

Extended PicoDOS® Reference Manual

SCHOOL OF OCEANOGRAPHY
and
APPLIED PHYSICS LABORATORY
UNIVERSITY OF WASHINGTON
Version 66.10 April 2012

Chapter 1 Introduction and Conventions

1.1 Introduction

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.

The version number of this document coincides with the version of the main Seaglider operating code in which these extensions exist.

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.

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.

```
picoDOS>>
```

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.

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.

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

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®

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.

Chapter 2.

List of Extended PicoDOS® Commands

Help and exit

?
 ??
 pdos
 tom8
 quit

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>+

Control

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

Diagnostics

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

Chapter 3

Extended PicoDOS[®] Command Reference

bathy

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.

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

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 2.1) to the specified level (table 2.2) and destination (table 2.3).

<i>Service</i>	<i>Description</i>
HPITCH	Pitch motor
HROLL	Roll motor
HVBD	VBD Pump and Valve
HPHONE	Modem hardware
HGPS	GPS receiver
HTT8	TT8 Computer
HCF8	Flash hardware
HANALOG	Analog circuits and control
HCOMPASS	Compass hardware
HRAFOS	RAFOS hardware
HSBECT	Seabird CT sensor
HSBEO2	Seabird O2 sensor
HWLBB2F	Wetlabs sensor
HOPTODE	Optode O2 sensor
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 2.1: Available services for capturing.

<i>Output Level</i>	<i>Description</i>
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 diagnostics. 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 2.2: Levels at which output can be captured for a given service.

<i>Destination</i>	<i>Description</i>
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 .

Table 2.3: Available destinations for capture output of any service.

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.

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

cat <filespec>+ [[> | >>]<outfile>]

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*

clearnv <passphrase>

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.

[del | rm] [v] <filespec>+

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

Example: `del SG01*LZ.A`

dumpnv

Dumps contents of non-volatile RAM.

flash_errors

Reports CF8 file open, write and close retries and errors.

gunzip <zipped_file> <file>

Uncompresses file compressed with gzip.

Example: `gunzip chunk.GZ MAIN.RUN`

gzip <file> <zipped_file>

Compresses file with gzip.

Example: `gzip MAIN.RUN MAIN.GZ`

Isleep <seconds>

Pause execution in low power sleep for up to 60 seconds.

md5 [<signature>] <file>

Generates 128-bit md5 hashes 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`

menu <menuspec> [<arguments>]

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/ct`

\$PARAM,value

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

Example: `$T_DIVE,330`

parse_capvecfile <file>

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`

pdos

Exits the Seaglider code to native picoDOS® on the TT8/CF8.

quit

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

readnv <varname>

Read the value of the specified variable, 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*.

reboot <run_file>

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 fallback safety feature and should be noted.

Example: *reboot REVA*

[ren | rename] <file1> <file2>

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

Example: *ren REVA.RUN MAIN.RUN*

resend_dive [/l /d /c /t] <dive> <fragment>

The mechanism by which the pilot can ask the Seaglider to resend individual pieces of previous log (/l), data (/d), capture (/c) or tar (/t) files. This is 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 zeros on the fragment number are optional.

Example: *resend_dive /d 289 1*

science

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

sleep <seconds>

Pause execution for up to 60 seconds.

split <file>

Splits files into pieces of size \$N_FILEKB (kBytes).

Example: *split sg0115dz.r*

strip1a <file>[<size>]

Strips trailing XMODEM padding characters ('1a') off the end of a file, optionally to a specified size (in bytes).

Example: *strip1a chunk.U31 1898*

sumasc <file>

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.

tar [c | x] <tarfile> <filespec>+

Implementation of standard Unix-style tar (tape archive) utility. Admits wild-card (*) expressions.

Example: `tar c data.tar SG01*DZ.A`

[target | targets] [<new_target> [<radius>]]

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.

Example: `target NE_CORNER`

Example: `target SW_CORNER 1852`

tom8

Exits the Seaglider main program and native picoDOS® to the TT8 monitor program, TOM8. PicoDOS is started by issuing the command 'go 2bcf8' at the TOM8 prompt.

usage

Provides a summary of disk usage on the compact flash.

ver

Provides versioning and configuration information for software and hardware installed (and running) on the Seaglider.

[xs | put | xr | get] <file1> <file2>

XMODEM protocol file transfer commands (issued from the glider).

Example: `xr chunk.U30`

Example: `xs sg0150DZ.A`

writenv <varname> <value>

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*.

?

Top-level help command, which only displays methods of exiting the Seaglider code's PicoDOS® mode, and documents the ?? command.

??

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