

Extended PicoDOS® Reference Manual

SCHOOL OF OCEANOGRAPHY

and

APPLIED PHYSICS LABORATORY

UNIVERSITY OF WASHINGTON

Version 66.12 May 2012; Revised by Kongsberg July 2017

1. INTRODUCTION AND CONVENTIONS

1.1. Introduction

- 1.1.1. This manual is a reference to extensions to the PicoDOS® operating system command set implemented in the Seaglider operating code. PicoDOS® is a registered trademark of Persistor Instruments, Inc., Bourne MA, USA.
- 1.1.2. The version number of this document coincides with the version of the main Seaglider operating code in which these extensions exist.
- 1.1.3. These extensions either make new functions available from the PicoDOS® prompt, or extend the capabilities of existing functions. Commands are only available through the Seaglider operating code, which intercepts and interprets the commands, passing them to PicoDOS® as appropriate. While in this mode, the Seaglider code passes any command not explicitly recognized as an extended PicoDOS® command on to PicoDOS® itself for execution. Limited error reporting exists in this case.
- 1.1.4. The extensions are accessible at the PicoDOS® prompt available from the main menu when connected directly to a Seaglider (exceptions as noted below), or by uploading the **pdoscmds.bat** file when the Seaglider is operating autonomously. In the former case, the Seaglider code displays a standard PicoDOS® prompt (`picoDOS>`) with an extra '>', as follows.

1.1.4.1. `picoDOS>>`

- 1.1.5. In the latter case, results of the extended PicoDOS® commands are captured to a file and transferred to the Seaglider basestation (in compressed form, named **sg0055pz.000**, using "0055" as a placeholder for dive number, and "000" for increment number) following execution of the commands. The basestation renames this file per the **p1230055.000.pdos** convention.
- 1.1.6. The **pdoscmds.bat** file is a plain text sequence of extended PicoDOS® commands, one per line. Lines that begin with a forward slash ('/') are interpreted as comments and are ignored.
- 1.1.7. Chapter 2 is a list of the extended PicoDOS® commands, grouped by functional area. Chapter 3 is an alphabetical-order reference for each command. Note that standard PicoDOS® commands are not documented here.

- 1.2. Document Conventions
- 1.2.1. Extended PicoDOS® commands are shown in ***bold italic*** type below. Options and arguments are shown on the same line. Exposition follows in plain type. File names are given in **lower-case bold** font. Click on the name of a command to jump to its description. Use the “Back” button to return to the previous location.
- 1.3. PicoDOS®
- 1.3.1. PicoDOS® is Persistor Instrument’s DOS-like operating system for the CF8/TT8 combination used on Seaglider. It provides access to the DOS FAT file system on the Compact Flash, as well as some simple file manipulation utilities. The TOM8 and PicoDOS® commands are documented in the PicoDOS® User’s Guide of November, 2000 (which is incorporated into this document by reference). The extensions below provide additional functionality, either to extend PicoDOS® generally or to provide Seaglider-specific functions.

2. LIST OF EXTENDED PICODOS® COMMANDS

- 2.1. Help and exit

```
?  
??  
pdos  
tom8  
quit
```

- 2.2. File manipulation and data

```
bathy  
cat <filespec>+ [<> | >>] <outfile>  
[del | rm] [/v] <filespec>+  
gunzip <zipped_file> <file>  
gzip <file> <zipped_file>  
md5 [<signature>] <file>  
[ren | rename] <file1> <file2>  
resend_dive [/l | /d | /c | /t] <dive> <fragment>  
science  
split <filename>  
strip1a <filename> [<size>]  
sumasc <file>  
tar [c | x] <file> <filespec>  
[xs | put | xr | get] <filespec>+
```

2.3. Control

```
$PARAM,value  
clearnv <passphrase>  
dumpnv  
readnv <varname>  
reboot <run_file> <arguments>  
[target | targets] [<new_target> [<radius>]]  
writenv <varname> <value>
```

2.4. Diagnostics

```
capvec [<service> <level> <dest>]  
flash_errors  
menu <menuspec> [<arguments>]  
parse_capvecfile <file>  
usage  
ver
```

3. EXTENDED PICODOS® COMMAND REFERENCE

3.1. *bathy*

- 3.1.1. Causes the on-board bathymetry files to be re-read, checked, and, if appropriate, loaded into memory. Useful for checking the integrity of the bathymetry files on the compact flash.

3.2. *capvec [<service><level><dest>]*

- 3.2.1. Without arguments, displays the current capture vector. The capture vector describes the capture output level and destination for each of the Seaglider's hardware and software services. Every output line in the Seaglider source code that is capturable is assigned a service and an output level (verbosity). The capture vector controls where the output is routed to and what the output level (verbosity) is for each service. With arguments, sets the capture vector for a specific service (table 3.1) to the specified level (table 3.2) and destination (table 3.3).

Table 3.1: Available services for capturing

Service	Description
HPITCH	Pitch motor
HROLL	Roll motor
HVBD	VBD Pump and Valve
HPHONE	Modem hardware
HGPS	GPS receiver
HTT8	TT8 Computer
HCF8	Flash hardware
HANALOG	Analogue circuits and control
HCOMPASS	Compass hardware

HRAFOS	RAFOS hardware
HSBECT	Sea-Bird CT sensor
HSBEO2	Sea-Bird O2 sensor
HGPCTD	Sea-Bird Pumped CTD
HWLBB2F	Wetlabs sensor
HOPTODE	Optode O2 sensor
HNCP	Current profiler
HPA	Passive acoustics
HRS	RSI Micro-Structure
HBATT	Battery hardware and charging
HPRES	Pressure Sensor
HXPDR	Transponder hardware
SPOWER	Software managing power
SBATHY	Software managing bathymetry
SNAV	Software managing navigation (primarily targets)
SKALMAN	Software kalman filter
SMOTOR	Software controlling all motor movements (primarily GC interrupt handler)
SSENSOR	Software controlling all sensors (primarily data sampling)
SDIVE	Software controlling dive and flight
SSURF	Software controlling surface activities
SEPDOS	Extended PicoDOS® support
SSYS	Software utilities and infrastructure
SUSR	Software dealing primarily with human console interaction (primarily tests and menus)
SGLMALLOC	Seaglider's heap implementation

Table 3.2: Levels at which output can be captured for a given service

Output Level	Description
CRITICAL	Only trigger output that is critical in nature – usually associated with an extreme hardware problem (such as motor errors) or a critical software problem such as buffer or heap overrun.
NORMAL	The vast majority of Seaglider output falls in this category currently.
DEBUG	Extended diagnostic. This output can be quite voluminous and it is recommended that this only be set on specific services when it is known that some needed output will be captured. Most services do not have output in this level currently, but work is ongoing to add extended diagnostics under this output level.

Table 3.3: Available destinations for capture output of any service.

Destination	Description
NONE	Do no output for this service.
PRINT	Send output only to the operator console.
FILE	Send output only to the capture file .
BOTH	Send output to both the operator console and to the capture file .

- 3.2.2. Note that a higher level of output also implies the lower levels. For example, setting a services output to **NORMAL** implies that **CRITICAL** output will also be triggered.

Example: *capvec HVBD DEBUG PRINT*

Sets the hardware VBD service output to debug level (most verbose) and routes the output for that service to the operators console only.

Example: *capvec HVBD DEBUG FILE*

Sets the hardware VBD service output to debug level (most verbose) and routes the output for that service to the **capture file**.

Example: *capvec HVBD DEBUG BOTH*

Sets the hardware VBD service output to debug level (most verbose) and routes the output for that service to the operators console and to the **capture file**.

Example: *capvec HVBD NORMAL BOTH*

Default setting. Sets the hardware VBD service output to normal level (standard level) and routes the output for that service to the operators console and to the **capture file**.

- 3.2.3. By default, the Seaglider software sets the output levels to **NORMAL** and output destinations to **BOTH** for all services. As a side effect of this change, the **capture file** is almost always being filled with output.

3.3. ***cat <filespec> [[> | >>] <outfile>***

- 3.3.1. Concatenates files in the same general way as the standard Unix cat command. Admits use of wildcards (*) in the file specification.

Example: *cat chunk.u30 chunk.u31>>chunk.gz*

3.4. ***clearnv <passphrase>***

- 3.4.1. Clears contents of non-volatile RAM utility storage. The passphrase must exactly match one of the strings hard-coded in epdos.c: I_really_mean_it! Clears all utility storage (including password and telephone numbers), I_mean_it! Only clears latched target and flight state information.

3.5. ***[del | rm] [/v] <filespec>***

- 3.5.1. Extends the standard delete (del) command by admitting use of wildcards (*) in the specification.

Example: *del SG01*LZ.A*

Deletes all zipped log files whose name starts with SG01 that have been transferred to the basestation by the glider. DZ = zipped data files, KZ = zipped capture files, LZ = zipped log files, A = files that have been transferred to the basestation, X = files that have not been transferred to the basestation.

- 3.6. ***dir***
- 3.6.1 Lists the contents of the CF card
Example: dir
Lists all content on the Cf card
Example: dir *.cnf
Lists all cnf files on the CF card
- 3.7. ***dumpnv***
- 3.7.1. Dumps contents of non-volatile RAM.
- 3.8. ***flash_errors***
- 3.8.1. Reports CF8 file open, write and close retries and errors.
- 3.9. ***gunzip <zipped_file> <file>***
- 3.9.1. Uncompresses a file compressed with gzip.
Example: gunzip MAIN.GZ MAIN.RUN
Uncompresses MAIN.GZ to MAIN.RUN
- 3.10. ***gzip <file> <zipped_file>***
- 3.10.1. Compresses file with gzip.
Example: gzip MAIN.RUN MAIN.GZ
Compresses MAIN.RUN to MAIN.GZ
- 3.11. ***lsleep <seconds>***
- 3.11.1. Pause execution in low power sleep for up to 60 seconds.
- 3.12. ***md5 [<signature>] <file>***
- 3.12.1. Generates 1280bit md5hashes of the specified file. If a signature (hash) is specified, md5 compares the specified signature (hash) with the one it generates for the file and generates an error if they are not identical (in the character-by-character sense). Used to verify the integrity of files uploaded to the Seaglider. In particular, md5 is the verification part of the protocol used to upload, verify and reboot new executable code on the Seaglider.
Example: md5 082ab2b60d626181e73b17429c55dd8f chunk.GZ
- 3.13. ***menu <menuspec> [<arguments>]***
- 3.13.1. Execute commands from the Seaglider code menu tree, by specifying the absolute menu path to the command and any required arguments. The menu is specified by menu names, separated by forward slashes, '/'. The arguments are specified in a whitespace-separated list: arg1=val1 arg2=val2
Example: menu /hw/battery/read

Reads the battery voltage.

3.14. **\$PARAM, value**

- 3.14.1. Changes the specified parameter to the specified value. Parameters are specified as three-digit numbers (nnn), and are in lineal order as they appear in the Seaglider menus (or in the code source file *parms.h*).

Example: *\$T_DIVE,330*

Sets the target dive depth to 330 m.

3.15. **parse_capvecfile <file>**

- 3.15.1. Parses a capture vector file and updates the capture vector. This command is normally used during glider provisioning or testing – the preferred way to set the capture vector is through the *capvec* command.

Example: *parse_capvecfile capvec.new*

3.16. **pdos**

- 3.16.1. Exits the Seaglider extended picoDOS® mode to native PicoDOS® on the TT8/CF8. To exit from native PicoDOS® and return to the Seaglider main menu type **main**.

3.17. **quit**

- 3.17.1. Exits the Seaglider extended picoDOS® mode and returns to the Seaglider main menu.

3.18. **readnv <varname>**

- 3.18.1. Read the value of the specified variable in non-volatile RAM, where the variable is one of the following: *target_name, password, selftest_count, boot_count, last_known_lon, last_known_lat, last_last_fix_time, magvar, fly_escape_route, fly_safe_depth, device0, device1, device2, device3, telnum, or altnum*.

Example: *readnv password*

Example: *readnv telnum*

3.19. **reboot <run_file>**

- 3.19.1. Reboots the Seaglider using the specified file name as the executable file. Note that the .run suffix is not used in this command. It is important to verify that the file that is the target executable is not corrupted. A way to prevent unrecoverable hangs is to never rename new unproven executables **main.run**. Name them **mainnew.run**, for example, and issue the **reboot** command as **reboot mainnew**. If the Seaglider hangs during the reboot, the watchdog timer should initiate another reboot, but to the executable code called *main*, which presumably was running when the **reboot** command was issued. This is a fall-back safety feature and should be noted.

Example: *reboot REVA*

3.20. ***[ren | rename] <file1> <file2>***

- 3.20.1. Rename command which calls the CF8 rename command directly, and is aliased to work with either ***ren*** or ***rename***.

Example: *ren REVA.REN MAIN.RUN*

Renames the file REVA.REN to MAIN.RUN.

3.21. ***resend_dive [/l | /d | /c | /f] <dive> <fragment>***

- 3.21.1. The mechanism by which the pilot can ask the Seaglider to resend individual pieces of previous log (/l), data (/d), or capture (/c) files or another file (/f). The /l, /d, and /c options are used to recover missing pieces of data, which are most often caused by protocol confusion between the Seaglider and the basestation (on whether or not a particular data piece was successfully transferred). The data are specified by type, dive number, and fragment number (the third 4kB piece of dive 123's data file would be specified "/d 123 2" (data piece counter starts at 0). Leading zeroes on the fragment number are optional. The /f option allows other file types to be sent from the glider to the basestation.

Example: *resend_dive /d 289 1*

Resend the second 4kB chunk of dive 289's data file. Chunk numbering begins with 0.

Example: *resend_dive /d 289*

Resend the entire data file from dive 289.

Example: *resend_dive /l 289*

Resend the entire log file from dive 289.

Example: *resend_dive /c 289*

Resend the entire capture file from dive 289.

Example: *resend_dive /d /l 289*

Resend the entire data and log files from dive 289.

Example: *resend_dive 289*

Resend all files (data, log and capture) from dive 289.

Example: *resend_dive /f tcm2mat.000*

Resend the file tcm2mat.000.

3.22. ***science***

- 3.22.1. Parses the on-board **science file**. Used mainly in the laboratory or test tank to establish sampling intervals for testing.

3.23. ***sleep <seconds>***

- 3.23.1. Pause execution for up to 60 seconds.

Example: *sleep 30*

- 3.24. ***split <file>***
- 3.24.1. Splits files into pieces of size **\$N_FILEKB** (kBytes).
Example: *split sg0115dz.r*
- 3.25. ***strip1a <file>[<size>]***
- 3.25.1. Strips trailing XMODEM padding characters ('1a') off the end of a file, optionally to a specified size (in bytes).
Example: *strip1a chunk.u31 1898*
Strips the padding characters off the end of the file named chunk.u31 to the specified size of 1898 bytes.
Example: *strip1a chunk.u31*
Strips the padding characters from the end of the file named chunk.u31
*This function should be performed on all files transferred to the Seaglider's CF card using XMODEM.
- 3.26. ***sumasc <file>***
- 3.26.1. Sums succeeding lines of data file to recreate original observed values. This is the inverse of the simple successive difference compression scheme used for glider data files.
- 3.27. ***Tar [c | x] <tarfile> <filespec>***
- 3.27.1. Implementation of standard Unix-style tar (tape archive) utility. Admits wild-card (*) expressions. c = create a tar file; x = extract from a tar file
Example: *tar c data.tar SG01*DZ.A*
Create a tar file called data.tar that includes all files beginning with SG01 and ending with DZ.A.
Example: *tar x glider.tar*
Extracts all files included in the glider.tar file.
- 3.28. ***[target | targets] [<new_target> [<radius>]]***
- 3.28.1. The mechanism by which the pilot can change the current Seaglider target and the target radius from among the targets listed in the current **targets file** on the Seaglider. The **new_target** must be in the list of targets in the **targets file** on the Seaglider flash. The radius is in meters. Either the command '**target**' or '**targets**' can be used.
Example: *target NE_CORNER*
Changes the current Seaglider target to NE_CORNER
Example: *targets SW_CORNER 1852*
Changes the current Seaglider target to SW_CORNER and the radius to 1852 meters.

- 3.29. ***tom8***
- 3.29.1. Exits the Seaglider main program and native PicoDOS® to the TT8 monitor program, TOM8. PicoDOS® is started by issuing the command ‘go2bcf8’ at the TOM8 prompt.
- 3.30. ***usage***
- 3.30.1. Provides a summary of disk usage on the compact flash.
- 3.31. ***ver***
- 3.31.1. Provides versioning and configuration information for software and hardware installed (and running) on the Seaglider.
- 3.32. ***[xs | put | xr | get] <file1> <file2>***
- 3.32.1. XMODEM protocol file transfer commands (issued from the glider).
Example: *xr chunk.U30*
The Seaglider receives the file named chunk.U30 during XMODEM transfer.
Example: *xs sg0150DZ.A*
The Seaglider sends the file named sg0150DZ.A during XMODEM transfer.
Example: *get chunk.U30*
The Seaglider receives the file named chunk.U30 during XMODEM transfer.
Example: *put sg0150DZ.A*
The Seaglider sends the file named sg0150DZ.A during XMODEM transfer.
- 3.33. ***writenv <varname> <value>***
- 3.33.1. Write specified value of specified variable to non-volatile RAM (NVRAM) utility storage. Variables stored in NVRAM are *target_name*, *password*, *selftest_count*, *boot_count*, *last_known_lon*, *last_known_lat*, *last_last_fix_time*, *magvar*, *fly_escape_route*, *fly_safe_depth*, *device0*, *device1*, *device2*, *device3*, *telnum*, and *altnum*.
Example: *writenv telnum 881690000000*
Changes the telnum to the RUDICS number 881690000000.
Example: *writenv telnum 14251234567*
Changes the telnum to PSTN number 14251234567, where 1 is the country code (note the glider adds the 00 before the 1 automatically so it is not included in this command), 425 is the area code and 1234567 is the phone number.
Example: *writenv target_name WP1*
Changes the target name to WP1.
- 3.34. ***?***
- 3.34.1. Top-level help command, which only displays methods of exiting the Seaglider code’s PicoDOS® mode, and documents the ?? command.

3.35. **??**

- 3.35.1. Extensive help command, which displays methods of exiting the Seaglider code's PicoDOS® mode, lists the available extended PicoDOS® commands, and gives usage hints.