# PDSI Users Manual
**Version 2.0**

Nathan Wells
National Agricultural Decision Support System
http://nadss.unl.edu
University of Nebraska-Lincoln
nwells@cse.unl.edu

July 15, 2003

# Contents

## Introduction

This PDSI program was written based on a FORTRAN program that calculated the monthly PDSI that was received from the National Climatic Data Center (NCDC). The program was translated to C++ and several modifications were made, including making the index Self-Calibrating. This PDSI program is capable of calculating the Original Monthly PDSI, the Self-Calibrating Monthly and Weekly PDSI, as well as the Weekly Crop Moisture Index (CMI). A description of how to use the program is given in the **Usage** and **Usage Examples** sections. The necessary input files are explained in the **Input Files** section and the **Output Files** section explains what files the program writes to and where they are.

This document is intended to provide information about how to use the PDSI program as it is offered from http://nadss.unl.edu. It is not intended to explain how the PDSI calculations are performed or how the program's code is structured. There is separate documentation for both the PDSI calculations and the PDSI code. Check http://nadss.unl.edu/info/ for the most current documentation on the PDSI calculations.

### *Changes Since Version 1*

- The metric flag's functionality was added. March 12, 2003
- The PET calculations were altered to work in both the northern and southern hemispheres. April 1, 2003
- The calculations for the weekly CMI were added. July 1, 2003
- A more intuitive format of the parameter file can now be used. A user can now supply either the AWC and latitude (in decimal degrees) or the AWC and TLA. July 15, 2003.

## Usage

| Command Line Argument | Description |
|---|---|
| `--help` | Causes the program to output the options available. |
| `-v` | Verbose mode. Causes the program to print intermediate calculations to the screen. |
| `-s` | Silent mode. Suppresses all screen output. |
| `-m` | Reads the input files as if they were given in metric units(ºC and mm). By default, the program reads the input in English units (ºF and in). |
| `-b` | The program will reproduce the bugs in the FORTRAN program. |
| `-e` | This flag is used to specify that the given year is the end year and not the start year. By default, when a single year is given as an argument, it is assumed to be the start year. If two years are given, the smaller is the start year and the larger is the end year. |
| `-f` | Specifies the output format. Must be accompanied by one of the three strings 'table', 'column', or 'both', which is to be given as separate arguements. |
| `-x` | Specifies what extra files to be given. Must be accompanied by one of the four strings 'wb', 'Xtable', 'potentials', or 'all', which is to be given as separate arguements. |
| `-n` | The nadss flag. It will cause the program to read the parameter files as if it contains the AWC and the TLA. By default, the program reads the parameter file as if it contains the AWC and the latitude of the station, and then calculates the TLA based on that. |
| `-i` | Specifies the location of the input files. The path to the directory containing the input files should be given by attaching it to the '-i' flag. Example: `./pdsi -imonthly/` |
| `-o` | Specifies the location of the output files. . The path to the directory containing the input files should be given by attaching it to the '-o' flag. Example: `./pdsi -omonthly/` If the output directory does not exist, it will be created. |
| `A decimal number` | Any decimal number is interpreted to be a tolerance specification, which controls what tolerance will be used in the comparisons. The default is 0.00001 |
| `A 4-digit integer` | Any four digit integer is interpreted to be a year. If one year is given, it is assumed to be the start year, unless the '-e' flag is used. If two years are given, the start and end year are set accordingly. |
| `An integer with 1,2 or 3 digits` | Any integer between 1 and 999 (inclusive) can be used to specify the total number of years to output. The number must be no greater than the total number of available years. The number of years is always measured from the start year. |

Any group of single character flags can be given together, with the exception of the ‘-i’ and ‘-o’ flags, which must be separate. The following are all valid sets of command line arguments that produce the same thing.

```
./pdsi –x all –f both –imonthly/ -otest/output/dir/ -v –b –s –n 1948

./pdsi –x all –s –b –otest/output/dir/ -i/monthly/ -n 1948

./pdsi 1948 –xfbvsn –imonthly/ -otest/output/dir/ all both
```

## Usage Examples

This section gives some examples of how to use the PDSI program. The input files are the same as those that come with the program from **http://nadss.unl.edu**. The data comes from Hartington, NE. The examples are given using the UNIX version of the program, for the simple reason that it is easier to show the contents of files, directory structure, and especially the command line arguments in a UNIX environment. The same operations can be done in Windows, but they require more graphical interaction.

### Running the program with no flags

```
[~/pdsi/test]$ ./pdsi
   * Self-Calibrated Weekly PDSI written to ./weekly/1/
   * Self-Calibrated 2-Week PDSI written to ./weekly/2/
   * Self-Calibrated 4-Week PDSI written to ./weekly/4/
   * Self-Calibrated 13-Week PDSI written to ./weekly/13/
   * Monthly PDSI written to ./monthly/original/
   * Self-Calibrated Monthly PDSI written to ./monthly/self_cal/
   * Weekly CMI written to ./weekly/CMI/
Calculations Complete
[~/pdsi/test]$ head -3 monthly/self_cal/PDSI.tbl
 1893   0.18   0.65   0.98   1.57  -0.15  -0.67  -0.66  -0.68  -0.87  -0.98  -0.00   0.62
 1894   0.02  -0.09  -0.34  -0.01  -0.22  -0.73  -1.24  -1.63  -2.01  -1.95  -2.18  -2.12
 1895  -2.12  -2.26  -2.60  -2.44  -2.63   0.40   0.18   0.37   0.84  -0.21  -0.20  -0.44
```

Notes: If the program is run without using the flags to specify the location of the input files, it looks in the current working directory. Thus, all input files should be located in the same directory.

### Using the '-i' and '-o' flags

```
[~/pdsi/test]$ mv weekly_T weekly_P wk_T_normal weekly/
[~/pdsi/test]$ cp parameter weekly
[~/pdsi/test]$ mv monthly_T monthly_P T_normal parameter monthly/
[~/pdsi/test]$ ls
makefile  monthly  pdsi  pdsi.cpp  weekly
[~/pdsi/test]$ ./pdsi -iweekly/ -otest/output/directory/
   * Self-Calibrated Weekly PDSI written to test/output/directory/weekly/1/
   * Self-Calibrated 2-Week PDSI written to test/output/directory/weekly/2/
   * Self-Calibrated 4-Week PDSI written to test/output/directory/weekly/4/
   * Self-Calibrated 13-Week PDSI written to test/output/directory/weekly/13/
   * Cannot calculate the Monthly PDSI.
   * Cannot calculate the Self-Calibrated Monthly PDSI.
   * Weekly CMI written to test/output/directory/weekly/CMI/
Calculations Complete
[~/pdsi/test]$ ls
makefile  monthly  pdsi  pdsi.cpp  test  weekly
[~/pdsi/test]$ ls test/output/directory/
weekly
```

Notes: Since the monthly input files were not in the weekly directory, the monthly PDSI was not calculated.

## Using the '-f' flag

```
[~/pdsi/test]$ ./pdsi -f both -imonthly/
   * Cannot calculate the weekly PDSI.
   * Cannot calculate the weekly PDSI.
   * Cannot calculate the weekly PDSI.
   * Cannot calculate the weekly PDSI.
   * Monthly PDSI written to ./monthly/original/
   * Self-Calibrated Monthly PDSI written to ./monthly/self_cal/
   * Cannot calculate the weekly CMI.
Calculations Complete
[~/pdsi/test]$ ls monthly/self_cal/
PDSI.clm  PDSI.tbl  PHDI.clm  PHDI.tbl  WPLM.clm  WPLM.tbl  ZIND.clm  ZIND.tbl
[~/pdsi/test]$ head -3 monthly/original/PDSI.tbl
 1893   0.25   0.90   1.35   2.16  -0.28  -1.29  -1.21  -1.23  -1.55  -1.73  -0.00   0.86
 1894   0.02  -0.16  -0.64   0.59  -0.41  -1.37  -2.31  -2.98  -3.64  -3.42  -3.77  -3.56
 1895  -3.48  -3.69  -4.26  -3.87  -4.16  -3.18  -3.08  -2.49  -1.53  -1.77  -1.62  -1.94
[~/pdsi/test]$ head -3 monthly/original/PDSI.clm
 1893    1   0.25
 1893    2   0.90
 1893    3   1.35
```

## Setting the start year

```
[~/pdsi/test]$ ./pdsi 1895 -imonthly/ -s
[~/pdsi/test]$ head -3 monthly/original/PDSI.tbl
 1895  -3.48  -3.69  -4.26  -3.87  -4.16  -3.18  -3.08  -2.49  -1.53  -1.77  -1.62  -1.94
 1896  -2.06  -2.32  -2.20   1.15   1.82   2.96   3.81   3.37   3.09   3.69   4.92   4.48
 1897   5.37   4.98   5.58   6.03   4.65   4.84   4.79   4.12   3.02   2.70   2.35   3.39
```

Notes: The program still calculates the PDSI from the start year, but just outputs the data starting from the specified year.

## Using the '-e' flag

```
[~/pdsi/test]$ ./pdsi -se 1989 -imonthly/
[~/pdsi/test]$ tail -3 monthly/self_cal/PDSI.tbl
 1987   1.06   0.77   2.02  -0.39  -0.69  -0.63  -0.44  -0.54   0.05  -0.31  -0.30  -0.13
 1988  -0.03  -0.23  -0.61   0.17  -0.15  -0.90  -1.07  -1.03  -1.02  -1.33  -1.42  -1.65
 1989  -1.69  -1.87  -2.08  -2.46  -2.92  -3.24  -3.54  -3.64  -3.51  -3.72  -3.78  -3.75
```

## Setting the Tolerance

```
[~/pdsi/test]$ ./pdsi 0.2 -se 1989 -imonthly/
[~/pdsi/test]$ tail -3 monthly/self_cal/PDSI.tbl
 1987   1.06   0.77   2.02  -0.39  -0.69  -0.63  -0.44  -0.54   0.05  -0.31  -0.30   0.20
 1988   0.30  -0.23  -0.61   0.17  -0.15  -0.90  -1.07  -1.03  -1.02  -1.33  -1.42  -1.65
 1989  -1.69  -1.87  -2.08  -2.46  -2.92  -3.24  -3.54  -3.64  -3.51  -3.72  -3.78  -3.75
```

Notes: Notice the difference in Dec 1987 and Jan 1988.

## Setting both start and end year

```
[~/pdsi/test]$ ./pdsi -imonthly -s 1995 1999
[~/pdsi/test]$ cat monthly/self_cal/PDSI.tbl
 1995  -0.31  -0.53   0.25   1.19   1.94   1.33   0.95   1.25   1.28   2.02   2.07  -0.09
 1996  -0.00  -0.13  -0.12  -0.34  -0.28  -0.42   0.69   0.79   0.98   0.81   1.80   1.94
 1997   1.95   2.07   1.78   1.86   2.46   1.78   2.23   1.82   2.17   2.80   2.42   2.28
 1998   2.13   1.81   2.36   3.06   2.56   2.80   2.58   2.42   1.64   2.29   2.74   2.51
 1999   2.29   2.30   2.02   2.69   2.93   4.10   5.77  -0.26  -0.59  -0.77  -1.08  -1.26
```

## Setting the total number of years

```
[~/pdsi/test]$ ./pdsi 3 -imonthly/ -s
[~/pdsi/test]$ cat monthly/self_cal/WPLM.tbl
 1893   0.18   0.65   0.98   1.57   0.84  -0.67  -0.63  -0.68  -0.87  -0.98  -0.84   0.62
 1894   0.30  -0.09  -0.34   0.42  -0.22  -0.73  -1.24  -1.63  -2.01  -1.95  -2.18  -2.12
 1895  -2.12  -2.26  -2.60  -2.43  -2.63  -1.75  -1.84  -1.32  -0.28  -0.70  -0.62  -1.08
[~/pdsi/test]$ ./pdsi 3 -imonthly/ -s 1990
[~/pdsi/test]$ cat monthly/self_cal/WPLM.tbl
 1990  -3.68  -3.85  -4.01  -4.17  -3.21  -3.89  -4.01  -4.06  -4.23  -3.97  -3.88  -3.81
 1991  -3.75  -3.87  -3.97  -3.77  -3.51  -3.10  -3.37  -3.62  -3.69  -3.50  -2.59  -2.63
 1992  -2.48  -2.15  -1.18  -1.31  -1.78  -1.26  -0.29   1.96   2.38   3.59   3.78   3.94
[~/pdsi/test]$ ./pdsi 3 -imonthly/ -s 1948 2000
[~/pdsi/test]$ cat monthly/self_cal/WPLM.tbl
 1948  -0.18   0.48   0.19  -0.42  -0.90   0.21   0.19   0.16  -0.94  -0.91   0.41   0.56
 1949   1.23   0.43   1.79   1.00   0.95  -0.10   0.43  -0.45   1.04   0.58  -0.34  -0.40
 1950  -0.40  -0.24   0.28  -0.21   0.49   1.06   1.30   1.28   0.66   0.13  -0.13  -0.40
```

## Using the ‘-x’ flag

```
[~/pdsi/test]$ ./pdsi -imonthly -s -x wb
[~/pdsi/test]$ ls -1 monthly/self_cal/
PDSI.tbl
PHDI.tbl
WB.tbl
WPLM.tbl
ZIND.tbl
[~/pdsi/test]$ ./pdsi -imonthly -s -x potentials
[~/pdsi/test]$ ls -1 monthly/self_cal/
dvalue
PDSI.tbl
PHDI.tbl
potentials
WPLM.tbl
ZIND.tbl
[~/pdsi/test]$ ./pdsi -imonthly -s -x Xtable
[~/pdsi/test]$ ls -1 monthly/self_cal/
bigTable.tbl
PDSI.tbl
PHDI.tbl
WPLM.tbl
ZIND.tbl
[~/pdsi/test]$ ./pdsi -imonthly -s -x all
[~/pdsi/test]$ ls monthly/self_cal/
bigTable.tbl  dvalue  PDSI.tbl  PHDI.tbl  potentials  WB.tbl  WPLM.tbl  ZIND.tbl
```

## Input Files

Currently, the program requires several specifically named input files. The download package from http://nadss.unl.edu comes with the seven input files that are needed to calculate both the monthly PDSI and the weekly PDSI. These seven input files are:
monthly_T, monthly_P, mon_T_normal, weekly_T, weekly_P, wk_T_normal, and parameter.

To just calculate the monthly PDSI, four of the seven files are needed:

- monthly_T
- monthly_P
- mon_T_normal or T_normal
- parameter

To just calculate the weekly PDSI, four of the seven files are needed:

- weekly_T
- weekly _P
- wk_T_normal or T_normal
- parameter

### *monthly_T and weekly_T*

These files hold the temperature data for a station. Each line starts with the year and is followed by 12 (52) temperature entries. The temperature is the average monthly (weekly) temperature for each of the 12 months (52 weeks) of that year. Shown below are the first three lines from monthly_T.

```
1893 14.371 14.000 26.435 44.250 54.645 70.683 73.355 69.323 63.600 48.806 32.867 19.161
1894 12.705 14.979 37.984 49.700 61.209 71.463 77.935 74.312 65.283 51.516 34.767 29.548
1895 12.519 17.964 33.994 54.506 60.411 66.172 70.548 69.622 65.288 44.795 32.433 23.333
```

### *monthly_P and weekly_P*

These files hold the precipitation data for a station. It's the same format as the temperature files; each line starts with the year and is followed by 12 (52) temperature values. The precipitation values are the total monthly (weekly) precipitation for each of the 12 months (52 weeks) of that year. Shown below are the first three lines from monthly_P.

```
1893  0.610  1.500  1.730  4.050  1.950  0.790  3.020  2.570  1.430  0.850  1.260  2.350
1894  0.700  0.550  0.580  4.240  2.430  1.150  0.580  1.480  0.550  1.760  0.050  1.000
1895  0.650  0.540  0.520  3.980  2.380  6.240  2.320  3.920  4.770  0.060  1.040  0.000
```

## mon_T_normal and wk_T_normal

These files have the normal temperature data for a station. It has only 12 (52) entries, all on one line. The values in the file are the normal, or average, temperature over all the years on record for each of 12 months (52 weeks). This file can be named T_normal, to comply with the original FORTRAN program. The program will open the file named 'T_normal' first (if it exists), using it only if there are the correct number of entries in it. Shown below are the entire contents of mon_T_normal.

```
19.693  23.849  34.988  49.082  60.467  70.074  75.505  73.478  64.484  52.634  36.201  24.267
```

## parameter

The parameter file contains two numbers. The first number should always be the Available Water Holding Capacity (referred to in this document as the AWC, but also referred to elsewhere as AWHC or RZWHC). The second number, by default, should be the latitude of the station. The latitude should be given in decimal degrees rather than degrees, minutes, seconds. In other words, the value 42.60 is valid while 42:36:00 is not valid. Below is an example of the parameter file in normal format.

```
12.180000   42.60
```

If the '-n' flag is used, the second number should be the negative tangent of the latitude, which is known as the TLA (for unknown reasons). Below is what the information given in the example above would look like in this format.

```
12.180000   -0.919547
```

## Missing Values

If data is not available, it can be represented in the input files by using the value –99.00. This will cause the program to skip that period as if it didn't exist. Since the PDSI is an accumulating index (that is, one value depends directly on the previous value), large gaps in the data will have adverse effects. The PDSI also depends on long-term averages used in the water balance equation, so using a small amount of data is also not a good idea. The program requires at least 25 years of data in order to do any calculations, but it does not check to see how many holes in the data there are, or how big they are.

You cannot insert the –99.00 value into either the normal temperature or the parameter file.

The AWC is very important to the calculations of the PDSI, so try to use the most accurate information available. A good source for AWC information is the USDA's STATSGO dataset. If you do not have AWC information, do not try to guess what it might be. You should contact a local agronomist or the county extension office.

## Output Files

The PDSI calculations actually result in four different indices, the PDSI, the Weighted PDSI (WPLM), the Palmer Hydrological Drought Index (PHDI) and the Palmer Z-index (ZIND). In addition to the different indices produced, the PDSI program will also attempt to calculate several different types of the PDSI (the original monthly PDSI, the self-calibrated monthly PDSI, the weekly PDSI, etc.). To prevent the program from overwriting output files created for a different PDSI type, the program will always output each type of PDSI to its own subdirectory.

| PDSI Type | Subdirectory |
|---|---|
| Original Monthly | `monthly/original/` |
| Self-Calibrated Monthly | `monthly/self_cal/` |
| Self-Calibrated Weekly | `weekly/1/` |
| Self-Calibrated 2-Week | `weekly/2/` |
| Self-Calibrated 4-Week | `weekly/4/` |
| Self-Calibrated 13-Week | `weekly/13/` |
| Weekly CMI | `weekly/CMI/` |

### *Default PDSI Output Files*

There are two different formats that the PDSI program uses when creating output files from PDSI calculations. The default format is the table format, which is the same as the format of the monthly_T or weekly_T input files. Each line contains one year of data. The first number is the year, which is followed by a number for each of the 12 months or 52 weeks of the year. The four different indices are written into PDSI.tbl, WPLM.tbl, PHDI.tbl, and ZIND.tbl.

The second format is column format. In this format, each index value has its own line. Each line contains the year, the month (or week), and the value for the index. The four different indices are written to PDSI.clm, WPLM.clm, PHDI.clm, and ZIND.clm.

An example of what the two formats of the output files look like is given in the **Usage Examples** section.

### *Extra PDSI Output Files*

There are a handful of extra files that can be output by the program if specified using the '-x' command line flag. How the '-x' flag is used and what files it creates are shown in the section **Using the '-x' flag**. The contents of each file are described below.

#### -x potentials

This option creates two additional files, one call 'potentials' and one called 'dvalue'. The first part of each of the two files is shown below.

potentials

```
Year  MONTH        P          PE          PR          PRO         PL        P-PE
 1893     1    0.610000   0.000000   0.000000  12.180000   0.000000   0.610000
 1893     2    1.500000   0.000000   0.000000  12.180000   0.000000   1.500000
 1893     3    1.730000   0.000000   0.000000  12.180000   0.000000   1.730000
```

The potentials file holds the calculated potential values for each month (or week). The first column, labeled **P**, is the precipitation for that period. The second, labeled **PE**, is the potential evapotranspiration (Thornthwaite's). The third column is labeled **PR** and is the potential recharge of the soil moisture. Fourth is the potential runoff, labeled **PRO**, followed by **PL**, which is potential loss. The last column is **P-PE**.

dvalue

```
1893 1 0.427254
1893 2 1.264103
1893 3 1.245328
1893 4 2.395174
```

The file called dvalue holds the moisture departure for each month (or week). The moisture departure roughly described the difference between the precipitation that fell and what was needed to maintain normal soil moisture conditions.

## -x wb

This produces a file containing the water balance coefficients, which are called **α**, **β**, **γ**, and **δ**. These values are calculated based on long-term averages and then combined with the potential values to determine the amount of precipitation needed to maintain normal soil moisture conditions. This is then used to compute the moisture departure. Below is the complete monthly version of WB.tbl.

```
PERIOD   ALPHA     BETA    GAMMA    DELTA
  1     1.0000   0.1179   0.0150   0.0000
  2     1.0000   0.1771   0.0194   0.0599
  3     0.9984   0.3006   0.0398   0.1533
  4     0.9893   0.2280   0.0613   0.1300
  5     0.9654   0.1949   0.0693   0.2151
  6     0.9353   0.1042   0.0319   0.2643
  7     0.8476   0.0177   0.0131   0.5180
  8     0.7923   0.0453   0.0053   0.4662
  9     0.7799   0.0703   0.0129   0.3663
 10     0.8392   0.0804   0.0135   0.3789
 11     0.8986   0.1447   0.0139   0.2458
 12     1.0000   0.1228   0.0104   0.0000
```

## -x Xtable

This option will produce a file called bigTable.tbl.

```
YEAR  MONTH      Z     %Prob     X1       X2       X3
 1893     1     0.65    0.00     0.18     0.00     0.00
 1893     2     1.74    0.00     0.00     0.00     0.65
 1893     3    1.42    0.00    0.00    0.00    0.98
```

The column labeled ‘Z’ is the Z-index, which is also called the moisture anomaly. The next column is labeled ‘%Prob’ and shows the probability of the current spell ending. The ‘X1’ value is the severity of a wet spell that may be becoming established, ‘X2’ is the severity of a dry spell that may be becoming established, and ‘X3’ is the severity of a currently established spell. This information is quite useful in giving clues about whether or not the current spell will end, which will trigger backtracking.

### -x all

This option is used to produce all extra PDSI output files.

## *CMI Output Files*

The CMI calculations will always produce four output files:

- CMI.tbl
- potentials
- WB.tbl
- CMI_calc.tbl

CMI.tbl has the same format as the PDSI table output files. The potentials and WB.tbl files are exactly the same as those output from the weekly PDSI calculations.

The file named CMI_calc.tbl is designed to give the information from all the intermediate calculations performed. There is one line for each week, and each line contains the following information:

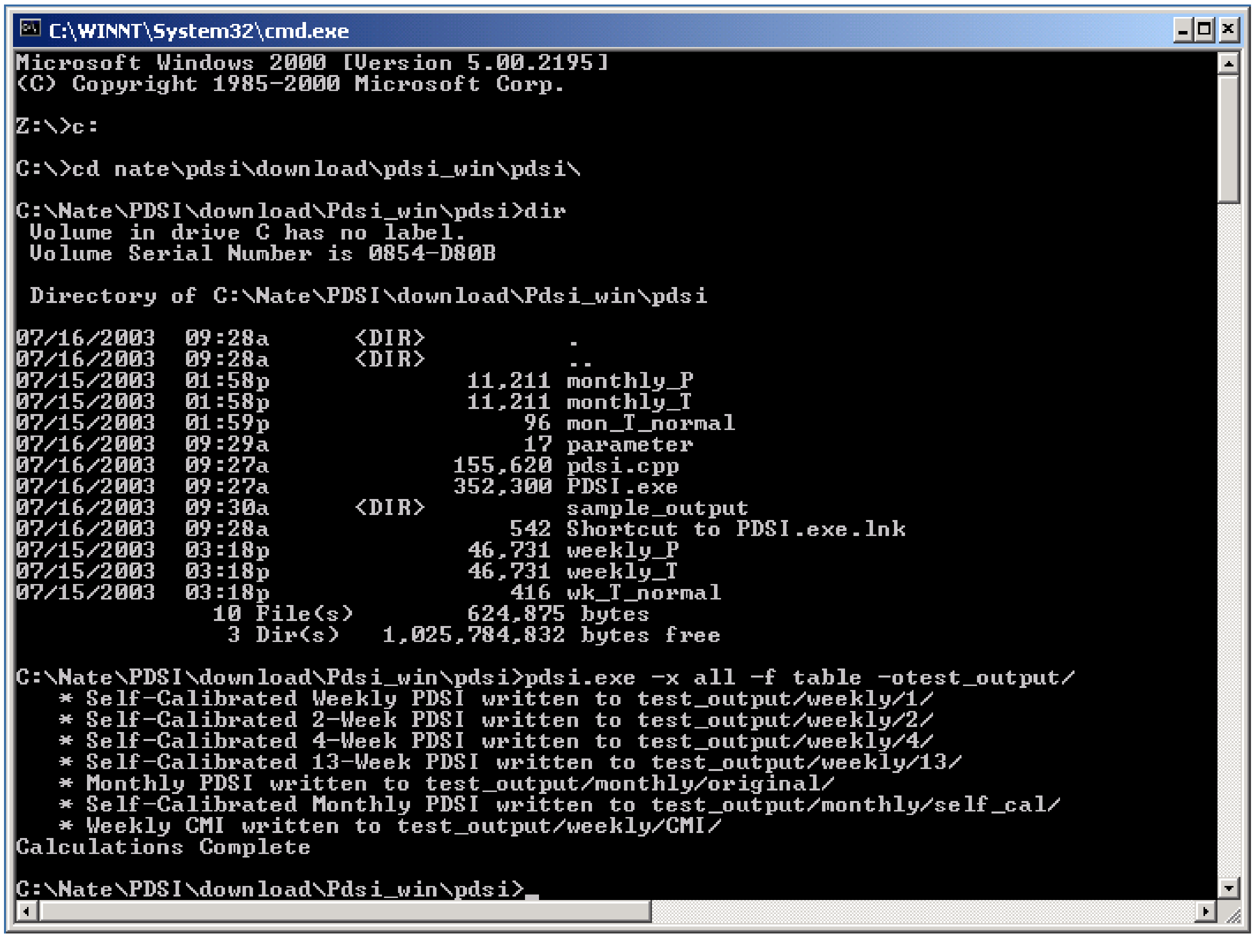
| | |
|---|---|
| year | |
| week | |
| PET | Potential Evapotranspiration |
| ET | Computed 'Actual' evapotranspiration |
| Alpha | The water balance coefficient of evapotranspiration |
| R | The total computed recharge |
| RO | The total computed runoff |
| Ss | Inches of water stored in the top layer of soil |
| Su | Inches of water stored in the lower layer of soil |
| M | Average percent of field capacity during the week |
| DE | Relative evapotranspiration anomaly for the week |
| Yprime | First approximation of Y |
| Y | Index of evapotranspiration deficit |
| H | A "return to normal" factor |
| G | Index of excessive moisture |
| CMI | The Crop Moisture Index |

## Working in Windows

The PDSI program was designed and tested in a Unix environment, so some of its features are not very well suited to Windows. For example, there is no easy way to use the command line flags that add so much functionality to the program. Here are some tips and tricks to get the most out of the PDSI program while running it under Windows.

### *Running the PDSI program via a command window*

One way to run the PDSI program with command line options is to open a command window. This is done by running the command "`cmd`" from the "`Run`" dialog box, reached from the start menu. By using DOS commands, like "`cd`" and "`dir`", you can navigate to the directory where the PDSI program is located.

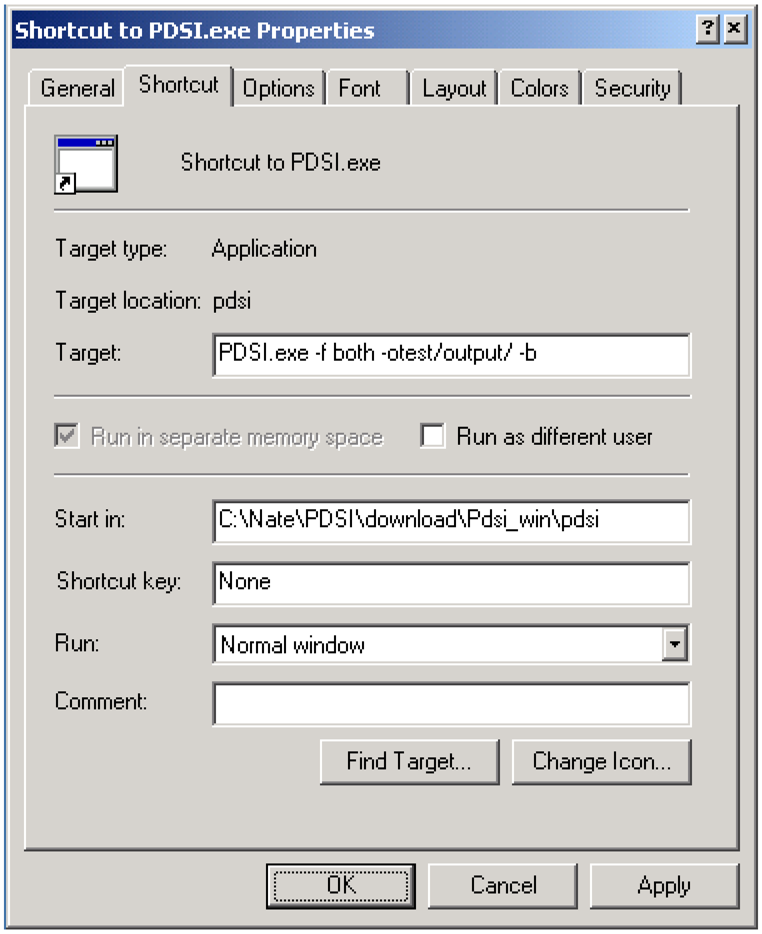


```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

Z:\>c:

C:\>cd nate\pdsi\download\pdsi_win\pdsi\

C:\Nate\PDSI\download\Pdsi_win\pdsi>dir
 Volume in drive C has no label.
 Volume Serial Number is 0854-D80B

 Directory of C:\Nate\PDSI\download\Pdsi_win\pdsi

07/16/2003  09:28a      <DIR>          .
07/16/2003  09:28a      <DIR>          ..
07/15/2003  01:58p              11,211 monthly_P
07/15/2003  01:58p              11,211 monthly_T
07/15/2003  01:59p                  96 mon_T_normal
07/16/2003  09:29a                  17 parameter
07/16/2003  09:27a             155,620 pdsi.cpp
07/16/2003  09:27a             352,300 PDSI.exe
07/16/2003  09:30a      <DIR>          sample_output
07/16/2003  09:28a                 542 Shortcut to PDSI.exe.lnk
07/15/2003  03:18p              46,731 weekly_P
07/15/2003  03:18p              46,731 weekly_T
07/15/2003  03:18p                 416 wk_T_normal
              10 File(s)        624,875 bytes
               3 Dir(s)   1,025,784,832 bytes free

C:\Nate\PDSI\download\Pdsi_win\pdsi>pdsi.exe -x all -f table -otest_output/
    * Self-Calibrated Weekly PDSI written to test_output/weekly/1/
    * Self-Calibrated 2-Week PDSI written to test_output/weekly/2/
    * Self-Calibrated 4-Week PDSI written to test_output/weekly/4/
    * Self-Calibrated 13-Week PDSI written to test_output/weekly/13/
    * Monthly PDSI written to test_output/monthly/original/
    * Self-Calibrated Monthly PDSI written to test_output/monthly/self_cal/
    * Weekly CMI written to test_output/weekly/CMI/
Calculations Complete

C:\Nate\PDSI\download\Pdsi_win\pdsi>_
```

One of the benefits of running the PDSI program in this manner is that if any error occurs, the error message will remain on the screen, describing the problem in detail. Conversely, when the program is run normally in Windows and an error occurs, Windows will close the execution window immediately, leaving the user no idea what the error was.

## *Using a shortcut to add command line options*

Another way to use the command line options of the PDSI program in a windows environment is to add the options in a shortcut to the program. A shortcut can be created by right-clicking on the icon of the PDSI program. Once a shortcut has been created, options can be added to the end of the 'Target" field in the properties of the shortcut.



Shortcut to PDSI.exe

Target type: Application

Target location: pdsi

Target: PDSI.exe -f both -otest/output/ -b

[✓] Run in separate memory space

[ ] Run as different user

Start in: C:\Nate\PDSI\download\Pdsi_win\pdsi

Shortcut key: None

Run: Normal window

Comment:

## Bugs

After extensive testing, the PDSI program is performing within a 0.011 tolerance of the FORTRAN version, using the '-b' flag to replicate the bugs in the FORTRAN program. A few discrepancies have shown up in the testing however. These are all due to miniscule differences in various computations throughout the program. All identified discrepancies are defined below.

### *Backtracking Discrepancies*

Backtracking occurs in two basic instances. The first instance is when the probability of an established wet or dry spell ending reaches 100%. The program then backtracks to decide what the previous PDSI values should be. The second instance happens on transition from no established wet or dry spell to an established spell. The program backtracks to decide upon the actual PDSI for previous months. The backtracking process itself is simple. The program chooses the appropriate index (X1 or X2) to start the backtracking. Then the backtrack function uses that index until it reaches zero. It then switches to the other index and repeats. The problem occurs when the index is approximately zero. The FORTRAN and C++ programs do not always agree on if this value is zero or not. This causes the backtracking function to stay with the incorrect index, thus incorrectly assigning PDSI values for previous months.

### *Probability Discrepancies*

This discrepancy exists in the process of calculating the probability of a wet or dry spell ending. When this probability is approximately 100%, the FORTRAN program calculates the percentage at around 99.999%. The C++ program calculates the value as 100%. This causes the PDSI values to differ considerably for months, and possibly years afterwards. Theoretically this could also happen as the probability approached 0%, but it has not yet been seen in testing.

### *WPLM Discrepancies*

This discrepancy shows up in calculations of the WPLM when the book keeping indices X1 and X2 are approximately equal. Since the different programs use different internal computations, what each decides is the larger index does not always coincide. This error has no effect on the following months because the WPLM is only dependent on the current months data. These discrepancies are all caused by the same basic dilemma. When doing a comparison, should significant figures be considered? The temperature inputs are all given with 2 decimal places of significance. This limits the accuracy of any value calculated to 2 decimal places. In general, no rounding is done to account for this until final results are reached. This poses a problem when dealing with inequalities. To provide better handling of these comparisons a tolerance flag was added. This allows the user to decide how accurate these comparisons are.