

# LJK/CDROM Reference Manual

This software can be used to format and verify volumes according to ISO-9660 and related standards. In the case of formatting volumes, this software preserves all RMS file attributes so the files can be read from ISO-9660 volumes and volume sets using normal VMS mechanisms.

This software does not actually write to CD-R or DVD-R devices. Volumes created as disc images or entire magnetic disks should be transferred to CD-R or DVD-R using an Infoserver or various tools that run directly on VMS.

**Revision/Update Information:** This is a new manual

**Operating System and Version:** VAX/VMS Version 6.1 or higher  
Alpha VMS Version 6.1 or higher

**Software Version:** LJK/CDROM V1.0

---

**2002**

---

Copyright ©2002 by LJK Software, 1 Broadway, Suite 600, Cambridge, MA 02142-1100

The following are trademarks of LJK Software:

LJK/CDROM  
the LJK/CDROM logo

The following are trademarks of Hewlett-Packard Company:

Alpha  
VAX  
VMS  
VMScLuster

---

# Contents

<b>PREFACE</b>	<b>ix</b>
<b>NEW AND CHANGED FEATURES</b>	<b>xiii</b>
<hr/>	
<b>CHAPTER 1 GETTING STARTED</b>	<b>1-1</b>
<hr/>	
1.1 VERIFYING ISO-9660 FORMATTING	1-1
<hr/>	
1.2 FORMATTING A CDROM VOLUME	1-2
<hr/>	
<b>CHAPTER 2 INTRODUCTION TO ISO-9660</b>	<b>2-1</b>
<hr/>	
2.1 WHY WOULD ONE USE ISO 9660 CDROMS ON VMS?	2-1
2.1.1 What About Hybrid Volumes ?	2-2
<hr/>	
2.2 THE NATURE OF ISO-9660 VOLUMES	2-3
2.2.1 Hierarchies	2-3
2.2.2 File Identifiers	2-4
2.2.3 Directory Identifiers	2-5
2.2.4 Character Sets	2-5
2.2.4.1 D1-Characters • 2-5	
2.2.4.2 A1-Characters • 2-5	
2.2.4.3 Character Set Agreements • 2-5	
2.2.5 Primary Hierarchy	2-6
2.2.6 Levels of Interchange	2-7
2.2.7 Beyond ISO-9660	2-7
2.2.7.1 Posix Rock Ridge Extensions • 2-7	
2.2.7.2 VMS-specific RMS File Attributes • 2-7	
2.2.7.3 Microsoft Joliet Hierarchies • 2-8	
<hr/>	
2.3 THE NATURE OF ISO-9660 VOLUME SETS	2-8
2.3.1 The ISO-9660 Volume Group	2-8

# Contents

---

<b>CHAPTER 3</b>	<b>DEFINING CHARACTER SETS</b>	<b>3-1</b>
------------------	--------------------------------	------------

---

3.1	PREDEFINED CHARACTER SETS	3-1
-----	---------------------------	-----

---

<b>CHAPTER 4</b>	<b>CREATING HIERARCHIES</b>	<b>4-1</b>
------------------	-----------------------------	------------

---

4.1	HIERARCHY NAMES AND VOLUME SETS	4-2
-----	---------------------------------	-----

---

4.2	PREDEFINED HIERARCHIES	4-2
-----	------------------------	-----

---

<b>CHAPTER 5</b>	<b>FORMATTING A VOLUME</b>	<b>5-1</b>
------------------	----------------------------	------------

---

<b>CHAPTER 6</b>	<b>FORMATTING A VOLUME SET</b>	<b>6-1</b>
------------------	--------------------------------	------------

---

6.1	FORMATTING MULTIPLE VOLUMES WITH ONE COMMAND	6-1
-----	--	-----

---

6.2	CONTROLLING PLACEMENT ONTO VOLUMES WITH SMALLER VOLUME GROUPS	6-1
-----	---	-----

---

6.3	LINKING NEW VOLUMES INTO AN EXISTING VOLUME SET	6-1
-----	---	-----

---

<b>CHAPTER 7</b>	<b>VERIFYING A VOLUME</b>	<b>7-1</b>
------------------	---------------------------	------------

---

7.1	IMPLICIT VERIFICATION PROCESSING	7-1
-----	----------------------------------	-----

---

7.2	SUPPRESSING ERRORS	7-2
7.2.1	Suppressing Multiple Errors _____	7-2
7.2.2	Suppressing Particular Errors _____	7-2

---

**CHAPTER 8 COMPARING TWO VOLUMES 8-1**

---

---

**CHAPTER 9 COMMAND REFERENCE 9-1**

---

---

**9.1 COMMAND SUMMARY 9-1**

---

---

**9.2 COMMAND FORMATS 9-1**

---

---

**9.3 SPECIFYING ARBITRARY CHARACTERS 9-2**

---

---

**9.4 DETAILED SPECIFICATION OF INDIVIDUAL COMMANDS 9-3**

---

ADD	9-4
COMPARE	9-8
CREATE HIERARCHY	9-10
DEFINE CHARACTER_SET	9-14
EXIT	9-16
HELP	9-17
LINK GROUP	9-19
SET DEFAULT	9-20
SHOW VERSION	9-24
SUBSYSTEM	9-25
VERIFY	9-26
WRITE	9-30

---

**APPENDIX A LJK/CDROM INSTALLATION A-1**

---

---

**A.1 VMSINSTAL INSTALLATION COMMANDS A-1**

---

---

**A.2 PCSI INSTALLATION COMMANDS A-1**

---

---

**A.3 ENABLING HELP/MESSAGE AFTER INSTALLATION A-2**

---

---

**A.4 SAMPLE VMSINSTAL INSTALLATION A-2**

---

---

**A.5 SAMPLE PCSI INSTALLATION A-3**

---

## Contents

---

<b>APPENDIX B</b>	<b>BUG REPORTS</b>	<b>B-1</b>
<hr/>		
B.1	WHO CAN SUBMIT BUG REPORTS ?	B-1
<hr/>		
B.2	FORMAT OF A BUG REPORT	B-1
<hr/>		
B.3	INFORMAL QUESTIONS	B-2
<hr/>		
<b>APPENDIX C</b>	<b>COPYRIGHT</b>	<b>C-1</b>
<hr/>		
<b>APPENDIX D</b>	<b>LICENSING</b>	<b>D-1</b>
<hr/>		
D.1	STANDARD LICENSE	D-2
<hr/>		
D.2	VMS HOBBYIST LICENSE	D-2
<hr/>		
D.3	NO-CHARGE LICENSE	D-3
<hr/>		
<b>APPENDIX E</b>	<b>VMS ISSUES</b>	<b>E-1</b>
<hr/>		
E.1	MULTIPLE HIERARCHIES	E-1
<hr/>		
E.2	VOLUME SWITCHING	E-1
<hr/>		
<b>APPENDIX F</b>	<b>MESSAGES</b>	<b>F-1</b>
<hr/>		
F.1	PRIMARY MESSAGES	F-1
<hr/>		
F.2	SECONDARY MESSAGES	F-11
<hr/>		
F.3	EMBEDDED MESSAGES	F-15

---

F.4	INTERNAL ERROR MESSAGES	F-15
-----	-------------------------	------

---

ISO-9660 GLOSSARY	Glossary-1
-------------------	------------

---

INDEX

---

FIGURES

2-1	Multiple Hierarchies Pointing to the Same Files _____	2-4
-----	---	-----

---

TABLES

2-1	Comparison of ISO-9660 to ODS-n for CDROM/DVD Storage on VMS _____	2-1
2-2	ISO-9660 Levels of Interchange _____	2-7
3-1	Predefined Character Sets _____	3-1
4-1	Predefined Hierarchies _____	4-2



---

## Preface

LJK/CDROM has two major capabilities:

- Format according to ISO-9660 and related standards
- Verify ISO-9660 CDROMs or disc images created by LJK/CDROM or by other tools for conformance to ISO-9660 and related standards

LJK/CDROM does **not** actually write to CD-R or DVD-R devices. Volumes created as disc images or entire magnetic disks should be transferred to CD-R or DVD-R using an Infoserver or various tools that run directly on VMS.

---

## Document Structure

If you are interested in making casual use of LJK/CDROM, you should concentrate on the Chapter 1, Getting Started.

### Chapters

- Chapter 1, Getting Started, gives abbreviated instructions for casual use of LJK/CDROM without using more advanced capabilities.
- Chapter 2, Introduction to ISO-9660, explains the general nature of the ISO-9660 standard as preparation for making effective use of LJK/CDROM.
- Chapter 3, Defining Character Sets, describes how to define Character Sets within LJK/CDROM.
- Chapter 4, Creating Hierarchies, describes how to create Hierarchies in an ISO-9660 Volume by using LJK/CDROM.
- Chapter 5, Formatting a Volume, tells how to use LJK/CDROM to create an ISO-9660 Volume.
- Chapter 6, Formatting a Volume Set, tells how to use LJK/CDROM to create an ISO-9660 Volume Set.
- Chapter 7, Verifying a Volume, tells how to use LJK/CDROM to validate the ISO-9660 format of a Volume.
- Chapter 8, Comparing Two Volumes, tells how to use LJK/CDROM to perform a logical comparison between two Volumes.
- Chapter 9, Command Reference, describes the commands for controlling LJK/CDROM in a reference format.

### Appendices

- Appendix A, LJK/CDROM Installation, shows how to install LJK/CDROM.
- Appendix B, Bug Reports, tells how to report LJK/CDROM defects.
- Appendix C, Copyright, describes the copyright restrictions for LJK/CDROM, which differ from typical copyright restrictions.

- Appendix D, Licensing, gives the license texts for LJK/CDROM.
- Appendix E, VMS Issues, discusses VMS version-related ISO-9660 issues.
- Appendix F, Messages, gives an explanation of messages that might be issued during use of LJK/CDROM.
- ISO-9660 Glossary gives an alphabetical-order explanation of various terms (capitalized throughout this manual) that have specialized meanings under ISO-9660 and related standards.

---

## Intended Audience

This manual is intended for an individual who is interested in the creation of ISO-9660 and related Volume structures.

There is no particular requirement that the individual using LJK/CDROM be a VMS system manager or have privileged access to the machine on which LJK/CDROM is installed, but in some cases a CDROM or magnetic disk device might need to be protected so as to allow the LJK/CDROM user to mount the device `/FOREIGN`. A VMS ACL is usually preferable for that.

---

## Associated Documents

LJK/CDROM is designed to comply with the following standards:

- ISO 9660 : 1988 (E)—15-Apr-1988
- Rock Ridge Interchange Protocol Revision 1.09—14-Aug-1991
- IEEE P1281 System Use Sharing Protocol Version 1.12—8-Jul-1994
- IEEE P1282 Rock Ridge Interchange Protocol Version 1.12—8-Jul-1994
- Microsoft Joliet Specification Version 1—22-May-1995
- ISO/IEC 2022 Character code structure and extension techniques, Fourth Edition—1-Dec-1994

References in this document to “ISO-9660 and related standards” mean the above set of documents. In most cases you will not need to have copies of those standards. The major exception would be if you are trying to resolve a discrepancy between the results of the command `LJK/CDROM VERIFY` and some other formatting tool, particularly one for which you have the source.

Of course LJK Software welcomes citation of particular sections of the relevant standards within bug reports submitted as described in Appendix B, Bug Reports, but going to that length is not required for most bug reports.

---

## Conventions

Within this *LJK/CDROM Reference Manual* the term “ODS-n” means either VMS ODS-2 or VMS ODS-5, since the two format are almost the same as compared to ISO-9660.

Within this *LJK/CDROM Reference Manual* the term “disk” means means a fixed disk (typically magnetic) while the term “disc” means a removable disc such as CDROM, CD-R, DVD or DVD-R.

Within this *LJK/CDROM Reference Manual* unexpectedly capitalized terms within normal text paragraphs have specific meanings in the context of ISO-9660 and related standards. Such terms are defined in ISO-9660 Glossary. For instance, the term “volume set” might mean an ISO-9660 Volume Set or an ODS-n volume set, but when capitalized as “Volume Set” within this document it always means an ISO-9660 Volume Set.



---

## **New and Changed Features**

This is a new product.



# 1

---

## Getting Started

Serious users of LJK/CDROM will need to read subsequent chapters of this manual to understand the full capabilities of the program. But casual users might be able to succeed by reading just this chapter, at least to get started.

In particular, if you are using the no-charge license<sup>1</sup> to verify the integrity of ISO-9660 CDROMs or disc images formatted with other tools you will be able to get by with just the following section, Verifying ISO-9660 Formatting, so long as you do not encounter any errors.

---

### 1.1 Verifying ISO-9660 Formatting

You can use LJK/CDROM to check the following formats written either by LJK/CDROM or by another program:

- ISO 9660 : 1988 (E)—15-Apr-1988
- Rock Ridge Interchange Protocol Revision 1.09—14-Aug-1991
- IEEE P1281 System Use Sharing Protocol Version 1.12—8-Jul-1994
- IEEE P1282 Rock Ridge Interchange Protocol Version 1.12—8-Jul-1994
- Microsoft Joliet Specification Version 1—22-May-1995

The command to verify the integrity of a disc is:

```
LJK/CDROM VERIFY ddcu:
```

where “ddcu:” is the name of the physical disc, which must be mounted /FOREIGN to allow LJK/CDROM to get direct access to the Volume metadata.

The command to verify the integrity of a disc image is:

```
LJK/CDROM VERIFY ddcu:[dir.subdir]file.ext
```

There is a possibility that errors LJK/CDROM discovers regarding a particular file on the Volume being verified will in fact occur for multiple files (potentially all files) on that Volume. To suppress repetitive errors, consider using the following qualifiers:

- /SHOW=NOMULTIPLE\_ERRORS
- /TOLERATE=<VARIOUS>

Those qualifiers are documented in Chapter 9, Command Reference under the command VERIFY.

---

<sup>1</sup> As described in Appendix D, Licensing, no Product Authorization Key (PAK) is required for the no-charge license.

If you receive error messages with the command *LJK/CDROM VERIFY*, you should read the discussion at the start of Chapter 7, *Verifying a Volume*, to help you come to a decision regarding how serious those errors are in your environment.

---

## 1.2 Formatting a CDROM Volume

You can use LJK/CDROM to write the following formats:

- ISO 9660 : 1988 (E)—15-Apr-1988
- Microsoft Joliet Specification Version 1—22-May-1995

with just a simple command.

The command to write to an entire magnetic disk is:

```
LJK/CDROM WRITE ddcu: source-filespec-1,source-filespec-2,...
```

where the magnetic disk must be mounted /FOREIGN to allow LJK/CDROM to create the Volume metadata.

The command to create a disc image is:

```
LJK/CDROM WRITE target-filespec source-filespec-1,source-filespec-2,...
```

**Note: The LJK/CDROM WRITE command takes the target file specification before the source file specifications because the source file specifications are optional.**

That command presumes the files are to be written to the ISO-9660 image with the same names and directory locations as on the source disk. Altering that requires use of the following qualifier:

- /NAMES

That qualifier is documented in Chapter 9, *Command Reference* under the command WRITE.

You must use “subsystem mode” described in Section 9.2, *Command Formats*, to issue commands like *CREATE HIERARCHY* before the *WRITE* command to write a Volume using any of the following formats:

- Rock Ridge Interchange Protocol Revision 1.09—14-Aug-1991
- IEEE P1281 System Use Sharing Protocol Version 1.12—8-Jul-1994
- IEEE P1282 Rock Ridge Interchange Protocol Version 1.12—8-Jul-1994

That will require a deeper understanding of LJK/CDROM, beyond what is provided in this chapter.

# 2

## Introduction to ISO-9660

This chapter explains the general nature of the ISO-9660 standard as preparation for making effective use of LJK/CDROM.

### 2.1 Why would one use ISO 9660 CDROMs on VMS?

Of course exchange of data with other operating systems is the first reason most people consider using the ISO-9660 format on VMS, but that must be tempered by certain considerations:

- RMS relative and indexed files are VMS-specific formats that are not typically readable in other environments. In fact, some information critical to reading relative and indexed files is stored in a system-specific area set aside by ISO-9660 standard.
- Even though RMS sequential file record structures (including variable length records) are directly supported by the ISO-9660 standard, many other platforms do not adhere to that portion of the standard.

But even in a pure-VMS environment, there are reasons why one might want to store data in the ISO-9660 format, as described in Table 2-1, Comparison of ISO-9660 to ODS-n for CDROM/DVD Storage on VMS.

**Table 2-1 Comparison of ISO-9660 to ODS-n for CDROM/DVD Storage on VMS**

Reasons to create ISO-9660 Volumes	
Characteristic	Description
Optimized for CDROM and DVD	Structures internal to the ISO-9660 format speed up access to data on a medium where seeks are relatively slow compared to transfer times. By contrast, the ODS-n data structure was designed for a read-write environment supporting careful update so that data would not become inconsistent in the event of a system crash. ISO-9660, as a format not allowing updates once a Volume has been written, is free of those needs.
Volume Switching	ISO-9660 support on VMS is designed to allow mounting of just some of the members of a Volume Set, switching to other members only if their data is required. This allows support of extremely large (up to 65535-member) Volume Sets without excessive numbers of drives.
Adding to a Volume Set	The ISO-9660 standard is designed to allow easy addition of new Volumes to an existing Volume Set, with the later Volumes containing full directory information about contents of the earlier Volumes. ODS-n, by contrast, was designed for read-write disks and requires the ability to write to older Volumes—not quite the model for CDROM and DVD

---

Reasons to create ODS-n Volumes	
Characteristic	Description
A simpler MOUNT command	One must specify the /UCS_SEQUENCE="(B" qualifier to mount an ISO-9660 Volume whose Directory Identifiers and File Identifiers contain dollar-signs or dashes or the /UCS_SEQUENCE="-A" to mount an ISO-9660 Volume whose Directory Identifiers and File Identifiers use the full ODS-5 character set.
CDROM support back to VMS 5.0	ISO-9660 format is not supported on VMS versions prior to V6.1 (Alpha) or V6.2 (VAX).
Booting from CDROM	The VMS operating system itself will not boot from an ISO-9660 format disc. While most people do not need to write discs from which VMS will boot, for those who do, the need is critical.

---

### 2.1.1 What About Hybrid Volumes ?

An initial reaction might be to suggest use of a hybrid volume format, containing both ISO-9660 and ODS-n directory structures. That would merge some advantages of the two formats, but with certain drawbacks:

- Data Sharing is required to conserve space

Except for volumes that would otherwise be less than half full, it takes some tricky tool integration to share the file data between the ISO-9660 directory tree and the ODS-n directory tree.

- Inability to add to Volume Sets

The ODS-n model does not provide for adding members to a read-only volume set. Thus while the ISO-9660 side of a hybrid volume set could be extended, the ODS-n side could not.

- Differences in the Volume Set model

An ODS-n volume set is designed to be used with all members of the volume set mounted at the same time. This is only suitable for situations where one can be guaranteed that those reading the discs have as many disc drives as are contained in the volume set.

For example, one might presume that if the VMS layered product software distribution were done on ISO-9660 a reasonable approach would be to make it a single Volume Set, but making it an ODS-n volume set would certainly cause problems.

In summary, a hybrid ISO-9660/ODS-n volume (which LJK/CDROM does not currently support), would be useful for single volumes but not for volume sets.

---

## 2.2 The Nature of ISO-9660 Volumes

An ISO-9660 Volume bears certain similarities to a traditional VMS ODS-n volume, but also has certain differences.

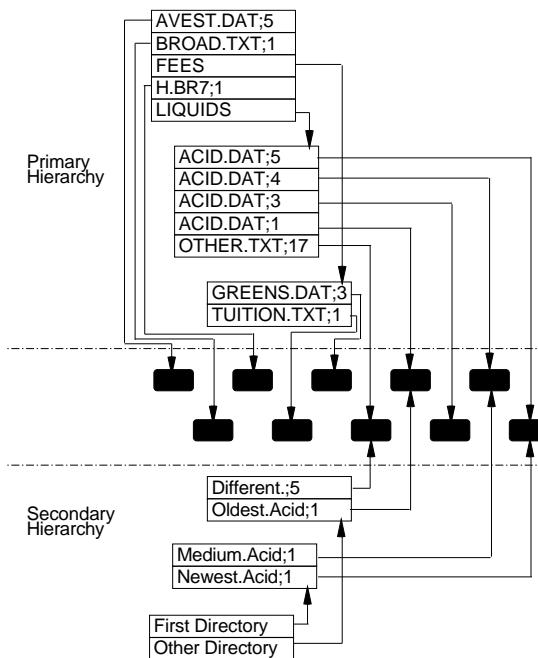
---

### 2.2.1 Hierarchies

In recognition of the reality that different operating systems will have different needs, an ISO-9660 Volume can have a variety (limited only by disc space) of Hierarchies, each consisting of a user-visible directory tree and additional metadata that may or may not be apparent to the user. This is a considerable departure from ODS-n, where there is a single hierarchy based at the master file directory (MFD, [000000]). All directories can be found in that one tree on ODS-n.

One prominent use of these separate Hierarchies is to provide different Character Sets for diverse computing environments. One Hierarchy might have a limited structure restricting File Identifiers to uppercase alphanumeric characters while another Hierarchy had diverse characters. These various Hierarchies can have their directory entries point to the same File data, sharing it in the same way that an alias directory entry does on a traditional ODS-n disk. While Files can be shared between Hierarchies, there is no requirement that a File be accessible from all Hierarchies.

Figure 2–1 Multiple Hierarchies Pointing to the Same Files



ISO-9660 has nothing comparable to an alias entry that points to a directory. Directories are not shared at all between Hierarchies, since the entries within each Directory must conform to the Hierarchy to which that Directory belongs.

## 2.2.2 File Identifiers

ISO-9660 specifies File Identifiers quite similar to those for VMS, consisting of three parts and two separators:

- File Name
- SEPARATOR 1 (.)
- File Name Extension
- SEPARATOR 2 (;)
- File Name Version (a decimal number between 1 and 32767 inclusive)

with two additional rules:

- 1 The File Name and File Name Extension cannot both be null.
- 2 The total length of the File Name plus the total length of the File Name Extension cannot exceed 30.

Thus the ISO-9660 File Identifier look familiar to VMS users, and it can be customized to look familiar to users of platforms that have different

characteristics, such as lacking support for file versions. There is no ISO-9660 requirement that display of File Identifier information look the same in every environment. Some versions of MacOS allow the user to control whether ISO-9660 File Name Versions will be displayed or suppressed, while Windows NT in some cases suppresses display of filename extensions even from its own native disk formats.

---

## 2.2.3 Directory Identifiers

ISO-9660 Directory Identifiers, on the other hand, consist of a single field, up to 31 characters in length. ISO-9660 Directory Identifiers have no extension<sup>2</sup> or version components.

---

## 2.2.4 Character Sets

Every Hierarchy on an ISO-9660 Volume has a specific Character Set allowed for metadata. That Character Set is known as the “C-Characters” and is defined for the Hierarchy according to the ISO-2022 standard using a set of Escape Sequences.

---

### 2.2.4.1 D1-Characters

A subset of the “C-Characters” are called D1-Characters and are the only ones allowed for certain metadata fields:

- Directory Identifiers
- File Identifiers
- Volume Identifier
- Volume Set Identifier

---

### 2.2.4.2 A1-Characters

A different subset of the “C-Characters” are called A1-Characters and are the only ones allowed for the remaining character metadata fields:

- System Identifier
- Publisher Identifier
- Data Preparer Identifier
- Application Identifier

---

### 2.2.4.3 Character Set Agreements

The ISO-9660 specification provides no enforcement of the subsets for D1-Characters and A1-Characters. Those restrictions are only by agreement between those producing Volumes and those reading them.

---

<sup>2</sup> An extension **is** allowed in a Directory Identifier by the Microsoft Joliet format, which is just one of several reasons that Joliet does not really conform to the ISO-9660 specification.

## 2.2.5 Primary Hierarchy

---

There is a requirement for one primary Hierarchy that must exist on every ISO-9660 Volume. The primary Hierarchy allows only the following for D-Characters:<sup>3</sup>

- A-Z (uppercase-only)
- 0-9
- \_ (underscore)

The primary Hierarchy allows only the following for A-Characters:<sup>4</sup>

A-Z (uppercase-only)

0-9

\_ (underscore)

space

!

"

%

&

,

(

)

\*

+

,

- (dash)

.

/

:

;

<

=

>

?

While a given operating system may or may not be able to interpret the characters used to name Files and Directories in various Hierarchies on an ISO-9660 Volume, all operating systems should be able to interpret the characters used to name Files and Directories in that primary Hierarchy.

---

<sup>3</sup> The term D-Characters is used for the list of characters specified for the primary Hierarchy, analogous to the term D1-Characters for lists of characters defined for other Hierarchies.

<sup>4</sup> The term A-Characters is used for the list of characters specified for the primary Hierarchy, analogous to the term A1-Characters for lists of characters defined for other Hierarchies.

## 2.2.6 Levels of Interchange

There is another dimension of ISO-9660 compatibility across operating systems, as shown in Table 2–2, ISO-9660 Levels of Interchange:

**Table 2–2 ISO-9660 Levels of Interchange**

Level	Restrictions
Level 1	Files can contain only one File Section. File Names can contain no more than 8 characters. File Name Extensions can contain no more than 3 characters. Directory Identifiers can contain no more than 8 characters.
Level 2	Files can contain only one File Section.
Level 3	No restrictions.

For most ISO-9660 Hierarchies those levels of interchange are a matter of understanding between those creating a Volume and those reading that Volume. There is an exception in the case of Microsoft Joliet Hierarchies, where the interchange level is actually encoded into the Escape Sequences.

## 2.2.7 Beyond ISO-9660

The ISO-9660 standard provides multiple mechanisms for extensions, and luckily three that are of interest to diverse operating systems do not conflict with each other.

### 2.2.7.1 Posix Rock Ridge Extensions

The Rock Ridge extensions provide storage for data elements needed for Unix-like environments. On any given Hierarchy, LJK/CDROM can support the following standard:

- Rock Ridge Interchange Protocol Revision 1.09—14-Aug-1991

or alternatively it can support the combination of a following two standards:

- IEEE P1281 System Use Sharing Protocol Revision 1.12—8-Jul-1994
- IEEE P1282 Rock Ridge Interchange Protocol Revision 1.12—8-Jul-1994

### 2.2.7.2 VMS-specific RMS File Attributes

The ISO-9660 standard itself supports RMS file attributes for sequential files, but for relative and indexed files VMS-specific RMS attributes must be added to the information for a File. Since this information is associated with a File, it is independent of Hierarchies.

---

### 2.2.7.3 Microsoft Joliet Hierarchies

Unlike the Unix and VMS extensions described above, the Microsoft Joliet extensions actually force a violation of the ISO-9660 standard. In fairness, ISO-9660 did not provide a central feature that Microsoft needed to support Windows NT—two-byte characters. The Microsoft Joliet extensions require use of a separate Hierarchy.

---

## 2.3 The Nature of ISO-9660 Volume Sets

Each ISO-9660 Volume has a particular Volume Sequence Number from 1 to 65535 indicating its position in its Volume Set. Even a single Volume written in isolation has a Volume Sequence Number of 1, and could be used as the basis for writing additional members of the same Volume Set.

Unlike for traditional VMS ODS-n volume sets, with ISO-9660 Volume Sets you can add Volumes without writing onto existing Volumes. This is particularly suited to read-only media like CDROM and DVD.

---

### 2.3.1 The ISO-9660 Volume Group

Each time Volumes are added to a Volume Set, the Volumes (possibly only one) being added form a new Volume Group. Each Volume in the new Volume Group contains directory information for all the Hierarchies in all the prior Volume Groups in the Volume Set, as well as all the Volumes in the new Volume Group being written.

Looking at this from the VMS perspective, it means that, in order to have full access to the Volume Set, the first Volume you mount must be a member of the latest Volume Group for the Volume Set.<sup>5</sup> Once that mount has been done, VMS has information about all files in the Volume Set and can engage in volume switching to get the right disc for retrieving any file.

---

<sup>5</sup> If the first Volume you mount is a member of some earlier Volume Group, the Volume Set will appear as it did just after that Volume Group was written, and volume switching would have no occasion to attempt to access Volumes in subsequent Volume Groups.

# 3 Defining Character Sets

This chapter describes how to define Character Sets within LJK/CDROM.

As described in Section 2.2.4, Character Sets, ISO-9660 restricts the Character Sets that can be used for metadata within a Hierarchy. There are two levels of such restrictions:

- C-Characters  
Selection of an ISO-2022 G0 (and optionally G1) character set, enforced by the Escape Sequences field within the ISO-9660 format.
- D1-Characters and A1-Characters  
Further restriction of which characters can be used for particular metadata within a Hierarchy. This is not enforced by the ISO-9660 format directly, but is the subject of an agreement between those producing ISO-9660 Volumes and those reading them.

The *DEFINE CHARACTER\_SET* command within LJK/CDROM associates a user-chosen name with the attributes of a Character Set:

- Escape Sequences
- A1-Characters
- D1-Characters

## 3.1 Predefined Character Sets

LJK/CDROM predefines several Character Sets (and thereby reserves those names for Character Sets) to reduce the need for custom definitions of Character Sets. The predefined Character Sets are as follows:

**Table 3–1 Predefined Character Sets**

LJK/CDROM Name	Escape Sequences	D/D1-Characters	Other Restrictions
LJK\$ISO9660	none	A-Z, digits and underscore	none
LJK\$MICROSOFT_L1	%/@	Microsoft characters	8.3 names, files cannot span Volumes
LJK\$MICROSOFT_L2	%/C	Microsoft characters	files cannot span Volumes
LJK\$MICROSOFT_L3	%/E	Microsoft characters	none
LJK\$VMS	(B	A-Z, digits, underscore, dollar sign and dash	none
LJK\$VMS_EXTENDED	-A	all ODS-5 characters (i.e., not * or ?)	none

## Defining Character Sets

To define simple Character Sets you can use the same Escape Sequences as one of the Character Sets predefined by LJK/CDROM. For more complex Character Sets, however, you will need a copy of the ISO-2022 standard, since LJK/CDROM does not enforce all the restrictions of all the Escape Sequences allowed by ISO-2022. Using the *DEFINE CHARACTER\_SET* command is what specifies to LJK/CDROM the fact that particular A1-Characters and D1-Characters are legal under a particular set of Escape Sequences.

It is important to remember that the name of a Character Set is just an internal reference used during one invocation of LJK/CDROM.

# 4

## Creating Hierarchies

---

This chapter describes how to create Hierarchies in an ISO-9660 Volume by using LJK/CDROM.

The combination of Volume Identifier and Escape Sequences for each Hierarchy on a Volume must be unique, and during actual use of a Volume that combination will suffice for identification.

However, within LJK/CDROM a user-specified Hierarchy name is used to describe a Hierarchy, providing an implicit link to a particular Character Set definition.

The *CREATE HIERARCHY* command within LJK/CDROM associates with a user-chosen name the attributes desired for the Hierarchy. The full range of possible attributes is described in LJK/CDROM, under the command *CREATE HIERARCHY*, but the most important ones are as follows:

- Character Set†  
The name of the Character Set to be used for the Volume Set being created.
- Effective Date and time  
The date and time when the information in the Volume may be used. If this is not specified, the information may be used immediately.
- Expiration Date and Time  
The date and time when the information in the Volume may be regarded as obsolete. If this is not specified, the information shall not be regarded as obsolete.
- Posix extension type‡  
The type of Posix extension (if any) to be used in the Hierarchy. Type names available are IEEE\_P1282 and RRIP\_1\_09.
- Set Identifier‡  
The Volume Set Identifier for the new Volume Set being created.
- Set Size  
The maximum Volume Sequence Number to be created for the current Volume Group. If the specified data will not fit within that limit, an error message will be written before writing any data.
- Space  
The number of Logical Blocks to be created on the Volumes.

---

† The Escape Sequences for this value must match those for any prior Volume Groups of a Volume Set. That is the mechanism whereby LJK/CDROM associates a Hierarchy name with an existing Hierarchy on an existing Volume Set.

‡ This value can only be specified for the initial Volume Group of a Volume Set.

## Creating Hierarchies

- Volume Identifier

A set of identifications for the Volumes, used as the “volume-label” by the VMS *MOUNT* command.

---

### 4.1 Hierarchy Names and Volume Sets

There is no ISO-9660 requirement that the Volume Identifier for one Hierarchy on a Volume differ from the Volume Identifier for other Hierarchies on that Volume so long as the Escape Sequences within the Character Set provides a distinction.

Likewise there is no ISO-9660 requirement that the Volume Identifier for a Hierarchy on one Volume differ from the Volume Identifier for the same Hierarchy on a different Volume in the same Volume Set. But the use of Volume Identifiers in the VMS *MOUNT* command makes it preferable to use different Volume Identifiers for successive Volumes, particularly when mounting multiple Volumes from the same Volume Set at the same time. Therefore, LJK/CDROM defaults to Volume Identifiers that include the Volume Sequence Number within the first 12 characters of the Volume Identifier (since VMS Mount deals mainly with the first 12 characters).

The Hierarchy name provides a single user-friendly name within LJK/CDROM for describing a Hierarchy.

Hierarchies can be added to, but not removed from successive Volume Groups of a Volume Set. If there are no new Files provided for a Hierarchy when extending a Volume Set, metadata for the Hierarchy will still be present on the new Volumes, pointing to files on previous Volumes, ready for access via volume switching.

---

### 4.2 Predefined Hierarchies

If no Hierarchies have been defined at the time the command *ADD* or the command *WRITE*<sup>1</sup> is issued, LJK/CDROM automatically defines the following four Hierarchies and proceeds using those Hierarchies.

**Table 4–1 Predefined Hierarchies**

LJK/CDROM Hierarchy	Character Set Used
LJK\$OPEN	LJK\$ISO9660
LJK\$MICROSOFT	LJK\$MICROSOFT_L3
LJK\$ODS2	LJK\$VMS
LJK\$ODS5	LJK\$VMS_EXTENDED

If Hierarchies already have been defined at the time that the command *ADD* or the command *WRITE* is issued but none of those Hierarchies use

---

<sup>1</sup> This will always be true if the command *WRITE* is issued directly from DCL rather than from subsystem mode.

a Character Set corresponding to the Character Set LJK\$ISO9660, just the LJK\$OPEN Hierarchy above will be automatically defined.



# 5 Formatting a Volume

---

This chapter tells how to use LJK/CDROM to create an ISO-9660 Volume.

As discussed in Section 1.2, Formatting a CDROM Volume, the simplest method of formatting a Volume with LJK/CDROM is to use one of the commands:

```
LJK/CDROM WRITE ddcu: source-filespec-1,source-filespec-2,...
```

or

```
LJK/CDROM WRITE target-filespec source-filespec-1,source-filespec-2,...
```

**Note: The LJK/CDROM *WRITE* command takes the target file specification before the source file specifications because the source file specifications are optional.**

But for more precise control you should just issue the DCL command

LJK/CDROM

to enter subsystem mode and from there use the following commands (typically in the order shown):

1 *DEFINE CHARACTER\_SET*

With this command you can define your own sets of legal characters for the Hierarchies you will create, if the predefined Character Sets described in Table 3-1, Predefined Character Sets, do not meet your needs.

2 *SET DEFAULT*

This command is an adjunct to the *CREATE HIERARCHY* command, intended to avoid repetitive typing and limitations on command line length. You can issue one or more *SET DEFAULT* command prior to each *CREATE HIERARCHY* command, providing default values for the *CREATE HIERARCHY* command.

3 *CREATE HIERARCHY*

With this command you define a Hierarchy, using the qualifiers you provide as well as those you have used before with the *SET DEFAULT* command.

4 *ADD*

This command specifies one or more input files for inclusion on the Volume being formatted.

5 *WRITE*

## Formatting a Volume

This command causes the actual formatting of one or more Volumes, possibly including additional input files specified as you did with the *ADD* command.

---

## 6 Formatting a Volume Set

This chapter tells how to use LJK/CDROM to create an ISO-9660 Volume Set.

Volume Sets are key to the power of ISO-9660 allowing efficient access to much more data than will fit on a single disc. There are two general scenarios in which you format multiple Volumes to form a Volume Set.

---

### 6.1 Formatting Multiple Volumes With One Command

Using just the same commands as for a single Volume you can format a multiple member Volume Set. The major accommodation you need to make for the existence of multiple Volumes is to ensure the considerations in Section 4.1, Hierarchy Names and Volume Sets, are taken into account for smooth use of the resulting Volume Set on VMS. Using a */VOLUME* qualifier for the *CREATE HIERARCHY* command that is similar to the default (or leaving the default) is the easiest way to do this.

---

### 6.2 Controlling Placement onto Volumes with Smaller Volume Groups

The Volumes written in response to a single command are called a Volume Group. By default LJK/CDROM arranges files on these various Volumes to minimize the number of Volumes required. If you would prefer to have precise control over which Files are on which Volumes, write each Volume as a separate Volume Group, followed by writing the next Volume using the method described in Section 6.3, Linking New Volumes into an Existing Volume Set.

---

### 6.3 Linking New Volumes into an Existing Volume Set

LJK/CDROM can add to an existing Volume Set through use of the *LINK GROUP* command. The list of commands used to format a Volume Set thus becomes:

1 *DEFINE CHARACTER\_SET*

With this command you can define your own sets of legal characters for the Hierarchies you will create, if the predefined Character Sets described in Table 3-1, Predefined Character Sets, do not meet your needs.

If you are adding Volumes to an existing Volume Set, you should define any custom Character Sets with this command before issuing the *LINK GROUP* command. By doing that, you ensure the Character Sets needed by the Hierarchies on the existing Volume Set.

2 *SET DEFAULT*

## Formatting a Volume Set

This command is an adjunct to the *CREATE HIERARCHY* command, intended to avoid repetitive typing and limitations on command line length. You can issue one or more *SET DEFAULT* commands prior to each *CREATE HIERARCHY* command, providing default values for the *CREATE HIERARCHY* command.

*SET DEFAULT* commands issued prior to a *LINK GROUP* command also affect the following defaults for that command:

- Volume Set Size
- Volume Space Size
- Volume Identifiers

### 3 *LINK GROUP*

This command reads Hierarchies from the last Volume Group of an existing Volume Set. A Volume from that Volume Group must be mounted /FOREIGN in the device specified by the *LINK GROUP* command. The *LINK GROUP* command must be issued before any *CREATE HIERARCHY* command.

### 4 *CREATE HIERARCHY*

With this command you define a Hierarchy, using the qualifiers you provide as well as those you have used before with the *SET DEFAULT* command.

In the particular case where one has issued the *LINK GROUP* command, the *CREATE HIERARCHY* command first makes an attempt to match an existing anonymous Hierarchy that was imported from the previous Volume Group. If that succeeds, the existing Hierarchy is renamed, rather than creating a new Hierarchy.

### 5 *ADD*

This command specifies one or more input files for inclusion on the Volume being formatted.

### 6 *WRITE*

This command causes the actual formatting of one or more Volumes, possibly including additional input files specified as you did with the *ADD* command.

# 7

---

## Verifying a Volume

This chapter tells how to use LJK/CDROM to validate the ISO-9660 format of a Volume.

The longstanding rules of computer data interchange are:

- 1 Be conservative in the data your program generates.
- 2 Be liberal in the data your program accepts.

In the arena of ISO-9660, many operating systems, including VMS, are quite liberal in what they will accept, correctly reading Volumes that have errors they can work around. That good compliance with rule 2 is important because many nominally ISO-9660 Volumes are written without following rule 1, generating whatever will be accepted by the operating systems on which the producing organization happens to test reading the data.

This obviously can lead to problems when data is transferred to someone using a different operating system or a different version of a tested operating system.

The purpose of the command *LJK/CDROM VERIFY* is to provide a strict reading of ISO-9660 Volumes so that authors can determine whether what they write correctly adheres to the relevant specifications, rather than whether it just happens to work on a limited number of operating systems where they test reading their Volumes.

---

### 7.1 Implicit Verification Processing

In addition to the explicit use of the *VERIFY* command, LJK/CDROM also performs verification processing in three other circumstances.

- 1 on each Volume after it has been formatted

This verification can be suppressed with the */NOVERIFY* qualifier.

- 2 on input Volumes to the *COMPARE* command

This verification cannot be suppressed, since LJK/CDROM cannot make a meaningful comparison between Volumes that are not in a proper ISO-9660 format.

- 3 on input Volumes to the *LINK GROUP* command

This verification cannot be suppressed, since LJK/CDROM cannot merge in data from an exiting Volume that is not in a proper ISO-9660 format.

---

### 7.2 Suppressing Errors

When you explicitly use the *VERIFY* command and encounter errors, you may want to get a thorough understanding of the exact nature of the errors. By default LJK/CDROM will display every error it encounters, but at times that can get in the way, since the same fault that caused an error on one File from the input Volume might very well cause the same error on every other File as well!

There are two mechanism for suppressing errors from an explicit *VERIFY* command, allowing you to look more carefully to understand all the sources of such errors.

---

#### 7.2.1 Suppressing Multiple Errors

The *VERIFY* command qualifier */SHOW* defaults to *MULTIPLE\_ERRORS* to indicate each instance of a particular error. By specifying */SHOW=(NOMULTIPLE\_ERRORS,...* you can suppress many error reports, particularly those associated with Posix extensions.

---

#### 7.2.2 Suppressing Particular Errors

The *VERIFY* command qualifier */TOLERATE* can be used to suppress particular classes of errors, particularly for cases when a Volume was created with a mechanism that made a particular type of error repeatedly.

# 8

---

## Comparing Two Volumes

This chapter tells how to use LJK/CDROM to perform a logical comparison between two Volumes).

The *COMPARE* command makes a logical comparison between two Volumes. In doing so, it ignores certain differences that would be reported by a block-for-block comparison:

- Block addresses on disc  
Relocation on disc does not affect the logical contents.
- Lengths of directories  
Comparisons of directories discover differences between record sets.
- Dissimilar System Use Sharing Protocol entries  
Ordering only matters between entries of the same type.
- System Use Sharing Protocol CE and ST entries  
Those affect placement on an individual disc, rather than logical contents.

There are two particular circumstances where you might find it helpful to be able compare two Volumes.

- A Volume has been slightly modified  
If you use LJK/CDROM to change just a few files on a disk produced by another tool (for instance, to provide RMS file attributes) a comparison may be in order to ensure that just the intended changes were made.
- A Volume has been added to an existing Volume Set  
While new Files will be added, all the metadata from the prior Volume Group should be intact on the new Volume.

The *COMPARE* command was provided within LJK/CDROM in order to check the operations of LJK/CDROM, but of course it can be used to check the operations of other ISO-9660 formatting tools as well.



---

# 9 Command Reference

This chapter describes the commands for controlling LJK/CDROM in a reference format.

---

## 9.1 Command Summary

LJK/CDROM commands can be divided into three basic groups:

- Miscellaneous

HELP  
SHOW VERSION  
EXIT

- Validation Commands

Those used to check the formatting of existing Volumes:

VERIFY  
COMPARE

- Creation Commands

Those used to create new Volumes:

DEFINE CHARACTER\_SET  
LINK GROUP  
SET DEFAULT  
CREATE HIERARCHY  
ADD  
WRITE

---

## 9.2 Command Formats

There are two distinct methods for issuing LJK/CDROM commands.

- DCL Command Format

Commands are given directly from the DCL prompt (\$ by default).

This format has the advantage of allowing DCL symbol substitution for parameterized execution of rote functions.

- Subsystem Command Format

The command *LJK/CDROM* is given first to get into LJK/CDROM, where subsequent commands are given in response to the prompt LJKC>.

This format has the performance advantage of only incurring one VMS Image Activation for a whole series of LJK/CDROM functions.

## Command Reference

In the command descriptions shown in the following pages both the DCL Command Format and the Subsystem Command Format are shown where applicable.

**Requirement:** You must use the Subsystem Command Format when using any of the commands:

- **DEFINE CHARACTER\_SET**
- **LINK GROUP**
- **SET DEFAULT**
- **CREATE HIERARCHY**
- **ADD**

because those commands merely set up definitions for a subsequent WRITE command. If you exit from Subsystem mode back to DCL before you issue the WRITE command, all results of the 5 listed commands will be lost.

**The LJK/CDROM WRITE command can be issued directly from DCL, but only when creating very simple Volumes, since the 5 listed commands cannot be used for preparation.**

An additional format is shown for the HELP command, since help information about LJK/CDROM is stored in the main HELP library and can therefore be accessed by the DCL HELP command.

While LJK/CDROM returns meaningful status to DCL, the code may have the INHIB\_MSG bit set or it may be clear. Those who write command procedures handling the status returned by LJK/CDROM should always discount that bit before making comparisons.

---

### 9.3 Specifying Arbitrary Characters

Qualifiers described in any of the forms:

```
/QUALIFIER_NAME=character-list  
/QUALIFIER_NAME=a/a1-character  
/QUALIFIER_NAME=d/d1-character
```

can take a list of elements which will be concatenated into a single set of characters as follows:

- Quoted string

The contents of the quoted string are taken as a list of characters.

- Number

The ASCII value of the number is taken as a character.

This mechanism allows a full range of ANSI characters to be specified, but does not depend on VMS extended parsing which is not available on earlier versions of VMS or on VAX.

LJK/CDROM **does** check to ensure that the characters specified conform to ISO-9660 and ISO-2022 by being restricted to the G0 and G1 ranges (i.e., not having decimal values 0-31 or 128-159)

---

## 9.4 Detailed Specification of Individual Commands

The following pages in this chapter contains full documentation of individual commands available for LJK/CDROM.

# Command Reference

## ADD

---

## ADD

Add one or more VMS files to the set which will be written to a Volume.

---

**FORMAT**            **\$ LJK/CDROM ADD**    *input-filespec,...*

---

Positional Qualifiers	Defaults
<i>/EFFECTIVE=date</i>	<i>None.</i>
<i>/[NO]EXPIRATION</i>	<i>/EXPIRATION</i>
<i>/NAMES=list-of-hierarchy-name-pairs</i>	<i>None.</i>
<i>/SHOW=(keyword[,...])</i>	<i>See below.</i>
<i>/[NO]VMS_ATTRIBUTES</i>	<i>/VMS_ATTRIBUTES</i>

---

- restrictions**
- This command is only effective from the subsystem prompt (LJKC>).
  - You must have read access to the specified files.
  - You must have read access to any directories for which input-filespecs are wildcarded.

---

**PARAMETERS**    *input-filespec*  
Possibly wildcarded file specification for files to be added.

---

**DESCRIPTION**    Add the specified VMS file(s) to the set which will be written to a Volume when the next WRITE command is given.

The ADD command and the corresponding WRITE command must be issued in the same invocation of LJK/CDROM subsystem mode (with LJK/CDROM issuing the prompt "LJKC>").

---

**QUALIFIERS**    */EFFECTIVE=date*  
Specifies the date that should be recorded for the effective date of the file(s). Such a date will be recorded in the ISO-9660 Extended Attribute Record, and in Hierarchies for which Rock Ridge information is being recorded the date will also be recorded in the TF System Use Entry.

***/EXPIRATION (D)***

***/NOEXPIRATION***

Specifies that the expiration date from the VMS file is to be recorded for the expiration date of the file on the Volume being written. Such a date will be recorded in the ISO-9660 Extended Attribute Record, and in Hierarchies for which Rock Ridge information is being recorded the date will also be reported in the TF System Use Entry.

This qualifier is provided because the meaning of an Expiration Date on a VMS file is locally defined and such a date may not be appropriate for distribution to other sites.

***/NAMES=(*[hierarchy-name=]*ISO-filespec[,...])***

Specifies the File Identifiers and Directory Identifiers under which files are to be stored on various Hierarchies of the Volume being created.

The Directory Identifiers and File Identifiers for the Primary Hierarchy must consist of D-Characters (no dollar signs allowed) while the Directory Identifiers and File Identifiers for supplementary Hierarchies must consist of D1-Characters (as specified by the Character Set for the Hierarchy).

If the “[hierarchy-name=]” string is omitted, LJK/CDROM attempts to store under the specified filespec in all Hierarchies, producing an error message if that effort fails in any Hierarchy.

If this qualifier is not provided, LJK/CDROM attempts to use the input Directory Identifiers and File Identifiers for all Hierarchies, producing an error if any of the characters are illegal for any of the Hierarchies.

Besides the VMS-style, specification of Directory Identifiers for a Hierarchy can also be done in two other manners, if the initial character is not “[” or “<”:

- Colon separations

Each Directory Identifier is followed by a colon, similar to Macintosh filespecs.

- Slash separations

Each Directory Identifier is followed by a left angle bracket, similar to Unix filespecs.

If no directory is specified, the directory from the input filespec is used for the output volume. To put a file into the root directory, as is required for files specified on the *CREATE HIERARCHY* or *SET DEFAULT* command qualifiers:

- */ABSTRACT\_FILE=file.ext;version*
- */APPLICATION=\_file.ext;version*
- */BIBLIOGRAPHIC\_FILE=file.ext;version*
- */COPYRIGHT\_FILE=file.ext;version*
- */DATA\_PREPARER=\_file.ext;version*
- */PUBLISHER=\_file.ext;version*

specify the directory as “[” in the */NAMES* qualifier.

***/SHOW=DEBUG***

***DETAILS***

***MULTIPLE\_ERRORS (D)***

***PROGRESS (D)***

***SUMMARY (D)***

Specifies what information should be displayed during the command.

## Command Reference

### ADD

- **DEBUG**  
Display extremely detailed information suitable for submitting a bug report.
- **DETAILS**  
Display detailed information suitable for analyzing a problem.
- **MULTIPLE\_ERRORS**  
Display even those errors that were encountered on a previous file.
- **PROGRESS**  
Display indications of progress in processing the command.
- **SUMMARY**  
Display a summary of major results of the command.

### ***/VMS\_ATTRIBUTES (D)***

### ***/NOVMS\_ATTRIBUTES***

Specifies that VMS-specific RMS attributes are to be stored on the Volume being created. This is the default.

Omitting the VMS-specific RMS attributes will inhibit reading of relative and indexed files from the Volume, but will not interfere with sequential files, including those with variable-length records.

Omitting the VMS-specific RMS attributes will not save any space on the Volume. The major use for omitting the VMS-specific RMS attributes would be to test on VMS (by mounting the resulting Volume) that the standard ISO-9660 support for sequential file record formats are working correctly. That is the purpose for which LJK Software uses the */NOVMS\_ATTRIBUTES* form of this qualifier.

---

## EXAMPLE

```
LJKC> ADD FILE.DAT/NAMES=(QWX=AA:BB:,WER=[QWT.OYT])
```

Include FILE.DAT in the [AA.BB] Directory of the QWX Hierarchy and the [QWT.OYT] Directory of the WER Hierarchy.

```
LJKC> ADD DISK$A:[*... ]FILE.DAT/NAMES=(QWX=FOO.TXT,WER=FIE.LOG)
```

Include all instances of FILE.DAT in the corresponding Directories naming them FOO.TXT in the QWX Hierarchy and FIE.LOG in the WER Hierarchy.

```
LJKC> ADD DATA.TXT/NAMES=(EWOY=FIE.LOG;)
```

Include DATA.TXT in the corresponding Directory of the EWOY Hierarchy naming it FIE.LOG and giving it a File Name Version one greater than the highest existing File Name Version.)

## Command Reference

### ADD

```
LJKC> ADD DATA.TXT/NAMES=(EWOY=FIE.LOG)
```

Include DATA.TXT in the corresponding Directory of the EWOY Hierarchy naming it FIE.LOG and preserving the initial version number as the File Name Version (failing if a file with that File Name, File Name Extension and File Name Version previously was included in that Directory.)



- **DEBUG**  
Display extremely detailed information suitable for submitting a bug report.
- **DETAILS**  
Display detailed information suitable for analyzing a problem.
- **MULTIPLE\_ERRORS**  
Display even those errors that were encountered on a previous file.
- **PROGRESS**  
Display indications of progress in processing the command.
- **SUMMARY**  
Display a summary of major results of the command.

---

### EXAMPLE

```
$ LJK/CDROM COMPARE DKA400: TEST.DAT
```

Compare the Volume on DKA100: (currently mounted /FOREIGN) to the volume image in file TEST.DAT of the current default directory.

```
$ LJK/CDROM COMPARE TEST.DAT/DATA TEST.DIT
```

Compare the volume images in files TEST.DAT and TEST.DIT including the data within individual files.

## Command Reference

### CREATE HIERARCHY

---

## CREATE HIERARCHY

Create (and name for use within LJK/CDROM) a Hierarchy to be used on the Volumes to be written.

---

### FORMAT **LJKC> CREATE HIERARCHY** *hierarchy-name*

---

Command Qualifiers	Defaults
<i>/ABSTRACT_FILE=d/d1-characters</i>	<i>None.</i>
<i>/APPLICATION=a/a1-characters</i>	<i>None.</i>
<i>/APPUSE=filespec</i>	<i>None.</i>
<i>/BIBLIOGRAPHIC_FILE=d/d1-characters</i>	<i>None.</i>
<i>/CHARACTER_SET=name</i>	<i>LJK\$ISO9660</i>
<i>/COPYRIGHT_FILE=d/d1-characters</i>	<i>None.</i>
<i>/CREATION=date-time</i>	<i>Current time.</i>
<i>/DATA_PREPARER=a/a1-characters</i>	<i>None.</i>
<i>/EFFECTIVE=date-time</i>	<i>None.</i>
<i>/EXPIRATION=date-time</i>	<i>None.</i>
<i>/LOGICAL_BLOCK_SIZE=value</i>	<i>2048</i>
<i>/MODIFICATION=date-time</i>	<i>Current time.</i>
<i>/POSIX=keyword</i>	<i>/NOPOSIX</i>
<i>/PUBLISHER=a/a1-characters</i>	<i>None.</i>
<i>/SET_IDENTIFIER=string</i>	<i>current-date-time</i>
<i>/SETSIZE=maximum</i>	<i>65535</i>
<i>/SPACE=logical-block-count</i>	<i>262500</i>
<i>/SYSTEM=character-list</i>	<i>None.</i>
<i>/VOLUME=volume-name-list</i>	<i>VOLUME_*</i>

---

#### restrictions

- This command is only effective from the subsystem prompt (LJKC>).

---

### PARAMETERS ***hierarchy-name***

Name of the Hierarchy to be created.

---

**DESCRIPTION** Create (and name for use within LJK/CDROM) a Hierarchy to be used on the Volumes to be written.

---

### QUALIFIERS

#### ***/ABSTRACT\_FILE=d- or d1-characters***

Specifies the Abstract File Identifier for this Hierarchy on each Volume in the Volume Group being written.

This field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. The field must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/APPLICATION=a- or a1-characters***

Specifies the Application Identifier for this Hierarchy on each Volume in the Volume Group being written.

If this field starts with the character underscore ( ), the rest of the field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. That remainder of the field after the underscore must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/APPUSE=input-filespec***

Specifies a file containing data that is to be included in the Application Use field for this Hierarchy on each Volume in the Volume Group being written.

***/BIBLIOGRAPHIC\_FILE=d- or d1-characters***

Specifies the Bibliographic File Identifier for this Hierarchy on each Volume in the Volume Group being written.

This field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. The field must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/CHARACTER\_SET=name***

Specifies the Character Set for this Hierarchy.

***/COPYRIGHT\_FILE=d- or d1-characters***

Specifies the Copyright File Identifier for this Hierarchy on each Volume in the Volume Group being written.

This field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. The field must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/CREATION=date-time***

Specifies the Volume Creation Date and Time for this Hierarchy on each Volume in the Volume Group being written.

***/DATA\_PREPARER=a- or a1-characters***

Specifies the Data Preparer Identifier for this Hierarchy on each Volume in the Volume Group being written.

If this field starts with the character underscore ( ), the rest of the field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. That remainder of the field after the underscore must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

## Command Reference

### CREATE HIERARCHY

#### ***/EFFECTIVE=date-time***

Specifies the Volume Effective Date and Time for this Hierarchy on each Volume in the Volume Group being written.

#### ***/EXPIRATION=date-time***

Specifies the Volume Expiration Date and Time for this Hierarchy on each Volume in the Volume Group being written.

#### ***/LOGICAL\_BLOCK\_SIZE=value***

Specifies an alternate Logical Block Size to the standard 2048 bytes with which Hierarchies are created. The Logical Block Size must be the same for all the Hierarchies on a Volume.

#### ***/POSIX=IEEE\_P1282 (D) RRIP\_1\_09***

Specifies the POSIX Rock Ridge version to be used in this Hierarchy. If no value is specified for the /POSIX qualifier, IEEE\_P1282 is used. If the /POSIX qualifier is not specified, no Rock Ridge format will be used. Using a Rock Ridge format does consume extra space on the Volume.

#### ***/MODIFICATION=date-time***

Specifies the Volume Modification Date and Time for this Hierarchy on each Volume in the Volume Group being written.

#### ***/PUBLISHER=a- or a1-characters***

Specifies the Publisher Identifier for this Hierarchy on each Volume in the Volume Group being written.

If this field starts with the character underscore ( ), the rest of the field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. That remainder of the field after the underscore must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

#### ***/SET\_IDENTIFIER=string***

Specifies the Volume Set Identifier for this Hierarchy on each Volume in the new Volume Set being created. This qualifier should not be specified when the command *LINK GROUP* has previously been given, since the Volume Set Identifier in that case must come from the prior Volume Group.

The default value for this qualifier is the current VMS date-time with punctuation removed and truncated to 13 characters, for example: "09SEP20021234".

For those using the VMS Hobbyist license, an identification number will be appended to the specified (or defaulted) Volume Set Identifier.

#### ***/SETSIZE=maximum***

Specifies the maximum number of Volumes that can be included in the Volume Set, including both those being created now and any used to create previous Volume Groups within the Volume Set. If the files specified with the commands *LINK GROUP*, *ADD* and *WRITE* exceed the capacity of this number of Volumes, an error message will be displayed before any output data is written.

The ISO-9660 specification limits the maximum Volume Set Size to 65535.

***/SPACE=logical-block-count***

Specifies the number of logical blocks on the output Volume. When the output Volume is a volume image file, this is the size with which it will be created (taking into account the difference in size between Logical Blocks and VMS blocks).

The maximum number of Logical Blocks that can be specified is the number that would result  $2^{31}$  (about 2 billion) VMS blocks.

***/SYSTEM=character-list***

Specifies the System Identifier that is to be included in this Hierarchy, identifying the system which is able to act on the System Area (Logical Sectors 0-15) of the created Volume.

Generally this qualifier should be omitted.

***/VOLUME=volume-name-list***

Specifies the list of Identifiers to be used as the Volume Identifier in this Hierarchy on successive Volumes of the Volume Group being written in this session.

If a single name is provided for the list and it contains an asterisk (\*), the asterisk will be replaced by the Volume Sequence Number in each Volume Identifier as it is written.

If a single name is provided for the list and it contains no asterisk (\*), that name will be used intact for the Volume Identifier in this Hierarchy on each Volume of the current Volume Group. There is no requirement in ISO-9660 that the Volume Identifier of various Volumes differ from one another.

---

## EXAMPLE

```
LJKC> CREATE HIERARCHY SPECIAL/VOLUME=(ONE,TWO)
```

Create a Hierarchy which will use “ONE” as the Volume Identifier of the first Volume created and “TWO” as the Volume Identifier for all subsequent Volumes created.

```
LJKC> CREATE HIERARCHY MY_NAME/VOLUME=(DISC*)
```

Create a Hierarchy which will use “DISCnnn” as the Volume Identifier of Volume nnn of the Volume Set, where nnn is a variable width decimal representation.

---

## DEFINE CHARACTER\_SET

Define the characters than can be used for Directory Identifiers, File Identifiers and other metadata within one or more Hierarchies.

---

**FORMAT**            **LJKC> DEFINE CHARACTER\_SET -**  
*character-set-name*

---

Command Qualifiers	Defaults
<i>/ESCAPE_STRINGS=character-list</i>	<i>None.</i>
<i>/A1_CHARACTERS=character-list</i>	<i>See below.</i>
<i>/D1_CHARACTERS=character-list</i>	<i>See below.</i>

---

**restrictions**            • This command is only effective from the subsystem prompt (LJKC>).

---

**PARAMETERS**        *character-set-name*  
Name used within LJK/CDROM for the Character Sets.

---

**DESCRIPTION**        Define the characters than can be used for Directory Identifiers, File Identifiers and other metadata within one or more Hierarchies.

The need to use the DEFINE CHARACTER\_SET command is reduced by the fact that LJK/CDROM already provides the following six built-in Character Sets:

- LJK\$ISO9660 - A-Z, digits and underscore
- LJK\$MICROSOFT\_L1 - Microsoft characters, 8.3 File Identifiers, Files cannot span Volumes
- LJK\$MICROSOFT\_L2 - Microsoft characters, Files cannot span Volumes
- LJK\$MICROSOFT\_L3 - Microsoft characters
- LJK\$VMS - A-Z, digits, underscore, dollar sign and dash
- LJK\$VMS\_EXTENDED - ODS-5 characters (i.e., all 8-bit characters but "\*" and "?")

The primary Hierarchy will always use the LJK\$ISO9660 Character Set.

The DEFINE CHARACTER\_SET command establishes a connection between the escape strings value to be recorded on the Volume and the characters that will be accepted on ADD and READ commands. LJK/CDROM does **not** check the legality of the characters specified against the escape strings specified based on the ISO-2022 standard; that is left to the user.

LJK/CDROM **does** check to ensure that the characters specified by the /A1\_CHARACTERS and /D1\_CHARACTERS are restricted to the G0 and G1 ranges (i.e., not having decimal values 0-31 or 128-159) as specified by ISO-9660 and ISO-2022.

---

## QUALIFIERS

### ***/ESCAPE\_STRINGS=character-list***

Specify ISO-2022 G0 and optionally G1 graphic characters sets that form the C-Characters set available for Directory Identifiers, File Identifiers and other metadata.

### ***/A1\_CHARACTERS=character-list***

Specify which of the C-Characters can be used for metadata other than File Identifiers and Directory Identifiers.

If no /A1\_CHARACTERS qualifier is specified, the A1-Characters set will be the A-Characters set of A-Z (uppercase only), digits, underscore, space and the characters “!%&'()\*+,-./:;<=>?@”.

### ***/D1\_CHARACTERS=character-list***

Specify which of the C-Characters can be used for File Identifier and Directory Identifier metadata.

If no /D1\_CHARACTERS qualifier is specified, the D1-Characters set will be the D-Characters of A-Z (uppercase only), digits and underscore.

---

## EXAMPLE

```
LJKC> DEFINE CHARACTER_SET MY_CHARS/D1_CHARACTERS=0123456789
```

Define a Character Set where File Identifiers and Directory Identifiers can only contain digits (along with the required SEPARATOR 1 (.) and SEPARATOR 2 (;) for a File Identifier.

## Command Reference

### EXIT

---

## EXIT

Exit from the LJK/CDROM subsystem back to the VMS command language.

---

### FORMAT

#### LJKC> EXIT

---

Command Qualifiers	Defaults
--------------------	----------

*None.*

*None.*

---

### restrictions

*None.*

---

### PARAMETERS

*None.*

---

### DESCRIPTION

Exit from the LJK/CDROM subsystem back to the VMS command language.

---

### QUALIFIERS

*None.*

---

### EXAMPLE

LJKC> EXIT

Exit from the LJK/CDROM subsystem back to the VMS command language.



# Command Reference

## HELP

---

### EXAMPLE

\$ LJK/CDROM HELP VERIFY/TOLERATE

Display information about the /TOLERATE qualifier to the command VERIFY.

\$ LJK/CDROM HELP VERIFY \*

Display information about all subtopics for the VERIFY command.

---

## LINK GROUP

Specify an existing Volume Set onto which additional Volumes will be added.

---

**FORMAT**            **LJKC> LINK GROUP**    *device-name*

---

Command Qualifiers	Defaults
<i>None.</i>	<i>None.</i>

---

- restrictions**
- This command is only effective from the subsystem prompt (LJKC>).
  - The LINK GROUP command must be issued before any DEFINE HIERARCHY command.
  - You must have read access to the ISO-9660 CDROMs or disc images.
  - If parameter is a device rather than a file image, it must be mounted foreign.

---

**PARAMETERS**    ***device-name***  
A ISO-9660 CDROM or disc image from the highest existing Volume Group of the existing Volume Set.

---

**DESCRIPTION**    Specify an existing Volume Set onto which additional Volumes will be added.

LJK/CDROM reads the existing Volume Set to gather existing metadata that must be included on the Volumes it writes in order to make those new Volumes be part of the Volume Set.

The LINK GROUP command automatically defines within LJK/CDROM all the Hierarchies that exist on preceding members of the Volume Set, so those Hierarchies will be written to subsequent Volumes. Those Hierarchies, however, do not have names, so a subsequent DEFINE HIERARCHY command with matching Volume Identifier and Escape Sequences will affix its name to the inherited Hierarchy.

---

**QUALIFIERS**        *None.*

---

## EXAMPLE

LJKC> LINK GROUP *file-name*

Specify a Volume from the final existing Volume Group of the Volume Set to which additional new Volumes will be added.

---

## SET DEFAULT

Create default values for subsequent CREATE HIERARCHY commands.

---

### FORMAT

### LJKC> SET DEFAULT

---

Command Qualifiers	Defaults
<i>/ABSTRACT_FILE=d/d1-characters</i>	<i>None.</i>
<i>/APPLICATION=a/a1-characters</i>	<i>None.</i>
<i>/APPUSE=filespec</i>	<i>None.</i>
<i>/BIBLIOGRAPHIC_FILE=d/d1-characters</i>	<i>None.</i>
<i>/CHARACTER_SET=name</i>	<i>LJK\$ISO9660</i>
<i>/COPYRIGHT_FILE=d/d1-characters</i>	<i>None.</i>
<i>/CREATION=date-time</i>	<i>Current time.</i>
<i>/DATA_PREPARER=a/a1-characters</i>	<i>None.</i>
<i>/EFFECTIVE=date-time</i>	<i>None.</i>
<i>/EXPIRATION=date-time</i>	<i>None.</i>
<i>/LOGICAL_BLOCK_SIZE=value</i>	<i>2048</i>
<i>/MODIFICATION=date-time</i>	<i>Current time.</i>
<i>/POSIX=keyword</i>	<i>/NOPOSIX</i>
<i>/PUBLISHER=a/a1-characters</i>	<i>None.</i>
<i>/SET_IDENTIFIER=string</i>	<i>current-date-time</i>
<i>/SETSIZE=maximum</i>	<i>65535</i>
<i>/SPACE=logical-block-count</i>	<i>262500</i>
<i>/SYSTEM=character-list</i>	<i>None.</i>
<i>/VOLUME=volume-name-list</i>	<i>VOLUME_*</i>

---

### restrictions

- This command is only effective from the subsystem prompt (LJKC>).

---

### PARAMETERS

*None.*

---

### DESCRIPTION

Set default values in advance for the qualifiers would otherwise be specified with the CREATE HIERARCHY command.

The CREATE HIERARCHY command can take so many qualifiers of such potential length that it would be impossible to specify them all within DCL limits, or for that matter, the limits of accurate human typing.

The SET DEFAULT command creates accumulative values for the various qualifiers it shares with the CREATE HIERARCHY command, and those values take effect on subsequent CREATE HIERARCHY commands until the next WRITE command.

---

**QUALIFIERS**

***/ABSTRACT\_FILE=d- or d1-characters***

Specifies the Abstract File Identifier for this Hierarchy on each Volume in the Volume Group being written.

This field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. The field must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/APPLICATION=a- or a1-characters***

Specifies the Application Identifier for this Hierarchy on each Volume in the Volume Group being written.

If this field starts with the character underscore (\_), the rest of the field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. That remainder of the field after the underscore must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/APPUSE=input-filespec***

Specifies a file containing data that is to be included in the Application Use field for this Hierarchy on each Volume in the Volume Group being written.

***/BIBLIOGRAPHIC\_FILE=d- or d1-characters***

Specifies the Bibliographic File Identifier for this Hierarchy on each Volume in the Volume Group being written.

This field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. The field must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/CHARACTER\_SET=name***

Specifies the Character Set for this Hierarchy.

***/COPYRIGHT\_FILE=d- or d1-characters***

Specifies the Copyright File Identifier for this Hierarchy on each Volume in the Volume Group being written.

This field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. The field must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

***/CREATION=date-time***

Specifies the Volume Creation Date and Time for this Hierarchy on each Volume in the Volume Group being written.

***/DATA\_PREPARER=a- or a1-characters***

Specifies the Data Preparer Identifier for this Hierarchy on each Volume in the Volume Group being written.

## Command Reference

### SET DEFAULT

If this field starts with the character underscore ( ), the rest of the field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. That remainder of the field after the underscore must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

#### ***/EFFECTIVE=date-time***

Specifies the Volume Effective Date and Time for this Hierarchy on each Volume in the Volume Group being written.

#### ***/EXPIRATION=date-time***

Specifies the Volume Expiration Date and Time for this Hierarchy on each Volume in the Volume Group being written.

#### ***/LOGICAL\_BLOCK\_SIZE=value***

Specifies an alternate Logical Block Size to the standard 2048 bytes with which Hierarchies are created. The Logical Block Size must be the same for all the Hierarchies on a Volume.

#### ***/POSIX=IEEE\_P1282 (D) RRIP\_1\_09***

Specifies the POSIX Rock Ridge version to be used in this Hierarchy. If no value is specified for the /POSIX qualifier, IEEE\_P1282 is used. If the /POSIX qualifier is not specified, no Rock Ridge format will be used. Using a Rock Ridge format does consume extra space on the Volume.

#### ***/MODIFICATION=date-time***

Specifies the Volume Modification Date and Time for this Hierarchy on each Volume in the Volume Group being written.

#### ***/PUBLISHER=a- or a1-characters***

Specifies the Publisher Identifier for this Hierarchy on each Volume in the Volume Group being written.

If this field starts with the character underscore ( ), the rest of the field is interpreted as the File Identifier for a file in the Root Directory of each Volume being created. That remainder of the field after the underscore must follow the syntax rules for a File Identifier in this Hierarchy and must represent a file that exists in that Directory. The File Name must have a maximum length of 8 and the File Name Extension must have a maximum length of 3.

#### ***/SET\_IDENTIFIER=string***

Specifies the Volume Set Identifier for this Hierarchy on each Volume in the new Volume Set being created. This qualifier should not be specified when the command *LINK GROUP* has previously been given, since the Volume Set Identifier in that case must come from the prior Volume Group.

The default value for this qualifier is the current VMS date-time with punctuation removed and truncated to 13 characters, for example: "09SEP20021234".

For those using the VMS Hobbyist license, an identification number will be appended to the specified (or defaulted) Volume Set Identifier.

### ***/SETSIZE=maximum***

Specifies the maximum number of Volumes that can be included in the Volume Set, including both those being created now and any used to create previous Volume Groups within the Volume Set. If the files specified with the commands *LINK GROUP*, *ADD* and *WRITE* exceed the capacity of this number of Volumes, an error message will be displayed before any output data is written.

The ISO-9660 specification limits the maximum Volume Set Size to 65535.

### ***/SPACE=logical-block-count***

Specifies the number of logical blocks on the output Volume. When the output Volume is a volume image file, this is the size with which it will be created (taking into account the difference in size between Logical Blocks and VMS blocks).

The maximum number of Logical Blocks that can be specified is the number that would result  $2^{31}$  (about 2 billion) VMS blocks.

### ***/SYSTEM=character-list***

Specifies the System Identifier that is to be included in this Hierarchy, identifying the system which is able to act on the System Area (Logical Sectors 0-15) of the created Volume.

Generally this qualifier should be omitted.

### ***/VOLUME=volume-name-list***

Specifies the list of Identifiers to be used as the Volume Identifier in this Hierarchy on successive Volumes of the Volume Group being written in this session.

If a single name is provided for the list and it contains an asterisk (\*), the asterisk will be replaced by the Volume Sequence Number in each Volume Identifier as it is written.

If a single name is provided for the list and it contains no asterisk (\*), that name will be used intact for the Volume Identifier in this Hierarchy on each Volume of the current Volume Group. There is no requirement in ISO-9660 that the Volume Identifier of various Volumes differ from one another.

---

## EXAMPLE

```
LJKC> SET DEFAULT/VOLUME=(ONE,TWO)/POSIX=IEEE_P1282
LJKC> SET DEFAULT/PUBLISHER="LJK Software"
LJKC> SET DEFAULT/COPYRIGHT_FILE=COPYING.TXT
LJKC> SET DEFAULT/EFFECTIVE=14-DEC-2525
```

Those commands store their qualifier values into memory for use on the next CREATE HIERARCHY command.

```
LJKC> CREATE HIERARCHY MY_NAME/POSIX=RRIP_1_09
```

Create a Hierarchy which will use the values specified via the SET DEFAULT command except substituting the other Posix Rock Ridge format for the one specified via SET DEFAULT.



---

## subsystem

Enter the LJK/CDROM command subsystem mode.

---

### FORMAT **\$ LJK/CDROM**

---

Command Qualifiers	Defaults
<i>None.</i>	<i>None.</i>

---

**restrictions** *None.*

---

**PARAMETERS** *None.*

---

**DESCRIPTION** The following commands preset information for use by the WRITE command, so they are only effective if entered from the subsystem prompt (LJKC>) prior to the WRITE command to which they apply.

- ADD
- CREATE HIERARCHY
- DEFINE CHARACTER\_SET
- LINK GROUP
- SET DEFAULT

To terminate subsystem mode, enter the command EXIT.

---

**QUALIFIERS** *None.*

---

### EXAMPLE

```
$ LJK/CDROM  
LJKC>
```

Enter LJK/CDROM subsystem mode.



**SUMMARY (D)**

Specifies what information should be displayed during the command.

- **DEBUG**  
Display extremely detailed information suitable for submitting a bug report.
- **DETAILS**  
Display detailed information suitable for analyzing a problem.
- **MULTIPLE\_ERRORS**  
Display even those errors that were encountered on a previous file.
- **PROGRESS**  
Display indications of progress in processing the command.
- **SUMMARY**  
Display a summary of major results of the command.

***/TOLERATE******=ISO9660=A\_TRAILING\_NULL******=ALL=A1\_TRAILING\_NULL******=MICROSOFT=A1\_TRAILING\_NULL******=ISO9660=D\_TRAILING\_NULL******=ALL=D1\_TRAILING\_NULL******=MICROSOFT=D1\_TRAILING\_NULL***

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 7.4.5 that pad fields with nulls rather than the required spaces.

***/TOLERATE******=MICROSOFT=ESCAPE\_TRAILING\_TRASH***

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 8.5.6 that pad fields with spaces rather than the required nulls.

***/TOLERATE******=ISO9660=A\_CHARACTERS=character-list******=ALL=A1\_CHARACTERS=character-list******=MICROSOFT=A1\_CHARACTERS=character-list******=ISO9660=D\_CHARACTERS=character-list******=ALL=D1\_CHARACTERS=character-list******=MICROSOFT=D1\_CHARACTERS=character-list***

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 7.4 that include particular illegal characters.

***/TOLERATE******=ISO9660=D\_MISSING\_SEPARATOR\_1******=ISO9660=D\_MISSING\_SEPARATOR\_2******=ALL=D1\_MISSING\_SEPARATOR\_1***

## Command Reference

### VERIFY

**=ALL=D1\_MISSING\_SEPARATOR\_2**  
**=MICROSOFT=D1\_MISSING\_SEPARATOR\_1**  
**=MICROSOFT=D1\_MISSING\_SEPARATOR\_2**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 7.5.1 that fail to include SEPARATOR 1 (.) or SEPARATOR 2 (;) characters in File Identifiers.

**/TOLERATE**

**=ISO9660=D\_DIRECTORY\_SEPARATOR\_1**  
**=ISO9660=D\_DIRECTORY\_SEPARATOR\_2**  
**=ALL=D1\_DIRECTORY\_SEPARATOR\_1**  
**=ALL=D1\_DIRECTORY\_SEPARATOR\_2**  
**=MICROSOFT=D1\_DIRECTORY\_SEPARATOR\_2**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 7.5.1 that incorrectly include SEPARATOR 1 (.) or SEPARATOR 2 (;) characters in Directory Identifiers.

There is no **/TOLERATE=MICROSOFT=D1\_DIRECTORY\_SEPARATOR\_1** since Microsoft's Joliet specification allows Directory Identifiers to contain SEPARATOR 1 (.).

**/TOLERATE**

**=ISO9660=DIRECTORY\_DEPTH=n**  
**=ALL=DIRECTORY\_DEPTH=n**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 6.8.2.1 that incorrectly create directory depths greater than 8.

There is no **/TOLERATE=MICROSOFT=DIRECTORY\_DEPTH=n** since Microsoft's Joliet specification allows greater directory depths.

**/TOLERATE**

**=ISO9660=SELF\_RECORDING\_DATE\_MISMATCH**  
**=ISO9660=PARENT\_RECORDING\_DATE\_MISMATCH**  
**=ALL=SELF\_RECORDING\_DATE\_MISMATCH**  
**=ALL=PARENT\_RECORDING\_DATE\_MISMATCH**  
**=MICROSOFT=SELF\_RECORDING\_DATE\_MISMATCH**  
**=MICROSOFT=PARENT\_RECORDING\_DATE\_MISMATCH**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 6.8.2.2 that include a Recording Date field in one of the first two Directory Records that differs from the Recording Date of the main Directory Record describing the same Directory.

**/TOLERATE=ISO9660**

**=CREATION\_DATE\_MISMATCH**  
**=MODIFICATION\_DATE\_MISMATCH**  
**=EXPIRATION\_DATE\_MISMATCH**

**=EFFECTIVE\_DATE\_MISMATCH**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 9.5 and 4.1.6 of one of the Rock Ridge specifications such that a different date is included in the Extended Attribute Record (ISO) structure from that in the TF System Use Entry (Rock Ridge) structure.

**/TOLERATE=ALL****=CREATION\_DATE\_MISMATCH****=MODIFICATION\_DATE\_MISMATCH****=EXPIRATION\_DATE\_MISMATCH****=EFFECTIVE\_DATE\_MISMATCH**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 9.5 and 4.1.6 of one of the Rock Ridge specifications such that a different date is included in the Extended Attribute Record (ISO) structure from that in the TF System Use Entry (Rock Ridge) structure.

**/TOLERATE=MICROSOFT****=CREATION\_DATE\_MISMATCH****=MODIFICATION\_DATE\_MISMATCH****=EXPIRATION\_DATE\_MISMATCH****=EFFECTIVE\_DATE\_MISMATCH**

Suppress error messages due to mistaken implementations of ISO-9660 : 1988 (E) 9.5 and 4.1.6 of one of the Rock Ridge specifications such that a different date is included in the Extended Attribute Record (ISO) structure from that in the TF System Use Entry (Rock Ridge) structure.

---

**EXAMPLE**

```
$ LJK/CDROM VERIFY DKA100: /TOLERATE=(ISO=(DIRECTORY_DEPTH=11,D_MISSING_SEPARATOR_2),
  microsoft=(dl_trailing_null,escape_trailing_trash))
```

( )Escape Sequences

VERIFY a Volume but do not complain about:

- An excessive directory depth up to 11 in an ISO Hierarchy
- The lack of a File Name Version in an ISO Hierarchy
- A field padded with null rather than space Microsoft Joliet Hierarchy
- An Escapes field that contains non-nulls after the ISO-2022 escape strings in an Microsoft Joliet Hierarchy

).



***/EXPIRATION (D)******/NOEXPIRATION***

Specifies that the expiration date from the VMS file is to be recorded for the expiration date of the file on the Volume being written. Such a date will be recorded in the ISO-9660 Extended Attribute Record, and in Hierarchies for which Rock Ridge information is being recorded the date will also be reported in the TF System Use Entry.

This qualifier is provided because the meaning of an Expiration Date on a VMS file is locally defined and such a date may not be appropriate for distribution to other sites.

***/NAMES=(*[hierarchy-name=]*ISO-filespec[,...])***

Specifies the File Identifiers and Directory Identifiers under which files are to be stored on various Hierarchies of the Volume being created.

The Directory Identifiers and File Identifiers for the Primary Hierarchy must consist of D-Characters (no dollar signs allowed) while the Directory Identifiers and File Identifiers for supplementary Hierarchies must consist of D1-Characters (as specified by the Character Set for the Hierarchy).

If the “[hierarchy-name=]” string is omitted, LJK/CDROM attempts to store under the specified filespec in all Hierarchies, producing an error message if that effort fails in any Hierarchy.

If this qualifier is not provided, LJK/CDROM attempts to use the input Directory Identifiers and File Identifiers for all Hierarchies, producing an error if any of the characters are illegal for any of the Hierarchies.

Besides the VMS-style, specification of Directory Identifiers for a Hierarchy can also be done in two other manners, if the initial character is not “[” or “<”:

- Colon separations

Each Directory Identifier is followed by a colon, similar to Macintosh filespecs.

- Slash separations

Each Directory Identifier is followed by a left angle bracket, similar to Unix filespecs.

If no directory is specified, the directory from the input filespec is used for the output volume. To put a file into the root directory, as is required for files specified on the *CREATE HIERARCHY* or *SET DEFAULT* command qualifiers:

- */ABSTRACT\_FILE=file.ext;version*
- */APPLICATION=\_file.ext;version*
- */BIBLIOGRAPHIC\_FILE=file.ext;version*
- */COPYRIGHT\_FILE=file.ext;version*
- */DATA\_PREPARER=\_file.ext;version*
- */PUBLISHER=\_file.ext;version*

specify the directory as “[” in the */NAMES* qualifier.

## Command Reference

### WRITE

***/SHOW=DEBUG  
DETAILS  
MULTIPLE\_ERRORS (D)  
PROGRESS (D)  
SUMMARY (D)***

Specifies what information should be displayed during the command.

- **DEBUG**  
Display extremely detailed information suitable for submitting a bug report.
- **DETAILS**  
Display detailed information suitable for analyzing a problem.
- **MULTIPLE\_ERRORS**  
Display even those errors that were encountered on a previous file.
- **PROGRESS**  
Display indications of progress in processing the command.
- **SUMMARY**  
Display a summary of major results of the command.

***/VERIFY (D)  
/NOVERIFY***

Indicates a VERIFY operation is to be run against the completed Volume.

***/VMS\_ATTRIBUTES (D)  
/NOVMS\_ATTRIBUTES***

Specifies that VMS-specific RMS attributes are to be stored on the Volume being created. This is the default.

Omitting the VMS-specific RMS attributes will inhibit reading of relative and indexed files from the Volume, but will not interfere with sequential files, including those with variable-length records.

Omitting the VMS-specific RMS attributes will not save any space on the Volume. The major use for omitting the VMS-specific RMS attributes would be to test on VMS (by mounting the resulting Volume) that the standard ISO-9660 support for sequential file record formats are working correctly. That is the purpose for which LJK Software uses the /NOVMS\_ATTRIBUTES form of this qualifier.

---

## EXAMPLE

LJKC> WRITE DKA100:

Write a Volume containing the collected results of ADD commands to the disk DKA100: (which has been mounted /FOREIGN).

```
$ LJK/CDROM WRITE SYS$LOGIN:FOO.DAT A.B
```

Write a Volume containing all files named FOO to the disc image at  
SYS\$LOGIN:FOO.DAT.



---

# A LJK/CDROM Installation

This appendix shows how to install LJK/CDROM.

There are two available methods to install LJK/CDROM:

- Section A.1, VMSINSTAL Installation Commands
- Section A.2, PCSI Installation Commands

But regardless of which method you use, the very first installation of LJK/CDROM on a system disk should be followed by a special step if you want to use the HELP/MESSAGE data specific to LJK/CDROM.

- Section A.3, Enabling HELP/MESSAGE After Installation

---

## A.1 VMSINSTAL Installation Commands

The two steps to install LJK/CDROM using VMSINSTAL are as follows:

```
$ MOUNT ddcu: LJK /MEDIA_FORMAT=CDROM /UCS_SEQUENCE="(B"  
$ @SYS$UPDATE:VMSINSTAL LJK_CDROM DISK$LJK:[KITS] OPTIONS I
```

Note the use of the qualifier:

```
/UCS_SEQUENCE="(B"
```

on the *MOUNT* command in order to access an ISO-9660 hierarchy that contains dollar signs in certain Directory Identifiers and File Identifiers—[DECW\$BOOK] in particular.

---

## A.2 PCSI Installation Commands

The three steps to install LJK/CDROM using PCSI are as follows:

```
$ MOUNT ddcu: LJK /MEDIA_FORMAT=CDROM /UCS_SEQUENCE="(B"  
$ BACKUP DISK$LJK:[KITS]LJK_CDROM_PCSI_KITS.BCK/SAVE [ ]/NEW_VERSION  
$ PRODUCT INSTALL LJK_CDROM /SOURCE=SYS$DISK:[ ]
```

Note the use of the qualifier:

```
/UCS_SEQUENCE="(B"
```

on the *MOUNT* command in order to access an ISO-9660 hierarchy that contains dollar signs in certain Directory Identifiers and File Identifiers—[DECW\$BOOK] in particular.

### A.3 Enabling HELP/MESSAGE After Installation

After using the *PRODUCT INSTALL* command for the very first installation of LJK/CDROM, you will not have access to the LJK/CDROM *HELP/MESSAGE* data. To enable that access you must modify a startup command procedure (such as *SYSSMANAGER:SYSTARTUP\_VMS.COM*) to include a definition like:

```
$ DEFINE/SYSTEM MSGHLP$LIBRARY -  
  SYS$HELP:MSGHLP$LIBRARY.MSGHLP$DATA,-  
  SYS$HELP:LJK$LIBRARY.MSGHLP$DATA
```

VMS provides no automated mechanism for layered products to merge their *HELP/MESSAGE* data such that it will persist beyond the next VMS upgrade. If LJK/CDROM were to attempt to modify the definition of that logical name on the fly, it would certainly fail at a site where the number of *HELP/MESSAGE* files to be included exceeded some limit like the maximum number of equivalence names for a logical name.

### A.4 Sample VMSINSTAL Installation

Here is a sample of LJK/CDROM installation using *VMSINSTAL*:

```
$ MOUNT DKA400: LJK /MEDIA_FORMAT=CDROM /UCS_SEQUENCE="(B"  
%MOUNT-I-WRITELOCK, volume is write locked  
%MOUNT-I-CDROM_ISO, LJK:LJK_CDROM (1 of 1) , mounted on _NODE$DKA400:  
$ @SYS$UPDATE:VMSINSTAL LJK_CDROM DISK$LJK:[KITS] OPTIONS I
```

OpenVMS AXP Software Product Installation Procedure K7.2

It is 1-NOV-2002 at 15:40.

Enter a question mark (?) at any time for help.

The following products will be processed:

LJK\_CDROM V1.0

Beginning installation of LJK\_CDROM V1.0 at 15:40

%VMSINSTAL-I-RESTORE, Restoring product save set A ...

%VMSINSTAL-I-REMOVED, Product's release notes have been moved to SYS\$HELP.

The following files will be added or replaced:

```
SYS$COMMON:[SYSEXE]LJK$CDROM.EXE;  
SYS$COMMON:[SYSHLP]LJK$CDROM010.RELEASE_NOTES;  
SYS$COMMON:[SYSHLP]LJK$LIBRARY.MSGHLP$DATA;  
SYS$COMMON:[SYSLIB]LJK$CDROM_SHARE.EXE;  
SYS$COMMON:[SYSMMSG]LJK$MESSAGES.EXE;
```

The following files will be modified:

```
SYS$COMMON:[SYSHLP]HELPLIB.HLB;  
SYS$COMMON:[SYSLIB]DCLTABLES.EXE; (new version created)
```

All questions have been asked.

%VMSINSTAL-I-RESTORE, Restoring product save set B ...

The remainder of the installation will take 5 minutes  
on a stand-alone MicroVAX-II.

```

%LJK_CDROM-I-DCLTABLES, Adding command LJK/CDROM to DCL tables
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

      Installation of LJK_CDROM V1.0 completed at 15:40

      Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

      Creating installation data file: VMI$ROOT:[SYSUPD]LJK_CDROM010.VMI_DATA

      VMSINSTAL procedure done at 15:41

$

```

## A.5 Sample PCSI Installation

Here is a sample of LJK/CDROM installation using PCSI, with /LOG qualifiers added to the *BACKUP* command, just for illustrative purposes:

```

$ MOUNT DKA400: LJK /MEDIA_FORMAT=CDROM /UCS_SEQUENCE="(B"
%MOUNT-I-WRITELOCK, volume is write locked
%MOUNT-I-CDROM_ISO, LJK:LJK_CDROM (1 of 1) , mounted on _NODE$DKA400:
$ BACKUP DISK$LJK:[KITS]LJK_CDROM_PCSI_KITS.BCK/SAVE [ ]/NEW_VERSION/LOG
%BACKUP-S-CREATED, created DISK$LOGIN:[LJK]LJK-AXPVMS-LJK_CDROM-T0001--1.PCSI;1
%BACKUP-S-CREATED, created DISK$LOGIN:[LJK]LJK-VAXVMS-LJK_CDROM-T0001--1.PCSI;1
$ PRODUCT INSTALL LJK_CDROM /SOURCE=SYS$DISK:[ ]

  1 - LJK AXPVMS LJK_CDROM V1.0          Layered Product
  2 - LJK VAXVMS LJK_CDROM V1.0          Layered Product
  3 - All products listed above
  4 - Exit

Choose one or more items from the menu separated by commas: 2

The following product has been selected:
  LJK VAXVMS LJK_CDROM V1.0          Layered Product [Installed]

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

LJK VAXVMS LJK_CDROM V1.0: LJK/CDROM ISO-9660 Formatter

  Copyright 2002 LJK Software, All Rights Reserved.

  LJK Software, Cambridge, MA USA

  A license PAK might be needed after installation.

Do you want the defaults for all options? [YES]

Do you want to review the options? [NO]

Execution phase starting ...

The following products will be installed to destinations:
  LJK VAXVMS LJK V2.9          DISK$VAX96A073:[VMS$COMMON.]
  LJK VAXVMS LJK_CDROM V1.0    DISK$VAX96A073:[VMS$COMMON.]

Portion done: 0%...20%...30%...60%...70%...80%...90%...100%

The following products have been installed:
  LJK VAXVMS LJK V2.9          Layered Product
  LJK VAXVMS LJK_CDROM V1.0    Layered Product

$

```

The “product” named LJK which is automatically installed provides **HELP/MESSAGE**, **HELP**, **Message** and **DCL Command** information that is shared between products from LJK Software.



---

## **B Bug Reports**

This appendix tells how to report LJK/CDROM defects.

There is opportunity for lengthy debate over what is a “bug”, what is a “feature”, when a “bug report” is really an “enhancement request” and similar issues.

Rather than semantic nit-picking, however, the purpose of this appendix is to discuss communications between you, the user of LJK/CDROM, and LJK Software, the vendor and maintainer.

Even in cases where there is no problem with the software, user reports of difficulties give LJK Software information as to where documentation or training can be improved, so we appreciate your input.

At the same time, both parties want to make these interactions as productive as possible, and it is to that purpose that these rules are directed.

---

### **B.1 Who Can Submit Bug Reports ?**

Anyone can submit a bug report for LJK/CDROM, but not everyone will get a response. LJK Software will respond to bug reports (including returning the discs provided as part of the bug report) for:

- Standard licensees whose right-to-use-new-versions has not expired.
- Select others who provide particularly useful bug reports (in the sole judgement of LJK Software).

Those who use the no-charge license (for VERIFY) or the hobbyist license might choose to limit their bug reports to those they feel are crucial enough to be worth the effort of sending a bug report even without getting a response.

---

### **B.2 Format of a Bug Report**

Bug reports for a particular version of LJK/CDROM are only accepted until 30 days after the successor version is released. Bug reports should be addressed to:

LJK Software  
Attention: LJK/CDROM Bug Reports  
1 Broadway, Suite 600  
Cambridge, MA 02142-1100

Most bug reports should be accompanied by CD-R media:

## Bug Reports

- For problems with the commands *VERIFY*, *COMPARE* or *LINK GROUP*

An ISO-9660 CD-R which can be used to demonstrate the problem.

- For problems with the commands *ADD* or *WRITE* commands

A CD-R containing a VMS Backup save set of the files needed to reproduce the problem.

In all cases, there should be a piece of paper:

- 1 Describing the problem in English
- 2 showing a log demonstrating the problem, including the LJK/CDROM command *SHOW VERSION*.
- 3 for standard licensees giving the Product Authorization Key authorization number.

---

### B.3 Informal Questions

Standard licensees will be given an individual email address for asking simple questions for the duration of their right-to-use-new-versions.

# C

## Copyright

---

This appendix describes the copyright restrictions for LJK/CDROM, which differ from typical copyright restrictions.

This appendix does not deal with licensing the use of LJK/CDROM. For licensing information consult Appendix D, Licensing, within the *LJK/CDROM Reference Manual* for this version of the product.

The information on the CDROM containing LJK/CDROM is copyrighted by:

LJK Software  
1 Broadway, Suite 600  
Cambridge, MA 02142-1100

### Copying for Use

A licensee under any of the licenses in Appendix D, Licensing, within the *LJK/CDROM Reference Manual* for this version of the product, may copy the software to magnetic disk for installation and use in accordance with that license. Said licensee may also copy the installed software to backup media in accordance with normal backup procedures, only for use by the licensee (individual or organization) that installed it.

### Copying for Sharing

This software may be copied for licensed use by another licensee (under any of the licenses in Appendix D, Licensing, within the *LJK/CDROM Reference Manual* for this version of the product) **only to a CD-R disc** under the following restrictions:

- in such a fashion that a bit by bit comparison with the VMS command *DIFFERENCES* with foreign mounted discs shows the copy is identical. In the absence of a clean termination to the *DIFFERENCES* command, it is sufficient that an error like:

```
%DIFF-F-READERR, error reading DKA400:[].;  
-RMS-F-RER, file read error  
-SYSTEM-F-ILLBLKNUM, illegal logical block number
```

occur in both direction of comparison.

- in such a fashion that all wording from the face of the original disc is reproduced in a legible fashion on the face of the copy.

## Copyright

### **Transfer for Sharing**

An original CDROM disc of this software may also be transferred (temporarily or permanently) for licensed use by another licensee under any of the licenses in Appendix D, Licensing, within the *LJK/CDROM Reference Manual* for this version of the product.

### **Overall Restriction**

Transfer of the files comprising LJK/CDROM to another individual or organization via any medium other than CDROM or CD-R is forbidden.

# D Licensing

---

This appendix gives the license texts for LJK/CDROM.

There are three kinds of licenses for using LJK/CDROM, as described in the sections below:

- Section D.1, Standard License
- Section D.2, VMS Hobbyist License
- Section D.3, No-charge License

## General License Terms

For each of those kinds of license, the following apply:

- Software provided under these licenses may only be used in accordance with the terms specified here. Variances to these terms may only be made in writing, and no representative of LJK Software is empowered to do otherwise.
- No ownership or interest in the subject software is provided by these licenses. The license is valid only for use on machines owned by or leased to the licensee. The license may not be transferred to another entity.
- All aspects of license compliance are subject to technical enforcement procedures (e.g., Release Date restrictions), but absence of such procedures does not relieve the licensee to honor the limitations of the license.

No other use of LJK/CDROM is permitted.

## Limitation of Liability

In case of any dispute, the maximum liability of LJK Software shall be the price paid by the licensee for the license.

## Choice of Law

Governing law for each of these license types shall be the laws of the Commonwealth of Massachusetts.

---

## D.1 Standard License

The recipient of a Product Authorization Key from issuer LJK for an LJK/CDROM Standard License may use LJK/CDROM:

- 1 Only to use the standard features of LJK/CDROM described in this document,
- 2 Only on a single VMS system or VMScluster,
- 3 Only for up to the licensed number of simultaneous users specified in the Product Authorization Key,
- 4 Only for a version of LJK/CDROM with a Release Date prior to the Release Date specified in the Product Authorization Key.

Prior to the Release Date specified in the Product Authorization Key, the licensee has the right:

- to expect responses to bug reports submitted to LJK Software in accordance with the directions in Appendix B, Bug Reports,
- to contact LJK Software via Internet electronic mail at a unique address to receive answers to informal questions.

The Release Date specified in the Product Authorization Key for a Standard License corresponds to the end of the Right-to-Use New Versions period purchased with that Standard License (a minimum of one year).

---

## D.2 VMS Hobbyist License

The holder of VMS Hobbyist Product Authorization Key with an expiration date less than one year in the future for product OPENVMS-HOBBYIST from issuer DEC who does not have a Standard License for LJK/CDROM may use LJK/CDROM:

- 1 Only to use the standard features of LJK/CDROM described in this document,
- 2 Only for home hobbyist purposes having no business (commercial or non-commercial) purpose,
- 3 Only during the duration of their VMS Hobbyist Product Authorization Key,
- 4 For the version covered by this manual or any earlier version of LJK/CDROM.

The hobbyist user has no guarantee of receiving responses to any bug reports.

---

### D.3 No-charge License

A user who does not have a Standard License for LJK/CDROM and whose usage is not eligible for the VMS Hobbyist License above may use LJK/CDROM:

- 1 Only to use the *VERIFY* command of LJK/CDROM,
- 2 without any requirement for a Product Authorization Key,
- 3 for the version covered by this manual or any earlier version of LJK/CDROM.

The no-charge user has no guarantee of receiving responses to any bug reports.



---

# E VMS Issues

This appendix discusses VMS version-related ISO-9660 issues.

---

## E.1 Multiple Hierarchies

On VMS V7.3 and at least some earlier versions, ISO9660 MOUNT works only for the first Hierarchy. That can be a Primary Hierarchy or a Supplementary Hierarchy, but Hierarchies after the first cannot be mounted.

As a workaround, make sure the Hierarchy you expect to be used from VMS is the first Hierarchy for which you issue the *CREATE HIERARCHY* command.

---

## E.2 Volume Switching

While VMS V6.2 does a reasonable job of handling "volume switching" when there are fewer CDROM drives available than the total number of discs in the Volume Set, VMS V7.0-VMS V7.3 have problems. Unless you have a VMS ECO from HP to correct this problem, you should restrict your use of Volume Sets on those versions to situations where you are able to mount all members of the Volume Set at the same time. Otherwise, in the worst cases, VMS may crash.

This problem only affects discs mounted as ISO-9660 Volume Sets. Mounting /FOREIGN causes no problem.



# F

---

## Messages

This appendix gives an explanation of messages that might be issued during use of LJK/CDROM.

There are four types of LJK/CDROM messages:

- Section F.1, Primary Messages
- Section F.2, Secondary Messages
- Section F.3, Embedded Messages
- Section F.4, Internal Error Messages

---

### F.1 Primary Messages

These are the message codes for LJK/CDROM that might be returned to DCL (after first displaying them if other than Success severity).

These explanations are also available via the *HELP/MESSAGE* command once your system has been set up according to the instructions in Section A.3, Enabling HELP/MESSAGE After Installation.

Another set of messages that might be returned to DCL are those from the LICENSE facility. Those are documented and supplied for the *HELP/MESSAGE* command by VMS itself.

LJK-F-ALLOSPACE, 'logical-block-count' per-disk allocation space is less than calculated 'logical-block-count'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports.

LJK-W-BADCDFORMAT, format errors were encountered in the ISO-9660 data

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** As indicated in messages displayed before this one, the format of this Volume did not properly verify against the relevant standards. Those earlier messages will indicate one or more of the following causes for this problem:

- Faulty formatting of a Volume created by some other software

## Messages

- An internal error in LJK/CDROM
- A hardware error preventing the reading of data

Further operations based on the commands(s) given will be skipped.

**User Action:** Report the problem to the appropriate software or hardware maintainer.

LJK-F-BADCMDCHAR, bad 'name' character "character" in value "string"

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** An illegal character was provided in a command line.

**User Action:** Review the relevant portion of Chapter 9, Command Reference within the *LJK/CDROM Reference Manual*.

LJK-F-BADCMDTEXT, bad 'name' value was "string"

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** A bad text value was provided in a command line.

**User Action:** Review the relevant portion of Chapter 9, Command Reference within the *LJK/CDROM Reference Manual*.

LJK-F-BADCMDVALUE, bad 'name' value was 'nnn' (!'hex-longword' hex)

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** A bad numeric value was provided in a command line.

**User Action:** Review the relevant portion of Chapter 9, Command Reference within the *LJK/CDROM Reference Manual*.

LJK-W-BADISO9660, 'string'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Data on the Volume for which verification was requested violates the ISO-9660 standard.

**User Action:** If the Volume was formatted by the latest version of LJK/CDROM consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*. If the Volume was formatted by some other software, contact the maintainer of that software.

LJK-W-BADJOLIET, 'string'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Data on the Volume for which verification was requested violates the Microsoft Joliet standard it claims to honor.

**User Action:** If the Volume was formatted by the latest version of LJK/CDROM consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*. If the Volume was formatted by some other software, contact the maintainer of that software.

LJK-F-BADLOGIC, internal logic error detected

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-F-BADLOGICPC, internal logic error detected at PC 'hex-longword'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-W-BADPOSIX, 'string'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Data on the Volume for which verification was requested violates the Rock Ridge or System Use Sharing Protocol standard it claims to honor.

**User Action:** If the Volume was formatted by the latest version of LJK/CDROM consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*. If the Volume was formatted by some other software, contact the maintainer of that software.

## Messages

LJK-W-DIFFERENT, the two volumes were logically different

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** The two Volumes for which the LJK/CDROM *COMPARE* command was issued are not logically the same.

**User Action:** Compare the areas of difference with your expectations.

LJK-I-ENQLMLOW, ENQLM quota for process is lower than recommended value of 100

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** The specified quota is low.

**User Action:** Raise the quota using AUTHORIZE or via system parameter PQL\_MENQLM.

LJK-F-EXISTS, 'file-identifier' already exists

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:**

- 1 You are attempting to specify an ISO-9660 File that already exists for a different source file.
- 2 LJK/CDROM got an internal error (when accompanied by BADFIELD or BADLOGIC).

**User Action:**

- 1 Revise your commands to avoid a conflict.
- 2 Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-F-FOREIGN, device 'device-name' is mounted /FOREIGN

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** A device currently mounted /FOREIGN was specified in a context where that is not permitted, such as in a ddcu:[dire]file.ext file specification.

**User Action:** Determine whether it is the mount status or the device name that is wrong and retry with corrections.

LJK-I-INSFVOLIDS, insufficient /VOLUME= values for Volumes 'volume-number' through 'volume-number'!/\_and a single wildcarded value was not supplied

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** The Volumes being written as a Volume Set cannot all have unique Volume Identifiers for at least one Hierarchy. This is only an informational message because the result is a legal ISO-9660 Volume Set, but mounting individual Volumes on VMS may be confusing.

**User Action:** Decide whether unique Volume Identifiers are important enough to you that you want to make adjustments.

LJK-S-INTERCHANGE\_1, ISO9960 Volume metadata conforms to Interchange Level 1

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Success

**Explanation:** The Volume was verified as compatible with ISO-9660 Interchange Level 1. This means it should be readable on any system that is specified as being able to read ISO-9660 Volumes.

**User Action:** Distribute without concern.

LJK-I-INTERCHANGE\_2, ISO9960 Volume metadata conforms to Interchange Level 2

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** The Volume was verified as compatible with ISO-9660 Interchange Level 2. This means it should be readable on systems that are specified as being able to read ISO-9660 Volumes and can also handle longer File Identifiers and Directory Identifiers.

**User Action:** Distribute with a warning that longer names are included.

LJK-I-INTERCHANGE\_3, ISO9960 Volume metadata conforms to Interchange Level 3

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** The Volume was verified as compatible with ISO-9660 Interchange Level 3. This means it should be readable on systems that are specified as being able to handle the entire ISO-9660 standard.

**User Action:** Distribute with a warning that full ISO-9660 compliance is required.

## Messages

LJK-F-IOSBERROR, unanticipated status block error from the 'name' system service

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** An I/O error was returned by VMS.

**User Action:** Investigate the hardware problem.

LJK-F-ISO9660ALLOCATED, Volume 'volume-number' Block 'logical-block-number' was previously allocated

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-F-ISO9660NOFIT, ISO9660 per-Volume overhead leaves no space for files!/\_within /SPACE='logical-block-count' Logical Blocks of /BLOCK\_SIZE='nnn' bytes each

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The required metadata to support Hierarchies and Directories is so large that it leaves no room for data.

**User Action:** Consider using larger Volumes or smaller Volume Sets.

LJK-F-ISO9660NOVOL, Providing space of 'count' Logical Blocks!/\_within the per-Volume available space of 'count' Logical Blocks!/\_would require Volumes 'volnum' through 'volnum',!/\_exceeding the imposed limit of /SETSIZE='count' Volumes

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The data specified cannot fit within the given constraints.

**User Action:** Consider allowing more Volumes in the Volume Group.

LJK-F-LINKNOTFIRST, the LINK GROUP 'name' command followed a CREATE HIERARCHY command

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The command *LINK GROUP* was given too late.

**User Action:** Issue the *LINK GROUP* command before the first *CREATE HIERARCHY* command.

LJK-F-NOCOMMAND, No command verb was present for parsing

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The command was not present for parsing.

**User Action:** Look for local modifications that created a duplicate command tables file before LJK/CDROM was installed. Make sure the process logged in after LJK/CDROM was installed.

LJK-F-NOFILES, no files selected

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The LJK/CDROM *WRITE* command was issued without specifying any Files and without any Files specified by prior *ADD* commands. LJK/CDROM will not format a Volume without including at least one File. The product is not aimed at the coaster fabrication market.

**User Action:** Try again, specifying at least one file to write to the Volume.

LJK-F-NOMATCH, Existing and Modified records do not match

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-F-NOROOTFILE, file 'file-specification' was not included in the root directory

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** A file was not provided for the [] root Directory for some Hierarchy even though it had been specified by one of the qualifiers /PUBLISHER=, /DATA\_PREPARER=, /APPLICATION=, /COPYRIGHT\_FILE=, /ABSTRACT\_FILE= or /BIBLIOGRAPHIC\_FILE=.

**User Action:** Use an appropriate *ADD* command, or additional filespec to the *WRITE* command, to provide the required file. Otherwise, refrain from specifying that the particular root file will be provided.

## Messages

LJK-F-NOTDISK, device 'device-name' is not a disk

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** LJK/CDROM can only deal with optical discs or magnetic disks, except for the /OUTPUT= qualifier, but some other device was specified.

**User Action:** Correct the specification in the command.

LJK-F-NOTFOREIGN, device 'device-name' is not mounted /FOREIGN

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** For all contexts except a full file specification (ddcu:[dir]file.ext), LJK/CDROM requires that devices be mounted /FOREIGN.

**User Action:** Determine whether it is the mount status or the device name that is wrong and retry with corrections.

LJK-F-NOVOLID, no /VOLUME= value for Volume 'volume-number'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

It is possible that modifying your use of the /VOLUME= qualifier might work around this error in LJK/CDROM.

LJK-F-OPENERR, error opening 'file-specification'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** LJK/CDROM was unable to open the specified ISO-9660 CDROM or disc image.

**User Action:** Analyze the problem using normal VMS techniques considering any subsidiary chained messages.

LJK-F-OPENIN, error opening 'file-specification' as input

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** LJK/CDROM was unable to open the file specified by /APPUSE=.

**User Action:** Analyze the problem using normal VMS techniques considering any subsidiary chained messages.

LJK-F-OPENOUT, error opening 'file-specification' as output

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** LJK/CDROM was unable to store under the specified File Identifier (including a new highest File Name Version because an entry already exists for the highest possible File Name Version).

**User Action:** Analyze the problem using normal VMS techniques considering any subsidiary chained messages.

LJK-F-POSIXTOOBIG, Too much Posix information for Rock Ridge V1.09 is provided!/\_for file "file-specification"

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The information provided for POSIX is more than will fit in the available space.

**User Action:** Provide less Posix information, refrain from specifying /POSIX, or specify /POSIX=IEEE\_P1282 rather than /POSIX=RRIP\_1\_09.

LJK-F-READERR, error reading 'volume-name'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** One of the two volumes specified with the LJK/CDROM *COMPARE* command could not be validated. LJK/CDROM is unable to compare unless the inputs specified are both valid ISO-9660 Volumes.

**User Action:** Re-create the faulty Volume.

LJK-F-SMALLDISK, only 'vms-block-count' of the specified 'vms-block-count' blocks are available!/\_on target 'device-name'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The /SPACE= qualifier specified more space that was available on the target Volume.

**User Action:** Specify a smaller amount of space, avoid specifying /SPACE=, or use a larger target Volume.

LJK-F-SSERROR, unanticipated error from the 'name' system service

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The error specified on following lines was returned from a VMS system service.

**User Action:** Take action according to normal VMS troubleshooting methods.

## Messages

LJK-F-THISSPACE, 'count' block!%S for the Extended Attribute Record plus!/\_!'count' block!%S for File Section 'number' of!/\_!'nnn'!/\_!\_is more than the calculated!/\_!'count' maximum blocks available on any volume

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-W-UNKCDFORMAT, unknown version indicators were encountered in the ISO-9660 data

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** As indicated by messages displayed earlier, the Volume has System Use Areas or Continuation Areas that cannot be interpreted by LJK/CDROM.

**User Action:** For the *VERIFY* command, decide whether a Volume with this status is acceptable. If the Volume was formatted by the latest version of LJK/CDROM consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-F-VALUENOTMATCH, 'name' value 'nnn' (!-'hex-longword' hex) does not match 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** There was an internal error in LJK/CDROM.

Two hierarchies contain incompatible values, and LJK/CDROM should have detected and prevented this earlier.

**User Action:** Consider submitting a bug report to LJK Software as described in Appendix B, Bug Reports within the *LJK/CDROM Reference Manual*.

LJK-F-VALUENOTP2, 'name' value 'nnn' (!-'hex-longword' hex) is not a power of 2

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The numeric value specified in a command must be a power of 2 and it was not.

**User Action:** Choose a number which is a power of two, e.g. 2048, 4096, 8192, etc.

LJK-F-VALUETOOHIGH, 'name' value 'nnn' (!-'hex-longword' hex) is larger than the limit of 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The numeric value specified in a command was too large.

**User Action:** Try again with a smaller value.

LJK-F-VALUETOOLONG, 'name' value "string" is longer than the limit of 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The text value specified in a command was too long.

**User Action:** Try again with a shorter value.

LJK-F-VALUETOOLOW, 'name' value 'nnn' (!-'hex-longword' hex) is smaller than the limit of 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The numeric value specified in a command was too small.

**User Action:** Try again with a larger value.

LJK-F-VALUETOOSHORT, 'name' value "string" is shorter than the limit of 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** The text value specified in a command was too short.

**User Action:** Try again with a longer value.

---

## F.2 Secondary Messages

These are the message codes that LJK/CDROM might display subsidiary to a Primary Message to form a VMS Chained Message. They give additional information about the situation described by the Primary Message.

LJK-F-ABSTFILE, specified as Abstract File Identifier

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

## Messages

LJK-F-APPLFILE, specified as Application Identifier

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-W-BADFIELD, 'name' field invalid at 'hex-longword'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-F-BADLONG, invalid data ('hex-longword') at 'hex-longword'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-S-BADWORD, invalid data ('hex-word') at 'hex-longword'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Success

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-F-BIBLFILE, specified as Bibliographic File Identifier

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-W-CLOSEDEL, error closing 'file-specification'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-W-CLOSEIN, error closing 'file-specification' as input

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-F-CLOSEOUT, error closing 'file-specification' as output

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-F-COPYFILE, specified as Copyright File Identifier

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-I-ESCAPES, with Escape Sequences "string"

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-I-HIERARCHY, for hierarchy 'name'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

## Messages

LJK-I-ISODIRECTORY, 'string' Directory 'directoryspec'!!\_!\_at Logical Block 'lbn' on Volume 'volnum'!!\_!\_has 'n' Logical Block EAR and 'nnn' byte File Section,!!\_!\_recorded 'date' with Flags 'hex'!+!+

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-I-ISOFILE, 'string' File Section from File 'filespec'!!\_!\_at Logical Block 'lbn' on Volume 'volnum'!!\_!\_has 'n' Logical Block EAR and 'nnn' byte File Section,!!\_!\_recorded 'date' with Flags 'hex' Unit Size 'n' and Gap 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-I-NOMATCHB, field 'name' New 'hex-number' Existing 'hex-number'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-I-NOMATCHN, field 'name' New 'nnn' Existing 'nnn'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-I-NOMATCHS, field 'name' New "string" Existing "string"

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Informational

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-F-PREPPFILE, specified as Data Preparer Identifier

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-F-PUBLFILE, specified as Publisher Identifier

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Fatal

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

LJK-W-TEXT, 'string'

**Facility:** LJK, LJK/CDROM Utility

**Severity:** Warning

**Explanation:** Additional information applies according to the Primary Message.

**User Action:** Evaluate in the context of the entire chained message.

---

### F.3 Embedded Messages

These are text messages included within another message to indicate the particular nature of a failure.

---

### F.4 Internal Error Messages

These messages indicate a defect in the LJK/CDROM software. They are distinguished by having underscores rather than spaces between the words.

<0a"cosymbol'0;!symbol!0a"w0a+32idr'c>



---

# ISO-9660 Glossary

This glossary gives an alphabetical-order explanation of various terms (capitalized throughout this manual) that have specialized meanings under ISO-9660 and related standards.

**A-Characters:** The characters defined by the ISO-9660 standard for use in miscellaneous metadata within the primary Hierarchy.

**A1-Characters:** The characters for use in miscellaneous metadata within a supplementary Hierarchy defined by agreement between those who produce a Volume and those who read that Volume.

**Application Identifier:** The identity of a specification of how the data are recorded on the current Volume Group.

**C-Characters:** The ISO-2022 character set specified by Escape Sequences to form an outer limit to the A1-Characters and D1-Characters.

**Character Set:** An LJK/CDROM construct combining a Character Set, A1-Characters and D1-Characters for use with a particular Hierarchy.

**D-Characters:** The characters defined by the ISO-9660 standard for use in identifier metadata within the primary Hierarchy.

**D1-Characters:** The characters for use in identifier metadata within a supplementary Hierarchy defined by agreement between those who produce a Volume and those who read that Volume.

**Data Preparer Identifier:** The identity of the person or other entity which controls the preparation of the data to be recorded on the current Volume Group

**Directory Identifier:** The name of an ISO-9660 Directory.

**Directory:** An ISO-9660 disc area listing individual Files and subordinate Directories within an Hierarchy.

**Escape Sequences:** A collection of ISO-2022 escape sequences (less the escape character) used to specify the C-Characters for a Hierarchy.

**File Identifier:** The full name of an ISO-9660 File within a Directory.

**File Name Extension:** The middle portion (between SEPARATOR 1 (.) and SEPARATOR 2 (:)) of the name of an ISO-9660 File within a Directory.

**File Name Version:** The final portion (after SEPARATOR 2 (:)) of the name of an ISO-9660 File within a Directory.

## ISO-9660 Glossary

**File Name:** The first portion (before SEPARATOR 1 (.)) of the name of an ISO-9660 File within a Directory.

**File:** An ISO-9660 disc area containing data.

**Hierarchy:** A set of ISO-9660 Directories adhering to a single set of naming conventions.

**Logical Block:** An allocation unit of Volume space under ISO-9660, defaulting to 2048 bytes.

**Publisher Identifier:** The identity of the person who specified what should be recorded on the current Volume Group

**SEPARATOR 1 (.):** The character which separates the File Name from the File Name Extension within a File Identifier.

**SEPARATOR 2 (;):** The character which separates the File Name Extension from the File Name Version within a File Identifier.)

**System Identifier:** The identification of a system which can recognize and act upon the initial Sectors of the Volume.

**Volume Group:** Volumes within a Volume Set that are written at the same time as each other.

**Volume Identifier:** The name of a Volume according to a particular Hierarchy.

**Volume Sequence Number:** The number of a Volume within a Volume Set.

**Volume Set Identifier:** The name of a Volume Set used in common across Volumes within a particular Hierarchy.

**Volume Set:** A sequence of Volumes sharing Hierarchies and with each member containing metadata for all previous Volume Groups along with its own Volume Group within the Volume Set.

**Volume:** ISO-9660 data as arranged on a disc or stored in a disc image file suitable for transfer to a single disc.

---

# Index

...  
  see ellipsis  
1281  
  see IEEE P1281  
1282  
  see IEEE P1282  
2022  
  see ISO-2022  
9660  
  see ISO-9660

---

## A

---

A1-Characters • 2-5, 3-1, 3-2, 9-15, Glossary-1  
Access  
  read • 9-4, 9-8, 9-19, 9-26  
  write • 9-30  
A-Characters • 2-6, 9-15, Glossary-1  
Application Identifier • 2-5, Glossary-1

---

## B

---

Bug reporting • B-1 to B-2

---

## C

---

C-Characters • 2-5, 3-1, 9-15, Glossary-1  
CD-R • i, ix, xi, B-1, B-2, C-1, C-2  
CDROM • x, xi, 1-2, 2-1, 2-2, 2-8, 9-8, C-1, C-2, E-1  
Character Sets • 2-3, 2-5, 4-1, 4-2, 4-3, 5-1, 6-1, 9-5, 9-11, 9-14, 9-21, 9-31, Glossary-1  
  defining • 3-1 to 3-2  
  LJK\$ISO9660 • 3-1, 9-14  
  LJK\$MICROSOFT\_L1 • 3-1, 9-14  
  LJK\$MICROSOFT\_L2 • 3-1, 9-14  
  LJK\$MICROSOFT\_L3 • 3-1, 9-14  
  LJK\$VMS • 3-1, 9-14  
  LJK\$VMS\_EXTENDED • 3-1, 9-14  
  predefined • 3-1  
Character Sets agreements • 2-5

Command format • 9-1 to 9-2

  DCL • 9-1  
  help • 9-2  
  subsystem • 9-1

Command procedures • 9-24, 9-26

Command Reference • 9-1 to 9-33

Commands

  DCL  
    BACKUP • A-3  
    DIFFERENCES • C-1  
    HELP/MESSAGE • A-2  
    LJK/CDROM • 5-1, 9-1  
    MOUNT • 2-2, 4-2, A-1, E-1  
    PRODUCT INSTALL • A-2

  LJK/CDROM

    ADD • 4-2, 5-1, 5-2, 6-2, 9-1, 9-4 to 9-7, 9-12, 9-23, 9-25, B-2, F-7  
    COMPARE • 7-1, 8-1, 9-1, 9-8 to 9-9, B-2, F-4, F-9  
    CREATE HIERARCHY • 1-2, 4-1, 5-1, 6-1, 6-2, 9-1, 9-10 to 9-13, 9-25, E-1, F-6  
    creation commands • 9-1  
    DEFINE CHARACTER\_SET • 3-1, 3-2, 5-1, 6-1, 9-1, 9-14 to 9-15, 9-25  
    EXIT • 9-1, 9-16  
    HELP • 9-1, 9-17 to 9-18  
    LINK GROUP • 6-1, 6-2, 7-1, 9-1, 9-12, 9-19, 9-22, 9-23, 9-25, B-2, F-6  
    SET DEFAULT • 5-1, 6-1, 6-2, 9-1, 9-5, 9-20 to 9-23, 9-25, 9-31  
    SHOW VERSION • 9-1, 9-24, B-2  
    subsystem command • 9-25  
    validation commands • 9-1  
    VERIFY • x, 1-1, 1-2, 7-1 to 7-2, 9-1, 9-26 to 9-29, B-2, D-3, F-10  
    WRITE • 1-2, 4-2, 5-1, 6-2, 9-1, 9-2, 9-4, 9-12, 9-23, 9-30 to 9-33, B-2, F-7

Copyright notice • C-1 to C-2

---

## D

---

D1-Characters • 2-5, 2-6, 3-1, 3-2, 9-5, 9-15, 9-31, Glossary-1

Data Preparer Identifier • 2-5, Glossary-1

## Index

D-Characters • 2-6, 9-5, 9-15, 9-31, Glossary-1  
DCL  
    Command procedures • 9-24, 9-26  
Directories • 2-4, 2-6, 9-6, 9-10, 9-12, 9-21, 9-22,  
    F-6, Glossary-1  
Directory Identifiers • 2-2, 2-5, 2-7, 9-5, 9-14, 9-15,  
    9-28, 9-31, A-1, F-5, Glossary-1  
Documentation • B-1  
DVD • xi, 2-1, 2-8, 9-8  
DVD-R • i, ix, xi

---

## E

---

Ellipsis • 9-17  
Enhancement request • B-1  
ENQLM  
    see Quotas  
    process  
        ENQLM  
Error messages  
    see Messages  
Escape Sequences • 2-5, 2-7, 3-1, 3-2, 4-1, 4-2,  
    9-14, 9-15, 9-19, 9-27, 9-29, F-13, Glossary-1  
Extended Attribute Record • 9-4, 9-29, 9-30, 9-31

---

## F

---

Feature • B-1  
File Identifiers • 2-2, 2-3, 2-4, 2-5, 9-5, 9-10, 9-11,  
    9-12, 9-14, 9-15, 9-21, 9-22, 9-28, 9-31, A-1,  
    F-5, F-9, Glossary-1  
File Name Extensions • 2-4, 2-7, 9-7, 9-10, 9-11,  
    9-12, 9-21, 9-22, Glossary-1  
File Names • 2-4, 2-7, 9-7, 9-10, 9-11, 9-12, 9-21,  
    9-22, Glossary-2  
File Name Versions • 2-4, 2-5, 9-7, 9-29, F-9,  
    Glossary-1  
Files • 2-3, 2-6, 4-2, 8-1, F-4, Glossary-2

---

## G

---

G0 character set • 3-1, 9-2, 9-15  
G1 character set • 3-1, 9-2, 9-15  
Glossary • Glossary-1

---

## H

---

HELP  
    library • 9-2, 9-17  
    Use of asterisk • 9-17  
HELP/MESSAGE • A-1, A-2, A-3, F-1  
Hierarchies • 2-3, 4-3, 5-1, 6-1, 6-2, 9-6, 9-12,  
    9-13, 9-22, 9-23, E-1, F-6, Glossary-2  
    creating • 4-1 to 4-3  
    LJK\$MICROSOFT • 4-2  
    LJK\$ODS2 • 4-2  
    LJK\$ODS5 • 4-2  
    LJK\$OPEN • 4-2  
    name • 4-1  
    predefined • 4-2  
    primary • 2-6, 9-5, 9-14, 9-31, E-1, Glossary-1  
Hierarchy  
    name • 9-10  
Hobbyist license • D-2  
Hybrid volume format • 2-2

---

## I

---

IEEE P1281 • x, 1-1, 1-2, 2-7, 8-1, F-3  
IEEE P1282 • x, 1-1, 1-2, 2-7, 8-1, 9-4, 9-12, 9-22,  
    9-23, 9-29, 9-30, 9-31, F-3, F-9, F-10  
Image activation  
    VMS • 9-1  
Input-filespec • 9-4  
Installation • A-1 to A-3  
Interchange Level 1 • 2-7, F-5  
Interchange Level 2 • 2-7, F-5  
Interchange Level 3 • 2-7, F-5  
Introduction to ISO-9660 • 2-1 to 2-8  
ISO-2022 • x, 2-5, 3-1, 3-2, 9-2, 9-14, 9-15, 9-29,  
    Glossary-1  
ISO-9660  
    Introduction to • 2-1 to 2-8

---

## J

---

Joliet  
    see Microsoft Joliet format

---

**L**

---

## License

- no-charge • 1-1, B-1, D-3

- Licensing • D-1 to D-3

- LJK/CDROM • A-1, B-1, C-1, C-2, D-1

- LJK Software • A-3, B-1, D-1

- Logical Blocks • 4-1, Glossary-2

- Logical names

- MSGHLP\$LIBRARY • A-2

---

**M**

---

- Maintainer • B-1

- Master file directory (MFD) • 2-3

- Messages • F-1 to F-15

- MFD (Master file directory) • 2-3

- Microsoft Joliet format • x, 1-1, 1-2, 9-28, 9-29, F-3

- Mount

- /FOREIGN • 9-8, 9-19, 9-26, 9-30

---

**N**

---

- No-charge license • 1-1, B-1, D-3

---

**O**

---

- ODS-2 • xi

- ODS-5 • xi, 2-2, 3-1, 9-14

- ODS-n • xi, 2-1, 2-2, 2-3, 2-8

---

**P**

---

- P1281

- see IEEE P1281

- P1282

- see IEEE P1282

- Performance • 9-1

- Placement onto Volumes

- controlling • 6-1

- Posix • 2-7, 9-12, 9-22, 9-23, F-3, F-9

- Problem

- see bug

- Prompt • 9-1

- Publisher Identifier • 2-5, Glossary-2

---

**Q**

---

## Qualifiers

## BACKUP

- /LOG • A-3

- /NEW\_VERSION • A-1

- /SAVE • A-1

## HELP

- /MESSAGE • A-1, A-2, F-1

## LJK/CDROM

- /A1\_CHARACTERS • 9-14, 9-15

- /ABSTRACT\_FILE • 9-5, 9-10, 9-20, 9-21, 9-31, F-7

- /APPLICATION • 9-5, 9-10, 9-11, 9-20, 9-21, 9-31, F-7

- /APPUSE • 9-10, 9-11, 9-20, 9-21, F-8

- /BIBLIOGRAPHIC\_FILE • 9-5, 9-10, 9-11, 9-20, 9-21, 9-31, F-7

- /BLOCK\_SIZE • F-6

- /CHARACTER\_SET • 9-10, 9-11, 9-20, 9-21

- Command • 9-8, 9-10, 9-14, 9-20, 9-26, 9-30

- /COPYRIGHT\_FILE • 9-5, 9-10, 9-11, 9-20, 9-21, 9-31, F-7

- /COPY\_COMMANDS • 9-26

- /CREATION • 9-10, 9-11, 9-20, 9-21

- /D1\_CHARACTERS • 9-14, 9-15

- /DATA • 9-8

- /DATA\_PREPARER • 9-5, 9-10, 9-11, 9-20, 9-21 to 9-22, 9-31, F-7

- /EFFECTIVE • 9-4, 9-10, 9-12, 9-20, 9-22, 9-30

- /ESCAPE\_STRINGS • 9-14, 9-15

- /EXPIRATION • 9-4 to 9-5, 9-10, 9-12, 9-20, 9-22, 9-30, 9-31

- for CREATE HIERARCHY compared to SET DEFAULT • 6-2, 9-20

- /LOGICAL\_BLOCK\_SIZE • 9-10, 9-12, 9-20, 9-22

- /MODIFICATION • 9-10, 9-12, 9-20, 9-22

- /NAMES • 1-2, 9-4, 9-5, 9-30, 9-31

- /OUTPUT • 9-24, 9-26, F-8

- Positional • 9-4, 9-30

- /POSIX • 9-10, 9-12, 9-20, 9-22, F-9

## Index

### Qualifiers

#### LJK/CDROM (cont'd)

/PUBLISHER • 9-5, 9-10, 9-12, 9-20, 9-22, 9-31, F-7  
/SETSIZE • 9-10, 9-12 to 9-13, 9-20, 9-23, F-6  
/SET\_IDENTIFIER • 9-10, 9-12, 9-20, 9-22  
/SHOW • 1-1, 7-2, 9-4, 9-5 to 9-6, 9-8 to 9-9, 9-26, 9-27, 9-30, 9-32  
=DEBUG • 9-5, 9-8, 9-27, 9-32  
=DETAILS • 9-5, 9-8, 9-27, 9-32  
=MULTIPLE\_ERRORS • 7-2, 9-5, 9-8, 9-27, 9-32  
=PROGRESS • 9-5, 9-8, 9-27, 9-32  
=SUMMARY • 9-5, 9-8, 9-27, 9-32  
/SPACE • 9-10, 9-13, 9-20, 9-23, F-6, F-9  
/SYSTEM • 9-10, 9-13, 9-20, 9-23  
/TOLERATE • 1-1, 7-2, 9-26, 9-27 to 9-29  
/VERIFY • 7-1, 9-30, 9-32  
/VMS\_ATTRIBUTES • 9-4, 9-6, 9-30, 9-32  
/VOLUME • 6-1, 9-10, 9-13, 9-20, 9-23, F-5, F-8

#### MOUNT

/FOREIGN • x, 1-1, 1-2, 6-2, E-1, F-4, F-8  
/MEDIA\_FORMAT • 2-2, A-1  
/UCS\_SEQUENCE • 2-2, A-1

#### PRODUCT INSTALL

/SOURCE • A-1

specifying arbitrary characters • 9-2

#### Quotas

process

ENQLM • F-4

---

## R

### RMS

file attributes • i, 2-7, 8-1, 9-6, 9-26, 9-32  
indexed files • 2-1  
relative files • 2-1  
sequential files • 2-1

Rock Ridge Interchange Protocol • x, 1-1, 1-2, 2-7, 8-1, 9-4, 9-12, 9-22, 9-23, 9-29, 9-30, 9-31, F-3, F-9, F-10

---

## S

SEPARATOR 1 (.) • 2-4, 9-15, Glossary-2

SEPARATOR 2 (: ) • 2-4, 9-15, Glossary-2

Subtopic

HELP • 9-17

System Identifier • 2-5, Glossary-2

System Use Entry • 8-1, F-3, F-10

Rock Ridge

TF • 9-4, 9-29, 9-30, 9-31

System Use Sharing Protocol

CE • 8-1

ST • 8-1

System Use Sharing Protocol • x, 1-1, 1-2, 2-7, 8-1, F-3

---

## T

Topic

HELP • 9-17

Training • B-1

Trouble reporting

see Bug reporting

Troubleshooting

see Messages

---

## U

User reports • B-1

---

## V

Vendor • B-1

VMS

hobbyist license • D-2

ISO-9660 handling

volume switching • 2-1, 2-8, 4-2, E-1

issues • E-1

version • E-1

Volume Groups • 2-8, 4-1, 4-2, 6-1, 6-2, 8-1, 9-11, 9-12, 9-13, 9-19, 9-21, 9-22, 9-23, F-6, Glossary-2

Volume Identifiers • 2-5, 4-1, 4-2, 9-13, 9-19, 9-23, F-5, Glossary-2

Volumes • 2-5, 5-2, 6-1, 6-2, 9-12, 9-23, F-1, Glossary-2

comparing two • 8-1

formatting • 5-1 to 5-2

Volumes (cont'd)

- verifying • 7-1 to 7-2
- Volume Sequence Number • 4-1, 4-2, 9-13, 9-23,  
Glossary-2
- Volume Set Identifiers • 2-5, 4-1, 9-12, 9-22,  
Glossary-2
- Volume Sets • 2-1, 2-8, 4-2, 6-1, 9-12, 9-19, 9-22,  
9-23, F-5, F-6, Glossary-2

formatting • 6-1 to 6-2

---

## W

---

Wildcard characters

- asterisk • 9-17
- percent sign • 9-17

