Installation of MIDAS on VMS Systems

Image Processing Group,
European Southern Observatory,
Karl-Schwarzschild-Straße 2,
D-85748, Garching, FRG.

December 1992
Update Jan 1994 Update Oct 1994

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. At present, MIDAS is distributed once a year in November. The system is available for both VAX/VMS and UNIX systems, and is available to non-profit research institutes free of charge. This document describes the installation procedure for MIDAS on VMS 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.

For VMS systems MIDAS is distributed by ESO by:

```
- Anonymous ftp from ftphost.hq.eso.org.  Compressed TAR format - CD-ROM
ISO9660 with RockRidge extensions.  Compressed TAR format
```

and

```
- 1/2" Tape, 1600/6250 bpi density magtape, BACKUP format.
- 4mm Cartridge tape, DDS/DAT, BACKUP format.
```

from the following address:
Jean-Pierre de Cuyper
Koninklijke Sterrewacht van Belgie
Ringland 3
B-1180 Brussel

This document describes the installation of MIDAS on computer systems running under the VMS operating system. It contains all information needed to ensure a smooth installation of the MIDAS software.

This version of MIDAS is built on the following set of interfaces:

- for access to the data base the so-called "standard and tables interfaces" defined by ESO. For a description see the MIDAS Environment Manual, revision 1.1.

- for graphics applications the interfaces of the AGL package from ASTRONET are used. For a description see the AGL User's Guide.
- for image display applications the IDI interfaces are used. For a description see the IDI interface document.

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

Chapter 2

Main Installation

2.1 System requirements

Please check these requirements before starting the installation of MIDAS:

1. MIDAS requires a disk space of 55000 blocks to read the distributed backup tape, plus 120000 blocks more for a complete installation with all MIDAS packages available, on a single disk. This amount can be reduced by using the *8 - Clean MIDAS* option within the *CONFIGMIDAS* procedure.
2. Check that the commands *DELETE* and *PURGE* are not redefined as *DELETE/CONFIRM* and *PURGE/CONFIRM* respectively.
3. The installation of MIDAS uses the following logical names and symbols:

   - *MID_DISK* to refer to the disk which contains the MIDAS software.
   - *MIDASHOME* defining the main directory for MIDAS.
   - *MIDVERS* defining the current version.

   These definitions are absolutely necessary for the subsequent steps!

   For demonstration purposes let's assume that the disk for MIDAS should be `DBA2:` and the directory under which you want to install MIDAS is called `SOFTWARE`, then proceed as follows:

   ```
   $ ASSIGN DBA2:  MID_DISK:
   $ MIDASHOME :== SOFTWARE
   $ MIDVERS :== 96NOV
   ```

4. The following libraries are device dependent. They are not included in the tape but they must exist.

LIBNAGB.OLB the NAG mathematical library.

IP8LIB.OLB Interface DeAnza for IP 8000 series.

If they do not exist, *INSTALLMIDAS* can create an *IP8LIB.OLB* empty library and a *NAGLIB.OLB* dummy library.

If they exist the operator should copy these libraries to the *MID_LIB* directory or reassign their logical names in the file *MID_INSTALL:LIBDEF.COM*.

5. Then check the size of the system pagefile and swapfile plus some system parameters (ask your system manager for assistance):

   - *PAGEFILE.SYS* should have at least *25000* blocks
   - *SWAPFILE.SYS* should have at least *18000* blocks.
   - system parameter *VIRTUALPAGECNT 60000* recommended.
   - *page file quota* for the installation should be at least *40960*.
   - *CPU limit*. Set it unlimited for the MIDAS installation.

   If the system files are not large enough you might get problems in compiling some large programs (e.g. in the agl library directory...).

   Use *SYSGEN* to check the value of the parameter *VIRTUALPAGECNT*. This parameter should be set to a rather high value in order to support the mapping of large images in MIDAS. If you have to modify *VIRTUALPAGECNT* you must reboot the system after having changed the value. **NOTE:** Do not forget to write the modified parameter *VIRTUALPAGECNT* to disk (see the doc about *SYSGEN*) otherwise the change is lost next time you reboot the system!

   Use the command *MC AUTHORIZE* and subcommands *SHOW* and *MOD* to check and modify the page file quota of the account you are using when installing MIDAS.

2.2 Requirements for users of MIDAS

1. All users working with MIDAS need a rather high page file quota in the authorization file (e.g. **30000**) due to the use of the mapping mechanism of MIDAS.

2. Also users of MIDAS should have at least a CPU limit time of **30** minutes.

3. Finally, they need a *BYTLIM* of **16384** (however, we already had cases where even 16384 was not sufficient and lead to problems when trying to create the foreground subprocess, so it may be set even higher).

4. The following symbol definitions are needed by users to run MIDAS:

```
$ SETMIDAS :== @ MID_DISK:['MIDASHOME'.96NOV.MONIT]SETMIDAS.COM
$ INMIDAS :== @ MID_DISK:['MIDASHOME'.96NOV.MONIT]START.COM
```

With *INMIDAS* you start each session of MIDAS. With *SETMIDAS* you once set up the symbol *MIDOPTION* and the logical name *MID_WORK* which are used in each subsequent INMIDAS.

- *MIDOPTION:* may be *PARALLEL* or *NOPARALLEL* (the default).
  If *MIDOPTION* is set to *NOPARALLEL*, all intermediate MIDAS working files in the MIDAS startup directory are deleted when starting MIDAS via *INMIDAS*.
  If *MIDOPTION* is set to *PARALLEL*, no intermediate files are deleted. So, the *PARALLEL* option is necessary if you want to run several sessions of MIDAS simultaneously.
- *MID_WORK:* specifies the MIDAS startup directory. MIDAS creates all temporary files like FORGRxy.KEY in this directory. Additionally, all the personal procedures may be stored in *MID_WORK*, in particular the *LOGIN.PRG* file. It does not matter where you are when you start MIDAS via INMIDAS, the intermediate files will be stored in *MID_WORK* and the *LOGIN.PRG* will be searched for in *MID_WORK*. Furthermore, when executing a "private" MIDAS procedure it may reside in the current default directory or in *MID_WORK*.

It may be convenient to put these two definitions into the system *LOGIN.COM* file - see your system manager for help.

2.3 Default configuration of MIDAS

The official MIDAS distribution tape from ESO provides, by default, the following configuration:

- Ready to use the VAX/VMS C compiler. Not valid for OPEN-VMS, therefore if your system is an OPEN-VMS, therefore and as indicated in section *Modifying the default configuration of MIDAS* you will**HAVE** to select option 2 (Select C compiler) and then option 1 (VMS C Compiler) if you want to use the OPEN-VMS C Compiler.
- Ready to use the VAX/VMS C compiler (not for OPEN/VMS)
- X11 display (DECwindows or Motif).
- No optional packages. Only the core of MIDAS will be installed.

2.4 Create directories for MIDAS

Now you have to create the main directory *['MIDASHOME']* if it does not already exist and the subdirectory *[.96NOV]*. You might need system privileges to do this. Type:

```
$ SET DEF MID_DISK:[0,0]
$ CREATE/DIR ['MIDASHOME']
$ SET DEF MID_DISK:['MIDASHOME']
$ CREATE/DIR [.96NOV]
```

and if you also want a directory for the demos and test data (highly recommended):

```
$ CREATE/DIR [.DEMO]
$ SET DEF MID_DISK:['MIDASHOME'.DEMO]
$ CREATE/DIR [.DATA]
```

2.5 Retrieving files from the MIDAS tape.

Mount the tape on, for instance tape unit *MTA2:* and type

```
$ SET DEF MID_DISK:['MIDASHOME']
$ MOUNT/FOR MTA2:
$ BACKUP/LOG MTA2:MIDAS.BCK MID_DISK:['MIDASHOME'...]*.*
$ DISMOUNT/UNLOAD MTA2:
```

2.6 Retrieving files from the MIDAS ftp account.

Our *midas* ftp account only accept connections from previously registered sites. Please, contact Resy de Ruijsscher, email address *resy@eso.org*, to activate your site and to provide you with the corresponding password.

```
$ SET DEF MID_DISK:['MIDASHOME']
$ FTP ftphost.hq.eso.org (134.171.6.9)
FTP server...  ready.
FTPHOST.HQ.ESO.ORG> user midas
FTPHOST.HQ.ESO.ORG> password ??????
FTPHOST.HQ.ESO.ORG> bin
FTPHOST.HQ.ESO.ORG> cd 96NOV/vms
FTPHOST.HQ.ESO.ORG> get README
```

and follow the instructions of the README file.

2.7 Reading the MIDAS compressed-tar file

Independently from tape or ftp you will get at least the following files:

```
TAR_VMS.EXE
COMPRESS_VMS.EXE
TAR_OPEN_VMS.EXE
COMPRESS_OPEN_VMS.EXE
96NOVPLX.TAR_Z
```

If you are on a OPEN/VMS machine, execute:

```
$ TAR :== $ MID_DISK:['MIDASHOME']TAR_OPEN_VMS.EXE
$ COMPRESS :== $ MID_DISK:['MIDASHOME']COMPRESS_OPEN_VMS.EXE
```

otherwise, execute:

```
$ TAR :== $ MID_DISK:['MIDASHOME']TAR_VMS.EXE
$ COMPRESS :== $ MID_DISK:['MIDASHOME']COMPRESS_VMS.EXE
```

To decompress and untar the MIDAS tar file, type:

```
$ COMPRESS -D 96NOVPLX.TAR_Z
$ TAR XVF 96NOVPLX.TAR
```

Execute the DCL file that defines some MIDAS logical files:

```
$ @ ['MIDASHOME'.96NOV.MONIT]MIDLOGS.COM
```

2.8 Setup of MIDAS

In order to adjust MIDAS to your local hardware configuration, you have to make the following modifications:

1. You must update the symbols *MIDASHOME* and *MIDVERS* in the files *MID_MONIT:IDISTART.COM* and *MID_MONIT:IDINSTALL.COM*
2. Edit the file *MID_MONIT:START.COM* and update the symbols *MIDASHOME* and *MIDVERS*, then and if you have a VAX cluster, modify the cluster node-name, and edit the file *STARTB.COM* as well. single node section and comment out the cluster section.

3. In the file *MID_MONIT:STARTA.COM* you will have to modify the set up of the work stations: *i.e.* changing the names of the terminals to test if you are sitting in front of an image display (stations A0, A1, B0 and B1) in case of fixed terminal lines. If terminal lines are assigned via a PBX / LAN system it is not possible to detect the physical location of a user from the terminal name. In that case the test for stations A0,... has to be done through asking for the MIDAS unit (in the same way as for the "dummy" unit).

4. The file *MID_MONIT:DEVICES.COM* holds the generic MIDAS names for the print/plot procedures on the default devices used by the MIDAS system, which are:

```
LPRINT       for printing on the main line printer
VERSA        for printing on an electro-static plotter
PENPLOT      for printing on a pen plotter
PLASER       for printing on Postscript black+white laser
COLOUR       for printing on Postscript color laser
SLIDE        for printing on PostScript slide (film) recorder
LASER        device name for PostScript black+white laser
TAPE0,TAPE1  for tape devices
```

   You have to edit this file in order to assign a MIDAS name to the default device (only one) for each device type existing at your site. All other devices of the same type (*e.g.* other Postscript Laser printers) have to be accessed via their usual "system" name. The format for these updates is explained in DEVICES.COM.

5. Edit the file *MID_HELP:PRINTE.ALQ* to reflect the setup of the printers and plotters at your site.

6. Finally edit the file *MID_HELP:TAPEDE.ALQ* to reflect tape devices at your site.

2.9 Modifying the default configuration of MIDAS

If your system is ready for the default installation, you can proceed with the next section.

If you want to modify the default configuration, you can use the user-friendly procedure *CONFIGMIDAS*. The following tasks can be invoked:

```
- MIDAS C compiler.
- MIDAS display configuration.
- MIDAS software configuration.
- MIDAS installation.
- MIDAS remove procedure.
```

```
- MIDAS help.
```

One possible sequence of how to use *CONFIGMIDAS* could be:

```
select: 1       List of MIDAS packages available:
                All optional packages will be listed,
                and a brief description can be displayed.
select: 2       Select C compiler: VAX/VMS or GNU C compiler.
select: 3       Select display: Deanza, X11 etc...
select: 4,5,6   Select only CORE, ALL or those packages of
                MIDAS of your own selection.
select: 7       Run INSTALLMIDAS.
select: 8       Clean MIDAS: To remove unnecessary files.
select: q       QUIT.
```

NOTE: See *['MIDASHOME'.96NOV.DOC.INSTALL]GCCVMS.PROBLEM* file if you want to use the GNU GCC compiler.

NOTE: You **HAVE** to run option 2 (Select compiler) and then option 1 (VMS C compiler) if your system is an OPEN-VMS machine.

2.10 Installation of MIDAS

Before starting the installation of MIDAS, please check again the section *system requirements* to avoid installation problems later.

To execute the installation procedure of MIDAS type:

```
$ SET DEF MID_INSTALL
$ @ CONFIGMIDAS
select:  7 (run INSTALLMIDAS)
```

INSTALLMIDAS will execute the following steps:

1. It checks first some requirements like the existence of configuration files, and libraries described in the section *system requirements*.
2. It will ask you whether you want the results to be written into the file *INSTALLMIDAS.LOG* or if they will be sent to the screen.
3. Then it will read the file *MID_INSTALL:PREINSTALLMIDAS.CNF* and will execute the command *PREINSTALL.COM* of each directory defined in this file.

4. Finally it will read the file *MID_INSTALL:INSTALLMIDAS.CNF* and will execute the command *MAKEFILE.COM* of each directory defined in this file.

5. INSTALLMIDAS will first compile all libraries belonging to the core of MIDAS, then it will compile and link the MIDAS monitor, and finally it will compile and link *PRIM* packages, then *APPLIC*, *STDRED* and *CONTRIB* packages only if they were selected.

If you have DECWindows or Motif you have to install the module *IDISERV.EXE* with certain privileges on each and every (!) VAXStation/node you want to use MIDAS on. All this is done when executing the command file *IDINSTALL*, but you need the VMS system privilege CMKRNL to be able to do it! Ask your system manager for help!:

```
$ SET DEF MID_DISK:['MIDASHOME'.96NOV.MONIT]
$ @ IDINSTALL
```

NOTE: This has to be done each time the system is rebooted (and for all systems/workstations MIDAS is running on). Therefore, it might be a good idea to add the relevant lines (see *IDINSTALL.COM*) to the system startup procedure (see your system manager).

2.11 Running MIDAS for the first time

1. Go to a "data" directory (e.g *[DEMO.DATA]* created in step 2.4), and execute:

   ```
   $ SET DEF MID_DISK:['MIDASHOME'.DEMO.DATA]
   $ INMIDAS
   ```
   (answer "zz" as MIDAS unit)

2. Once you are in MIDAS, execute the procedure *MID_PROC:COMPILE.ALL* in order to create the file *MID_PROC:INTERNAL.COD*. This file, which is very important, contains the "precompiled" MIDAS procedures (The ones which are executed via the " @% " command). Make sure that, once it is created, it has RWE protection for everybody!:

   `Midas 001> @ compile.all` (this command will bring you back to the DCL level)

3. Create the necessary LUT AND ITT files:

   ```
   $ GOMIDAS
   Midas 002> @ ascii_bin
   Midas 003> bye
   ```

4. If you have DECWindows run the XWindow test program *XTEST.EXE* which will display info about the capabilities of your display station.

```
$ SET DEF MID_DISK:['MIDASHOME'.96NOV.SYSTEM.IDISERV.SRC]
$ RUN XTEST
```

5. Finally and very important, you have to make sure the protections for the MIDAS files and directories are correctly set:

   To start with, all directories and files should have the protection RE (Read and Execute) for the world!

   The following procedure set the protections on all necessary MIDAS files:

```
$ @ MID_MONIT:MIDPROT
```

6. Now we are ready to go: Type *INMIDAS* and enter *00, 01, ..., 99* as MIDAS unit, if you want image display capability or enter *XA, ..., ZZ* for normal dummy MIDAS unit.

2.12 Verifying MIDAS

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

```
$ SET DEF MID_DISK:['MIDASHOME'.DEMO.DATA]
$ INMIDAS
(answer "00" as MIDAS unit)
Midas 001> @ vericopy
```

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

```
Midas 002> @@ veriall
```

If you want to run a particular verification procedure, type:

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

where *XX* can be *1,3,4,5,6,7,t1,t2 or dio*

The verification procedures are dessign 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 MIDAS commands.

NOTE: Distributed with the official tape and under ftp is now included a verification form. We kindly invite you to fill this form once you have finished the installation and verification of MIDAS and return it to the address written there (or via email by using the ASCII format file in *['MIDASHOME'.96NOV.doc.install]verification.doc*)

2.13 After the installation of MIDAS

That's it folks, to be on the safe side you should now backup the complete MIDAS system on tape:

```
$ SET DEF MID_DISK:['MIDASHOME'.96NOV.SYSTEM.VMS]
$ @ MIDSAVE
```

To reduce the total size of disk space used by MIDAS you could execute the option *8 - Clean MIDAS* of the *CONFIGMIDAS* procedure:

```
$ SET DEF MID_INSTALL
$ @ CONFIGMIDAS
select:  8 (clean MIDAS)
```

by selecting this you will get another menu:

```
          MIDAS DELETE MENU:
   ==============================
   1 - Delete source files.
   2 - Delete source files and libraries.
   3 - Delete documentation.
   4 - Delete executable files.
   5 - Purge all files.
   q - Quit.

 Select:
```

All selections will ask you for confirmation. Option 4 will remove all executable files created in the installation of MIDAS, therefore think twice before choosing it. Options 1 and 2 delete sources but they are not needed once the installation is finished (libraries are not necessary unless you want to use them to link your own applications with MIDAS).

Chapter 3

Installation of the 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 devices has to be added to the system.

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 to 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 a graphics device as the output device is done in the procedure *PLASSIGN.PRG*.Copying the plotfile onto a hardcopy device is done by the prodecure *SENDPLOT.PRG*. Both procedures make use of the separate procedure *CHECKDEV.PRG* that checks to device names and, if required, converts MIDAS logical names into system name. The system names of all devices should be included in the file *AGLDEVS.DAT* which can be found in the directory *['MIDASHOME'.96NOV.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 *['MIDASHOME'.96NOV.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 *['MIDASHOME'.96NOV.SYSTAB.ASCII.PLOT]* defines which drivers has to be used for the various devices. In the system environment variable *AGL3CONFIG* points to this directory. The variable is defined in the file *['MIDASHOME'.96NOV.MONIT]MIDLOGS.COM*.

The format in the *AGLDEVS.DAT* file is simple: each system name is followed by 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 C 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:

`% ASSIGN` *dev_driver* `AGL3DEV`

where *dev_driver* stands for the name of the device driver which AGL should use (provided it support that device) when it is made to plotfile. 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.COM*

This file can be found in *['MIDASHOME'.96NOV.MONIT]* and contains the logical MIDAS names for a limited number of the available hardcopy devices. The format is described in the file. Take care that the names in the file are unique. In addition, if names are added or modified also the procedures *CHECKDEV.PRG* may need to be modified (see below).

3.2.3 The procedure *CHECKDEV.PRG*

This procedure in the directory *['MIDASHOME'.96NOV.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.COM*:

```
mdevice :== laserprinter
```

Finally, in the procedures *SENDPLOT.PRG* and *PLASSIGN.PRG* a few lines should be included to recognize the new device. 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 missed or misinterpreted.

Finally, whenever your 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 *['MIDASHOME'.96NOV.PRIM.HELP]*.

Chapter 4

Panic Section

If you have DECWindows, experience has shown that the MIDAS commands *CREATE/DISPLAY* and *CREATE/GRAPH* are most likely to give you trouble. For example it might abort with an error message like

```
SYSTEM - F - Invalid device name ...
```

In that case check the following:

- the module *IDISERV.EXE* is installed at this CPU (use *$ INSTALL* and the option *LIST* in that utility to do that).
- the symbols *MIDASHOME* and *MIDVERS* are correctly set in *MID_MONIT:IDISTART.COM*.

In case of problems with the MIDAS installation use the Hot-line service either via phone: +49 89 320 06 456 or via electronic mail:

ESO::MIDAS (via SPAN)

midas@eso.org (via Internet)

MIDAS@DGAESO51 (via EARN/BITNET).

Chapter 5

Graphic Devices

```
# /$MIDASHOME/$MIDVERS/systab/ascii/plot/agldevs.dat
#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#            This file defines the graphics devices used by MIDAS
#
# 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.
# In principle all graphics units, except for terminals, should be put into
# this file. This file is read in the plot package in order to check if a
# valid device is used. For those who need to install devices not used at ESO,
# a number of examples are given at the end.
#
# 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.
#                       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.
#
# *** However, 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
#---------------------------------------------------------------------------
#
# *** 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
#graph_term=tkg.cit101   # CIT101 with CIG 201 card
#graph_term=tkg.hds22    # HDS 2200
#graph_term=tkg.4010     # tektronix 4010
#graph_term=tkg.4014     # tektronix 4014
#graph_term=tkg.cit101   # CIT101 with CIG 201 car
#graph_term=vt125        # DEC VT125 Terminal
#graph_term=vt125        # DEC VT240 Terminal
#
# *** Here for the default image display (e.g. Deanza)
image_displ:idi.sxw0     # default image display
#
# *** 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
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
#

# *** Here for the hardcopy systems device names
ps2usr0:pscript          # AGFA P400
ps2usr0.l:pscript.l      # AGFA P400
```

```
ps2usr0.p:pscript.p      # AGFA P400
ps2usr1:pscript          # HP LaserJet III
ps2usr1.l:pscript.l      # HP LaserJet III
ps2usr1.p:pscript.p      # HP LaserJet III
ps2vlt1:pscript          # HP LaserJet IIID
ps2vlt1.l:pscript.l      # HP LaserJet IIID
ps2vlt1.p:pscript.p      # HP LaserJet IIID
ps3vlt0:pscript          # AGFA P3400
ps3vlt0.l:pscript.l      # AGFA P3400
ps3vlt0.p:pscript.p      # AGFA P3400
ps4ast0:pscript          # HP LaserJet IIISi
ps4ast0.l:pscript.l      # HP LaserJet IIISi
ps4ast0.p:pscript.p      # HP LaserJet IIISi
ps4ipg0:pscript          # Apple Laserwriter
ps4ipg0.l:pscript.l      # Apple Laserwriter
ps4ipg0.p:pscript.p      # Apple Laerwriter
pc2usr0:pscript.c        # Tektronix Phaser, paper
pc2usr1:pscript.c        # Tektronix Phaser, overhead
sl2usr0:pscript.c        # Agfa Chromascript
hp2vlt2:hpgl.a           # Kyocera F-500
hp2usr0:hpgl.a           # HP 7550A penplotter, A4
hp2usr1:hpgl.c           # HP 7550A penplotter, A3
hp2usr3:hpgl.c           # HP 7550A penplotter, A3
versa:raster.200         # MIDAS default name for the Versatec plotter
#
# *** Below a number of examples for HARDCOPY device not available at ESO.
#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 the null device
null:null                # null device
unknown:null             # end of device file
```