

mkidsk recipe file format

mkidsk creates ISIS I, ISIS II 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 appears in any .IMD file it helps keep an audit trail
Note. irepo will make sure the first line follows this approach
```

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: <i>label_name</i>	The label name stored in ISIS.LAB. To be consistent with ISIS, <i>label_name</i> 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: <i>disk_format</i>	The disk format used. It can be one of ISIS II SD, ISIS II DD or ISIS PDS. Note for ISIS I use ISIS II SD
os: <i>os_system</i>	The operating system on the disk. mkidisk only uses this to provide a comment in the imd file. If os is missing, blank or has a value of NONE, then the comment records the disk as a non-system disk, otherwise it is a system disk.
interleave: <i>interleave_info</i>	Rarely used, 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 '<'
skew: <i>skew_Info</i>	Not used by mkidisk , but documents a non standard intertrack skew. Same format as interleave_info
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: <i>filename</i>	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.

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
```

The commas must appear even if the field is blank

Field	Notes
ISISName	The name used in the ISIS.DIR. It should be up to 6 alphanumeric characters optionally followed by a dot and up to 3 alphanumeric characters
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, the others have none
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 skip 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 ZEROHDR the file has zero length but the header is allocated. It is autogenerated path the location of the file to load. A leading ^ is replaced by the file repository path

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: ISIS II 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,0tk1WqAv6ITm5iWiEgaArpXUaSw,^Intel80/plm86/1.2/plm86.ov6
PLM86.LIB,W,QW0KTjXZ8+UJxb1vn65nGQL6Qsc,^Intel80/plm86/1.2/plm86.lib
IXREF,W,TdfNx21Hav891ycMS5MauUtSAYk,^Intel80/ixref/1.2/ixref
```

Creating your own recipe

Although original intent for capturing the information was to recreate an ISIS disk image, it is reasonably simple to write recipe files to create bespoke disks.

The two examples below illustrate this. **Note** the end of line comments beginning ! are for explanation and not part of the recipe file

Example 1

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 @
label: MYTEST
format: ISIS II DD
Files:
MYTEST,,,mytest          ! defaults to non system disk so nothing else needed
                           apart from local files
MYTEST.DAT,,,mytest.dat
```

Example 2

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

```
# MYTEST bootable          ! mirrors file name without the @
label: MYTEST
format: ISIS II DD
os: ISIS II 4.0             ! anything other than blank, NONE or CORRUPT
Files:
ISIS.BIN,,,^Intel80/isis_ii/4.0/isis.bin    ! the operating system ISIS II v4.0
ISIS.T0,,,^Intel80/isis_ii/4.0/isis.t0      ! the boot file
ISIS.CLI,,,^Intel80/isis_ii/4.0/isis.cli    ! needed to enter commands
MYTEST,,,mytest                          ! local files
MYTEST.DAT,,,mytest.dat
```

Note although the above are sufficient for **mkidisk**, 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: ISIS II DD
os: ISIS II 4.0
Files:
ISIS.BIN,,sRiKeMMS6BzoByw8JMfR602LkRc,^Intel80/isis_ii/4.0/isis.bin
ISIS.T0,,7zxc/qa5m2hRvuiSXG7NnaMB5Ow,^Intel80/isis_ii/4.0/isis.t0
ISIS.CLI,,FgoAa7lMzUi/LNa38wPB43R2Ddw,^Intel80/isis_ii/4.0/isis.cli
MYTEST,,4TZC4FGa38yJtp2fNGdPOa7bHmg,mytest
MYTEST.DAT,,L2SveCSD5U0t4bEKMndtgtHSj3E,mytest.dat
```

See Also

[Naming convention](#) and [disktools](#)

Update by Mark Ogden 30-Oct-2020