mkidsk recipe file format

mkidsk creates ISIS I / II / III and ISIS PDS disk images in .IMD or .IMG format using a text "recipe" file that provides instructions as to the content and format of the disk image.

unidsk automatically creates a recipe file when it decodes an existing .IMD or .IMG file to document the content. It is also possible to hand create files, see Creating your own recipe, in this file.

Although **mkidsk** is flexible on recipe filenames, the convention used by **unidsk** and **irepo** is that recipe files begin with an @ character. It is strongly recommended that this convention is followed.

Recipe

A recipe file is a simple text file that contains a small amount of meta data and a list of files to include in the disk image. It can also contain comment lines, to record supplementary information.

There are three types of information lines recorded in a recipe file, with blank lines being ignored.

1. Lines beginning with a #. These are comment lines and can be freely interspersed with other lines. See note below.
2. Metadata lines. These start with a keyword, then a colon, followed by a text value. These can occur in any order. The last metadata line is mandatory and has the keyword **Files** with no text value. This is used to signal that remaining content is the file list.
3. Disk content lines, these define the files to be included in ISIS.DIR order. Note key operating system files have fixed entries in ISIS.DIR and this is honoured by mkidsk.

Note: If **mkidsk** is creating a .IMD file, it copies some of the recipe comments as part of the IMD header. Specifically, the first block of comments at the start of the recipe file and, if the -s option is specified, any **source** metadata line and the immediately following comment lines.

First line

mkidsk puts no special meaning on the first line, except as noted in comments above. However to be consistent with **unidsk** and the **irepo** tools it is recommended that it has the format

```
# name
where name is the recipe name without the leading @.
As this apepars in any .IMD file it helps keep an audit trail
Note. irepo will make sure the first line follows this appoach
```

See naming.md for my personal naming convention.

Metadata

The following metadata lines are explicitly processed, others are ignored. The order here reflects the order unidsk uses. Note it is recommended to always provide the metadata marked with a *.

Metadata	Notes
*__label__: *label_name*	The label name stored in ISIS.LAB. To be consistent with ISIS, *label_name* should be up to 6 alphanumeric characters, followed optionally by a dot or dash and up to 3 alphanumeric characters. The name is converted to upper case.
version: nn	Record the version reference in ISIS.LAB. It can be up to 2 characters. Default is 2 nulls
*__format__: *disk_format*	Mandatory meta data item that specifies disk format to use. The main options are SD, DD and PDS, with the aliases ISIS II SD, ISIS II DD and ISIS PDS accepted for historical reasons. Note SD1, SD2, SD3, SD4, DD3 and DD4 can also be used to force defaults for interleave and skew, see mkidsk documentation for more information.
***os**: *os_system*	The operating system to be written on the disk, valid options are noted below If this meta data is missing, os: NONE is assumed.
interleave: *interleave_info*	Rarely specified, but provides an override for the default disk interleave. The info is 3 characters one each for track 0, track 1 and remaining tracks, the character value is '0' + the interleave so an interleave of 12 is the character '<' Note, mkidsk uses the format and os meta data to work out the interleave if not specified.
skew: *skew_Info*	Not used by **mkidsk**, but documents a non standard intertrack skew. Same format as interleave_info Note mkidsk uses the format and os meta data to work out the skew to use, if required.
crlf: *nn*	Only really of use for ISIS PDS disks where the crlf field in the ISIS.LAB file is not set during format This provides a way to force the value to match an original disk. Here each n is either an alphanumeric character, a dot or #hh, where h is a hex nibble 0-9,A-F
source: *filename*	Records the source image file that the information in the recipe was capture from. Although generated by **unidsk**, additional entries can be added to help record alternative image sources See notes above re use of comment lines after the source: line.
*__Files__:	Mandatory last meta data item

Note: Interleave is the spacing between sectors on a track and skew is the sector spacing added between tracks.

Current supported values for os meta data

Note user specified OS files will overwrite these, so it is advisable to either not specify the OS files in the content lines, or to use the AUTO location, see below.

Name (case insensitive)	**Default OS written to disk**
NONE or CORRUPT	Only ISIS.DIR, ISIS.LAB, ISIS.MAP and non system ISIS.T0 provided. Note ISIS.BIN should not be specified. Non standard ISIS.T0 can be specified
UNKNOWN	Only ISIS.DIR, ISIS.LAB, ISIS.MAP and non system ISIS.T0 provided. User specifies other files and bootable ISIS.T0.
ISIS 1.1 or ISIS I 1.1	ISIS I 1.1, a check is made that the format is SD
ISIS 2.2 or ISIS II 2.2	ISIS II 2.2, a check is made that the format is SD
ISIS m.n or ISIS II m.n	ISIS II m.m where m.n is one of 3.4, 4.2, 4.1, 4.2, 4.2w, 4.3, 4.3w. SD & DD formats supported
ISIS III(N) m.n	ISIS III(N) m.n where m.n is 2.0 or 2.2. SD & DD formats supported
PDS m.n	ISIS PDS m.n where m.n is 1.0 or 1.1 - requires PDS format
TEST m.n	Writes the special ISIS.BIN used by CONFIG m.n, where m.n is 1.0 or 1.1. SD & DD formats supported

Disk content lines

The individual description lines specify the files used in ISIS.DIR order. Each line is of the form

```
[ISISName,[attributes,[checksum,]]]location [# comment]
```

The location field is mandatory, others can be omitted, however if one of them is used, all previous prior to it must be provided, although they can be blank.

Field	Notes
ISISName	The name used in the ISIS.DIR file that is created. It should be up to 6 alphanumeric characters optionally followed by a dot and up to 3 alphanumeric characters If omitted the name is derived from the final filename part of the location field, see note below for details
attributes	The attributes to be set for the file. They can be any of **F** Format **I** Invisible **S** System **W** Write Protected If no attributes are supplied, the system files have their default format attributes as per the OS specified, the others have none To force no attributes for system files use a space character.
checksum	This is the SHA1 checksum of the file. Its main purpose is to allow lookup of files in a central file repository. if the file starts with a * character is is assumed the file has an error and **mkidsk** skips the file, otherwise the value is ignored. **unidsk** and **irepo** however use it to identify files in the repository and it is used to track changed items in the repository.
location	This is either a special marker or the location of the file to use as follows: **AUTO** the file is auto generated. The line is optional but can be used to specify non-standard attributes **ZERO** the file has zero length and is auto generated path the location of the file to load. A leading **^** is used to prefix the name with the file repository path
comment	Optional comment prefixed by a #. Note this does mean that location paths cannot have # in them.

Note a suitable set of system files, from the list below, are automatically included by mkidsk and need not be listed explicitly.

ISIS.DIR, ISIS.LAB, ISIS.MAP, ISIS.FRE, ISIS.T0, ISIS.BIN, ISIS.PDS, ISIS.OV0, ISIS.OV1, ISIS.OV2, ISISC.BIN

A potential reason for listing the above files is to override the default system attributes and it is generally recommended that if this is done, the location is set to AUTO.

For files in the list above from ISIS.T0 onwards, it is possible to specify alternative files to use to create non standard disks, e.g. when OS is specified as UNKNOWN.

Example

The example below was generated using **unidsk** disk on the file 95000333.imd, from the bitsavers web site.

```
# IN-950033-03-S
label: 950033-03
version: 34
format: SD
os: NONE
source: 95000333.imd
# IMD 1.17: 12/09/2009 18:51:04
# ISIS-II PL/M-86 COMPILER
# P/N 9500033-03 SD
# (C) INTEL 1978
Files:
ISIS.DIR,FI,,AUTO
ISIS.MAP,FI,,AUTO
ISIS.T0,FI,Fc/5LJAGrWS8Kz1X0GgSF8Dln5o,^Intel80/isis_ii/nonsys/isis.t0
ISIS.LAB,FI,,AUTO
PLM86,W,/60LQe2UCXMh8Dc942v3AvNF9Qg,^Intel80/plm86/1.2/plm86
PLM86.OV0,WI,VI29iYKVpy9GeqiX5R9Ca1bh63Y,^Intel80/plm86/1.2/plm86.ov0
PLM86.OV1,WI,KmLgmu1ogNu2ZD/lmEwh8R8tHDo,^Intel80/plm86/1.2/plm86.ov1
PLM86.OV2,WI,zsn/0+inB0y8MKp0t+pgW2eNCUo,^Intel80/plm86/1.2/plm86.ov2
PLM86.OV3,WI,6BHF8ZPZwNid/XmB+qM4HPOpSqA,^Intel80/plm86/1.2/plm86.ov3
PLM86.OV4,WI,S9YCuJ/7oe2g8RPK0G96pGUcVV0,^Intel80/plm86/1.2/plm86.ov4
PLM86.OV5,WI,vJv6r6B4WFNnIgcyiQbNnypA93M,^Intel80/plm86/1.2/plm86.ov5
PLM86.OV6,WI,0tklWqAv6ITm5iWiEgaArpXUaSw,^Intel80/plm86/1.2/plm86.ov6
PLM86.LIB,W,QW0KTjXZ8+UJxblvn65nGQL6Qsc,^Intel80/plm86/1.2/plm86.lib
IXREF,W,TdfNx2lHaV891ycMS5MauUtSAYk,^Intel80/ixref/1.2/ixref
```

Note, the version: 34, indicates that the original disk was formatted on an ISIS II 3.4 system, however examining the imd file reveals that none of the interleave and skew has been preserved.

Auto generation of ISISName

In the latest recipe format, it is possible to omit ISISName from the recipe line. The tools mkidsk and irepo generate a suitable value from the location information as follows.

1. The final file name part of the location path is located.
2. The leading characters of the file name are checked to find the longest match against following
    1. 1-6 alphanumeric characters, a dot, 1-3 alphanumeric characters
    2. 1-6 alphanumeric characters
3. It there is no match, then an error is reported, otherwise the matching text, converted to uppercase, is used as the ISISName.

Note the file name can have extra characters beyond the matching text. For example

```
^Intel80/isis_ii/nonsys/isis.t0           => ISIS.T0
^Intel80/isis_ii/nonsys/isis.t0_0xe5      => ISIS.T0      extra text ignored
file.1                                    => FILE.1
fileabc.123                               => FILEAB       no dot after 6
alphanumeric characters
file.1$$                                  => FILE.1
file.$$$                                  => FILE         $ not alphanumeric,
so no alphanumeric after dot
```

Creating your own recipe

Although original intent of recipes was to capture information to be able to recreate an ISIS disk image, it is reasonably simple to write recipe files to create bespoke disks. Either by starting from scratch as shown in the examples below, or by copying and amending an existing recipe.

Note the end of line comments beginning ! are for explanation and not part of the recipe file

Example 1

The mkidsk recipe to create an ISIS-II DD disk, with label MYTEST, no system and two local files mytest and mytest.dat. It assumes that the recipe file is ""@MYTEST simple"

```
# MYTEST simple         ! mirrors file name without the @ as per recommendation
label: MYTEST
format: DD
Files:
mytest                  ! remaining data is auto generated from these filenames
mytest.dat
```

Example 2

The mkidsk recipe to create an ISIS-II SD disk, with label MYTEST, ISIS-II V4.0, the DIR command and two local files mytest and mytest.dat. It assumes that the recipe file is ""@MYTEST bootable"

```
# MYTEST bootable           ! mirrors file name without the @ as per
recommendation
label: MYTEST
format: SD
os: ISIS 4.0
Files:
^Intel80/isis_ii/4.0/dir    ! include the dir command
mytest                      ! local files
mytest.dat
```

Note although the above are sufficient for **mkidsk**, running **irepo -u** *recipe_name* , will fill in missing information and also check the files exist.

```
Running irepo -u "@MYTEST bootable", with two randomly named files mytest and
mytest.dat, changes the file into

# MYTEST bootable
label: MYTEST
format: SD
os: ISIS 4.0
Files:
DIR,,IjfRkCY/72p5Se7le6CNq8Zh84U,^Intel80/isis_ii/4.0/dir
MYTEST,,4TZC4FGa38yJtp2fNGdPOa7bHmg,mytest
MYTEST.DAT,,L2SveCSD5U0t4bEKMndtgtHSj3E,mytest.dat
```

See Also

Naming convention and disktools

```
Update by Mark Ogden 16-Nov-2020
```