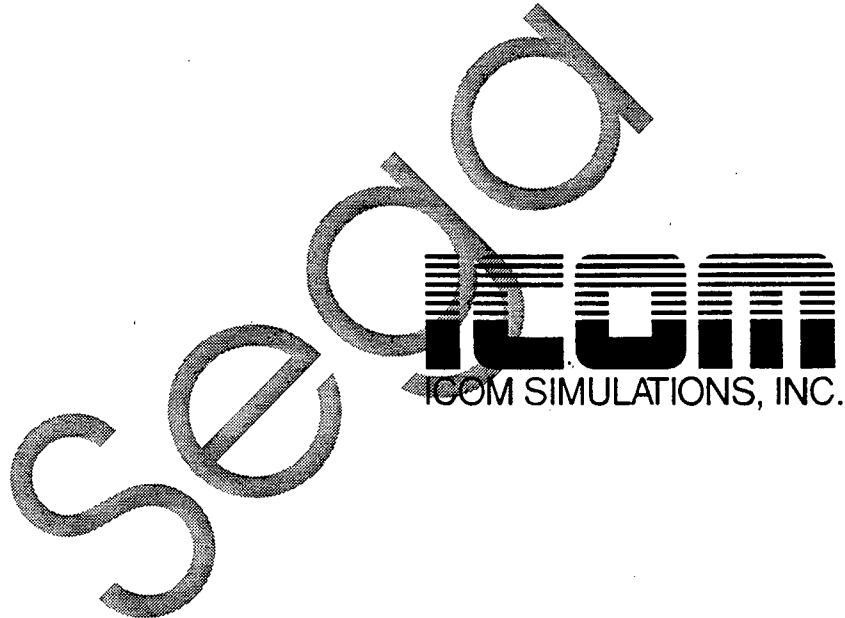


CTracTM

BuildTracks^{oft}

An ISO 9660 track image building tool.



Sega Ozisoft

Contents

Chapter 1 Introduction to BuildTrack

1.1 About BuildTrack.....	2
1.2 Running BuildTrack.....	3
1.2.1 Command Line Options.....	3

Chapter 2 Control Files

2.1 Command and File Format	6
2.2 An Example Control File.....	8
2.3 Control File Commands	10
2.3.1 Global Commands.....	13
2.3.2 Outermost Level Commands.....	19
2.3.3 Volume Commands	21
2.3.4 Primary Volume Commands.....	29
2.3.5 Supplementary Volume Commands.....	53
2.3.6 Volume Partition Commands.....	71
2.3.7 Boot Record Commands	79
2.3.8 Directory Hierarchy Commands	85
2.3.9 Directory Commands	93
2.3.10 File Commands.....	101

Appendices

Appendix A Parsing Algorithm.....	110
Appendix B Warning and Error Messages	112
Appendix C ISO Defined Characters.....	114

Index.....	115
-------------------	------------

Credits

Software: Todd Squires
Documentation: Donna Manning

This manual was produced using Microsoft Word from Microsoft Corp. and the index was produced using Sonar Bookends from Virginia Systems, Inc.

Copyright © 1991 ICOM Simulations, Inc.
All rights reserved.

Your rights of ownership are subject to the limitations and restrictions imposed by copyright laws as described below.

It is against the law to copy, reproduce or transmit including without limitation, electronic transmission over any network any part of the manual or program except as permitted by the Copyright Act of the United States, Title 17, United States Code. Under the law copying includes translation into another language or format. However, you are permitted by law to write the contents of the program into the machine memory of your computer so that the program may be executed provided, however, that in no event shall two users use the program at the same time, as in shared-disk systems. You are also permitted by law to make working copies of the program, solely for your own use, subject to the following restrictions; (1) Working copies must be treated in the same way as the original copy; (2) If you ever sell, lend or give away the original copy of the program, all working copies must also be sold, lent or given to the same person, or destroyed; (3) No copy (original or working) may be used while any other copy (original or working) is in use. If you make working copies of the program you should place on them the copyright notice that is on the original copy of the program.

The above is not an inclusive statement of the restriction imposed on you under the Copyright Act. For a complete statement of the restrictions imposed under the copyright laws of the United States of America see Title 17, United States Code.

With your convenience in mind, BuildTrack has been supplied without copy protection. Remember that our continued support of this software relies upon your respect for its creator's efforts.

The following are trademarks of ICOM Simulations, Inc.:
CTrac, ICOM Simulations and the ICOM Simulations, Inc. logo

ICOM Simulations, Inc.
648 South Wheeling Road
Wheeling, IL 60090

Chapter 1 Introduction to BuildTrack

- 1.1 About BuildTrack
- 1.2 Running BuildTrack
 - 1.2.1 Command Line Options

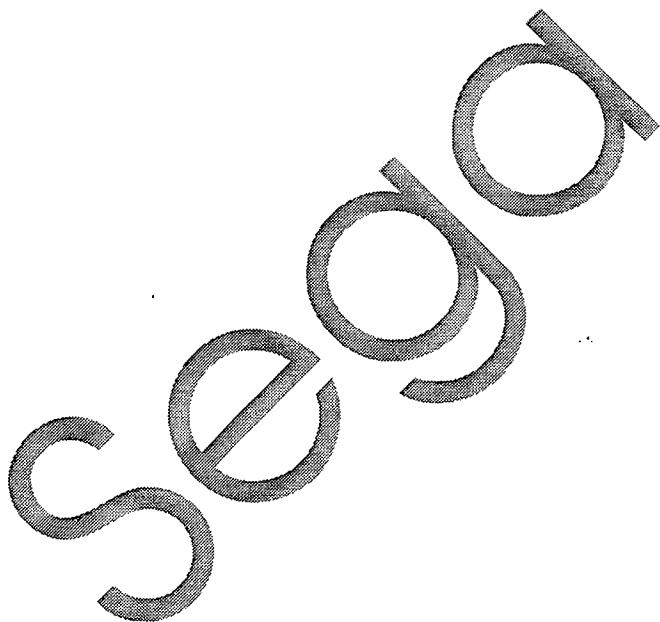
1.1 About BuildTrack

BuildTrack is a tool which is used to create an image of an ISO 9660 volume. ISO 9660 is an international standard (defined by the International Standards Organization (ISO)) that specifies the volume and file structure to be used with compact read only optical discs. ISO 9660 volumes usually reside on the first track of a compact disc. BuildTrack is used to create the volume image that is contained on this track.

ISO 9660 volumes contain a hierarchical directory system up to 8 levels deep. They also contain various structures, called "descriptors," that define the volume. ISO places restrictions on the length of file and directory names, as well as restrictions on the character set used for these names.

BuildTrack works by reading in a control file that specifies exactly what the ISO 9660 volume should look like. The control file is created by the user. It contains commands that describe the ISO volume image. When executed, BuildTrack parses through the control file and creates, in memory, all of the data structures needed to build the ISO volume image. If there are no errors, the volume image is written to an output file. The output file can then be used as input to the BuildDisc program. BuildDisc combines track images, along with the proper subcode information, and creates a compact disc image. For more information on BuildDisc, see the CTRAC BuildDisc documentation.

Setting up the control file for BuildTrack requires a working knowledge of the ISO 9660 volume format. BuildTrack is intended to be a low level programmer's tool that provides almost complete control of the layout of the ISO image. This amount of control can be a great benefit when trying to optimize an application for use on a CD-ROM disc. For more information on the ISO 9660 volume standard, refer to the *International Standard for CD-ROM Information Processing* booklet (reference number ISO 9660 : 1988 (E)).



1.2 Running BuildTrack

When BuildTrack is executed, it parses through the control file and creates, in memory, all of the data structures needed to build the ISO 9660 volume image. The image is not created, however, until the image definition in the control file has been completely read in and validated. Once validated, BuildTrack creates the volume image, writing its contents into an output file. To execute BuildTrack, use the following command format:

```
BuildTrack [options] controlFileName [options]
```

On systems that do not allow 9 character file names, use the command BT to execute BuildTrack. The invocation command, the control file name and any command line options must be separated by at least one space.

1.2.1 Command Line Options

All options are case sensitive. If an option is specified incorrectly, a listing of the options and a brief description of each will appear at the command prompt. At least one space must be used between a command line option and its parameters. If multiple options are specified, they must be separated by at least one space. Extra spaces are ignored.

-1

Using this option signifies that only ISO level 1 interchange compatible file names will be used. Warning messages will be generated if the level 1 interchange convention is violated. If the -w option is used also, the warning messages will not be reported.

-b <numSectors>

Use a buffer of the number of sectors specified when building the volume image. Each sector requires 2048 bytes of memory. If the -b command is not used, 64 sectors are used by default. Setting the number of sectors higher improves build time but uses more memory.

-d <variable> <value>

Defines the variable to the value specified. This option is useful for importing variable values into the control file. See also section 2.3.1 Global Commands for the Define command.

**-p
-w**

Prints diagnostic messages to the standard error device. This option is used mainly as a debugging tool and is not needed for normal operation.

Do not report warning messages. For a list of the warning messages that are generated by BuildTrack, see Appendix B Warning and Error Messages.

segagomisoft

Chapter 2 Control Files

- 2.1 Command and File Format
- 2.2 An Example Control File
- 2.3 Control File Commands
 - 2.3.1 Global Commands
 - 2.3.2 Outermost Level Commands
 - 2.3.3 Volume Commands
 - 2.3.4 Primary Volume Commands
 - 2.3.5 Supplementary Volume Commands
 - 2.3.6 Volume Partition Commands
 - 2.3.7 Boot Record Commands
 - 2.3.8 Directory Hierarchy Commands
 - 2.3.9 Directory Commands
 - 2.3.10 File Commands

2.1 Command and File Format

The control file provides the means by which the ISO 9660 volume image is described. The volume type is described first and then any descriptors that are needed. For each "primary" and "supplementary" volume descriptor, a directory hierarchy can be defined. Volume descriptors and the hierarchies that are contained within are written to the output volume image in the order that they are read from the control file definition. Also, files and directories are written to the output image in the order that they appear in the control file. For an explanation of the various commands that may appear in the control file, see section **2.3 Control File Commands**.

Command Format

All lines in the control file have the same structure. There must be at least one space between the control file command, the first parameter and any subsequent parameters. Extra spaces are ignored. Commands are not case sensitive.

```
command [parameter1] [parameter2] [parameters] ... ; comments
```

A comment begins immediately after a semi-colon (;). For a complete list and explanation of other BuildTrack special characters, see **Appendix A Parsing Algorithm**.

File Format

The control file for BuildTrack takes the following general format. The first string on a line is considered the command. Any leading white space is ignored.

```
Volume volType outputFileName  
PrimaryVolume  
; there can only be one primary volume  
; commands that further define the primary volume are placed here  
    Hierarchy  
; there can only be one hierarchy per volume descriptor  
; commands that further define the hierarchy are placed here  
        Directory theName  
; there can be an arbitrary number of directories  
; commands that further define the directory are placed here  
            other_directory_commands can be nested inside of directories up to the ISO defined -  
            ; limit of 8 deep - a warning is generated if this is exceeded  
                File theName {theVersion}  
; commands that further define the file are placed here  
            EndFile  
; an arbitrary number of files can be defined here  
; file commands can also be placed outside of directories (but inside the hierarchy -  
; command)  
            EndDirectory  
            EndHierarchy  
EndPrimaryVolume
```

Supplementary Volume
; there can be an arbitrary number of supplementary volumes
; commands that further define the supplementary volume are defined here
; the usage and layout of Hierarchy, Directory and File commands inside of the -
; supplementary volume follow the same format as in the primary volume

EndSupplementaryVolume
EndVolume

sega onisoft

2.2 An Example Control File

The following is a sample control file for building an ISO 9660 track.

```

; build CD-ROM based TIC-TAC-TOE game

Define WDO      "Sunday"      ; define some variables
Define WD1      "Monday"
Define WD2      "Tuesday"
Define WD3      "Wednesday"
Define WD4      "Thursday"
Define WD5      "Friday"
Define WD6      "Saturday"

Echo   "{program} version {version}.[edition]"
Echo   "It is [hour]:[minute]:[second] on ([WD{weekday}]) [month]/[day]/[year]"

; volume information
Volume ISO9660 "trk1"          ; output image to be placed in file trk1

SystemArea    "systemCode"       ; want this file in system area

PrimaryVolume                         ; there must be 1 primary vol. defined

; required information
SystemIdentifier
VolumeIdentifier
AbstractFileIdentifier
LogicalBlockSize 2048

; optional information
VolumeSetIdentifier
PublisherIdentifier
DataPreparerIdentifier
ApplicationIdentifier
CopyrightFileIdentifier
BibliographicFileIdentifier

; these need not normally be defined
VolumeCreationDate 03/16/1990 15:56:00:00 00
VolumeModificationDate 03/16/1990 15:56:00:00 00
VolumeExpirationDate 03/16/1990 15:56:00:00 00
VolumeEffectiveDate 00/00/0000 00:00:00:00 00

; required commands
LPath
MPath

```

2.2 An Example Control File

```
; the directory structure starts here
Hierarchy
File      TTT.EXE          ; there must be 1 hierarchy in prim. vol.
Source    TTT.EXE          ; this file exists on the root
EndFile

File      TTTCR.DOC
Source   TTTCR.DOC
EndFile

File      TTTAB.DOC
Source   TTTAB.DOC
EndFile

File      TTTB.DOC
Source   TTTB.DOC
EndFile
;-----

Directory GRAPHICS
File      THEBOARD.PIC
Source  'C:\GRAPHICS\THEBOARD.PIC'
Endfile

File      X.PIC
Source  'C:\GRAPHICS\X.PIC'
Endfile

File      O.PIC
Source  'C:\GRAPHICS\O.PIC'
Endfile
EndDirectory

EndHierarchy

EndPrimaryVolume

EndVolume
```

2.3 Control File Commands

There are ten types of commands that are valid in a BuildTrack control file. They are described briefly below. If a command is used incorrectly in the control file, a warning or an error message will be printed to the standard error device. If this occurs, BuildTrack will continue parsing through the control file until it reaches the end. It will continue printing all error or warning messages as necessary. If any error messages are generated, the volume image is not created. For a list of the warning and error messages that are generated by BuildTrack, see **Appendix B Warning and Error Messages**.

Global Commands

Valid at any point in the control file.

Outermost Level Commands

Valid before the beginning of a volume definition only.

Volume Commands

Valid within a volume definition only.

Primary Volume Commands

Commands that can be used while describing the primary volume portion of the ISO 9660 volume.

Supplementary Volume Commands

Commands that can be used while describing a supplementary volume portion of the ISO 9660 volume.

Volume Partition Commands

Valid within a volume partition definition only.

Boot Record Commands

Valid within a boot record definition only.

Directory Hierarchy Commands

Valid within a directory hierarchy definition only.

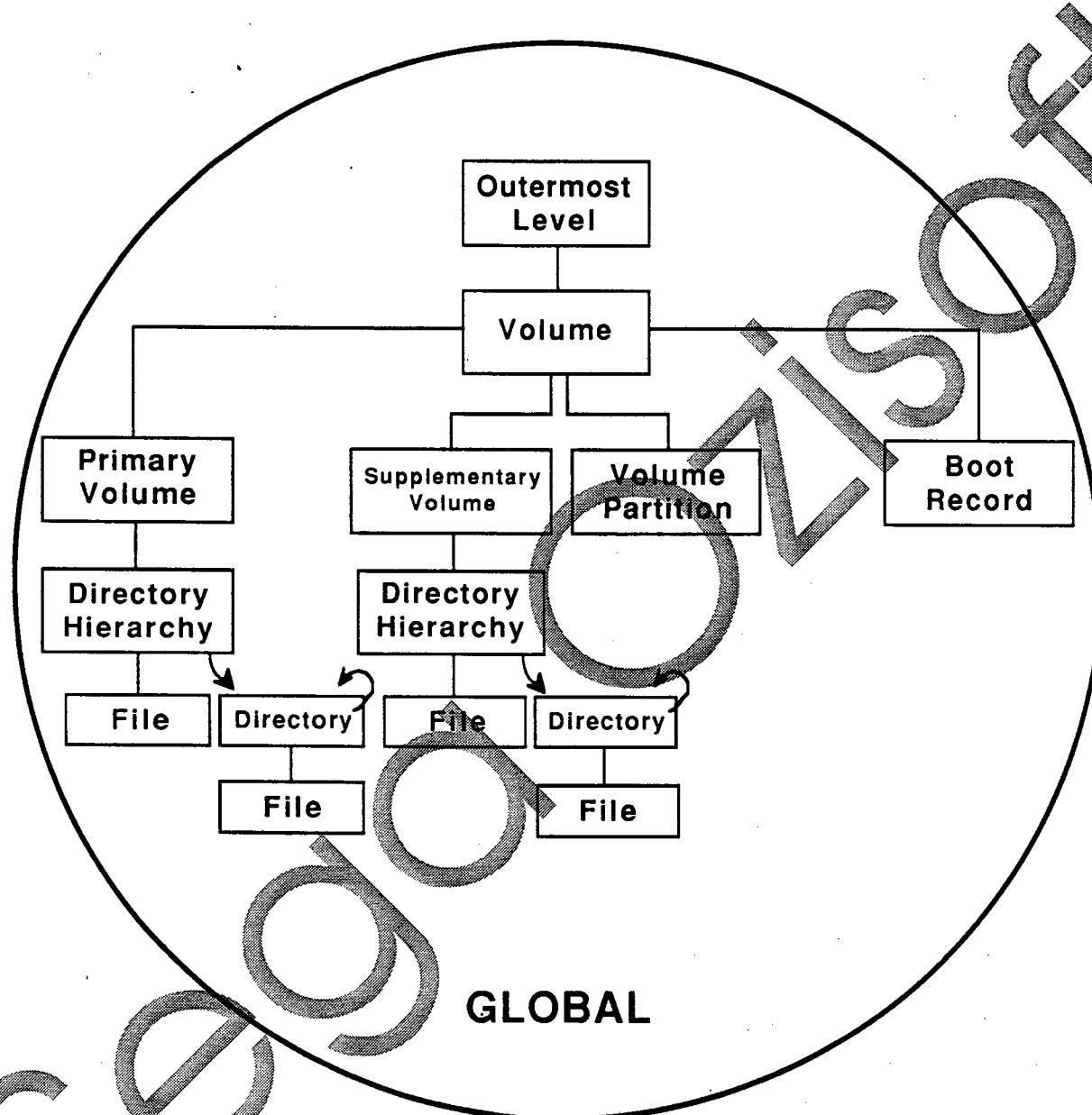
Directory Commands

Valid within a directory definition only.

File Commands

Valid within a file definition only.

The following diagram shows the hierarchical relationship of the BuildTrack control file commands.



Sega Ozisoft

2.3.1 Global Commands

SEGA OnSoft

Define

Syntax Define <variable> <value> [<variable> <value>] [...]

Example Define path C:IMAGES

Description Defines one or more variables to values. The variable and value can be any arbitrary ASCII strings. The variables and values are case sensitive.

- Define can be useful for aliasing values that would be repeated often in a control file or for importing values into the build. For more information, see section 1.2.1 Command Line Options for the -d option.
- To remove (undefine) a variable that was previously defined, define it to the empty string (""). For example, the command Define path "" undefines the variable in the example above.

Default When BuildTrack starts, the following variables are predefined:

program	- name of the executing program
version	- current version number
edition	- current edition
second	- the second of the minute
minute	- the minute of the hour
hour	- the hour of the day
day	- the day of the month
month	- the month of the year
year	- the year
weekday	- the day of the week
yearday	- the day of the year

See also

Appendix A Parsing Algorithm for an explanation of the notation used for accessing variables and for declaring strings.

Echo

Syntax Echo [<argument1>] [<argument2>] [...]

Example Echo this is a test

Description Reports all arguments to the console.

- Echo is normally used to display variables or messages while the control file is being processed.
- Using the ECHO command without an argument results in a carriage return being echoed.

Δ Note: Due to buffering of data sent to the console, it is possible that the output of ECHO may not appear until the BuildTrack program has stopped running.

second

GreenwichOffset

Syntax GreenwichOffset <theOffsetValue>

Example GreenwichOffset -23

Description This command is used to set the default Greenwich offset that will be used whenever a default date is placed into the output track file. The Greenwich offset is used in the ISO standard to define the time zones in which dates and times were recorded. Greenwich offsets are specified as the number of 15 minute intervals from -48 (West) to +52 (East) from Greenwich Mean Time.

- The Greenwich offset can be redefined as desired within the control file. Each GREENWICHOFFSET command takes effect until the next one is encountered.
- Greenwich offsets for some major cities are listed below.

Los Angeles	-32
Chicago	-23
New York	-20
Toronto	-21
London	0
Moscow	10
Tokyo	37
Sydney	40

Default

If a Greenwich offset is not specified within the control file, a default value of 0 is used.

Include

Syntax `Include <controlFileName1> [<controlFileName2>] [...]`

Example `Include MYFILE.CTL`

Description Includes the contents of each of the listed control files into the build at the point where the INCLUDE statement occurred. Each of the listed files is included in the order in which it appears in the list.

- Include files can be nested.
- To prevent recursive nesting of include files that could cause BuildTrack to hang up while parsing the control file, there is an imposed limit of 25 nested levels deep. If the control file attempts to nest include files to a level deeper than this, an error will be generated.

segad

onsg

ShowDefines

Syntax ShowDefines

Example ShowDefines

Description Reports all defined variables and their values to the console.

- This command can be useful when trying to determine what a variable is getting set to in a complex expression or to show the current state of variables from anywhere within the console.

- Δ Note: Due to buffering of data sent to the console, it is possible that the output of SHOWDEFINES may not appear until the BuildTrack program has stopped running.

See also *Appendix A Parsing Algorithm* for an explanation of the notation used for accessing variables and for declaring strings.

cooga

2.3.2 Outermost Level Commands

sega onsoft

Volume

Syntax Volume <volumeType> <outputFileName>

Example Volume ISO9660 IMAGE.TRK

Description Specifies that a volume of <volumeType> is being defined and that the output track image should be written to the output file with the name of <outputFileName>. Currently, the only volume type that can be specified is ISO9660. This command is used to begin a volume definition. Volume definitions are ended using the ENDVOLUME command.

- In the future, it is conceivable that BuildTrack might be able to build volumes that correspond to standards other than ISO9660. If this occurs, the <volumeType> field would be used to indicate which standard is in use.

See also Section 2.3.3 *Volume Commands* for the ENDVOLUME command.

segad

2.3.3 Volume Commands

sega onsoft

BootRecord

Syntax BootRecord

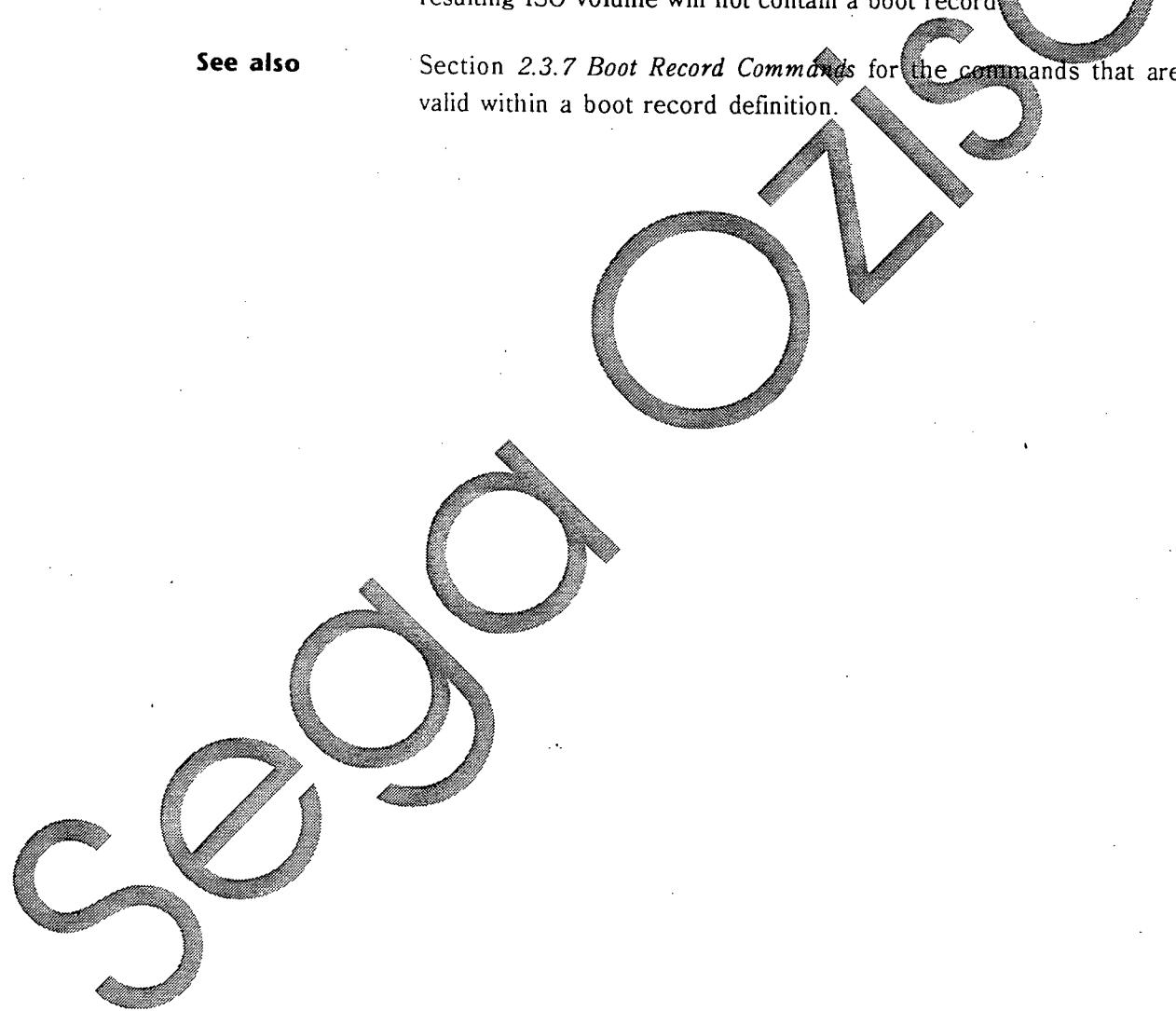
Example BootRecord

Description The definition of a boot record begins with the issuance of this command. Boot records can be used within ISO volumes to contain information that is needed at boot time.

- A boot record definition is ended when the ENDBOOTRECORD command is encountered.

Default If a BOOTRECORD command does not appear in the control file, the resulting ISO volume will not contain a boot record.

See also Section 2.3.7 *Boot Record Commands* for the commands that are valid within a boot record definition.



EndVolume

Syntax EndVolume

Example EndVolume

Description This is used to end the definition of a volume. This command usually appears at the end of a control file. When BuildTrack encounters this command, it checks to see if any errors have occurred so far, and if not, will proceed to build the specified volume. If errors have occurred, BuildTrack will not build the volume.

- There must be one matching ENDVOLUME command for every VOLUME command in the control file.

See also Section 2.3.2 *Outermost Level Commands* for the VOLUME command.

PrimaryVolume

Syntax PrimaryVolume

Example PrimaryVolume

Description This is used to indicate that the description of the primary volume portion of the ISO volume is beginning. Once this command is issued, the commands after it are used to define the primary volume portion of the ISO volume. The primary volume is the location where the main directory hierarchy exists.

- The description of the primary volume is ended by using the ENDPRIMARYVOLUME command.
- An ISO volume must contain exactly one primary volume.

See also Section 2.3.4 *Primary Volume Commands* for the commands that are valid within a primary volume definition.

secondary

SupplementaryVolume

Syntax SupplementaryVolume

Example SupplementaryVolume

Description This is used to indicate that the description of a supplementary volume within the ISO volume is beginning. Once this command is issued, the commands after it are used to define the supplementary volume portion of the ISO volume. Each supplementary volume contains its own distinct directory hierarchy.

- The description of the supplementary volume is ended by using the ENDSUPPLEMENTARYVOLUME command.
- Any number of supplementary volumes may be contained within an ISO volume.

Default If a SUPPLEMENTARYVOLUME command does not appear in the control file, the resulting ISO volume will not contain a supplementary volume.

See also Section 2.3.5 *Supplementary Volume Commands* for the commands that are valid within a supplementary volume definition.

SystemArea

Syntax SystemArea <theFileName>

Example SystemArea SYSTEM.DAT

Description Specifies an input file that contains the data to be placed into the system area of the current ISO 9660 volume. The system area of an ISO 9660 volume is made up of the first 16 logical sectors. There are 2048 bytes per sector for a total of 32768 bytes. The ISO standard does not specify what information (if any) should be placed there. System designers may chose to place boot code into these sectors.

- If the specified input file does not contain enough data to fill the first 16 logical sectors, the remaining bytes of those sectors will be padded with 0's. If the input file contains too much data, an error message will be generated and the track image will not be built.

Default If the SYSTEMAREA command is not used, the system area of the ISO volume will contain 32768 bytes of 0's.

sector

VolumePartition

Syntax VolumePartition

Example VolumePartition

Description This command marks the beginning of a volume partition descriptor record. The commands following are used to describe the volume partition.

- A volume partition descriptor definition is ended by using the ENDVOLUMEPARTITION command.
- Any number of volume partitions can exist in an ISO volume.
- ISO volumes normally do not contain volume partitions.

Default If a VOLUMEPARTITION command does not appear in the control file, the resulting ISO volume will not contain a volume partition.

See also Section 2.3.6 *Volume Partition Commands* for the commands that are valid within a volume partition definition.

segd

Sega Ozisoft

2.3.4 Primary Volume Commands

sega onisoft

AbstractFileIdentifier

Syntax AbstractFileIdentifier <theString>

Example AbstractFileIdentifier ABSTRACT.TXT

Description This command is used to fill in the abstract file identifier field of the primary volume descriptor. This field is used to contain the name of a file on the root directory of the ISO volume that contains an abstract statement for the volume.

- The file name must conform to ISO level 1 standards
- The file must appear in the root of the directory hierarchy. If the file does not appear in the root of the hierarchy, BuildTrack will generate a warning message.

Default If this command does not appear in the control file, the abstract file identifier string field of the primary volume descriptor will contain spaces (ASCII \$20). In this case, since an abstract file is not specified, none need exist on the root of the volume.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

ApplicationIdentifier

Syntax ApplicationIdentifier <theString>

Example ApplicationIdentifier TTT

Example ApplicationIdentifier _APID.DAT

Description Specifies the application identifier string of the primary volume descriptor record. According to ISO, "This field shall specify an identification of the specification of how the data are recorded on the Volume Group of which the volume is a member." If this string begins with an underscore (_) (ASCII \$5F), the remainder of the string is taken to be the name of a file on the root of the primary volume that contains application identification information. If a file name is specified, BuildTrack will check to make sure that the given file exists on the root and will generate a warning if it does not.

- The file name must conform to ISO level 1 standards.
- The application identifier is a string of up to 128 a-characters in length. For a list of the valid a-characters, see *Appendix C ISO Defined Characters*.

Default If this command does not appear in the control file, the application identifier string field of the primary volume descriptor will contain 128 spaces (ASCII \$20).

See also Section 2.3.3, *Volume Commands* for the PRIMARYVOLUME command.

ApplicationUse

Syntax ApplicationUse <theFile>

Example ApplicationUse APPUSE.DAT

Description Specifies an input file that contains the data that should be placed into the application use field of the primary volume descriptor. The application use field is 512 bytes long and can be filled with any values desired. The ISO standard does not specify what information (if any) should be placed there. Application designers may choose to place information into this field.

- If the specified input file does not contain enough data to fill the 512 bytes, the remaining bytes will be padded with 0's. If the input file contains too much data, an error message will be generated.

Default If the APPLICATIONUSE command is not issued, the application use field of the primary volume descriptor will contain 512 bytes of 0's.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

BibliographicFileIdentifier

Syntax	BibliographicFileIdentifier <theString>
Example	BibliographicFileIdentifier BIBLIO.TXT
Description	<p>This command is used to fill in the bibliographic file identifier field of the primary volume descriptor. This field is used to contain the name of a file on the root directory of the ISO volume that contains bibliographic information for the volume.</p> <ul style="list-style-type: none"> • The file name must conform to ISO level 1 standards. • The file must appear in the root of the directory hierarchy. If the file does not appear in the root of the hierarchy, BuildTrack will generate a warning message.
Default	If this command does not appear in the control file, the bibliographic file identifier string field of the primary volume descriptor will contain spaces (ASCII \$20). In this case, since a bibliographic file is not specified, none need exist on the root of the volume.
See also	Section 2.3.3 <i>Volume Commands</i> for the PRIMARYVOLUME command.

CopyrightFileIdentifier

Syntax CopyrightFileIdentifier <theString>

Example CopyrightFileIdentifier COPY.RGT

Description This command is used to fill in the copyright file identifier field of the primary volume descriptor. This field is used to contain the name of a file on the root directory of the ISO volume that contains a copyright message.

- The file name must conform to ISO level 1 standards.
- The file must appear in the root of the directory hierarchy. If the file does not appear in the root of the hierarchy, BuildTrack will generate a warning message.

Default If this command does not appear in the control file, the copyright file identifier string field of the primary volume descriptor will contain spaces (ASCII \$20). In this case, since a copyright file is not specified, none need exist on the root of the volume.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

DataPreparerIdentifier

Syntax	DataPreparerIdentifier <theString>
Example	DataPreparerIdentifier "TODD SMITH"
Example	DataPreparerIdentifier _DPID.DAT
Description	<p>Specifies the data preparer identifier field of the primary volume descriptor record. It is used to specify the person (or other entity) that prepared the data that is recorded on this volume. If this string begins with an underscore (_) (ASCII \$5F), the remainder of the string is taken to be the name of a file on the root of the primary volume that contains data preparer identification information. If a file name is specified, BuildTrack will check to make sure that the given file exists on the root and will generate a warning if it does not.</p> <ul style="list-style-type: none"> • The file name must conform to ISO level 1 standards. • The data preparer identifier is a string of up to 128 a-characters in length. For a list of the valid a-characters, see <i>Appendix C ISO Defined Characters</i>.
Default	If this command does not appear in the control file, the data preparer identifier string field of the primary volume descriptor will contain 128 spaces (ASCII \$20).
See also	Section 2.3.3 <i>Volume Commands</i> for the PRIMARYVOLUME command.

DescriptorWrites

Syntax DescriptorWrites <theAmount>

Example DescriptorWrites 3

Description The ISO standard allows the primary volume descriptor record to be written out to the ISO volume image more than one time for redundancy. This command allows the user to specify how many times the primary volume descriptor record should be written. Each descriptor record that is written takes one sector (2048 bytes) of disc space. There should be at most one occurrence of this command within the definition of the primary volume descriptor.

- There is no limit to the number of times that the primary volume descriptor record can be written. However, if the parameter given to this command is excessively large, a warning message will be generated.

Default If this command is not used within the definition of the primary volume descriptor, the primary volume descriptor will be written 1 time.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

EndPrimaryVolume

Syntax EndPrimaryVolume

Example EndPrimaryVolume

Description This command marks the end of the primary volume definition.
The primary volume definition must end with this command.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

sega

nzsoft

Hierarchy

Syntax Hierarchy

Example Hierarchy

Description When this command appears in the control file, it indicates that the root of the directory hierarchy for the primary volume is beginning.

- There must be exactly one HIERARCHY command within the definition of the primary volume.
- The hierarchy definition is ended by using the ENDHIERARCHY command.
- This command must be defined before the FILE or DIRECTORY commands can be used.

See also The DIRECTORY, ENDHIERARCHY, FILE and PRIMARYVOLUME commands.

LogicalBlockSize

Syntax LogicalBlockSize <theBlockSize>

Example LogicalBlockSize 2048

Description Logical sectors of an ISO volume are 2048 bytes long. ISO allows these sectors to be broken down into logical blocks. All files within ISO volumes are accessed using their logical block offset into the volume. Each file is made to begin on a logical block boundary. If the logical block size is large, information can be retrieved quickly, at the expense of wasted space between files. If the block size is set low, less space is wasted between files but information transfer can be slowed down.

- The only logical block sizes allowed by ISO are 512, 1024 and 2048.
- The logical block size must be specified exactly once for each primary volume descriptor that is defined.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

LPath

Syntax	LPath
Example	LPath
Description	To speed up access to files contained in ISO volumes, path tables are used. The path table contains a copy of the complete directory tree structure of the directory hierarchy. When a long path name is specified, it is possible to locate the directory at the end of the path more quickly by using the path table rather than by searching the CD for each directory in the path. There are two types of path tables. They are known as L and M types. This command instructs BuildTrack to place the L type path table for the primary volume into the output image at the current position. The position of this command in the control file defines the position of the L path table in the output image.
See also	<ul style="list-style-type: none"> • This command must appear within the primary volume description exactly once. <p>Section 2.3.3 <i>Volume Commands</i> for the PRIMARYVOLUME command.</p>

MPath

Syntax MPath

Example MPath

Description This command instructs BuildTrack to place the M type path table for the primary volume into the output image at the current position. The position of this command in the control file defines the position of the M path table in the output image.

- This command must appear within the primary volume description exactly once.

See also The LPATH command and section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

segd

ONISO

OptionalLPath

Syntax OptionalLPath

Example OptionalLPath

Description The optional L path table, if present, contains exactly the same information as the L path table. It is used for purposes of redundancy. The position of this command in the control file defines the position of the optional L path table in the output image.

- This command is not normally needed when building ISO volumes.
- This command can be issued at most one time within the primary volume definition.

Default If this command is not issued within the definition of the primary volume descriptor, an optional L path table will not be written into the output image.

See also The LPATH command and section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

OptionalMPath

Syntax OptionalMPath

Example OptionalMPath

Description The optional M path table, if present, contains exactly the same information as the M path table. It is used for purposes of redundancy. The position of this command in the control file defines the position of the optional M path table in the output image.

- This command is not normally needed when building ISO volumes.
- This command can be issued at most one time within the primary volume definition.

Default If this command is not issued within the definition of the primary volume descriptor, an optional M path table will not be written into the output image.

See also The MPATH command and section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

PublisherIdentifier

Syntax

PublisherIdentifier <theString>

Example

PublisherIdentifier "ICOM SIMULATIONS"

Example

PublisherIdentifier _PID.DAT

Description

Specifies the publisher identifier field of the primary volume descriptor record. It is used to specify the publisher of the information contained in the volume. If the string begins with an underscore (_) (ASCII \$5F), the remainder of the string is taken to be the name of a file on the root of the primary volume that contains publisher identification information. If a file name is specified, BuildTrack will check to make sure that the given file exists on the root and will generate a warning if it does not.

- The file name must conform to ISO level 1 standards.
- The publisher identifier is a string of up to 128 a-characters in length. For a list of the valid a-characters, see *Appendix C ISO Defined Characters*.

Default

If this command does not appear in the control file, the publisher identifier string field of the primary volume descriptor will contain 128 spaces (ASCII \$20).

See also

Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

SystemIdentifier

Syntax SystemIdentifier <theSystem>

Example SystemIdentifier CPM

Description Specifies a system that can recognize and act upon the contents of the first 16 sectors of the ISO volume (the system area). In effect, this command is used to specify the contents of the system identifier field of the primary volume descriptor record that is being defined.

- This command must be specified exactly once for each primary volume descriptor that is defined.
- The parameter to this command is a string that contains only a-characters. The string can be up to 32 characters in length. For a list of the valid a-characters, see Appendix C *ISO Defined Characters*.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME and SYSTEMAREA commands.

segd

VolumeCreationDate

Syntax

VolumeCreationDate <theDate> <theTime> <theOffset>

Example

VolumeCreationDate 10/17/1964 06:14:00:00 -23

Description

The creation date of the ISO volume is given in a field of the primary volume descriptor. BuildTrack will allow this field to be set to any valid date. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

Default

If this command does not appear in the control file, the creation date of the volume will be set to the date and time that the PRIMARYVOLUME command was read. The default Greenwich offset will be used.

See also

Section 2.3.1 *Global Commands* for the GREENWICHOFFSET command and section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

VolumeEffectiveDate

Syntax VolumeEffectiveDate <theDate> <theTime> <theOffset>

Example VolumeEffectiveDate 12/25/1990 00:00:00:00 -23

Description The effective date of the ISO volume is given in a field of the primary volume descriptor. This can be used to specify the date and time at which the information contained in the volume may be used. BuildTrack will allow this field to be set to any valid date. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

- BuildTrack will NOT generate a warning if the effective date of a volume is set later than the expiration date.

Default If this command does not appear in the control file, the effective date of the volume will be set to 00/00/0000 00:00:00:00 00. This indicates that the data in the volume may be used immediately.

See also The VOLUMEEXPIRATIONDATE command, section 2.3.1 *Global Commands* for the GREENWICHOFFSET command and section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

VolumeExpirationDate

Syntax

VolumeExpirationDate <theDate> <theTime> <theOffset>

Example

VolumeExpirationDate 1/1/2001 00:00:00:00 -23

Description

The expiration date of the ISO volume is given in a field of the primary volume descriptor. This can be used to specify the time and date at which the information contained in the volume may be regarded as obsolete. BuildTrack will allow this field to be set to any valid date. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

- BuildTrack will NOT generate a warning if the expiration date of a volume is set earlier than the effective date.

Default

If this command does not appear in the control file, the expiration date of the volume will be set to 00/00/0000 00:00:00:00 00. This indicates that no expiration date is set for the volume.

See also

The VOLUMEEFFECTIVEDATE command, section 2.3.1 *Global Commands* for the GREENWICHOFFSET command and section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

VolumIdentifier

Syntax VolumeIdentifier <theVolume>

Example VolumeIdentifier TICTACTOE

Description Specifies the identification of the ISO volume. In effect, this command is used to specify the contents of the volume identifier field of the primary volume descriptor record that is being defined.

- This command must be specified exactly once within the primary volume descriptor.
- The parameter to this command is a string that contains only d-characters. The string can be up to 32 characters in length. For a list of the valid d-characters, see *Appendix C ISO Defined Characters*.

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

segad

VolumeModificationDate

Syntax VolumeModificationDate <theDate> <theTime> <theOffset>

Example VolumeModificationDate 10/17/1964 06:14:00:00 -23

Description The modification date of the ISO volume is given in a field of the primary volume descriptor. BuildTrack will allow this field to be set to any valid date. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

Default If this command does not appear in the control file, the modification date of the volume will be set to the date and time that the ENDVOLUME command is read. The default Greenwich offset will be used.

See also Section 2.3.1 *Global Commands* for the GREENWICHOFFSET command and section 2.3.3 *Volume Commands* for the ENDVOLUME and PRIMARYVOLUME commands.

VolumeSetIdentifier

Syntax VolumeSetIdentifier <theString>

Example VolumeSetIdentifier TICTACTOE

Description Specifies the volume set identifier field of the primary volume descriptor record. It is used to specify the volume set of which this volume is a member.

- The volume set identifier is a string of up to 128 d-characters in length. For a list of the valid d-characters, see *Appendix C ISO Defined Characters*.

Default If this command does not appear in the control file, the volume set identifier string field of the primary volume descriptor will contain 128 spaces (ASCII \$20).

See also Section 2.3.3 *Volume Commands* for the PRIMARYVOLUME command.

segd

segad onisoft

2.3.5 Supplementary Volume Commands



AbstractFileIdentifier

Syntax AbstractFileIdentifier <theString>

Example AbstractFileIdentifier ABSTRACT.TXT

Description This command is used to fill in the abstract file identifier field of the supplementary volume descriptor. This field is used to contain the name of a file on the root directory of the ISO volume that contains an abstract statement for the volume.

- The file must appear in the root of the directory hierarchy. If the file does not appear in the root of the hierarchy, BuildTrack will generate a warning message.

Default If this command does not appear in the control file, the abstract file identifier string field of the supplementary volume descriptor will contain spaces (ASCII \$20). In this case, since an abstract file is not specified, none need exist on the root of the volume.

See also Section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command

ApplicationIdentifier

Syntax	ApplicationIdentifier <theString>
Example	ApplicationIdentifier "TIC TAC TOE"
Example	ApplicationIdentifier _AI.TXT
Description	<p>Specifies the application identifier string of the supplementary volume descriptor record. According to ISO, "This field shall specify an identification of the specification of how the data are recorded on the Volume Group of which the volume is a member." If this string begins with an underscore (_) (ASCII \$5F), the remainder of the string is taken to be the name of a file on the root of the supplementary volume that contains application identification information. If a file name is specified, BuildTrack will check to make sure that the given file exists on the root and will generate a warning if it does not.</p> <ul style="list-style-type: none"> • The file name must conform to ISO level 1 standards. • The application identifier is a string of up to 128 a1-characters in length. For a list of the valid a1-characters, see <i>Appendix C ISO Defined Characters</i>.
Default	If this command does not appear in the control file, the application identifier string field of the supplementary volume descriptor will contain 128 spaces (ASCII \$20).
See also	Section 2.3.3 <i>Volume Commands</i> for the SUPPLEMENTARYVOLUME command.

ApplicationUse

Syntax ApplicationUse <theFile>

Example ApplicationUse APPUSE.DAT

Description Specifies an input file that contains the data that should be placed into the application use field of the supplementary volume descriptor. The application use field is 512 bytes long and can be filled with any values desired. The ISO standard does not specify what information (if any) should be placed there. Application designers may choose to place information into this field.

- If the specified input file does not contain enough data to fill the 512 bytes, the remaining bytes will be padded with 0's. If the input file contains too much data, an error message will be generated.

Default If the APPLICATIONUSE command is not issued, the application use field of the supplementary volume descriptor will contain 512 bytes of 0's.

See also Section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command.

second

BibliographicFileIdentifier

Syntax	BibliographicFileIdentifier <theString>
Example	BibliographicFileIdentifier BIBLIO.TXT
Description	<p>This command is used to fill in the bibliographic file identifier field of the supplementary volume descriptor. This field is used to contain the name of a file on the root directory of the ISO volume that contains bibliographic information for the volume.</p> <ul style="list-style-type: none"> The file must appear in the root of the directory hierarchy. If the file does not appear in the root of the hierarchy, BuildTrack will generate a warning message.
Default	If this command does not appear in the control file, the abstract file identifier string field of the supplementary volume descriptor will contain spaces (ASCII \$20). In this case, since an abstract file is not specified, none need exist on the root of the volume.
See also	Section 2.3.3 <i>Volume Commands</i> for the SUPPLEMENTARYVOLUME command.

CopyrightFileIdentifier

Syntax CopyrightFileIdentifier <theString>

Example CopyrightFileIdentifier COPY.RGT

Description This command is used to fill in the copyright file identifier field of the supplementary volume descriptor. This field is used to contain the name of a file on the root directory of the ISO volume that contains a copyright message.

- The file must appear in the root of the directory hierarchy. If the file does not appear in the root of the hierarchy, BuildTrack will generate a warning message.

Default If this command does not appear in the control file, the copyright file identifier string field of the supplementary volume descriptor will contain spaces (ASCII \$20). In this case, since a copyright file is not specified, none need exist on the root of the volume.

See also Section 2.3.3, *Volume Commands* for the SUPPLEMENTARYVOLUME command

DataPreparerIdentifier

Syntax	DataPreparerIdentifier <theString>
Example	DataPreparerIdentifier "TODD SMITH"
Example	DataPreparerIdentifier _DPID.DAT
Description	<p>Specifies the data preparer identifier field of the supplementary volume descriptor record. It is used to specify the person (or other entity) that prepared the data that is recorded on this volume. If this string begins with an underscore (_) (ASCII \$5F), the remainder of the string is taken to be the name of a file on the root of the primary volume that contains data preparer identification information. If a file name is specified, BuildTrack will check to make sure that the given file exists on the root and will generate a warning if it does not.</p> <ul style="list-style-type: none"> The data preparer identifier is a string of up to T28 a1-characters in length. For a list of the valid a1-characters, see <i>Appendix C ISO Defined Characters</i>.
Default	If this command does not appear in the control file, the data preparer identifier string field of the supplementary volume descriptor will contain 128 spaces (ASCII \$20).
See also	Section 2.3.3 Volume Commands for the SUPPLEMENTARYVOLUME command.

DescriptorWrites

Syntax DescriptorWrites <theAmount>

Example DescriptorWrites 3

Description The ISO standard allows the supplementary volume descriptor record to be written out to the ISO volume image more than one time for redundancy. This command allows the user to specify how many times the supplementary volume descriptor record should be written. Each descriptor record that is written takes one sector (2048 bytes) of disc space. There should be at most one occurrence of this command within the definition of the supplementary volume descriptor.

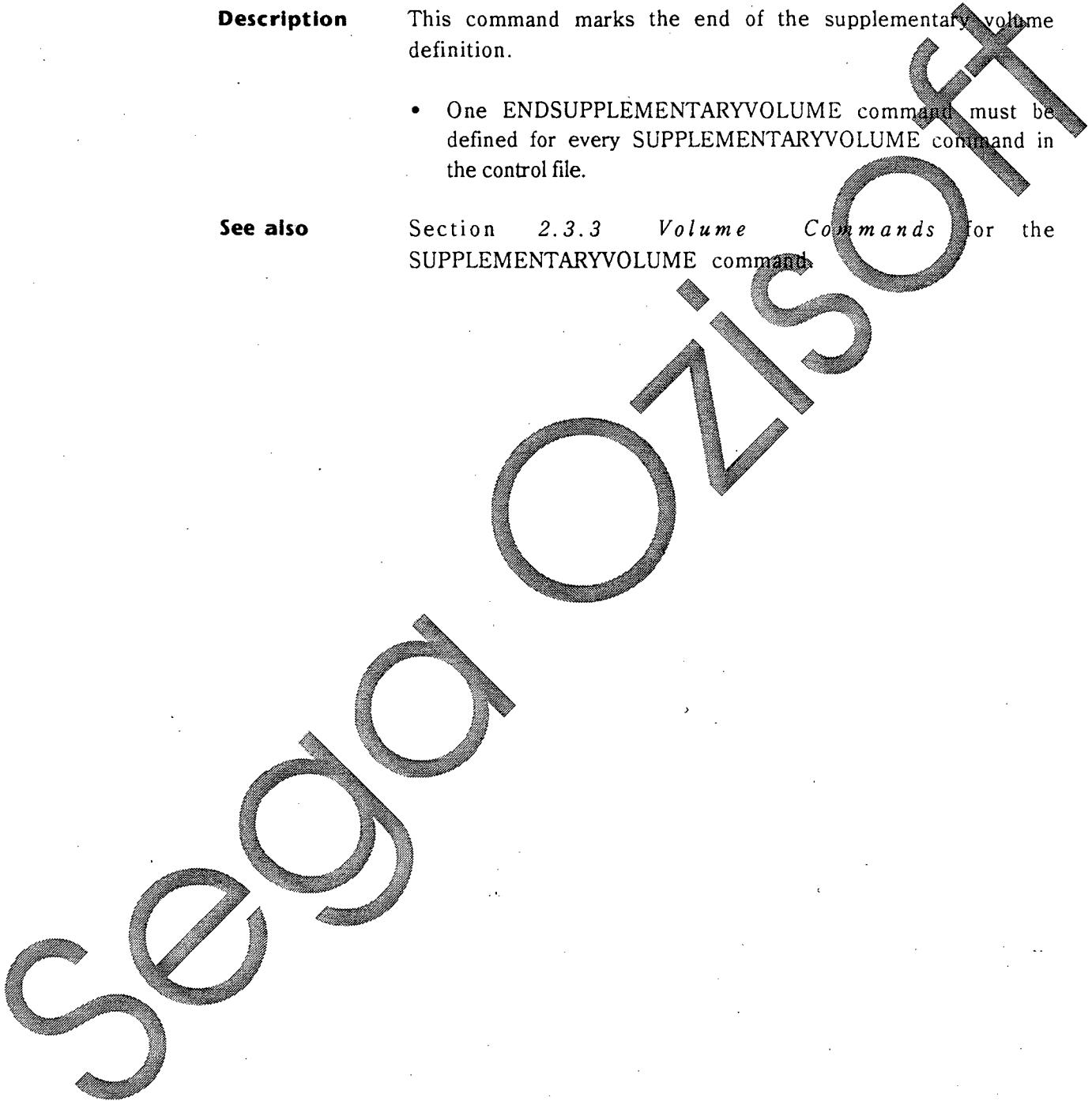
- There is no limit to the number of times that the supplementary volume descriptor record can be written. However, if the parameter given to this command is excessively large, a warning message will be generated.

Default If this command is not used within the definition of the supplementary volume descriptor, the supplementary volume descriptor will be written 1 time.

See also Section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command.

EndSupplementaryVolume

Syntax	EndSupplementaryVolume
Example	EndSupplementaryVolume
Description	This command marks the end of the supplementary volume definition.
See also	Section 2.3.3 <i>Volume Commands</i> for the SUPPLEMENTARYVOLUME command



Hierarchy

Syntax Hierarchy

Example Hierarchy

Description When this command appears in the control file, it indicates that the root of the directory hierarchy for the supplementary volume is beginning.

- The hierarchy definition is ended by using the ENDHIERARCHY command.
- There must be exactly one HIERARCHY command within the definition of the supplementary volume.
- This command must be defined before the FILE or DIRECTORY commands can be used.

See also The DIRECTORY, ENDHIERARCHY, FILE and SUPPLEMENTARYVOLUME commands

sega

LPath

Syntax LPath

Example LPath

Description To speed up access to files contained in ISO volumes, path tables are used. The path table contains a copy of the complete directory tree structure of the directory hierarchy. When a long path name is specified, it is possible to locate the directory at the end of the path more quickly by using the path table rather than by searching the CD for each directory in the path. There are two types of path tables. They are known as L and M types. This command instructs BuildTrack to place the L type path table for the supplementary volume into the output image at the current position. The position of this command in the control file defines the position of the L path table in the output image.

- This command must appear within the supplementary volume description exactly once.

See also Section 2.3.3 Volume Commands for the SUPPLEMENTARYVOLUME command.

MPath

Syntax MPath

Example MPath

Description This command instructs BuildTrack to place the M type path table for the supplementary volume into the output image at the current position. The position of this command in the control file defines the position of the M path table in the output image.

- This command must appear within the supplementary volume description exactly once.

See also The LPATH command and section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command.

segad

OptionalLPath

Syntax	OptionalLPath
Example	OptionalLPath
Description	<p>The optional L path table, if present, contains exactly the same information as the L path table. It is used for purposes of redundancy. The position of this command in the control file defines the position of the optional L path table in the output image.</p> <ul style="list-style-type: none"> • This command is not normally needed when building ISO volumes. • This command can be issued at most one time within each supplementary volume definition.
Default	If this command is not issued within the definition of the supplementary volume descriptor, an optional L path table will not be written into the output image.
See also	The LPATH command and section 2.3.3 <i>Volume Commands</i> for the SUPPLEMENTARYVOLUME command.

OptionalMPath

Syntax OptionalMPath

Example OptionalMPath

Description The optional M path table, if present, contains exactly the same information as the M path table. It is used for purposes of redundancy. The position of this command in the control file defines the position of the optional M path table in the output image.

- This command is not normally needed when building ISO volumes.
- This command can be issued at most one time within each supplementary volume definition.

Default If this command is not issued within the definition of the supplementary volume descriptor, an optional M path table will not be written into the output image.

See also The MPATH command and section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command.

PublisherIdentifier

Syntax	PublisherIdentifier <theString>
Example	PublisherIdentifier "ICOM SIMULATIONS"
Example	PublisherIdentifier _PID.DAT
Description	<p>Specifies the publisher identifier field of the supplementary volume descriptor record. It is used to specify the publisher of the information contained in the volume. If the string begins with an underscore (_) (ASCII \$5F), the remainder of the string is taken to be the name of a file on the root of the supplementary volume that contains publisher identification information. If a file name is specified, BuildTrack will check to make sure that the given file exists on the root and will generate a warning if it does not.</p> <ul style="list-style-type: none"> • The publisher identifier is a string of up to 128 a1-characters in length. For a list of the valid a1-characters, see <i>Appendix C ISO Defined Characters</i>.
Default	If this command does not appear in the control file, the publisher identifier string field of the supplementary volume descriptor will contain 128 spaces (ASCII \$20).
See also	Section 2.3.3 <i>Volume Commands</i> for the SUPPLEMENTARYVOLUME command.

SystemIdentifier

Syntax

SystemIdentifier <theSystem>

Example

SystemIdentifier CPM

Description

Specifies a system that can recognize and act upon the contents of the first 16 sectors of the ISO volume (the system area). In effect, this command is used to specify the contents of the system identifier field of the supplementary volume descriptor record that is being defined.

- This command must be specified exactly once for each supplementary volume descriptor that is defined.
- The parameter to this command is a string that contains only a-characters. The string can be up to 32 characters in length. For a list of the valid a-characters, see *Appendix C ISO Defined Characters*.

See also

Section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME and SYSTEMAREA commands.

VolumIdentifier

Syntax VolumIdentifier <theVolume>

Example VolumIdentifier TICTACTOE

Description Specifies the identification of the ISO volume. In effect, this command is used to specify the contents of the volume identifier field of the supplementary volume descriptor record that is being defined.

- This command must be specified exactly once for each supplementary volume descriptor that is defined.
- The parameter to this command is a string that contains only d-characters. The string can be up to 32 characters in length. For a list of the valid d-characters, see *Appendix C ISO Defined Characters*.

See also Section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command.

VolumeSetIdentifier

Syntax VolumeSetIdentifier <theString>

Example VolumeSetIdentifier TICTACTOE

Description Specifies the volume set identifier field of the supplementary volume descriptor record. It is used to specify the volume set of which this volume is a member.

- The volume set identifier is a string of up to 128 d-characters in length. For a list of the valid d-characters, see *Appendix C ISO Defined Characters*.

Default If this command does not appear in the control file, the volume set identifier string field of the supplementary volume descriptor will contain 128 spaces (ASCII \$20).

See also Section 2.3.3 *Volume Commands* for the SUPPLEMENTARYVOLUME command.

2.3.6 Volume Partition Commands

sega onisoft

DescriptorWrites

Syntax

DescriptorWrites <theAmount>

Example

DescriptorWrites 3

Description

The ISO standard allows the volume partition descriptor record to be written out to the ISO volume image more than one time for redundancy. This command allows the user to specify how many times the volume partition descriptor record should be written. Each descriptor record that is written takes one sector (2048 bytes) of disc space. There should be at most one occurrence of this command within the definition of the volume partition descriptor.

- There is no limit to the number of times that the volume partition descriptor record can be written. However, if the parameter given to this command is excessively large, a warning message will be generated.

Default

If this command is not used within the definition of the volume partition descriptor, the volume partition descriptor will be written 1 time.

See also

Section 2.3.3 *Volume Commands* for the VOLUMEPARTITION command.

EndVolumePartition

Syntax EndVolumePartition

Example EndVolumePartition

Description This command marks the end of a volume partition definition.

- There must be one matching ENDVOLUMEPARTITION command for every VOLUMEPARTITION command in the control file.

See also Section 2.3.3 *Volume Commands* for the VOLUMEPARTITION command.

sega

siso

SystemIdentifier

Syntax SystemIdentifier <theString>

Example SystemIdentifier CPM

Description Specifies a system that can recognize and act upon the contents of the system use field in the volume partition descriptor. In effect, this command is used to specify the contents of the system identifier field of the volume partition descriptor record that is being defined.

- This command must be specified exactly once for each volume partition that is defined.
- The parameter to this command is a string that contains only a-characters. The string can be up to 32 characters in length. For a list of the valid a-characters, see *Appendix C ISO Defined Characters*.

See also The SYSTEMUSE command and section 2.3.3 *Volume Commands* for the VOLUMEPARTITION command.

seg a

SystemUse

Syntax SystemUse <theFile>

Example SystemUse SYS.DAT

Description Specifies an input file that contains the data that should be placed into the system use field of the volume partition descriptor. The system use field is 1960 bytes long and can be filled with any values desired. The ISO standard does not specify what information (if any) should be placed there. System designers may choose to place information into this field.

- If the specified input file does not contain enough data to fill the 1960 bytes, the remaining bytes will be padded with 0's. If the input file contains too much data, an error message will be generated.

Default If the SYSTEMUSE command is not defined, the system use field of the volume partition descriptor will contain 1960 bytes of 0's.

See also Section 2.3.3 *Volume Commands* for the VOLUMEPARTITION command.

VolumePartitionData

Syntax VolumePartitionData <theFile>

Example VolumePartitionData PICT.DAT

Description Specifies an input file that contains the data that should be placed into the volume partition. The volume partition is of arbitrary length and can be filled with any values desired.

Default If the VOLUMEPARTITIONDATA command is not defined, the volume partition will be empty.

See also Section 2.3.3 *Volume Commands* for the VOLUMEPARTITION command.

Sega

VolumePartitionIdentifier

Syntax VolumePartitionIdentifier <theString>

Example VolumePartitionIdentifier PARTITION1

Description Specifies a system that can recognize and act upon the contents of the system use field in the volume partition descriptor. In effect, this command is used to specify the contents of the volume partition identifier field of the volume partition descriptor record that is being defined.

- This command must be specified exactly once for each volume partition descriptor that is defined.
- The parameter to this command is a string that contains only d-characters. The string can be up to 32 characters in length. For a list of the valid d-characters, see *Appendix G ISO Defined Characters*.

See also The SYSTEMUSE command and section 2.3.3 *Volume Commands* for the VOLUMEPARTITION command.

sega onisoft

2.3.7 Boot Record Commands

sega onisoft

BootIdentifier

Syntax BootIdentifier <theString>

Example BootIdentifier V1.0

Description Specifies the boot identifier for the current boot record. The boot identifier identifies the boot system specified in the boot system use field of the boot record.

- This command must be specified exactly once for each boot record that is defined.
- The parameter to this command is a string that contains only a-characters. The string can be up to 32 characters in length. For a list of the valid a-characters, see *Appendix C ISO-Defined Characters*.

See also The SYSTEMUSE command and section 2.3.3 *Volume Commands* for the BOOTRECORD command.

second

first

DescriptorWrites

Syntax DescriptorWrites <theAmount>

Example DescriptorWrites 3

Description The ISO standard allows the boot record to be written out to the ISO volume image more than one time for redundancy. This command allows the user to specify how many times the boot record should be written. Each boot record that is written takes one sector (2048 bytes) of disc space. There should be at most one occurrence of this command within the definition of the boot record.

- There is no limit to the number of times that the boot record can be written. However, if the parameter given to this command is excessively large, a warning message will be generated.

Default If this command is not used within the definition of the boot record, the boot record will be written 1 time.

See also Section 2.3.3 *Volume Commands* for the BOOTRECORD command.

sector

EndBootRecord

Syntax EndBootRecord

Example EndBootRecord

Description This command marks the end of the boot record. Each boot record must end with this command.

- There must be one matching ENDBOOTRECORD command for every BOOTRECORD command in the control file.

See also Section 2.3.3 *Volume Commands* for the **BOOTRECORD** command.

sega

visor

SystemIdentifier

Syntax SystemIdentifier <theString>

Alt. Syntax BootSystemIdentifier <theString>

Example SystemIdentifier CPM

Description Specifies a system that can recognize and act upon the boot identifier and the contents of the system use field in the boot record. In effect, this command is used to specify the contents of the boot system identifier field of the boot record that is being defined.

- This command must be specified exactly once for each boot record that is defined.
- The parameter to this command is a string that contains only a-characters. The string can be up to 32 characters in length. For a list of the valid a-characters, see *Appendix C ISO Defined Characters*.

See also The SYSTEMUSE command and section 2.3.3 *Volume Commands* for the BOOTRECORD command.

SystemUse

Syntax SystemUse <theFile>

Example SystemUse SYS.DAT

Description Specifies an input file that contains the data that should be placed into the boot system use field of the boot record. The system use field is 1977 bytes long and can be filled with any values desired. The ISO standard does not specify what information (if any) should be placed there. System designers may choose to place information into this field.

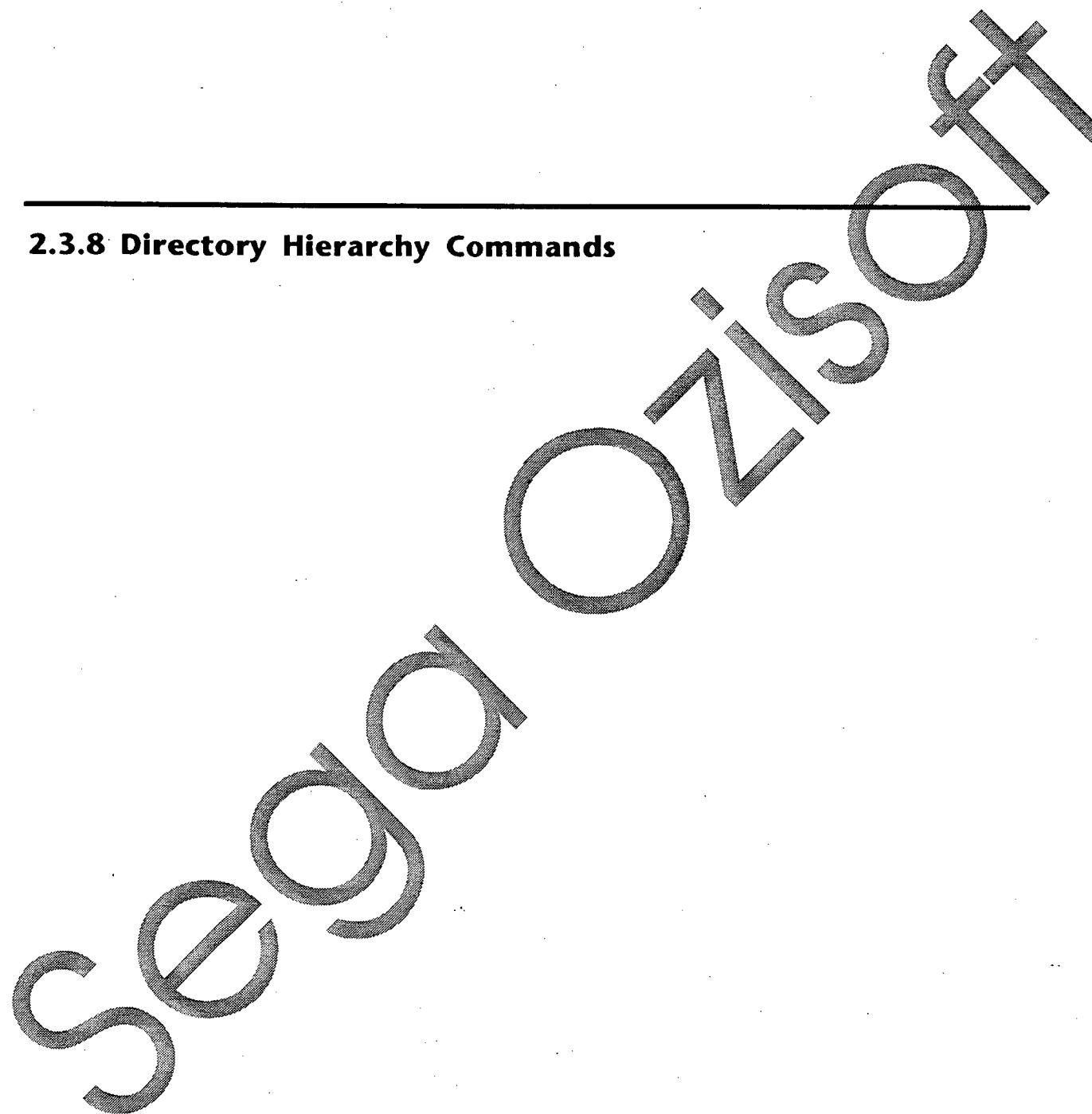
- If the specified input file does not contain enough data to fill the 1977 bytes, the remaining bytes will be padded with 0's. If the input file contains too much data, an error message will be generated.

Default If the SYSTEMUSE command is not issued, the boot system use field of the boot record will contain 1977 bytes of 0's.

See also Section 2.3.3 *Volume Commands* for the BOOTRECORD command.

sega

2.3.8 Directory Hierarchy Commands



AddLength

Syntax AddLength <theLength>

Example AddLength 1024

Description Each directory structure normally uses as much space in the ISO image as needed to contain the file and subdirectory records contained within it. However, there may be times when it is desired that additional space be used for the directory structure. This space could be used to facilitate updates and additions to the directory structure. The ADDLENGTH command is one way that more space can be requested. The parameter describes an additional number of bytes that the directory structure should occupy. These bytes are written out as 0's at the end of the directory structure.

Default If not specified, ADDLENGTH defaults to 0.

See also The MINLENGTH command.

Attributes

Syntax Attributes <attribute1> [attribute2] [attribute3] [...]

Example Attributes Hidden

Description This command will allow the attribute flags for the directory hierarchy to be defined. The attributes can be one or more of the following:

- | | |
|---|---|
| Hidden | - If this attribute is defined, the directory will not be made known to the user. |
| NotHidden | - If this attribute is set, the directory will be made known to the user. |
| <ul style="list-style-type: none"> • If two mutually exclusive attributes are defined, the later of the two is used. | |

Default If the ATTRIBUTES command is not used within the hierarchy description, the attributes will default to NotHidden.

See also Section 2.3.4 *Primary Volume Commands* or section 2.3.5 *Supplementary Volume Commands* for the HIERARCHY command.

Directory

Syntax Directory <theDirectoryName>

Example Directory GRAPHICS

Description This command is used to begin the definition of a new subdirectory within the directory hierarchy. The name of the subdirectory is given as the parameter. All commands that follow this command, until a corresponding ENDDIRECTORY command is seen, apply to this new subdirectory.

- ISO places some restrictions on the characters that can be used in directory names. Although BuildTrack will generate warnings when it encounters directory names that do not meet ISO standards, it will allow them to be placed into the output image. This can be useful on some systems which do not wish to be so restrictive.
- ISO defines "levels" of interchange. Interchange level 1 places even further restrictions on the length and format of directory names. If the **-l** option is given on the command line when running BuildTrack, it will additionally check to make sure that directory names fall within the level 1 definition. If they do not, a warning message will be generated.

See also Section 2.3.9 *Directory Commands* for the ENDDIRECTORY command

EndHierarchy

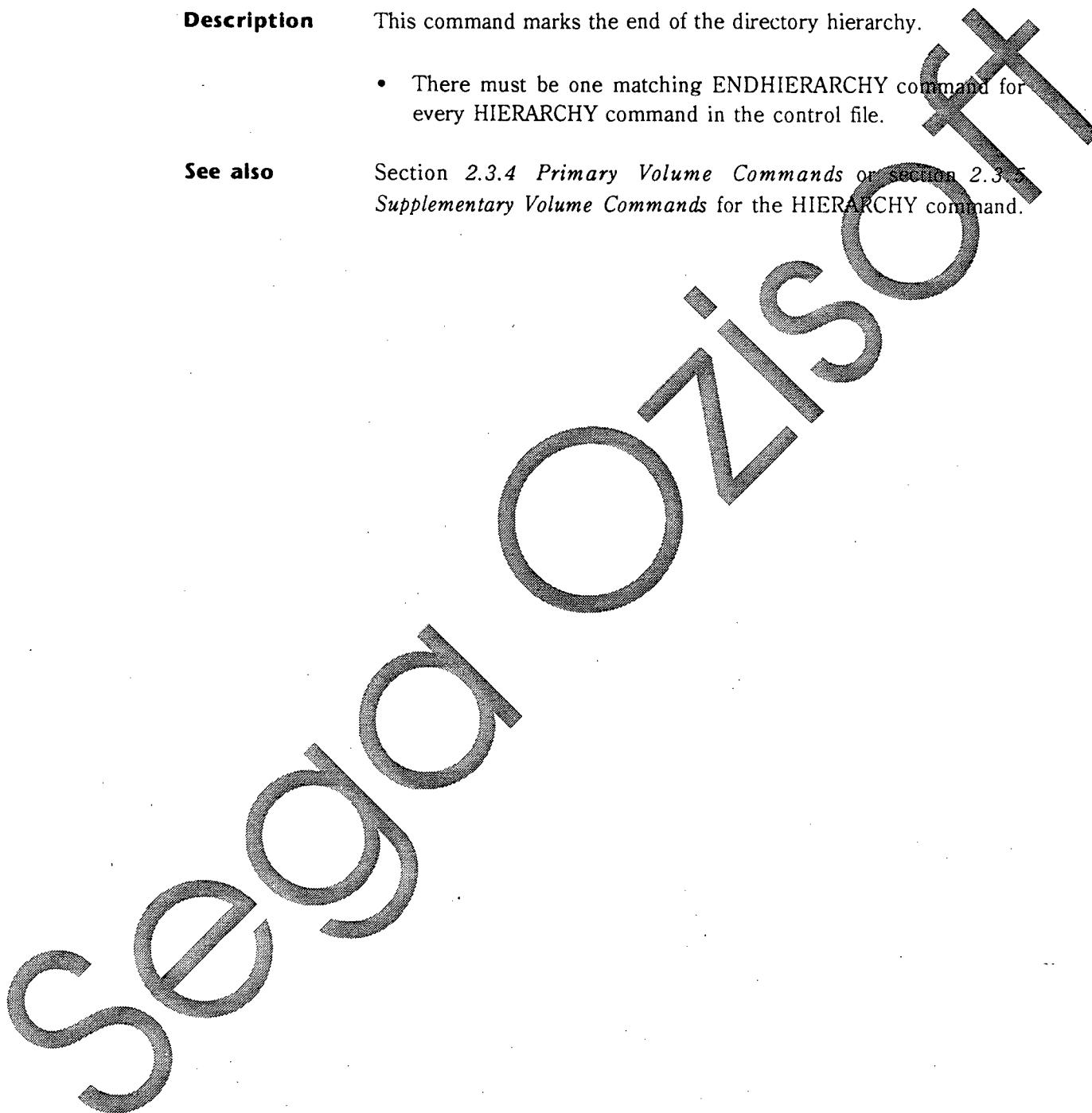
Syntax EndHierarchy

Example EndHierarchy

Description This command marks the end of the directory hierarchy.

- There must be one matching ENDHIERARCHY command for every HIERARCHY command in the control file.

See also Section 2.3.4 Primary Volume Commands or Section 2.3.5 Supplementary Volume Commands for the HIERARCHY command.



File

Syntax File <theISOName> [theVersion]

Example File AUTOEXEC.BAT 3

Description The definition of a file is begun by issuing this command. The name will be taken as the name of this file as it will appear in the ISO image. File names are checked for consistency with the ISO 9660 format and warnings are generated if the file name does not meet the specification. The version number is optional, and if specified, tells the version number of the file. Version numbers can range from 1 to 32767. The definition of the file will continue until the ENDFILE command is seen.

- ISO places some restrictions on the characters that can be used in file names. Although BuildTrack will generate warnings when it encounters file names that do not meet ISO standards, it will allow them to be placed into the output image. This can be useful on some systems which do not wish to be so restrictive.
- ISO defines "levels" of interchange. Interchange level 1 places even further restrictions on the length and format of file names. If the -1 option is given on the command line when running BuildTrack, it will additionally check to make sure that file names fall within the level 1 definition. If they do not, a warning message will be generated.
- If the version number is given as 0, a version number will not be written out (this is not legal for an ISO 9660 file name, but may sometimes be desirable).

Default If the version number is not specified, it defaults to 1.

See also Section 2.3.10 *File Commands* for the ENDFILE command.

MinLength

Syntax MinLength <theLength>

Example MinLength 16384

Description Each directory structure normally uses as much space in the ISO image as needed to contain the file and subdirectory records contained within it. However, there may be times when it is desired that additional space be used for the directory structure. This space could be used to facilitate updates and additions to the directory structure. The MINLENGTH command is one way that more space can be requested. The parameter describes the minimum number of bytes that the directory structure should occupy. If the directory structure would normally take up less space, it will be padded out with 0's until it takes at least as much space as is given by the parameter. If the directory would normally occupy more space than the parameter specifies, this command will have no effect. If any space is added to the directory structure, it will be padded with 0's.

Default If not specified, MINLENGTH defaults to 0.

See also The ADDLENGTH command.

RecordingDate

Syntax

RecordingDate <theDate> <theTime> <theOffset>

Example

RecordingDate 1/1/1990 12:34:56:00 -38

Description

Each directory structure has a recording date. This date specifies when the directory was recorded into the ISO image. BuildTrack will allow any date to be used. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

Default

If not specified, the current date, time and default Greenwich offset is used for the recording date.

See also

Section 2.3.1 *Global Commands* for the GREENWICHOFFSET command and Section 2.3.4 *Primary Volume Commands* or section 2.3.5 *Supplementary Volume Commands* for the HIERARCHY command.

2.3.9 Directory Commands

sega onisoft

AddLength

Syntax AddLength <theLength>

Example AddLength 1024

Description Each directory structure normally uses as much space in the ISO image as needed to contain the file and subdirectory records contained within it. However, there may be times when it is desired that additional space be used for the directory structure. This space could be used to facilitate updates and additions to the directory structure. The ADDLENGTH command is one way that more space can be requested. The parameter describes an additional number of bytes that the directory structure should occupy. These bytes are written out as 0's at the end of the directory structure.

Default If not specified, ADDLENGTH defaults to 0.

See also The MINLENGTH and DIRECTORY commands.

Attributes

Syntax Attributes <attribute1> [attribute2] [attribute3] [...]

Example Attributes Hidden

Description This command will allow the attribute flags for the current directory to be defined. The attributes can be one or more of the following:

- | | |
|---|---|
| Hidden | - If this attribute is defined, the directory will not be made known to the user. |
| NotHidden | - If this attribute is set, the directory will be made known to the user. |
| <ul style="list-style-type: none"> • If two mutually exclusive attributes are defined, the later of the two is used. | |

Default If the ATTRIBUTES command is not used within the directory description, the attribute will default to NotHidden.

See also The DIRECTORY command.

Directory

Syntax Directory <theDirectoryName>

Example Directory GRAPHICS

Description This command is used to begin the definition of a new subdirectory within the current directory. The name of the subdirectory is given as the parameter. All commands that follow this command, until a corresponding ENDDIRECTORY command is seen, apply to this new subdirectory.

- ISO places some restrictions on the characters that can be used in directory names. Although BuildTrack will generate warnings when it encounters directory names that do not meet ISO standards, it will allow them to be placed into the output image. This can be useful on some systems which do not wish to be so restrictive.
- ISO defines "levels" of interchange. Interchange level 1 places even further restrictions on the length and format of directory names. If the -1 option is given on the command line when running BuildTrack, it will additionally check to make sure that directory names fall within the level 1 definition. If they do not, a warning message will be generated.

See also The ENDDIRECTORY command.

EndDirectory

Syntax EndDirectory

Example EndDirectory

Description Ends the current directory definition.

- There must be one matching ENDDIRECTORY command for every DIRECTORY command in the control file.

See also The DIRECTORY command.

sega onisoft

File

Syntax File <theISOName> [theVersion]

Example File AUTOEXEC.BAT 3

Description The definition of a file is begun by issuing this command. The name will be taken as the name of this file as it will appear in the ISO image. File names are checked for consistency with the ISO 9660 format and warnings are generated if the file name does not meet the specification. The version number is optional, and if specified, tells the version number of the file. Version numbers can range from 1 to 32767. The definition of the file will continue until the ENDFILE command is seen.

- ISO places some restrictions on the characters that can be used in file names. Although BuildTrack will generate warnings when it encounters file names that do not meet ISO standards, it will allow them to be placed into the output image. This can be useful on some systems which do not wish to be so restrictive.
- ISO defines "levels" of interchange. Interchange level 1 places even further restrictions on the length and format of file names. If the -1 option is given on the command line when running BuildTrack, it will additionally check to make sure that file names fall within the level 1 definition. If they do not, a warning message will be generated.
- If the version number is given as 0, a warning will be generated and the version number will not be written out (this is not legal for an ISO 9660 file name, but may sometimes be desirable).

Default If the version number is not specified, it defaults to 1.

See also Section 2.3.10 *File Commands* for the ENDFILE command.

MinLength

Syntax MinLength <theLength>

Example MinLength 16384

Description Each directory structure normally uses as much space in the ISO image as needed to contain the file and subdirectory records contained within it. However, there may be times when it is desired that additional space be used for the directory structure. This space could be used to facilitate updates and additions to the directory structure. The MINLENGTH command is one way that more space can be requested. The parameter describes the minimum number of bytes that the directory structure should occupy. If the directory structure would normally take up less space, it will be padded out with 0's until it takes at least as much space as is given by the parameter. If the directory would normally occupy more space than the parameter specifies, this command will have no effect. If any space is added to the directory structure, it will be padded with 0's.

Default If not specified, MINLENGTH defaults to 0

See also The ADDLENGTH command.

segd

RecordingDate

Syntax RecordingDate <theDate> <theTime> <theOffset>

Example RecordingDate 1/1/1990 12:34:56:00 -38

Description Each directory structure has a recording date. This date specifies when the directory was recorded into the ISO image. BuildTrack will allow any date to be used. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

Default If not specified, the current date, time and default Greenwich offset is used for the recording date.

See also Section 2.3.1 *Global Commands* for the GREENWICHOFFSET command.

segad

2.3.10 File Commands

Sega Onisoft

AddLength

Syntax AddLength <theLength>

Example AddLength 1024

Description Each file normally uses as much space in the ISO image as needed to contain it. However, there may be times when it is desired that additional space be allocated for the file in the image. This space could be used to facilitate updates to the file. The ADDLENGTH command is one way that more space can be requested. The parameter describes an additional number of bytes that should be allocated in the ISO image for the file. Once the file size is known, the additional bytes are added to the size and the total number is allocated in the ISO image.

Default If not specified, ADDLENGTH defaults to 0.

See also The MINLENGTH command and section 2.3.9 *Directory Commands* for the FILE command.

Attributes

Syntax	Attributes <attribute1> [attribute2] [attribute3] [...]
Example	Attributes Hidden Record
Description	This command will allow the attribute flags for the current directory to be defined. The attributes can be one or more of the following:
Hidden	- If this attribute is defined, the file will not be made known to the user.
NotHidden	- If this attribute is set, the file will be made known to the user.
Record	- If this attribute is set, the file will be specified as a record.
NotRecord	- If this attribute is set, the file will not be specified as a record.
	<ul style="list-style-type: none"> • If two mutually exclusive attributes are defined, the later of the two is used.
Default	If the ATTRIBUTES command is not used within the file description, the attributes will default to NotHidden and NotRecord.
See also	Section 2.3.9 <i>Directory Commands</i> for the FILE command.

EndFile

Syntax EndFile

Example EndFile

Description The current file definition is ended with this command.

- There must be one matching ENDFILE command for every FILE command in the control file.

See also Section 2.3.9 *Directory Commands* for the FILE command

sega

zisot

ft

MinLength

Syntax MinLength <theLength>

Example MinLength 16384

Description Each file normally uses as much space in the ISO image as needed to contain it. However, there may be times when it is desired that additional space be allocated for the file in the image. This space could be used to facilitate updates to the file. The MINLENGTH command is one way that more space can be requested. The parameter describes the minimum number of bytes in the ISO image that should be set aside for the file. Once the file size is known, this number is compared against it and the larger of the two is allocated in the ISO image.

Default If not specified, MINLENGTH defaults to 0.

See also The ADDLENGTH command.

segd

ISO

RecordingDate

Syntax RecordingDate <theDate> <theTime> <theOffset>

Example RecordingDate 1/1/1990 12:34:56:00 -38

Description Each file has a recording date. This date specifies when the file was recorded into the ISO image. BuildTrack will allow any date to be used. Dates are specified in the following format:

MM/DD/YYYY HH:MM:SS:HH GG

MM is the month, DD is the day of the month, YYYY is the year, HH is the hour (military format), MM is the minute, SS is the second, HH is the hundredths of a second and GG is the Greenwich offset.

Default If not specified, the current date, time and default Greenwich offset is used for the recording date.

See also Section 2.3.1 *Global Commands* for the GREENWICHOFFSET command.

Source

Syntax Source <theFile>

Example Source C:CONTENTS.DAT

Description Specifies the input file whose contents will be placed into the ISO image as the contents of the file that is being defined. Full path names may be specified.

Default If a source is not specified for the current file, the file will be 0 bytes long.

See also Section 2.3.9 *Directory Commands* for the FILE command.

sega

ISO

Sega Ozisoft

Appendices

- Appendix A Parsing Algorithm
- Appendix B Warning and Error Messages
- Appendix C ISO Defined Characters

sega onsoft

Appendix A Parsing Algorithm

BuildTrack parses through the control file much like a UNIX™ shell parses input. It recognizes certain characters as special and performs various substitutions. Below is a list of the special characters and their meaning.

<u>Characters</u>	<u>Usage</u>
white space	Separates parameters.
return	Separates commands.
'...'	Single quotes protect the characters inside from substitution and groups the characters together so they are treated as one argument. Surround single quotes with double quotes to include them in an argument.
"..."	Double quotes group all characters inside as a single entity. Substitutions still occur and the results are grouped. Surround double quotes with single quotes to include them in an argument.
{...}	Braces force a recursive substitution on the variable/variables inside.
[...]	Brackets force a single substitution on the variable/variables inside.
;	Semi-colons begin comments.

Examples

- "{A}"
 - will produce 1 argument that is the replacement for the variable A
- {A}
 - will produce as many arguments as the replacement for A requires
- {"A"}
- {A B}
- {"A B"}
- "{"A B"}"
 - is the same as {A}{B}
 - is the same as {A}{B}
 - requests a substitution of the variable A B
- "{"A B"}"
 - will produce 1 argument that replaces A B (quotes nest through other characters)
- "{"A B"}"
 - will do the same as {"A B"}
- {(A)}
 - is C if {A} = B and {B} = C
- '{A}'
 - will result in the argument {A} being passed

- {A} - if A is defined as '{B}' and B is defined as "TEST", then {A} yields TEST, while
- [A] yields {B}
- "a "rip in" time" - will produce 2 arguments a rip and in time
- "" - will yield 1 argument that is an empty string
- '' - will yield 1 argument that is an empty string
- "don't do that" - will produce one argument don't do that

Sega Onisoft

Appendix B Warning and Error Messages

BuildTrack can potentially output a great variety of warning and error messages. Most of these are self explanatory and are not further detailed here. Below are listed a number of messages for which some additional information was thought to be helpful. In this section, whenever the characters STR are shown, they refer to a string that is filled in by the program at run time. This is usually that name of a command or file.

Warning Messages

Abstract file STR not located in root of hierarchy

Application identification file STR not located in root of hierarchy

Bibliographic file STR not located in root of hierarchy

Copyright identification file STR not located in root of hierarchy

Data preparer identification file STR not located in root of hierarchy

Publisher identification file STR not located in root of hierarchy

The ISO 9660 standard assumes that any of the files named above, if they exist, will be found in the root of the hierarchy of the volume. If the file has not been placed in the root during the build, this warning will be reported.

This parameter seems a bit excessive, but if you insist...

The value given for a particular command seems larger than would normally be expected for a parameter of this type. BuildTrack will allow you to use the large value but will give you this warning, just in case you have inadvertently used the wrong number.

Error Messages

Directory identifier STR is not unique

File identifier STR is not unique

It is not acceptable to have duplicate directory names at a given point in the directory hierarchy. Similarly, all files in a specific directory must have unique names. If any two directories, or any two file names are found to be identical, this error message will be issued.

Failed to allocate memory buffer for build (check the -b option)

The -b option allows the user to specify the number of memory buffers that will be used by BuildTrack during the build. If too many buffers are specified, it might not be possible for BuildTrack to obtain enough memory from the Operating System (OS) to satisfy the request. If you receive this message, reduce the number of buffers requested.

Failed to initialize
Failed to initialize file system
Failed to initialize volume parsing

These messages will usually be accompanied by another message indicating that not enough memory was available to perform the required action.

Failed to write data area
Failed to write system area

These messages will usually be accompanied by another message indicating that some sort of I/O error has occurred.

Could not open control file STR1. OS Reports: STR2
Failed to open STR1 during build phase. OS Reports: STR2
Failed to open STR1 during partition build phase. OS Reports: STR2
Failed to open file STR1. OS Reports: STR2
Failed to open output file STR1. OS Reports: STR2
Failed to open system data file STR1. OS Reports: STR2
Failed to read file STR1. OS Reports: STR2
Failed to read. OS Reports: STR
Failed to write. OS Reports: STR

All of these messages indicate that some type of I/O error occurred while attempting to perform the function noted. The phrase after 'OS Reports: ' will show what type of error the operating system reported to BuildTrack. Consult the OS manual for more information.

Appendix C ISO Defined Characters

a-characters

ISO defines a-characters as the following:

digits 0 - 9, upper case A - Z, space, !, ", %, &, ', (,), *, +,
comma (,), minus (-), period (.), /, semi-colon (;), colon (:), >,
<, =, ?, underscore (_)

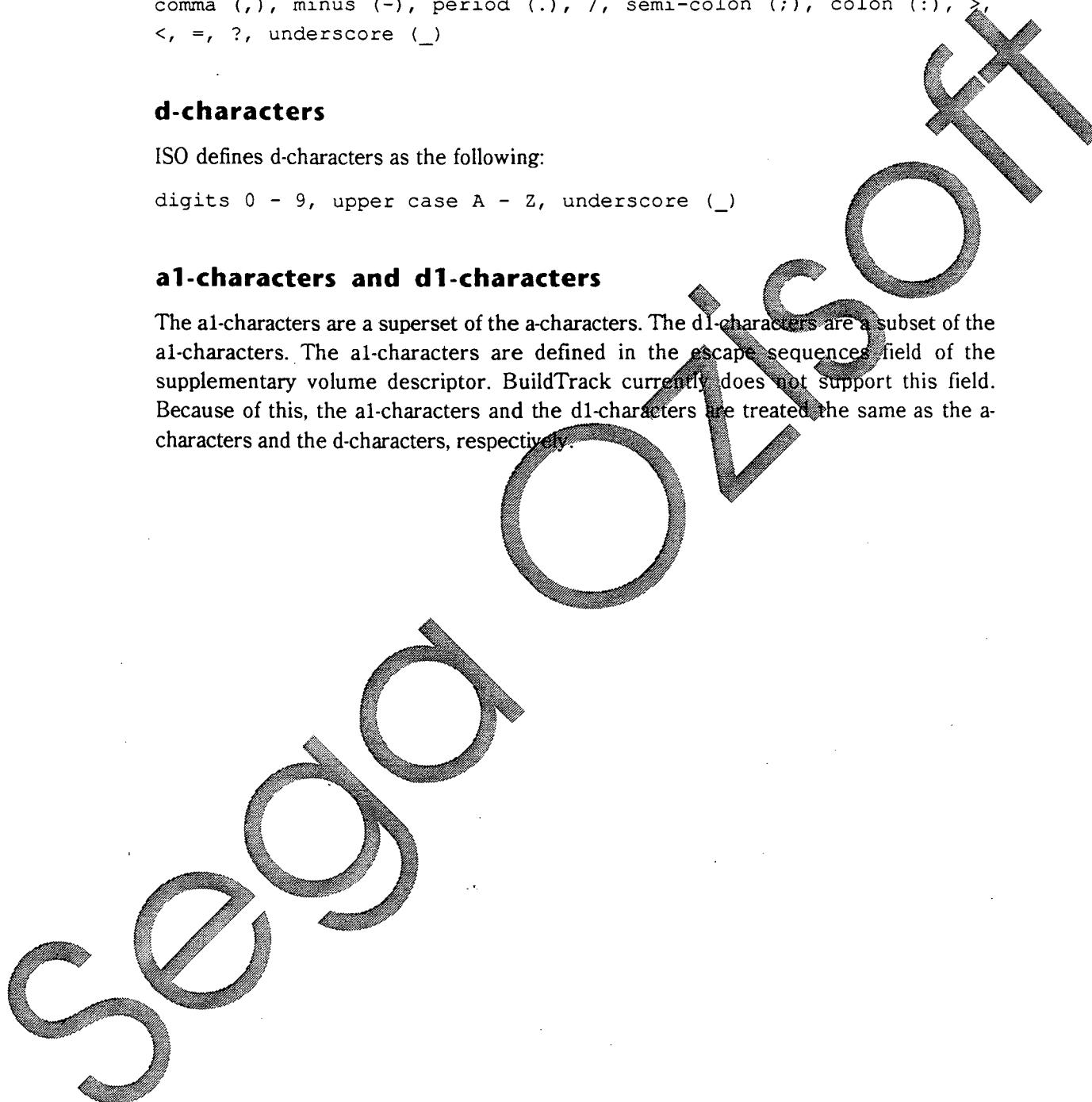
d-characters

ISO defines d-characters as the following:

digits 0 - 9, upper case A - Z, underscore (_)

a1-characters and d1-characters

The a1-characters are a superset of the a-characters. The d1-characters are a subset of the a1-characters. The a1-characters are defined in the escape sequences field of the supplementary volume descriptor. BuildTrack currently does not support this field. Because of this, the a1-characters and the d1-characters are treated the same as the a-characters and the d-characters, respectively.



Index

Symbols and Operators

' , 110
" " , 110
{}, 110
[], 110
; , 110

A

a-characters, 114
a1-characters, 114
AbstractFileIdentifier, 30, 54
AddLength, 86, 94, 102
Aliasing variables, 14
ApplicationIdentifier, 31, 55
ApplicationUse, 32, 56
Attributes, 87, 95, 103

B

BibliographicFileIdentifier, 33, 57
BootIdentifier, 80
BootSystemIdentifier, 83
Brackets
 square [], 110
 curly {}, 110

C

Comments, 110
Control file, 6
CopyrightFileIdentifier, 34, 58

D

d-characters, 114
d1-characters, 114
DataPreparerIdentifier, 35, 59
Day variable, 14
Define, 14
DescriptorWrites, 36, 60, 72, 81
Directory, 88, 96
Double quotes, 110

E

Echo, 15
Edition variable, 14
EndBootRecord, 82
EndDirectory, 97
EndFile, 104
EndHierarchy, 89
EndPrimaryVolume, 31
EndSupplementaryVolume, 61
EndVolume, 23
EndVolumePartition
Error messages, 112

F

File, 90, 98
G
GreenwichOffset, 16

H

Hidden attribute, 87, 95, 103
Hierarchy, 38, 62
Hour variable, 14

I

Include, 17

L

LogicalBlockSize, 39
LPath, 40, 63

M

MinLength, 91, 99, 105
Minute variable, 14
Month variable, 14
MPath, 41, 64

■ N

NotHidden attribute, 87, 95, 103
NotRecord attribute, 103

■ O

OptionalLPath, 42, 65
OptionalMPath, 43, 66

■ P

Parsing algorithm, 110
PrimaryVolume, 24
Program variable, 14
PublisherIdentifier, 44, 67

■ Q

Quotes
 single, 110
 double, 110

■ R

RecordingDate, 92, 100, 106
Record attribute, 103
Return, 110

■ S

Second variable, 14
Semi-colon
 ;, 110
ShowDefines, 18
Single quotes
 '', 110
Source, 107
Space, 110
SupplementaryVolume, 25
SystemArea, 26
SystemIdentifier, 45, 68, 74, 83
SystemUse, 75, 84

Undefining variables, 14

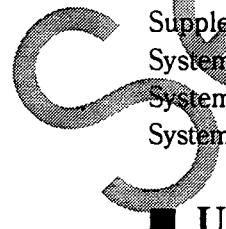
■ V

Version variable, 14
Volume, 20
VolumeCreationDate, 46
VolumeEffectiveDate, 47
VolumeExpirationDate, 48
VolumeIdentifier, 49, 69
VolumeModificationDate, 50
VolumePartition, 27
VolumePartitionData, 76
VolumePartitionIdentifier, 77
VolumeSetIdentifier, 51, 70

■ W

Warning messages, 112
Weekday variable, 14
White space, 110

Year variable, 14
Yearday variable, 14

**■ U**