MIDAS 17.00
A sytem for the interactive analysis of data from the Voyager Planetary Radio Astronomy Experiment

D. R. Evans

Radiophysics, Inc.
7912 Fairview Rd.
Boulder
CO 80303

devans@orion.colorado.edu

January 14, 1995

Contents

1 Introduction

This document describes MIDAS ("Microcomputer Interactive Data Analysis System") version 17.00 for the display and analysis of Voyager Planetary Radio Astronomy ("PRA") data.

1.1 Brief History

MIDAS was originally conceived in 1984, with the availability of the first microcomputers capable of replacing the minicomputers that had been used for PRA analysis until that time. These were NEC APC computers[1] running the CP/M-86 operating system. The early versions of MIDAS ran exclusively on these machines, and were written in Borland's Turbo Pascal. By the time of the Voyager 2 encounter with Uranus in January of 1986, MIDAS was sufficiently stable and capable that it was used by the Boulder PRA group to plot and analyze data in near real time during the encounter period.

Since then, the program has gone through many variations and upgrades, including two nearly complete rewrites, the first in 1987 and 1988 when it was converted to run under SunOS and recast in C, and the second in 1991 when, after many new features had been added in C++, the entire program was redesigned in terms of C++ objects.

The current version, MIDAS 17.00, is expected to be the final major version. It runs on PC class machines running MS-DOS or Windows, Sun workstations running SunView, and Apple Macintoshes running System 7. The program is relatively easily extensible to other systems; new modules to provide increased capabilities on all platforms can also be added relatively easily.

The program is divided into two independent modules. The first, called "core" code, is completely independent of hardware, and provides the basic features of the program; the second, the "display" code, varies from platform to platform. On some platforms (*e.g.*, MS-DOS) the display code is relatively simple and short, and represents only a tiny fraction of the complete program; on other platforms (*e.g.*, Macintosh) the display code constitutes the majority of the program.

[1]A computer demonstrably superior to the IBM PC, but which failed for lack of marketing skills.

This manual specifically pertains only to the non-Macintosh versions of MIDAS, although much of what is described here also pertains to that system. The display code for the Macintosh port was authored by Nick King[2]. Specific online help is available for the Macintosh version. Other versions do not have more than the simplest help online[3], and this document should be used for detailed reference. There is also a seperate Macintosh manual, in binhexed Word format, located in the file MACHELP.DOC.

1.2 Hardware requirements

The program has been tested under the following configurations. It is possible that it will work on less well equipped machines.

UNIX Sun workstation running SunOS 3.5 or later, with the SunView windowing system and a PostScript output device;

MS-DOS PC-compatible with 640K of memory running MS-DOS version 3.3 or later. The distributed binary runs only on 386 class machines or better, although the source should compile for machines of lesser power. A colour VGA card and monitor is necessary. For hardcopy, a PCL printer (*e.g.*, Hewlett Packard laser printer or compatible) is needed. A mouse is neither necessary nor used.

Windows Windows 3.1 or later, with a display of 800x600 or better and a mouse. The distributed binary runs only on 386 class machines or better, although the source should compile for machines of lesser power. For hardcopy, any Windows compatible printer may be used.

1.3 Memory management

MIDAS contains its own memory management routines (in various forms) to handle large internal arrays. In MS-DOS, a virtual memory scheme is provided so that very large data files can still be handled correctly even within the 640K MS-DOS limit. XMS and EMS are neither used nor needed nor supported. A less complicated scheme is used in Windows, and MIDAS

[2] reachable at nick@xvt.com

[3] because the author vastly prefers paper help to online so-called help

relies on the availability of space in the Windows swap file space to handle large arrays. In UNIX, MIDAS uses the default memory allocation scheme to handle large arrays. Large but not atypical (25MB) datafiles should be usable under any system, although considerable computing power is necessary for the initial startup indexing of such files to proceed in a reasonable amount of time.

1.4 Testing

In a package that runs on several platforms and that can handle and display such a wide variety of formats it has not been proved possible to test exhaustively all possible combinations of platform, filetype and display format. However, the program has been used in a production mode for a considerable period, especially in the SunView and Windows environments, and all known bugs have been removed. Such bugs as remain are most likely in individual display modules, and are probably easy to fix by a review of the equivalent code for other platforms; it is believed to be unlikely[4] that severe bugs remain in the core code.

1.5 What is provided

The full source, warts and all, is provided as part of this package. Executables for MS-DOS and Windows (along with the required redistributable Windows Dynamic Link Libraries ("DLLs") are also included.

To compile the source, the following compiliers are needed:

UNIX GNU C++. Any extant[5] version should work. The code does not require the C++ exception handling mechanism (see section 1.7 below); it does require the template mechanism.

Windows The compiler used to compile the distributed binaries was the Borland C++ compiler version 4.02. The code *will not* compile under compilers from competing software vendors, since it uses the Borland

[4] although, of course, possible
[5] *i.e.*, relatively recent

Object Windows Library ("OWL")[6]. Versions of the Borland C++ compiler prior to version 4.0 will not compile the code as it stands, as the Windows version of MIDAS is built on OWL 2.0, which was first supported with version 4.0 of the Borland compiler. The MIDAS source code can be compiled under version 4.02 of this compiler so as to function on a range of different target systems, including Windows NT. The distributed Windows binary *includes* full debugging information.

MS-DOS The compiler used was the Borland C++ compiler version 4.02. The only Borland specific code pertains to the Borland Graphics Interface ("BGI") and, if this small section of code is changed appropriately, the code should be compilable under compilers from competing vendors. It should compile unchanged under versions of the Borland compiler prior to 4.0. The distributed MS-DOS binary *includes* full debugging information.

Macintosh The Macintosh MPW workbench was used for the Macintosh coding effort. The distributed Macintosh binary *does not include* debugging information.

1.6 Packing List

MIDAS 17.00 is packaged as a pair of files in standard tar format. One file, MID17SRC.TAR contains only source code; the other file, MID17EXE.TAR contains binary code[7].

The file MID17SRC.TAR contains the following files[8]:

bdr.cc Class to handle data files in PDS browse data format [C];

bdr.h Header file for class to handle data files in PDS browse data format [C];

[6]With hindsight, it is now clear that this was a mistake, and the Microsoft Foundation Classes ("MFC") would have been a far better choice; however, the early versions of the Borland compiler seemed superior to those of Microsoft, it was only later that the reverse appeared to be the case, by which time a conversion to MFC would have been both costly and time consuming.

[7]Macintosh files contain the data fork only

[8]Files are marked as being either "C"ore code or "D"isplay code

binned_e.cc Class to bin low rate data files [C];

binned_e.h Header file for class to bin low rate data files [C];

canvas31.cc Windows 3.1 specific code [D];

canvas31.h Header file for Windows 3.1 specific code [D];

colour.cc Colour manipulation class for Sun [D];

colour.h Header class for Sun colour manipulation class [D];

config.dos Example MS-DOS configuration file [D];

config.mac Example Macintosh configuration file [D];

config.sun Example SunView configuration file [D];

config.win Example Windows configuration file [D]

data.cc Abstract base classes to handle PRA data [C];

data.h Header file for abstract base classes to handle PRA data [C];

defines.cc Globally useful deinitions [C];

defines.h Header file for globally useful definitions [C];

defs_dos.cc MS-DOS versions of useful definitions [D];

defs_dos.h MS-DOS header file for useful definitions [D];

defs_mac.cc Macintosh versions of useful definitions [D];

defs_mac.h Macintosh header file for useful definitions [D];

defs_sun.cc UNIX versions of useful definitions [D];

defs_sun.h UNIX header file for useful definitions [D];

defs_win.cc Windows versions of useful definitions [D];

defs_win.h Windows header file for useful definitions [D];

dosunix UNIX script file to change filenames to long format [D];

drestrin.cc String manipulation class[9] [C];

drestrin.h Header file for string manipulation class [C];

edr.cc Classes to handle EDR files [C];

edr.h Header file for class to handle EDR files [C];

event.cc SunView event handling class [D];

event.h Header class for SunView event handling class [D];

gf.cc Class to handle grey-scale rectangles used in spectral scans [C];

gf.h Header file for class to handle grey-scale rectangles used in spectral scans [C];

gf_lw.cc Laserwriter (PostScript) specific code for grey-scale rectangles [D];

gf_lw.h Header file for Laserwriter (postScript) specific code for grey-scale rectangles [D];

gf_mac.cc Macintosh specific code for grey-scale rectangles [D];

gf_mac.h Header file for Macintosh specific code for grey-scale rectangles [D];

gf_sun.cc SunView specific code for grey-scale rectangles [D];

gf_sun.h Header file for SunView specific code for grey-scale rectangles [D];

gf_vga.cc MS-DOS specific code for grey-scale rectangles [D];

gf_vga.h Header file for MS-DOS specific code for grey-scale rectangles [D];

gf_win.cc Windows specific code for grey-scale rectangles [D];

gf_win.h Header file for Windows specific code for grey-scale rectangles [D];

[9] Note that the new standard C++ string class was not used because the Macintosh compiler did not support it.

gmdr.cc Class to manipulate data records that have been converted to a genereic internal format [C];

gmdr.h Header file for class to manipulate data records that have been converted to a generic internal format [C];

gui_mac.cc Code for the Macintosh interface [D];

gui_mac.h Header file for code for the Macintosh interface [D];

headers_.h Macintosh header file [D];

high_rat.cc Class to manipulate high rate data frames [C];

high_rat.h Header file for class to manipulate high rate data frames [C];

icon.h Windows MIDAS icon [D];

ids_mac.h File of #define statements for the Macintosh [D];

list.cc Class to manipulate lists [C];

list.h Header file for class to manipulate lists [D];

mac_mida.mak Macintosh MPW script to build MIDAS from source [D];

magtape.cc Class to handle magnetic tape drive on Suns [D];

magtape.h Header file for class to handle magnetic tape drive on Suns [D];

makefile.dos MS-DOS makefile to build MIDAS from source [D];

makefile.sun UNIX makefile to build MIDAS from source [D];

makefile.win Automatically generated Windows makefile to build MIDAS from source [D];

menu.h Header file for SunView menu handling class [D];

mid17.ps This file in PostScript format [D];

mid17.tex This file in LaTeXformat [D];

panel_it.cc Class to handle items inside SunView panels [D];

panel_it.h Header file for class to handle items inside SunView panels [D];

pdsdr.cc Class to handle PDS format records [C];

pdsdr.h Header file for class to handle PDS format records [C];

resource.h Windows file to define resource IDs [D];

resource.r Macintosh resource file [D];

set.cc Class to manipulate sets [C];

set.h Header file for class to manipulate sets [C];

sm9.cc The MIDAS driver program [C];

sm_batch.cc Abstract base class for batch mode [C];

sm_batch.h Header file for the base batch mode class [C];

sm_mac_m.cc Macintosh specific operations [D];

sm_mac_m.h Header file for Macintosh specific operations [D];

sm_mode.cc Generic MIDAS operations [C];

sm_mode.h Header file for generic MIDAS operations [C];

sm_resou.h Windows resource file [D];

sm_sun_b.cc Batch mode on Suns [D];

sm_sun_b.h Header file for batch mode on Suns [D];

sm_sunvi.cc SunView specific operations [D];

sm_sunvi.h Header file for SunView specific operations [D];

sm_vga_m.cc MS-DOS specific operations [D];

sm_vga_m.h Header file for SunView specific operations [D];

sm_w_bat.cc Windows batch mode [D];

sm_w_bat.h Header file for Windows batch mode [D];

sm_w_mod.cc Windows specific operations [D];

sm_w_mod.h Header file for Windows specific operations [D];

surf_lw.cc Laserwriter (PostScript) graphical surface [D];

surf_lw.h Header file for laserwriter (PostScript) graphical surface [D];

surf_mac.cc Macintosh graphical surface [D];

surf_mac.h Header file for Macintosh graphical surface [D];

surf_sun.cc SunView graphical surface [D];

surf_sun.h Header file for SunView graphical surface [D];

surf_vga.cc MS-DOS graphical surface [D];

surf_vga.h Header file for MS-DOS graphical surface [D];

surf_win.cc Windows graphical surface [D];

surf_win.h Header file for Windows graphical surface [D];

surface.cc Abstract logical surface [C];

surface.h Header file for abstract logical surface [C];

t_ar_dos.h MS-DOS templates [D];

t_ar_mac.cc Macintosh template emulation [D];

t_ar_mac.h Header file for Macintosh template emulation [D];

t_ar_sun.h UNIX templates [D];

t_ar_win.h Windows templates [D];

t_array.h Generic templates [C];

timeclas.cc Class to manipulate times [C];

timeclas.h Header file for class to manipulate times [C];

utils_ma.cc Macintosh utilities [D];

utils_ma.h Header file for Macintosh utilities [D];

window.h SunView windowing class [D];

wmidas.def Windows definition file [D];

wmidas.rc Windows resource file [D].

The file MID17EXE.TAR contains the following files:

bc402rtl.dll Borland run time library;

bids402.dll Borland dynamic link library;

machelp.doc Macintosh help (in Word format), binhexed.

mid17.dvi This file in DVI format;

midas.exe MS-DOS executable;

midas.pif Sample Windows PIF file to run MIDAS in a DOS box;

mmidas Macintosh executable, data fork only;

wmidas.dsw Borland desktop file;

wmidas.exe Windows executable;

wmidas.ide Borland IDE file;

wmidas.prj Borland project file;

owl202.dll Borland OWL DLL;

1.7 The C++ Language

The C++ language represents somewhat of a moving target since there is, even now, no standard definition of the language. The author had the benefit of being a member of the ANSI committee during the period when the greatest changes were being introduced, which materially aided the design of MIDAS in C++ insofar as decisions regarding portability were concerned.

The version of the language as used in MIDAS represents the state of the language in mid 1992. To this base, templates have been added, but they are not specifically required for compilation. Templates help to reduce the quantity of coding necessary but, unfortunately, templates were not supported by the Macintosh compiler, and so have been used only sparingly and, wherever they are used, it is in the context of a conditional compilation. The Macintosh compiler also did not support exception handling. This was a greater loss and could not be circumvented by conditional compilation, as the availability of a robust exception handling mechanism would have required deep changes in the structure and flow of logic in the program, changes that cannot simply be produced by conditional compilation. I would have vastly preferred to have used the C++ exception handling mechanism, but there was no way that it could be done while still keeping a single set of core code that would run on all machines unchanged.

All the C++ compilers used had their quirks and differences[10]. Workarounds have been incorporated in the code, usually to the accompaniment of a more or less scathing comment. Most of these quirks[11] were found to be in the Macintosh compiler, but neither of the other compilers was immune.

2 Using MIDAS

2.1 PRA data formats

MIDAS can interpret almost any reasonable format of PRA data. These include:

[10]One hesitates to use the term "bugs" in referring to a language that is not yet completely specified.

[11]in the sense that the other two compilers agreed on one behaviour, while the third disagreed

- JPL EDR format low rate data with fixed record length of 1872 bytes;
- JPL EDR format low rate data with variable record length;
- PDS low rate ASCII browse format;
- PDS low rate binary browse format;
- PDS low rate non-browse format;
- JPL high rate format;
- PDS high rate format.

The only files in the above formats that may not be interpreted correctly by MIDAS are ones containing some of the very earliest, esoteric formats used for the pre- and post-launch instrument tests such as, for example, BC-1.

One format that is specifically *not* included is what used to be called DEDR[12] format. This would be simple to add, but since DEDR format was last used in Boulder in 1981, there seemed little point in including it. Additionally, the code would have been difficult to test as few readable tapes from that era are still available in Boulder[13]. Finally, the DEDR format was a simple one-to-one mapping of the JPL EDR data. For every DEDR tape there should exist a corresponding EDR tape which *can* be read by MIDAS.

2.2 Plotting modes

Data can be plotted in a variety of common PRA styles. These include:

- Spectral scans;
- Single channel plots ("tomplots");
- Raw point-by-point plots;
- Low resolution high rate;

[12] For "Decalibrated" EDR. I refrain from speculating on this unnecessarily coined word, which is used in a context where it means precisely the opposite of what it says.

[13] In fact, as of this date, the number of such is precisely zero.

- High resolution high rate.

From time to time, other plotting modes have been added to MIDAS, in particular with regard to displaying polarisation. However, these do not lend themselves to monochrome output[14] and so these are not present in this version of MIDAS.

2.3 Interfaces

I will not describe the interfaces in great detail, assuming instead a reasonable native intelligence on the part of the user. The UNIX and Windows interfaces are very similar to one another. On startup, two windows are produced on the screen: one contains labelled buttons that are pressed to initiate actions, the other is the canvas on which output is drawn. The MS-DOS interface comprises a prompt near the bottom of the screen, at which commands can be typed. The output is produced on the area of the screen above the prompt. In all systems there are mechanisms by which the program can provide textual output, messages and notifications.

2.4 Batch mode

MIDAS supports a batch mode on all platforms. In UNIX and Windows, a file containing MIDAS commands can be given as a parameter to the program name. If such a parameter is present, MIDAS will obtain all its input from the file. In MS-DOS, batch mode is supported through simple I/O redirection. That is, if the commands are contained in the file `COMMANDS.MID`, one types "`MIDAS < COMMANDS.MID`" to have MIDAS execute in batch mode. Any errors in a batch file are fatal and the program aborts immediately on reaching one. The default output device in batch mode is the printer, not the screen, except in MS-DOS mode, where the output is written to the screen first, then dumped to the printer.

Batch mode is very useful in multitasking environments for having MIDAS go through a file producing plots without intervention while running in the background.

[14] This was a difficult decision to make. However, since monochrome output is far more easily obtained than colour, and the mapping of screen-to-printer colour still varies widely, I decided to remove references to colour for this release.

2.5 Windows DOS boxes

MIDAS will run in a Windows DOS box. However, since it runs in graphics mode, it will execute only when the process has access to the entire screen. If the process is windowed, it will suspend operation until such time as the process returns to full screen display. No special PIF settings are necessary to run MIDAS in this way, although a sample PIF file (which also permits MIDAS to run in the background) is included in the distribution.

2.6 Multiple MIDAS processes

In UNIX, DOS and Windows, multiple MIDAS processes may be active simultaneously. In Windows, multiple MIDAS processes may also be present in executing DOS boxes. In DOS, mutliple MIDAS processes may be active simultaneously under the DOS Shell.

Note that it is not possible to run multiple processes from a single DOS Shell which is itself executing inside Windows. This is a constraint imposed by the way that Windows and DOS Shell are implemented and is a general prohibition not limited merely to MIDAS.

3 Commands

Not all commands are available in all environments, although the same basic, necessary subset is always present. In addition, the mechanism by which a command is entered necessarily varies from environment to environment. Rather than provide copious tedious detail, it is assumed that the user can work out the calling mechanisms for himself on the platform of most interest.

3.1 !

The "!" command does nothing. Any parameters are completely ignored. This command is useful for making comments in a batch file.

3.2 ?

The "?" command lists all available commands. This command is available only in MS-DOS mode and batch mode.

3.3 BIN

This command is used to bin a data file into 48-second bins. It is not necessary to perform an explicit BIN of an entire file in order to produce binned plots. If an explicit BIN command has not been issued, records are binned on an as-needed basis. Since binning is a realtively slow process, it is sometimes more convenient to issue a BIN command and work on other things until the command has completed rather than to use on-damand binning.

3.4 BRIGHTNESS

This command changes the brightness level of subsequent plots.

3.5 CLIPBOARD

This command copies the contents of the display canvas to the Windows clipboard.

3.6 COMMENT

This command is used to annotate subsequent plots with a comment.

3.7 DISPLAY

This command sets the output device for subsequent plots.

3.8 EXIT

The EXIT command is used to terminate the program. This command is available in MS-DOS and batch modes. Otherwise terminate the program from the windowing system.

3.9 FILETYPE

This command sets the type of a file to be OPENed. The default filetype is FIXED.

3.10 FIRST

This command produces the first plot of the currently open file.

3.11 INVERT

Inverts (in the sense of colour) the currently displayed image.

3.12 LAST

This command produces the last plot of the currently open file.

3.13 NEXT

This command produces the next plot of the currently open file. If there has been no plot produced from this file, the command behaves as a FIRST command.

3.14 OPEN

This command takes a filename as a parameter. It closes any open files, then attempts to open the named file for reading. If the file opens correctly, MIDAS scans the entire file to produce an internal index of records. This speeds up later access, but may take considerable time if the file is large. MIDAS displays how much of the file has been indexed as it performs this task.

3.15 PLOT

This command is used to produce a plot beginning at a specified time. It takes the parameters: year, day, hour, minute.

3.16 PLOTALL

This command steps through the complete file, producing all the plots for the currently selected spacecraft and PRA modes, producing hardcopy output.

3.17 PLOTRMDR

This command steps forward from the current point in the file, producing plots for the currently selected spacecraft and PRA modes, producing hard-copy output.

3.18 PRINT

This command sends the currently displayed plot to the printer. The hard-copy is a bit-for-bit replica (possibly magnified) of the display screen. Note that this is *not* the same as using the printer as the output device, which generally results in much higher quality output.

3.19 PRINTER

This command selects a new printer.

3.20 PRIOR

This command produces the prior plot of the currently open file. If there has been no plot produced from this file, the command behaves as a LAST command.

3.21 QUIT

QUIT is a synonym for the EXIT command. It is used to terminate the program. This command is available in MS-DOS and batch modes. Otherwise terminate the program from the windowing system.

3.22 REPLOT

Replots the image. This is useful after changing a plot parameter, to see how the change affects the output.

3.23 SATURATION

This command changes the saturation level of subsequent plots.

3.24 SET ?

This command displays the permissible parameters of the SET command. It is available only in MS-DOS and batch modes.

3.25 SET AUTOSCALE

This command has the format “SET AUTOSCALE ON—OFF” and controls the automatic scaling of ordinates when producing tomplots.

3.26 SET DURATION

This command sets the duration of subsequent plots. Its parameter is an integer, representing the duartion of plots in minutes.

3.27 SET ENDCHANNEL

This command sets the highest channel number (by default set to 200) which will appear on subsequent spectral plots. Its parameter is an integer.

3.28 SET NUMBER_OF_TOMPLOT_CHANNELS

This command is used to set the number of channels plotted in tomplots.

3.29 SET PLOTMODE

This command sets the type of plot (by default UNBINNED SPECTRAL) to be produced.

3.30 SET PRAMODE

This command sets the mode of the PRA instrument for which plotting will be enabled on subsequent plots. Its parameter is a string.

3.31 SET SCMODE

This command sets the spacecraft mode for which plotting will be enabled on subsequent plots. Its parameter is a string.

3.32 SET STARTCHANNEL

This command sets the lowest channel number (by default set to 3) which will appear on subsequent spectral plots. Its parameter is an integer.

3.33 SET START_TOMPLOT_CHANNEL

This command is used to set the highest frequency used in tomplots.

3.34 SET TOMPLOT_DN_RANGE

This command has the format "SET TOMPLOT_DN_RANGE channel_nr low_value high_value" and controls the upper and lower ordinate of the given channel number when producing tomplots. The default values for all channels are 0 and 255, which may be inappropriate (for example, in VIM).

3.35 SHOW DURATION

This command shows the current duration (in minutes) of plots. Default is 15.

3.36 SHOW PRAMODE

This command shows the current value of the PRA instrument mode for which plotting is enabled. Default is POLLO.

3.37 SHOW PRAMODES

This command lists all the modes in which the PRA instrument operated in the currently open file.

3.38 SHOW SCMODE

This command shows the current value of the spacecraft mode for which plotting is enabled. Default is GS-3.

3.39 SHOW SCMODES

This command lists all the spacecraft modes found in the currently open file.

3.40 SHOW TIMES

This command lists the earliest and latest times present in the currently opened file.