

# ***PicoDOS8 & TxBASIC*** ***Programming Guide***

**PicoDOS<sup>®</sup>**

© 2000 Persistor Instruments Inc.



**P E R S I S T O R**  
I n s t r u m e n t s I n c.

254-J Shore Road, Bourne, MA, 02532 USA  
Tel: 508-759-6434 Fax: 508-759-6436  
www.persistor.com info@persistor.com

## About the PicoDOS8 TxBASIC Guide

A Persistor plus PicoDOS8 gives your TxBASIC programs the ability to store huge amounts of data with the addition of just a few BASIC statements to your existing programs. You need to have PicoDOS8 installed in your TT8's onboard flash memory and version 4.10 or later of TxBASIC which has built-in PicoDOS8 support.

Complete PicoDOS8 installation instructions can be found in the Getting Started Guide that came with your Persistor or up on the [www.persistor.com](http://www.persistor.com) web site and updates to earlier versions of TxBASIC can be obtained by contacting technical support at [www.onsetcomp.com](http://www.onsetcomp.com).

You can access the five new TxBASIC commands for the Persistor by declaring them as extensions at the top of your program. There's a command to save all or part of your TxBASIC DATAFILE to a file on the flash memory card using DOS 8.3 file naming conventions. When your experiment is complete, you can move the memory card from the TT8 to your Windows PC equipped with a very inexpensive card reader and quickly copy the files to your PC for analysis. There's also a complimentary command to let you fill all or part of the DATAFILE from a named file resident on the flash memory card. Two other commands let your program monitor the total capacity and free space on the current memory card. Finally, there's a very powerful command that lets you invoke any of the PicoDOS8 shell commands from within your program.

## How to use the PicoDOS8 TxBASIC Guide

This guide is written as a follow-up to the printed Getting Started Guide that came with your Persistor and builds on information presented there. It, in turn, should have already convinced you of the necessity of mastering the TT8 using Onset's documentation. That done, you should be ready to start using the Persistor with TxBASIC.

For a quick introduction, we've annotated a simple TxBASIC program that demonstrates most of the techniques you'll need to learn to take advantage of the Persistor. We follow that with a reference section detailing the use of the PicoDOS8 extensions.



## **Other PicoDOS8 Related Documentation**

### **Getting Started Guides**

This is the key must-read document if you're to have successful experiments with the Persistor. There's a separate Getting Started Guide for each of the various Persistors and it's one of the only two printed documents that come with your Persistor. If you don't have this on hand, the latest version is always available from our [www.persistor.com](http://www.persistor.com) web site and you'll find PDF and html copies on the installation diskette. If you haven't yet read and worked through the installation procedures, do that first before attempting any of the programming described in this guide.

### **Persistor Data Sheets**

This is the other printed document that comes with your Persistor and describes the electrical, mechanical, and environmental specifications you may need to design your experiment.

### **PicoDOS8 User's Manual**

PicoDOS8 is our DOS-like operating system for the CF8/TT8 combination that provides both a command line user interface for common card and file operations as well as the underlying DOS FAT file system. It's this that lets your C and BASIC programs easily create and manipulate files that can later be read directly by your PC using inexpensive flash memory card readers. Here you will find descriptions of how to use the DOS-like commands and details of how PicoDOS8's presence impacts the amount of TT8 onboard flash and ram memory usage available to your applications.

### **PicoDOS8 C Programming Guide**

The PicoDOS8 C Programming Guide shows you how to adapt your C programs to take advantage of standard ANSI C file system capabilities that become possible with the addition of a Persistor and memory cards. This guide assumes that you are familiar with C programming and the ANSI C file functions, while describing the subtle differences between coding for desktop applications with unlimited processing power, memory space, and disk speed and coding for embedded applications where processing power is limited, battery power is precious, and file I/O takes considerably longer.



## Annotated TxBASIC Example

Below is an abbreviated version the picotest.tx example TxBASIC program that gets installed with your PicoDOS8 tools. It demonstrates how to use the

PicoDOS8 extensions in just a couple of dozen lines of code. The line numbers preceding the TxBASIC code are just used as reference points for explaining how the commands work in the text that follows the program code. You should type this program into TxTools and run the program as your introduction to the Persistor.

```
1   Model 800
2
3   extension CFSsize, CFAvail, CFSave, CFRead, CFExec
4
5   print "Checking card capacity..."
6   size = CFSsize()
7   avail = CFAvail()
8   PRINT
9   PRINT size, "byte card with ", avail, " bytes free"
10
11  CFExec("DIR")          // show starting directory
12  print "Set the system clock..."
13  CFExec("DATE")        // set it now so files will proper date
14  print
15
16  // PUT SOME TEXT IN THE DATAFILE FOR TESTING
17  dfptr = 0              // start at beginning
18  print "Type some text, end with <return>"
19  itext dfptr
20  if dfptr = 0 written = 0 // in case we're reporting
21  if dfptr > 0 written = CFSave("test.dat", dfptr)
22  print "Wrote ", written // show how we made out
23  CFExec("DIR")          // show updated directory
24  CFExec("TYPE test.dat") // show what we saved as text
25  CFExec("DUMP test.dat") // now show it as hex
26  print "starting XMODEM transfer..."
27  print CFExec("XS test.dat")
```



Line 1 is the standard TxBASIC declaration required by TxTools to identify which Tattletale model will be receiving the generated BASIC code.

Line 3 declares the five new TxBASIC extensions for PicoDOS8. Each is described in detail just ahead in the reference section.

Line 6 calls the CFSize extension function which reads the card to determine the total number of bytes the card can hold and returns this as a 32-bit number. This gets saved in the variable named “size” which will be used later to print the card’s vitals.

Line 7 calls the CFAvail extension function which computes how much free space is available on the card and returns this as a 32-bit number. This gets saved in the variable named “avail” which will also later to print the card’s vitals.

Lines 5, 8, and 9 format the newly discovered card information into a meaningful message which will look something like:

```
Checking card capacity...
64049152 byte card with 63940608 bytes free
```

Lines 16 through 19 ask you to type characters which will be automatically stored in the TT8 DATAFILE using the itext command. These should be easily recognized when we play them back in the next steps.

Line 20 is a test to keep you from needlessly creating an empty file if no characters were sent.

Line 21 calls the CFSave extension function to create a file called test.dat that will contain the characters you typed in the previous steps. It returns the number of bytes written for error checking or computing resource usage as shown by line 22.

Lines 23, 24, and 25 demonstrate the CFExec extension function which lets you invoke any of the PicoDOS8 commands from within your BASIC program. In this case, it first displays a list of files (which should now contain test.dat), then displays the file contents as text, and finally shows the test files contents in hexadecimal.

Line 27 again demonstrates the CFExec extension function - this time accessing the PicoDOS8 XMODEM file transfer capability to send the test file back to your PC without having to move the memory card around.

## PicoDOS8 Extensions Reference

### CFSize & CFAvail

These first two PicoDOS8 TxBASIC extensions are virtually identical in behavior and differ only in the count value that they return. Either one of these is an excellent candidate function to call at startup confirm the availability and status of the attached memory card.

CFSize returns the total capacity in bytes of the installed flash memory card while CFAvail tells you the total free space in bytes still available for use. Both of these calls cause PicoDOS8 to read multiple sectors into ram buffers to perform the computation which takes both time and power and increases as the card fills up.



If the returned value from either has a negative value, then there's a serious problem which will have to be corrected before you can read or write to the card. The table below gives the exact meaning of each of the negative error code:

## CFSave & CFAvail Error Codes

- 1 PicoDOS8 initialization failed
- 2 No PicoDOS installed in high flash
- 3 No Persistor attached
- 4 No CompactFlash card inserted
- 5 Undefined initialization error

## CFSave

CFSave (filename [,length [,dfofs [,append]]])

This extension saves all or part of the DATAFILE to a new or existing file on the flash memory card.

**filename** is a string constant or string variable that must conform to DOS 8.3 file naming conventions, where 8.3 refers to a maximum of eight characters for the first part of the name, and a maximum of three characters for the filename extension, and the period separating the two. Examples of valid filenames are: "test.dat", "temp", "data.0". The special filename "?" will cause CFSave to generate a unique new name so that you don't have to deal with that in your program.

**length** is an optional argument that can specify a byte count less than or equal to DFMAX, which is the default and is equal to the size of the datafile. If length is less than zero, or greater than DFMAX minus the start offset, CFSave will attempt to write the maximum number of bytes from offset to end.

**dfofs** is an optional argument that can specify a byte offset into the datafile where writing is to begin, and must be greater than or equal to zero, and less than DFMAX. If dfofs is specified, the length parameter will be shortened as needed to not write past the end of the datafile.

**append** is an optional flag argument that, when non-zero, tells CFSave to attempt to append to an existing file with the same name, but to create a new file, if the specified file does not yet exist.

**returns** the actual number of bytes transferred to the flash memory card, or a negative error code which is described ahead.

## CFRead

CFRead (filename [,length [,fileofs [,dfofs]]])

This extension reads all or part of an existing flash memory card file into the DATAFILE.

**filename** is a string constant or string variable that must conform to DOS 8.3 file naming conventions, as described for the CFSave command.

**length** is an optional argument that can specify a byte count less than or equal to DFMAX, which is the default and is equal to the size of the DFMAX minus the start offset, or the size of the file, whichever is less.

**fileofs** is an optional argument that can specify a byte offset into the file where reading is to begin, and which must be greater than or equal to zero.



**dfofs** is an optional argument that can specify a byte offset into the datafile where reading is to begin, and must be greater than or equal to zero, and less than DFMAX. If **dfofs** is specified, the length parameter will be shortened as needed to not read past the end of the datafile.

**returns** the actual number of bytes transferred from the flash memory card, or a negative error code which can be interpreted as described for CFSave.

### **CFSave & CFRead Error Codes**

- 1 CompactFlash failed initialization
- 2 Invalid filename specified
- 3 Can't open the specified file
- 4 Can't position to the requested offset

### **CFExec**

CFExec (command)

This extension lets you execute a PicoDOS intrinsic command as described above for the Command Shell.

**command** is a string constant or string variable of sixty characters or less that gets directly passed to PicoDOS.

**returns** zero for success, or -1 if the command failed. Note that executing a "RUN" will never return control to your TxBASIC program.

## **Contacts:**

### **Persistor Instruments Inc.**

**Tel:** 508-759-6434  
**Fax:** 508-759-6436  
**e-mail:** info@persistor.com

### **Onset Computer Corporation**

**Tel:** 800-LOGGERS  
**Fax:** 508-759-9100  
**e-mail:** sales@onsetcomp.com

## **Trademarks:**

**Persistor**<sup>®</sup>, **PicoDOS**<sup>®</sup>, and **MotoCross**<sup>®</sup> are registered trademarks of Persistor Instruments Inc. **Tattletale** and **CrossCut** are trademarks of Onset Computer Corporation. **CompactFlash**<sup>™</sup> is a trademark of the CompactFlash Association.



