncorrectable data check or repeatable correctable data check is encountered and ASSIGN is specified, an alternate is assigned and the process is restarted to analyze the surface of the alternate.

This is repeated n times (from CHECK(n)) until it is performed without encountering any uncorrectable data checks or any repeatable correctable data checks. Each pass through uses a different device-dependent bit pattern. If all the patterns for the device are used, the sequence of patterns is repeated starting with the first pattern used. See "CHECK(n) Values" on page 317 for the values of n that write each bit pattern at least once.

Skip Displacement Checking

Skip displacement checking of a track can be performed on 3340, 3344, 3350, 3375, and 3380 devices. Rigorous surface checking is done to locate and skip over defective areas on a track.

The maximum number of defects that can be skipped on any one track is dependent upon the device type. A track is declared defective only if the number of defects exceeds this maximum.

Skip displacement checking is performed using the INIT or INSPECT command as follows:

INIT command

- For the 3340, 3344, and 3350 when:
  - SKIP is specified with CHECK(n).
  - Primary surface checking detects a data check.
  - RECLAIM is specified and the track is currently flagged defective.

INSPECT command

- For all supported devices including the IBM 3375 and 3380 when:
  - SKIP is specified with CHECK(n).
  - Primary surface checking detects a data check.
  - RECLAIM is specified and the track is currently flagged defective.

Detection and assignment of a skip displacement for a track are complicated and can be time-consuming. When skip displacement processing is invoked, the emphasis is on defect detection.

For the NOSKIP process, the total running time can vary considerably, depending upon the number of tracks that might invoke skip displacement analysis.

Additionally, for low visibility defects, an error might not be detected during one NOSKIP run (and therefore no skip displacement checking takes place), but might be detected during a subsequent NOSKIP run.

Note, however, that, after a defect is skipped, that defect becomes invisible to all subsequent processing and will not be encountered again.

CHECK(n) Values

The n values shown below give one pass through each bit pattern. Specifying a value of n greater than these values causes the same bit pattern to be rewritten. As this is a time-consuming option, you should think about performance considerations.

- 2305, 2311, 2314, 3310, 3330, 3340, 3344, 3350, 3370, 9335. n=3 gives one pass through each bit pattern.
- 3375, 3380. n=2 gives one pass through each bit pattern if you specify INSPECT CHECK(n).

If you specify SKIP for supported devices, skip displacement checking is automatically performed and n is ignored.

Appendix H. User Security Exit Module

The data security function in Device Support Facilities provides for a user exit module which can tailor security processing to installation needs. A default module is shipped with Device Support Facilities.

The user exit module is invoked by the Device Support Facilities security/authorization module for online processing under MVS/370 and MVS/XA. The intended purpose of the user exit module is only to control the security checking function by setting return codes. Attempts to alter anything other than return codes with the module may have unpredictable results.

The exit module is link-edited with ICKDSF into SYS1.LINKLIB. The link-edit must be done with SETCODE AC(1) as Device Support Facilities is an authorized program. See Figure 23 on page 322 for an example of link-editing an exit module.

ICKUSER1 *must* be the name of your user exit module. The source for ICKUSER1 is contained in SYS1.SAMPLIB.

Device Support Facilities loads the following registers as input to ICKUSER1:

Register	Content
1	Pointer to a 3-word parameter list
13	Pointer to a standard 18-word save area
14	Return address
15	Address of ICKUSER1 entry point

The 3-word parameter list contains:

- Word 1 - Pointer to a data set name
- Word 2 - Pointer to a 6-character volume serial number
- Word 3 - Pointer to a 2-character field containing:
  - A 1-byte data set type code
  - A 1-byte Device Support Facilities command code

The data set type codes are:

- **1** - VSAM data set
- **2** - Password-protected data set
- **3** - Unexpired data set

The command codes are:

- **1** - reserved
- **2** - INIT command
- **3** - INSPECT command
- **4** - reserved

Note: The user exit module must set a return code in register 15 indicating the action required.

For password-protected data sets, the return codes are:

Hex	Dec	Meaning
00	0	Bypass checking for this volume, do not return to ICKUSER1 for this volume
04	4	Bypass checking for this data set, but return to ICKUSER1 for the next data set
08	8	Check this data set normally and return to ICKUSER1 for the next data set
0C	12	Check this volume normally and do not return to ICKUSER1 for this volume
10	16	Reject access to this data set without checking and return to ICKUSER1 for the next data set
14	20	Terminate this command immediately

Normal checking for a password-protected data set consists of taking the user-supplied password for this data set from the PASSWORDS parameter and verifying that this password provides authority to alter the data set.

For VSAM and unexpired data sets the return codes are:

Hex	Dec	Meaning
00	0	Bypass checking for this volume and do not return to ICKUSER1
0C	12	Check this volume normally and do not return to ICKUSER1
14	20	Terminate this command immediately

Normal checking for VSAM and unexpired data sets consists of checking whether the PURGE option has been specified. (For INSPECT with NOPRESERVE, the PURGE option is always absent.) If a code other than the ones shown is returned, Device Support Facilities terminates the function.

The IBM-supplied module always returns a code of 12. Codes returned by the IBM module are arranged as shown in Figure 21.

Hex Offset	Code	Command	Data Set Type
+4*	12		
+8*	12		
+C*	12		
+10	12	INIT	VSAM
+14	12	INIT	Password-Protected
+18	12	INIT	Unexpired
+1C	12	INSPECT	VSAM
+20	12	INSPECT	Password-Protected
+24	12	INSPECT	Unexpired
+28*	12		
+2C*	12		
+30*	12		

Figure 21. Offsets of Return Codes for the IBM-Supplied User Exit Module

(* — Reserved.)

However, the IBM-supplied return codes can easily be changed with superzap. Figure 22 is an example of overriding ICKUSER1, in which the IBM-supplied return code is changed in this manner. This example shows how ICKUSER1 can be overridden to permit use of the INSPECT command with the NOPRESERVE option on volumes containing VSAM data sets. This JCL changes the IBM-supplied return code of 12 at offset X'1C' as shown in Figure 21.

```
//DSFP62D     JOB     ,CLASS=A,MSGLEVEL=(1,1)
//ZAPOFF      EXEC    PGM=AMASPZAP
//SYSPRINT    DD      SYSOUT=A
//SYSLIB      DD      DSN=SYS1.LINKLIB,DISP=SHR,UNIT=3330-1,
//           VOL=SER=VS2D57
//SYSIN       DD      *
 NAME         ICKDSF  ICKUSER1
 VER          001C    0000000C
 REP          001C    00000000
/*
```

Figure 22. Example of Overriding ICKUSER1 to Change Return Codes

When an installation wants to replace the IBM-supplied user exit routine:

1. The replacement routine must be named ICKUSER1 with an entry point of ICKUSER1; and
2. ICKDSF must be relink-edited.

Figure 23 shows an example of how to relink-edit Device Support Facilities to include a new ICKUSER1. In the figure, ICKUSER1 is a member of MYLIB.

```
//LINKLIB      EXEC    PGM=IEWL,PARM='XREF,LIST,LET,RENT,REFR'
//SYSPRINT     DD      SYSOUT=A
//SYSUT1       DD      UNIT=SYSDA,SPACE=((CYL,2,2))
//MYLIB1       DD      DISP=SHR,DSN=MYLIB,VOL=SER=xxxxx,UNIT=3330-1
//SYSLMOD      DD      DISP=SHR,DSN=SYS1.LINKLIB
//SYSLIN       DD      *
     INCLUDE   MYLIB1(ICKUSER1)
     INCLUDE   SYSLMOD(ICKDSF)
     ENTRY     ICKSA01
     SETCODE   AC(1)
     NAME      ICKDSF(R)
/*
```

Figure 23. Example of Link-Edit for ICKUSER1

Glossary of Terms and Abbreviations

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the IBM *Dictionary of Computing* , SC20-1699.

A

A-unit. The direct access storage unit that contains the controller functions to attach to the storage control. An A-unit controls the B-units that are attached to it and is often referred to as a head of string.

access mechanism. See actuator.

actuator. A set of access arms and their attached read/write heads, which move as an independent component within a head and disk assembly (HDA). For example, the 3380 Model AK4 has two HDAs, each containing two actuators. See also device and volume.

ALLOC. A space allocation parameter which indicates type, such as cylinders, tracks, etc.

alternate track. On a direct access storage device, a track designated to contain data in place of a defective primary track.

ANALYZE drivetest. ICKDSF command used to determine if a problem exists in the drive itself rather than in the media located on a volume.

ANALYZE scan. ICKDSF command used to determine if problems exists in the media which is located on a volume.

APF. Authorized program facility

B

B-unit. A direct access storage unit that attaches to the subsystem through an A-unit. A B-unit has no controller functions.

BDAM. Basic direct access method.

Block length. A subparameter of the SPACE parameter in a DD statement, specifying the average block length of the data in a data set. The system computes how many tracks to allocate.

bootstrap. IPL program records

BPI. Bits per inch

C

CA. Control area, VSAM

C-unit. A direct channel attach 3380 direct access storage unit that contains both the storage control functions and the DASD controller functions. A 3380 C-unit functions as a head of string and controls the B units that are attached to it.

cache fast write. A form of fast write where the data is written directly to cache storage without using non-volatile storage and is available for later destaging. This 3990 Model 3 Storage Control function should be used for data of a temporary nature, or data which is readily recreated, such as the sort work files created by the appropriate release of DFSORT.

cache storage. A random access electronic storage in selected storage controls used to retain frequently used data for faster access by the channel. For example, 3880 Model 23 and 3990 Model 3 contain cache storage.

CCHH. Cylinder/head (track address)

CCW. Channel control word

CI. Control interval, VSAM

CKD. Count, key, and data record format

CLIST. Command list

CMS. Conversational Monitor System (interactive terminal oriented system under VM)

command retry. A combined channel and storage director procedure that allows a command in a channel program to be re-executed without causing an I/O interrupt.

controller. The hardware component of a DASD head of string unit that provides the path control and data transfer functions. For example, there are two controllers in a 3380 Model AE4 or AK4.

correctable errors. See errors, correctable. See also errors, uncorrectable.

count-key-data (CKD). A DASD data recording format employing self-defining record formats in which each record is represented by a count area, that identifies the record and specifies its format, an optional key area that may be used to identify the data area contents, and a data area that contains the user data for the record. CKD is also used to refer to the set of

channel commands that are accepted by a device that employs the CKD recording format.

CSECT. Control section

CVAF. Common VTOC Access Facility

CYL. A subparameter of the SPACE parameter in a DD statement, specifying that space is to be allocated by cylinder.

D

DADSM. Direct access device space management program that maintains the VTOC, VTOCIX, and the space on a volume.

DASD. Direct access storage device; for example, a 3380.

DASD fast write. A form of fast write to cache storage where the data is written concurrently to cache storage and nonvolatile storage and automatically scheduled for destaging to the DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. This feature is available with a 3990 Model 3 Storage Control which has nonvolatile storage.

DASD subsystem. One or more DASD strings and the storage control(s) to which the the DASD are attached.

DCB. Data control block

DD. Data definition

DDA. Direct Disk Adapter

defective track condition. When a track of an IBM 3340/3344 is flagged defective, a read or write to such a track causes a seek to the alternate track to be performed by the system ERPs. For devices other than IBM 3340/3344, the seek is performed by the device and is transparent to the user.

delta directory. On the IBM 3340/3344/3350 the delta directory is on track X'A' of the SA cylinder. The delta directory is used to store HAs that have had additional skips added to them by ICKDSF since the SA directory was last built.

device. In this book, the term device is used when referring to a single, uniquely addressable part of a Direct Access Storage, particularly in a context that includes the geometry (track length, number of tracks per cylinder, and number of cylinders) of the device, or the accessing of the device through channel paths. See volume, unit.

device address. Three or four hexadecimal digits that uniquely define a physical I/O device on a channel path in System/370 mode. The one or two left-most digits are the address of the channel to which the device is attached. The two right-most digits represent the unit address. See device number.

device ID. An 8-bit identifier that uniquely identifies a physical I/O device in an extended architecture system.

Device level selection (DLS). A DASD function available with 3380 Models AD4, AE4, AJ4, AK4, and CJ2. With DLS, each of the two controllers in the DASD string has a path to all devices in the string (up to 16 addresses), and any two devices in the 2-path DASD string can read or write data simultaneously.

device level selection enhanced (DLSE). A DASD function available with 3380 Models AJ4 and AK4. With DLSE, each of the four controllers in the DASD string (as a result of interconnecting two A-units), has a path to all devices in the string (up to 32 addresses), and any four devices in the 4-path DASD string can read or write data simultaneously.

device number. Three or four hexadecimal digits that logically identify an I/O device in an extended architecture system. See device address.

DFDSS. Data Facility Data Set Services

DFHSM. Data Facility Hierarchical Storage Manager

DFP. Data Facility Product

DFP/370. Data Facility Product for MVS/370

DFP/XA. Data Facility Product for MVS/Extended Architecture

DIRF. DADSM interrupt recording facility. If a system fails, or a permanent I/O error occurs during allocation of space or during execution of a routine that updates the VTOC, the VTOC may be in error. To ensure that an error is recorded, the DADSM routines turn on a bit in the VTOC upon entry to a DADSM function, and, if no errors occur during processing, turn off that bit upon exiting from that function. This "DIRF bit" is bit 5 of the DS4VTOCI field of the format-4 DSCB.

DLS. See device level selection.

DLSE. See device level selection enhanced.

DOS. See VSE.

DOS bit. On a volume without an indexed VTOC, this bit indicates that the free space map (format-5 DSCB) is invalid.

DPR. Dynamic path reconnection

DPS. Dynamic path selection

DRC. Data recording control of MSS

DRD. Data recording device of MSS

DSCB. Data set control block

dyadic processor. A dyadic processor consists of two integrated processors sharing channels and a central storage

dynamic path selection. A set of functions for 3380 units providing:

- Alternate controller: The capability of one string controller to back up the other and provide a path to the data in the case of a string controller failure.
- System-related device reserve: A device reserved to a particular system can be accessed by that system over any of the paths to that system.
- String switch: The capability to connect a string of devices to two different storage directors.
- Dynamic path reconnection: Disconnected DASD operations reconnect over any available channel path rather than being limited to the one on which the I/O was started.

E

ECC. Error correction code

emulated device. A device (for example, a 3330) that is logically formatted on to a different device type (for example, a 3350) such that the data and programming required by the 3330 can be used without modification on the 3350.

ENQ. A system macro used to serialize the access to a data set.

EREP. IBM Environmental Record Editing, and Printing program

errors, correctable. When a data check is referred to as 'correctable', the following are present:

- The format 5 sense record has the correctable bit set (byte 2, bit 1)
- The error is correctable by use of ECC correction bytes.

The storage director or the system ERPs can correct this error with the aid of the ECC correction code without rereading the record. All the information needed is contained in the format 5 sense record. Restart displacement, error displacement, and error correction pattern.

When ECC correctable, the data is reconstructed to be the same as the data original written on disk storage. The data correction may be done by the subsystem as data is transferred to processor storage, or it may be done by the system ERP after the data is in processor storage. Although the data in processor storage is correct for use in processing, the data on the disk is not changed. The next attempt to read the data may again result in a data error. See also errors, uncorrectable

errors, permanent. A permanent error cannot be corrected by ECC processing or by retrying the operation. A permanent error gives an I/O interruption at the system and causes the subsystem to send unit check status to the system. A permanent error usually terminates a job step and may terminate the job. The action depends on the error recovery procedures at the application level.

errors, temporary. A temporary error is an error that is corrected by ECC correction or by retrying the operation. In each case the operation in process is completed. If the subsystem performs the recovery procedures, the action does not cause an I/O interruption. If the system (ERP) performs the recovery procedure, an I/O interruption is given. A temporary error is also called a *soft* error.

If the error is correctable using the ECC bytes or readable with command retry, it is a temporary error. If the program issuing the I/O operation considers the error as having been corrected, then it is a temporary error.

The term applies to any type of DASD error (data checks, equipment checks, etc.).

errors, uncorrectable. A data check that the storage director cannot correct using the ECC correction code bytes. Rereading or rereading with head offset is the only method available. Also referred to as "ECC uncorrectable error."

EXCP. Execute channel program. A macro used to access a data set without specifying the organization.

F

FBA. Fixed block architecture

FBAVTOC. The volume table of contents (VTOC) for a fixed block architecture FBA device. See also *VTOC*

fence. To separate one or more paths or elements from the remainder of the logical DASD subsystem. The separation is by logical boundaries rather than power boundaries. This separation allows isolation of failing compononts so that they do not affect customer operation.

Format-1 DSCB (F1 DSCB). An identifier DSCB. Describes a data set or VSAM data space and its first three extents. See the appropriate system data areas manual.

Format-2 DSCB (F2 DSCB). An index DSCB. Describes the indexes of an ISAM data set. See the appropriate system data areas manual.

Format-3 DSCB (F3 DSCB). An extension DSCB. Describes the 4th through 16th extents of a data set or VSAM data space. See the appropriate system data areas manual.

Format-4 DSCB (F4 DSCB). A VTOC DSCB. Describes the extent and contents of the VTOC and volume and device characteristics. See the appropriate system data areas manual.

Format-5 DSCB (F5 DSCB). A free space DSCB. Describes the space on a volume that has not been allocated to a data set or to a VSAM data space (available space). See the appropriate system data areas manual.

FTA. File Tape Adapter

G

Gb. gigabyte (1024 Mb) (approximately 1 billion or 10^9 bytes)

H

HA. Home address (of track)

HDA. Head and disk assembly

Head and disk assembly (HDA). A field replaceable unit in a direct access storage device containing the disks and actuators. A 3380 Model AK4 has two HDAs.

head of string. The unit in a DASD string that contains controller functions. For example, a 3380 model AE4, AK4, or CJ2.

home address (HA). The first field on a CKD track that identifies the track and defines its operational status. The home address is written after the index point on each track.

HSM. Hierarchical Storage Manager

I

ICKDSF. Device Support Facilities

identifier (ID). A sequence of bits or characters that identifies a program, device, controller or system.

IFA. Integrated File Adapter

indexed VTOC. The data set on which the location of the format-1 DSCBs of the VTOC are kept in an index for quick access by DADSM.

index point. The reference point on a disk surface that determines the start of a track.

initial microprogram load (IML). The act of loading microprogram(s).

Integrated catalog facility. A facility by which VSAM data set volume-related fields are separated from the catalog and maintained in the VVDS on the volume on which the data set resides.

I/O. Input/output

I/O device. An addressable input/output unit, such as a direct access storage device, magnetic tape device, or printer.

IOCDS. Input/output control data set

IOCP. Input/Output Configuration Program

IPL. Initial program load

IPL text. IPL program records

IXVTOC. Indexed VTOC.

J

JCL. Job control language.

JES. Job Entry Subsystem

K

K. The number 1024

KSDS. Key-sequenced data set, VSAM

L

like DASD devices. Devices that have the same track capacity, and the same number of tracks per cylinder; that is, like devices can differ only in the number of cylinders per volume.

Like device. A (DASD) device with identical geometry; that is, number of bytes per track, number of tracks per cylinder, and number of cylinders per actuator.

M

maximal initialization. Maximal initialization includes:

- All the functions of medial initialization
- Checking every track surface for defects
- Conditional alternate track assignment
- Conditional reclamation of tracks

Mb. Megabytes (1024K)

Mb/sec. Megabyte per second (a unit of measure for data transfer rate)

medial initialization. Medial Initialization includes the following:

- All the functions of minimal initialization
- Validation of every track

migrate. To begin using a different type of storage device. Migration often includes the movement of existing data to the new devices.

minidisk. See minivolume.

minimal initialization. Minimal initialization includes:

- Constructs the first track of the volume (IPL text and volume label)
- Creates the VTOC
- Checks primary/alternate track associations

minivolume. When an MVS system runs on VM, you can have MVS-formatted VM minidisks, with their size equal to or less than that of the real volume. These are known as "minivolumes."

ms. Milliseconds

MSC. Mass storage control of MSS

MSF. 3851 Mass Storage Facility of MSS

MSS. Mass Storage System

MSS staging pack. The IBM 3330 or 3350 volume that is used as a buffer for data that has been read from an MSS cartridge or for data that is to be written to a cartridge.

MSV. Mass storage volume of MSS

multipath storage director. A storage director in a 3990 Storage Control operating in DLSE support mode. Each multipath storage director in a storage cluster is associated with two storage paths. All storage paths in a multipath storage director respond to the same range of control unit addresses on a channel.

MVS. Multiple Virtual Storage, see MVS/370

MVS/370. Multiple Virtual Storage/370

MVS/SP. IBM Multiple Virtual Storage/System Product

MVS/XA (MVS/Extended Architecture). An operating system that supports a virtual memory address space larger than 16 megabytes.

O

OS/VS. Operating System/Virtual Storage

OS/VS2 MVS. Operating System/Virtual Storage 2, with multiple virtual address space support

P

PCUA. Primary channel unit address

permanent errors. See errors, permanent.

Primary surface checking. Performed by ICKDSF when NOSKIP is specified. Consists of writing and reading back bit patterns for the specified device. Skip displacement checking is performed (for those devices that support skip displacement) only if a data check occurs

PSP. Preventive service planning

PSW. Program status word

PTF. Program temporary fix

R

R0. Record zero

RACF. Resource Access Control Facility program product

RAS. Reliability, availability, and serviceability

RBA. Relative byte address, VSAM

RBN. Relative block number

RELCREDT. Relative creation date

RELEXPDT. Relative expiration date

RESERVE. A method of serializing DADSM update accesses to the VTOC. It is also a method of serializing CPU accesses to a shared DASD volume.

RMF. Resource Management Facility

RPS. Rotational position sensing

rotational position sensing (RPS). A function that permits a DASD to reconnect to a block multiplexer channel when a specified sector has been reached. This allows the channel to service other devices on the channel during positional delay.

S

SA. Surface analysis also see surface analysis cylinder

SCUA. Secondary channel unit address

Similar device. A (DASD) device with the same number of bytes per track and tracks per cylinder.

single-path storage director. A storage director in a 3990 or 3380 Model CJ2 operating in DLS support mode. Each single-path storage director in the storage cluster is associated with one storage path. A storage path on a single-path storage director responds to a unique control unit address on the channel. A single-path storage director in a 3990 is like a storage director in a 3880.

SIO (F). Start I/O Fast Release

Skip displacement. A method of shifting data so that any surface defect causing an error can be positioned in a gap rather than in a data area, eliminating the error.

SLSS. System Library Subscription Service

SMB. Speed Matching Buffer, see also *speed matching buffer*

SMF. System Management Facility

speed matching buffer. An option available on some models of the IBM 3880 or IBM 3990 Storage Control, which facilitates the connection of DASD devices with high data transfer rates to channels which have lower data transfer rates. The data is read from the device at a high rate, then transferred to the channel at a lower rate, and vice versa for write operations.

storage cluster. In the 3990 Storage Control and 3380 Model CJ2, a power and service region containing two independent data paths and either one multipath storage director or two single-path storage directors. It is designed so that should a failure or maintenance action occur, it will be independent of the other storage cluster in a 3990 Model 2 or Model 3 Storage Control. The 3990 Model 1 and the 3380 Model CJ2 each have a single storage cluster; the 3990 Model 2 and Model 3 each have two storage clusters. See also storage director, single-path storage director, and multipath storage director.

storage control. The component in a DASD subsystem that connects the DASD to the host channels. It performs channel commands and controls the DASD devices. For example, the 3990 Model 2 and Model 3 are storage controls.

storage director. In a 3990 storage director, a logical entity consisting of one or more physical storage paths in the same storage cluster. In a 3880, a storage director is equivalent to a storage path. See also storage path, single-path storage director, and multipath storage director.

storage director ID. An 8-bit designation that uniquely identifies the storage director regardless of its selection address. It identifies to the service representative, by means of EREP, a failing subsystem component (storage director) without his having to translate a selection address (which may have little relation to a physical address) to a physical component. The storage director ID is the number shown on the operator panels of 3880s and the attached DASD units.

storage facility. See 4-path string.

storage path. The hardware within the 3990 Storage Control that transfers data between the DASD and a channel. See also storage director.

storage subsystem. One or more storage controls and their attached storage devices.

string. A series of connected DASD units sharing one or more controllers (or heads of string). For example, a 3380 Model AE4 with the attached B-units is one string.

surface analysis cylinder. The surface analysis (SA) cylinder is reserved by the system to contain a record of all the home addresses (HA) that have a skip displacement assigned at the time of manufacture or assigned subsequently by ICKDSF (depends on the device).

T

temporary errors. See errors, temporary.

TRK. A subparameter of the SPACE parameter in a DD statement, specifying that space is to be allocated by tracks.

TSO. Time Sharing Option

U

UCB. Unit control block

uncorrectable errors. See errors, uncorrectable.

unit. In this book, the term unit is used to refer to an orderable unit of IBM 3380 Direct Access Storage. Each unit contains four access mechanisms. Model AA4, AD4, and AE4 units also contain two controllers.

unlike DASD devices. Devices that have different track capacities, or that have a different number of tracks per cylinder.

Unlike device. A (DASD) device with a different number of bytes per track, tracks per cylinder, or both.

user-assigned group name. A 1 to 8 character name that identifies a particular group of devices. The user-assigned group name and the devices that make up a group are specified during system generation (also called "esoteric name").

V

VIER. VTOC index entry record contains pointers to each format-1 DSCB in the VTOC.

VIO. Virtual Input/Output

VIXM. VTOC index map is the record that maps the VTOC index

VM. Virtual Memory

VM/370. Virtual Machine Facility/370 Operating System

VMDS. VTOC map of DSCBs contains a bit map indicating free and allocated DSCBs in the VTOC.

VMMAP. Virtual Machine Monitor Analysis Program

VM/SP. IBM Virtual Machine/System Product

VM/SP HPO. IBM Virtual Machine/System Product High Performance Option

VM/XA M. A. Virtual Machine/Extended Architecture Migration Aid

volume. In this book, the term volume is used to refer to the DASD space which is accessible by one actuator, in a context where the data stored is of concern, not the geometry of the device.

volume label. The volume label uniquely identifies the volume to the operating system. Additional optional user labels are available. containing information such as user name, serial number, etc.

VPSM. VTOC pack space map contains separate bit maps which show the free and allocated space for cylinders and tracks respectively.

VSAM. Virtual storage access method

VSE/AF. Virtual Storage Extended/Advanced Functions

VSE/SP. Virtual Storage Extended/System Package

VSE/VSAM. Virtual Storage Extended/Virtual Storage Access Method program product

VTOC. Volume Table Of Contents

VTOCIX. The data set on which the location of the format-1 DSCBs of the VTOC are kept in an index for quick access by DADSM.

VVDS. The VSAM volume data set. It describes the VSAM data sets, on a volume, that are cataloged in the integrated catalog facility catalog. The name of the data set is SYS1.VVDS.Vvolser.

W

WTO. Write to operator

Numeric

2-path string. A series of physically connected DASD units in which the head of string unit provides two data transfer paths that can operate simultaneously.

4-path string. A series of physically connected DASD units in which the heads of string provide four data transfer paths that can operate simultaneously. A four path string requires two 3380 Enhanced Subsystem model A-units.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Device Support Facilities
User's Guide and Reference

GC35-0033-13

Reader's Comment Form

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Do not use this form to request IBM publications. If you do, your order will be delayed because publications are not stocked at the address printed on the reverse side. Instead, you should direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

If you have applied any technical newsletters (TNLs) to this book, please list them here: _____

Chapter/Section _____

_____ Page No. _____

Comments:

Note: Staples can cause problems with automatic mail-sorting equipment.
Please use pressure-sensitive or other gummed tape to seal this form.

If you want a reply, please complete the following information.

Name _____ Phone No. (_____) _____

Company _____

Address _____

_____

Thank you for your cooperation. No postage is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address in the Edition Notice on the back of the title page.)

GC35-0033-13

Reader's Comment Form

Fold and tape | Please do not staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Programming Publishing
P.O. Box 49023
San Jose, CA 95161-9023

Fold and tape | Please do not staple | Fold and tape

IBM®

File Number
S370-30

GC35-0033-13

Printed in U.S.A.