

Installation of MIDAS on UNIX/Linux Systems

Data Management Division,
European Southern Observatory,
Karl-Schwarzschild-Straße 2,
D-85748, Garching, Germany

September 2011

Abstract

The Munich Image Data Analysis System, MIDAS, developed and maintained by ESO, provides general tools for image processing and data reductions with emphasis on astronomical applications. MIDAS is at present distributed once a year in September.

Please, note that by now Midas has entered a phase where only maintenance and minimal bug fixes are done.

The official name, ESO-MIDAS, is a registered trademark. ESO-MIDAS is available under the GNU General Public License (GPL), and can be implemented on VMS, OpenVMS and UNIX (= Unix/Linux/Mac OSX) systems.

This document describes the installation procedure for MIDAS on Unix/Linux/Mac OSX systems.

Chapter 1

Introduction

The Munich Image Data Analysis System, MIDAS, is developed and maintained by the European Southern Observatory. Its main goal is to provide the astronomical community with software tools for both general image processing and analysis, and for the reduction of data acquired at ESO telescopes. Although originally developed on a VAX/VMS system, since 1988 MIDAS is available for systems with a VAX/VMS operating systems, or with one of the main flavors of the UNIX systems.

MIDAS is made available to the public by ESO via anonymous ftp from ESO's ftp server: ftp.eso.org.

This document describes the installation of MIDAS on computer systems running under the UNIX ¹ operating system. It contains all information needed to ensure a smooth installation of the MIDAS software. In Chapter 2 the actual installation of MIDAS is described, as well as how MIDAS should be set up properly. In addition this chapter describes a few items related with file initializations. In Chapter 3, the installation procedures of graphic devices can be found. Chapter 4 describes other optional stand-alone utilities. Chapter 5 contains the address of the MIDAS problem reporting system in case of problems.

Throughout this document a number of typographic conventions are used. These conventions are given below.

- *Italic type style* is used to emphasize a term or statement in the body of a paragraph:
- `Typewriter type style` is used:
 - to show, within the body of the text, command lines or options that should be typed verbatim on the screen;
 - to show, within examples, computer-generated input/output or the contents of files.

¹this includes Unix, Linux and Mac OSX systems

- **Bold type style** only used to emphasize a very important note or remark for the installation of MIDAS.

In addition the following symbols are used:

- # for the standard prompt when executing as superuser.
- % for the standard prompt when executing as a normal user.

Chapter 2

Main Installation

2.1 Create a directory for MIDAS

First create a directory for MIDAS into which you will read the distribution tape (usually named something like `/midas`, which also can be referred as the **MIDASHOME** directory).

2.2 Retrieve MIDAS sources from the *midas* ftp account

To retrieve MIDAS sources:

```
% cd /midas
% ftp ftp.eso.org
Name: ftp
Password: your_email_address
ftp> cd /midaspub/11SEP/sources
ftp> get README.sources
ftp> binary (necessary for binary files)
ftp> get 11SEPp11.0.tar.gz
ftp> get any_other_file
ftp> quit
```

Or visit the MIDAS homepage on the web:

<http://www.eso.org/sci/data-processing/software/esomidass/> .
and click on the link **ftp-retrieval** to get to the MIDAS ftp area.

To decompress and un-tar the MIDAS sources:

```
% cd /midas
% tar -zxvf 11SEPp11.0.tar.gz
```

```
% ln -s 11SEPp11.0 11SEP
```

All unloaded files will come under the release directory `11SEP`, also referred to as **MIDVERS**¹.

User support is offered on a best effort basis, only. We can provide help for problems, questions during installation or usage of the MIDAS system. If answers to questions cannot be obtained locally (e.g. through the manuals) or after consulting our FAQ on the WWW, please use the web interface of the MIDAS problem reporting system. This can be reached via the button `ProblemReportForm` on top of the MIDAS home page.

2.3 The AutoConfig Script

A fully automated shell script, `autoconfig`, in the directory `/midas/11SEP/install/unix` wraps around the script, `config`, described below, and performs all the intermediate steps explained in the following sections. If you execute `autoconfig` you can skip all the sections until `Verifying Midas`.

To execute the `autoconfig` script type:

```
% cd /midas/11SEP/install/unix
% sh autoconfig
```

2.4 The Config Script

The shell script `/midas/11SEP/install/unix/config` is the front-end of the installation of MIDAS. It offers a menu of different options that are executed when selected by the option number. To execute the `config` script type:

```
% cd /midas/11SEP/install/unix
% sh config
```

The `config` script will generate the following output on the screen:

```
MIDAS CONFIG MENU:
=====
1 - list of MIDAS packages available
2 - select only MIDAS core
3 - select all MIDAS packages
```

¹If you do not use the standard `MIDASHOME` (`/midas`) and `MIDVERS` (`11SEP`) settings, you have to update the file `/midas/11SEP/local/default.mk`, because `MIDVERS` and `MIDASHOME` are hard coded in there.

```
4 - select MIDAS core and packages of your choice
5 - preinstall MIDAS
6 - install MIDAS
61 - install MIDAS - force 32bit executables
62 - install MIDAS - force 64bit executables
66 - install MIDAS - prepare own MAKE_OPTIONS
67 - install MIDAS - use the MAKE_OPTIONS above
7 - update MIDAS
8 - setup MIDAS
9 - clean MIDAS
10 - help
q - quit
```

Select: (option number or q to quit)

2.5 Preinstallation of MIDAS

From the `config` menu, select the option 5 - preinstall MIDAS. If your system matches one of the system names located in the directory `/midas/11SEP/install/unix/systems`, the procedure will use it automatically to complete the preinstallation of MIDAS, otherwise it will show you a list of system names for a selection. If your machine is not included in the list type the option `unknown`.

NOTE: For some systems, the preinstallation will print out, on your screen, important notes for the installation of MIDAS on your system. Please, read these notes carefully because they could indicate a specific environment that you should check before you continue with the installation of MIDAS. The file containing these notes is located in `/midas/11SEP/local/README`.

2.6 Selection of optional packages

MIDAS sources are divided in the *core* of MIDAS (indispensable and therefore always included and installed) and several *optional* packages, grouped in three categories: applications (*applic*), standard reduction packages (*stdred*) and contribution packages (*contrib*). The selection is done in the `config` script using the following options:

```
MIDAS CONFIG MENU:
=====
1 - list of MIDAS packages available
2 - select only MIDAS core
```

- 3 - select all MIDAS packages
- 4 - select MIDAS core and packages of your choice

The first option gives you the complete list of optional packages available in the MIDAS release and an indication whether they have been selected. A brief description of each package is displayed when the name of the package is entered.

NOTE: By default, and without selecting any of the above options, only the *core* of MIDAS will be installed.

2.7 Installation of MIDAS

If you are satisfied with the selection of packages, you can start with the installation of MIDAS.

From the `config` menu, select option 6 - `install` MIDAS. The options 61, 62, 66 and 67 are to be used in case of problems with the standard option 6 - for details see below. The procedure will check for the existence of the C and Fortran compiler (the Fortran compiler is usually just a frontend to the C compiler), the ncurses library, the X11 library and the Motif library. Since Midas needs the header files of these libraries, the development packages of these libraries have to be installed on your Linux system. Without the ncurses library no command line editing will be possible in Midas, otherwise it has no effect. Without the X11 library the installation will give some errors in `/midas/11SEP/system/idiserv/src`. MIDAS will be installed anyway but without graphic and display capabilities. And without the Motif library the Graphical User Interfaces (GUIs) of MIDAS cannot be installed, e.g. the help GUI. On many Linux systems there is no Motif library available, instead the lesstif library has to be installed. On Mac OSX the Fortran compiler/frontend is not included with the system and has to be obtained from the Web, e.g. via Fink or MacPorts.

The installation of MIDAS runs in the background. A log file is written in the temporary directory `/midas/11SEP/tmp` which can be checked for errors when finished. If you want to trace the output of the installation try the following command:

```
% tail -f /midas/11SEP/tmp/install.here_is_a_number
```

The options 61 and 62 are mainly for Midas installations on Mac OSX. Depending upon the version of Darwin, there is a standard 64bit C compiler that produces 64bit executables, but running these executables crashes immediately (Abort trap). So, with these options you can force the standard C compiler to produce 32- or 64-bit modules. Options 66 and 67 give the possibility to edit the `make_options` file manually and use it in the installation. First, choose option 66 and don't continue with the installation, then edit the file `/midas/11SEP/local/make_options` manually, and start the installation with option 67.

2.8 Update of MIDAS

If, for whatever reason, the installation of MIDAS is stopped, and you want to finish it, you can continue from the point where it was stopped by selecting the option 7 - **update** MIDAS, from the **config** menu.

This option is useful to avoid re-compilation of files. For instance, when selecting other packages of MIDAS that were not included in a first selection.

The update procedure also runs in the background and the output is written in the file `/midas/11SEP/tmp/update.here_is_a_number`.

2.9 Setup of MIDAS

While the installation or update of MIDAS is still running, you can execute this step from the **config** menu by selecting option 8 - **setup** MIDAS. This script will ask you for five definitions that will be updated in your file `/midas/11SEP/monit/syskeys.unix` (default values are shown in parenthesis):

Name of the computer/system: Max. 20 characters. The string will be appended to the message 'MIDAS version 11SEP on ' and will welcome you when starting a MIDAS session.

Debugger command: Max. 8 characters. Name of the debugger in your system (gdb, dbx, sdb, xdb, etc) that will be invoked when running a MIDAS command with the **DEBUG** option.

Command to print a Postscript file: Max. 20 characters. Command to send a Postscript file to a Postscript printer.

Command to print an ASCII file: Max. 20 characters. Name of Command to convert an ASCII file into Postscript and then send it to a Postscript printer.

Email of the MIDAS administrator at your site (problem-report service Max. 40 characters. Problem reports generated by the **REPORT/PROBLEM** and the **Feedback...** option in the Graphical User Interface **HELP** will be sent to this address. If your site does not have a person responsible for such a service, the default is the MIDAS problem-report service supported at ESO.

The **setup** MIDAS will now configure some files needed for the start-up of MIDAS:

```
/midas/11SEP/system/unix/inmidas
/midas/11SEP/system/unix/gomidas
/midas/11SEP/system/unix/helpmidas
```

`inmidas` and `gomidas` are the commands to start and continue MIDAS in a user terminal (no GUI exists for that). `helpmidas` is a script to start a stand-alone GUI help for MIDAS, equivalent to execute `create/gui help` within MIDAS. These files should be copied by the system administrator to a public directory (included in the *PATH* list) where users of MIDAS can access them by name (e.g. `/usr/local/bin`).

The command `inmidas` starts a MIDAS session. Type `inmidas -help` for a description of which options are available. Options can also be passed to MIDAS through their equivalent environment variables, but options passed by argument take preference over those defined in the environment.

Next is a list of nroff files for the UNIX `man` command. describing the syntax for the previous scripts and should be copied to a local *man* directory (e.g `/usr/local/man`):

```
/midas/11SEP/system/unix/inmidas.1
/midas/11SEP/system/unix/gomidas.1
/midas/11SEP/system/unix/helpmidas.1
/midas/11SEP/system/unix/readline.1
```

2.10 Running MIDAS for the first time

Once the installation of MIDAS is successfully completed, you still need to initialize some MIDAS files by executing the following sequence, once:

```
% mkdir /midas/tmp
% cd /midas/tmp
% inmidas
Midas 001> @ compile.all
(this will produce about 40 lines of output and it will exit to the Shell)
% gomidas
Midas 002 > @ ascii_bin
(this will create the LUTs and ITTs files)
Midas 003 > bye
```

Now you are ready to run MIDAS. Type `inmidas` for your first MIDAS session or `gomidas` to resume a previous interrupted MIDAS session.

NOTE: If you've used `autoconfig` instead of `config` for the installation, all the steps described above are done automatically for you.

2.11 Verifying MIDAS

To verify the correct installation of MIDAS you can now execute a set of verification procedures:

```
% cd /midas/tmp
% inmidas
Midas 001> @ vericopy
```

The `vericopy` procedure will copy the verification procedures and some data files to your working directory. Some of these procedures use, by default, the display facilities but they can be disabled with a given argument. To run a particular verification procedure:

```
Midas 002> @@ verifyXX
```

have a look into your directory to see what `XX` can be, or if you want to run all of them (it stops if something fails), just type:

```
Midas 003> @@ veriall
```

The verification procedures are designed to exercise MIDAS commands and to compare results with internal and correct values. They should stop and report to the operator if the tests detect problems with some MIDAS commands.

Alternatively, you can run the verifications plus all tutorials via the Midas procedure `superverify.prg`. You need to download the file `demo_data.tar.gz` with Midas test files from `/midaspub` on `ftp.eso.org` and create and fill a directory `demo/data` in e.g. `/midas`, and have the env. var. `MID_TEST` point to that directory, first:

```
% export MID_TEST=/midas/demo/data
% cd /midas/tmp
% inmidas
Midas 004> @ superverify
```

2.12 Cleaning MIDAS

If you are satisfied with the installation, you could clean the MIDAS directory and get some disk space by removing installation files or even packages of MIDAS that are not needed. A backup of the whole MIDAS directory is recommended before executing this step.

From the `config` menu, select the option `9 - clean MIDAS`. You will get a menu of choices, and for any option you select, confirmation will be requested before it starts removing files:

MIDAS DELETE MENU:

=====

- 1 - Delete only object files.
- 2 - Delete object and source files.
- 3 - Delete object, source files and libraries.
- 4 - Delete executable files.

Option 1 deletes all object files (.o) under the MIDAS release directory (upto 20% of MIDAS size) but they do not compromise MIDAS performance. It's the safest option.

Option 2 deletes all object and source files (.o .c and .for) under the MIDAS release directory (upto 45% of MIDAS size). They are not needed for MIDAS performance, however you will not be able to install new patches without sources. You still can compile and link your own MIDAS code.

Option 3, like Option 2, removes also all static library files (.a) (upto 55% of MIDAS size). Since MIDAS uses shared libraries the static ones are not required for development with MIDAS.

Option 4, removes only executables files (.exe) for the optional package categories: applic, stdred and contrib. The core of MIDAS remains, you only remove packages you do not need anymore. Source files remain so executables could be regenerated if necessary.

If what you really want is to reduce your MIDAS installation to the minimum but still operational expression, you could also use our *cleanmidas* script used to generate the only binaries releases of MIDAS (it is like Option 3 plus other unnecessary files). You use the script in this way:

```
% cd /midas/11SEP
% ./system/unix/cleanmidas
```

2.13 FITS Data Decompression on the Fly

Compressed FITS files on disk can also be accessed with the MIDAS command `intape/fits` (`indisk/fits`). Compressed FITS files are recognized by extra extensions(e.g `.z` or `.gz`) to their names. If `intape/fits` matches a filename with one of its expected compressed extensions, the file will be decompressed with the command indicated for such extension, e.g. `gunzip`). The decompression of the FITS file is done on-the-fly, that means that the decompress command is spawned in a piped mode, sending the decompressed data directly to the input of the `intape` command. This improves speed access and reduces disk space because it does not create temporary files.

The decompression code uses, by default, a *built-in* table of compressed extensions and decompress commands as indicated in the following table:

Built-in table:	
Extension	Command
.gz	zcat
.z	gunzip -c

The algorithm to apply the decompression is:

- If the file exists and the filename matches on the compressed extensions the file will be decompressed.
- If the file does not exist, the first file with a filename+extension found in the same directory will be decompressed.
- In any other case the file will be opened if it exists or an error will be returned if it does not exist.
- Whenever a decompress command is executed it will also be displayed in the standard output.

Example: In a directory with no `myfits.mt` file but where a `myfits.mt.gz` file exists, MIDAS will execute something like `"zcat myfits.mt.gz | INTAPE/FITS"`

```
Midas 001> $ls myfits*
myfits.mt.gz
Midas 002> INTAPE/FITS 1 xxx myfits.mt
zcat myfits.mt.gz
Image xxx0001 ...
Midas 003>
```

The *built-in* decompression table is ignored and overwritten by another user-defined table in a file pointed by the environment variable `DECOMPRESS_TABLE` if it exists, otherwise in a MIDAS-defined table located in the file `/midas/11SEP/incl/decompress.dat` if it exists. The syntax of these two files is described in the header of the `/midas/11SEP/incl/decompress.dat` file, which could also be used as a template.

2.14 Access to printer-devices in MIDAS

Printer-devices in MIDAS are used to print out text, graphics (plots) and images loaded onto the display window of MIDAS. In order to ease the installation a number of generic definitions of printer-devices used by MIDAS are defined in the file `/midas/11SEP/monit/devices.sh`, like:

LPRINT for the main line printer (text output only)
LASER for Postscript b+w laser printer
PLASER has to be the same as LASER !!
COLOUR for Postscript color laser printer
SLIDE for film recorder (slides)
PENPLOT for a pen plotter (HP Graphics Language format)

You have to edit this file and assign the printer system names to the logical names above (if existing at your site).

Apart from these logical names also system names for printers can be used. However, the use of printer-devices for graphics output (plots) involves some extra installation work. A separate chapter, Chapter 3 *Installation of Graphics Devices* describes in detail how extra devices for graphics can be added.

2.15 Line editing in MIDAS

The GNU `readline` is a public domain library widely supported on many UNIX platforms and enhances the line-editor capabilities of MIDAS like a history stack of commands, Emacs or Vi editing mode, command and filename completion functions and a communication channel with the MIDAS Graphical User Interface HELP for `help-on-line` functions.

The editing mode by default with GNU `readline` is Emacs but it can be switched to mode Vi by setting the variable `editing-mode` to `vi` in the initialization file `$INPUTRC` if exists otherwise in the file `$HOME/.inputrc`. As a brief introduction to GNU `readline` here is list of most common key-sequences:

Self-insert	CTR-L	->	clear-screen
CTR-E -> end-of-line	CTR-A	->	beginning-of-line
CTR-U -> unix-line-discard	CTR-K	->	kill-line
TAB -> complete	TABs	->	possible-completions
CTR-X -> (also F1)			on-line-help through the GUI HELP

More detailed information about editing commands with GNU `readline` in the man-page `/midas/11SEP/system/unix/readline.l` Please, note, that you need the development version of the ncurses library for editing.

2.16 Stand-alone MIDAS utilities

Chapter 5 `Tapeserver Installation` and Chapter 6 `Other MIDAS Utilities` describe the stand-alone MIDAS utilities, that is, programs that run, outside a MIDAS session, at

the system level. These are optional programs located in the `/midas/11SEP/util` directory and are not included in the standard installation of MIDAS.

2.17 Calibration and Demo Data for MIDAS

The distribution file containing the sources of MIDAS does not include any calibration data required by some packages or the demo data required with some tutorials. The reason for this is to reduce the distribution size of MIDAS via Internet to the essentially minimum required, leaving the retrieval of this bulky data to the discretion of the operator deciding which optional packages are installed.

In our *anonymous* and *midaspub* ftp accounts the calibration and demo data is located under the `./calib` and `./demo` subdirectories respectively.

Data is distributed in compressed-tar files. The whole calibration data is located in one single file called `calib_src.tar.gz` and demo data is offered in FITS format in `demo_data.tar.gz` for all tutorials available in MIDAS, e.g. *TUTORIAL/EXTRACT*, *TUTORIAL/FILTER*, *TUTORIAL/GRAPHICS*, *TUTORIAL/ITT*, *TUTORIAL/LUT*, *TUTORIAL/SPLIT*, *TUTORIAL/TABLE*, *TUTORIAL/ALIGN* and *TUTORIAL/FIT*.

Calibration and demo data are stored in FITS format, independent of the specific release, therefore they must be installed, not under the `/midas/11SEP` directory, but under `/midas/calib` and `/midas/demo` or equivalent. Tutorial and calibration procedures expect to find their data under these two directories.

For a proper installation of the demo data for all tutorials in MIDAS, e.g. in `/midas/demo/data` get the file `demo_data.tar.gz`, decompress and un-tar it in the `midas` directory.

```
% cd /midas/  
% tar -zxvf demo_data.tar.gz
```

2.18 MIDAS documentation

The file containing the sources of MIDAS does not include the MIDAS documentation. It is located in our *anonymous* ftp accounts under the subdirectory `./11SEP/sources/doc`.

All MIDAS documentation is classified in three volumes and distributed in compressed Postscript files or `.html` files:

```
volume_a.ps.Z - ESO-MIDAS Volume A: System  
volume_b.ps.Z - ESO-MIDAS Volume B: Data Reduction  
volume_a.html.tar.Z - ESO-MIDAS Volume A: System  
volume_b.html.tar.Z - ESO-MIDAS Volume B: Data Reduction
```

`env.ps.Z` - *ESO-MIDAS Environment*
`env.html.tar.Z` - *ESO-MIDAS Environment*

For details see the `README.doc` file in that directory.

Chapter 3

Installation of Graphic Devices

Below you will find a description of how to implement (new) graphic devices or how to make adjustments according to the local situation. Section 1 gives an overview of how the MIDAS plot system works. Section 2 describes the various files which play a role in the system. Finally, section 3 gives an example of how and where to make modifications in case a new device has to be added to the system.

The following, rather elaborate setup only makes sense if you install Midas for a group of users on a shared system. If you are installing Midas just on your own PC, this is probably overkill. Just use the system names for the devices. E.g. if your colour printer has the name `mycol` in your Linux system, use `mycol` also in the Midas commands. See the help of `copy/display` for examples of that.

3.1 The MIDAS Graphics System

All information about the status of the MIDAS plot package is stored in MIDAS keywords. This information describes the layout of the plot(s) to be produced, the last plot file created and the graphic device to which the result will be sent. Once a plot file (with extension `.plt`) is created the user can send this file to another device than the default one. These two possibilities (default device name stored in keyword or device name from user input) make life somewhat more complicated, since the user may use MIDAS device names rather than the system names.

If the setup of the system is correctly done MIDAS knows three sets of device names. The first set contains the names of the graphics terminal, the graphic windows, and the display windows. The second one contains the system device names of the hardcopy devices, or a subset of that. Finally, a third set of names contains logical names (MIDAS device names) for a few of the devices in the second set.

To solve this complication the names for the hardcopy devices stored in the MIDAS

keywords are identical to the system names. If a MIDAS device (logical) name is used, MIDAS will first convert this name into the system name. If no corresponding system name is found a fatal error will be given. Once the system name is established the plot system knows which of the drivers has to be used in order to produce the plot. If a window or terminal name is used, or an abbreviation of that, MIDAS will expand that name to its full name known by the plot system.

The assignment of the hardcopy devices as the output device is done in the procedure *plassign.prg*. Copying the plot-file onto a hardcopy device is done by the procedure *send-plot.prg*. Both procedures make use of the separate procedure *checkdev.prg* that checks device names and, if required, converts MIDAS logical names into system names. The system names of all devices should be included in the file *agldevs.dat* which can be found in the directory */midas/11SEP/systab/ascii/plot*. For each device the file also gives the driver that will be used for that device.

3.2 Modification or Installation of New Devices

Within the MIDAS directory structure you will find a directory */midas/11SEP/libsrc/agl* that contains all source code for building the lower level AGL library which the MIDAS plot system needs. The code can roughly be divided into device independent routines and the device drivers. Information about the first part can be found in the AGL References manual; the latter is described in the set of small device documents. However, this graphics directory is not relevant for changes in the graphics devices. For that purpose only minor modifications are needed elsewhere. They are in different subdirectories of the MIDAS directory tree.

Below, in a few small sections you will find what to do in order to install a new hardcopy graphics device.

3.2.1 The file *agldevs.dat*

MIDAS can handle a number of devices. The file *agldevs.dat* in the directory */midas/11SEP/systab/ascii/plot* defines which drivers have to be used for the various devices. In the system environment variable `AGL3CONFIG` points to this directory. The variable is defined in the file */midas/11SEP/monit/midlogs.sh*.

The format in the *agldevs.dat* file is simple: each system name is followed by a colon and then followed by a driver name. Comments are preceded by a hash (#) character. In this file you will first find a number of lines describing graphic terminals, graphics windows and display windows. Hereafter, you will find a number of lines describing the available hardcopy devices at ESO and their associated drivers. At the end in the Appendix A an example is given.

In case one wants to use a device that is not included in the *agldevs.dat* file the device

can be defined by the command:

```
% setenv AGL3DEV dev_driver # (C-Shell)
% AGL3DEV=dev_driver ; export AGL3DEV # (Bourne-Shell)
```

where `dev_driver` stands for the name of the device driver which AGL should use (provided it supports that device) when it is made to plot-file. A useful application of this possibility is if one wants to use a graphics terminal that is not of the default MIDAS type (VT640). The file *agldevs.dat* contains a listing of all available terminal drivers.

3.2.2 The file *devices.sh*

This file can be found in */midas/11SEP/monit* and contains the logical MIDAS names for a limited number of the available hardcopy devices. The format is described in the file. Make sure that the names in the file are unique. In addition, if names are added or modified also the procedure *checkdev.prg* may need to be modified (see below).

3.2.3 The procedure *checkdev.prg*

This procedure in the directory */midas/11SEP/prim/proc* converts MIDAS logical names into system names and checks these on their existence in the *agldevs.dat* table. Hence it relies on the complete and correct information in this file. In case you want to install (a) new MIDAS logical name(s) the procedure only needs minor modifications. Comments indicate where the changes have to be made.

3.3 Examples

3.3.1 Adding a hardcopy device (system name only)

Suppose one wants to add a hardcopy device with a system name `laserprinter` and suppose this device is a postscript device. The only thing that is needed is to include the system name in the *agldevs.dat* file:

```
laserprinter:pscript          # new device name
```

3.3.2 Adding a hardcopy device (MIDAS name and system name)

Suppose one wants to add the MIDAS device name `mdevice` to the system, which should point to the system name `laserprinter`.

The first step is the same as in the previous example: Add an extra line in the file *agldevs.dat*:

```
laserprinter:pscript          # new device name
```

The second step would be to include the MIDAS name and system name in the file *device.sh*:

```
mdevice=laserprinter; export mdevice
```

Finally, in the procedure *checkdev.prg* a few lines should be included in order to recognize the new logical name. The addition should be done in the section “MIDAS Logical names”. For this particular example these line should look like:

```
ELSEIF P1(1:3) .EQ. "MDE" THEN    ! here for mdevice  
  P1 = M$SYMBOL("MDEVICE")
```

In case of problems you could use the MIDAS command `DEBUG/PROCEDURE`. This enables you to execute the `ASSIGN/GRAPHICS` and the `COPY/GRAPHICS` command step by step and to find out where a device name is missing or misinterpreted.

Finally, whenever you modify your system please document this. E.g. add or modify some help documentation in the *assigngrap.hlq*, *copy_grap.hlq*, and the *printe.alq* files in */midas/11SEP/prim/help*.

Chapter 4

Other MIDAS Utilities (Optional)

4.1 bdf2ps: convert MIDAS-bdf files to Postscript format

- **Description:** The command `bdf2ps` reads a single MIDAS-bdf file or a specification file with a set of images and creates a Postscript file depending on the options.
- **Synopsis:**
`bdf2ps [-wbneavfN] [-i bdf-file] [-o ps-file] [-l spec-file] [-p page-format] [-r resolution] [-c lut] [-s xscale,yscale] [-z lcut,hcut]`
- **Installation:**

```
% cd /midas/11SEP/util/bdf2ps
% make
```

And, as superuser, copy the binary and man-page files to standard system directories (e.g.: `/usr/local/bin` and `/usr/man/man1`):

```
# cp /midas/11SEP/util/exec/bdf2ps.exe /usr/local/bin/bdf2ps
# cp /midas/11SEP/util/bdf2ps/bdf2ps.1 /usr/man/man1/bdf2ps.1
```

4.2 ltape: list headers of FITS format files on tape or disk

- **Description:** The command `ltape` reads files in FITS format from disk or tape and list their headers. Both a one line summary and a full listing are available. A number of comment fields can be provided and will be printed in the list header.
- **Synopsis:**
`ltape [-hmv] [-i input] [-o output-file] [-l list-file]`
- **Installation:**

```
% cd /midas/11SEP/util/ltape
```

```
% make
```

And, as superuser, copy the binary and man-page files to standard system directories (e.g.: */usr/local/bin* and */usr/man/man1*):

```
# cp /midas/11SEP/util/exec/ltape.exe /usr/local/bin/ltape
```

```
# cp /midas/11SEP/util/ltape/ltape.1 /usr/man/man1/ltape.1
```

Chapter 5

Panic Section and other services

- **ESO-MIDAS on the World Wide Web**

The ESO-MIDAS WWW pages can be reached using the ESO-MIDAS page:

`http://www.eso.org/sci/data-processing/software/esomidass`

This URL contains an overview of the MIDAS hard- and software requirements, distribution policy, documentation and support, and User Guides.

- **Panic Section**

In case of problems with the MIDAS installation please have a look at our MIDAS WWW Page. Helpful information can be found in the FAQs or in our problem reporting database. Especially chapter 2 and 3 of the MIDAS User's Guide Volume A contain a lot of information about how to start a MIDAS session and how to start writing procedures.

If you could not find any information related to your problem there, you can submit your problem via the web interface. The MIDAS problem reporting database can be accessed via the link **ProblemReportForm** on top of the MIDAS home page. Please fill out all fields on the form. This is the safest and fastest way to get help on your specific problem, because the system will automatically forward your problem to the right person.

Finally, if you cannot use the web form, you can also send an e-mail to

`midas@eso.org`

Appendix A

Graphic Devices

```
#
#+++++
#.Identification:
# /$MIDASHOME/$MIDVERS/systab/ascii/plot/agldevs.dat
#
#.Purpose:
# Defines the graphics devices used by MIDAS.
#
#.Description:
# Every line not starting with a hash (#) contains two entries, separated by
# a colon and a comment. The first entry is the system name for the device,
# the second one is the AGL driver name.
# All graphics output devices, except for Postscript printers, should be
# put into this file. This file is read in the plot package of Midas in order
# to check if a device is used. Some examples are given at the end.
#
#.Syntax:
# The syntax for each line is simple:
#     <device>:<driver>:<node>:=<command>           # comment
#     where:
#         <device> Is the identification of the physical device
#                 according to the rules of the operating system.
#                 For postscript devices the name may include '.l'
#                 or '.p' in order to indicate landscape or portrait
#                 orientation of the graph. Deafult (without) the
#                 extension is '.l'.
#                 The special device TT stands for the current
#                 user's terminal and it is meaningful only in the
```



```

#           user's local configuration file.
#           <driver> Is the identification of the corresponding driver
#                   as specified in the driver configuration module.
#                   Some drivers require further specifications to
#                   select specific characteristics of the device
#                   (this is device dependent) (e.g.: tkg.vt640).
#
#           <node>   Is the optional specification of the remote node
#                   (provided for future support of device in a
#                   distributed environment).
#
#           <command> Is the optional command to be executed when the
#                   device is closed. The = sign is required and is
#                   not part of the command. The % sign may be used
#                   within the command to indicate the place where
#                   the driver generated file name must be put into
#                   the command line.
#
#.Note:
# At ESO we only use the <device> and <driver> entries ***
# For further information please consult the documentation in the file
# /$MIDASHOME/$MIDVERS/doc/agl/aglinsta.doc
#
# 020228 last modif KB
#
#-----
#
# *** Here for the general terminal. Other terminal devices supported by AGL
#      and MIDAS are below.
graph_term:tkg.vt640    # default DEC VT100 terminal with retrographics
#
# *** Some examples for devices not available at ESO.
#graph_term:tkg.4010    # tektronix 4010
#graph_term:tkg.4014    # tektronix 4014
#graph_term:vt125       # DEC VT125 Terminal
#graph_term:vt125       # DEC VT240 Terminal
#graph_term:tkg.cit101  # CIT101 with CIG 201 card
#graph_term:tkg.hds22   # HDS 2200
#graph_term:tkg.wyse    # Wyse graphics terminal
#your_ln03:tkg.ln03    # your local LN03 plus laser printer (name your_ln03)
#exam_qms:tkg.qms      # your local qms laser printer (system name exam_qms)
#my_apple:pscript       # your Apple Laser Writer (system name my_apple)

```

```

#
# *** Here for workstations
#
graph_wnd0:idi.sxw0g      # IDI driver for workstation graphics window 0
graph_wnd1:idi.sxw1g      # IDI driver for workstation graphics window 1
graph_wnd2:idi.sxw2g      # IDI driver for workstation graphics window 2
graph_wnd3:idi.sxw3g      # IDI driver for workstation graphics window 3
graph_wnd4:idi.sxw4g      # IDI driver for workstation graphics window 4
graph_wnd5:idi.sxw5g      # IDI driver for workstation graphics window 5
graph_wnd6:idi.sxw6g      # IDI driver for workstation graphics window 6
graph_wnd7:idi.sxw7g      # IDI driver for workstation graphics window 7
graph_wnd8:idi.sxw8g      # IDI driver for workstation graphics window 8
graph_wnd9:idi.sxw9g      # IDI driver for workstation graphics window 9
#
image_wnd0:idi.sxw0       # IDI driver for workstation image window 0
image_wnd1:idi.sxw1       # IDI driver for workstation image window 1
image_wnd2:idi.sxw2       # IDI driver for workstation image window 2
image_wnd3:idi.sxw3       # IDI driver for workstation image window 3
image_wnd4:idi.sxw4       # IDI driver for workstation image window 4
image_wnd5:idi.sxw5       # IDI driver for workstation image window 5
image_wnd6:idi.sxw6       # IDI driver for workstation image window 6
image_wnd7:idi.sxw7       # IDI driver for workstation image window 7
image_wnd8:idi.sxw8       # IDI driver for workstation image window 8
image_wnd9:idi.sxw9       # IDI driver for workstation image window 9
#
# ***                      Hardcopy systems device names
# *** Here for the paper devices
ps2usr0:pscript          # the generic printer from symbol PLASER, LASER
ps2usr0.l:pscript.l
ps2usr0.p:pscript.p
#
# *** here for the colour postscript printers
pc2usr0.c:pscript.c      # the generic printer from symbol COLOUR
pc2usr0:pscript
pc2usr0.l:pscript.l
pc2usr0.p:pscript.p
#
# *** here for HP printers (HPGL)
ppbla28:hpgl.a          # Kyocera F-5000
#
# *** generic postscript device
postscript:pscript      # postscript device standard DIN A4 format

```

```

postscript.c:pscript.c # postscript device colour
postscript.l:pscript.l # postscript device DIN A4 landscape
postscript.p:pscript.p # postscript device DIN A4 portrait
postscript.u:pscript.u # postscript device US-legal standard format
postscript.ul:pscript.ul # postscript device US-legal landscape
postscript.up:pscript.up # postscript device US-legal portrait
postscript.ul:pscript.ul # postscript device US-legal landscape
postscript.up:pscript.up # postscript device US-legal portrait
postscript.4l:pscript.4l # postscript device A4 format landscape
postscript.4p:pscript.4p # postscript device A4 format portrait
postscript.3l:pscript.3l # postscript device A3 format landscape
postscript.3p:pscript.3p # postscript device A3 format portrait
#
# *** here for the null device
null:null # null device
#
# *** The following lines MUST remain the last ones of the file.
# No reading will be done beyond that point.
#*****
unknown:null # end of device file
#*****

```