

APPLICATION NOTE



A Better Way to Write High-Frequency Data to 16 GB and Smaller CF Cards

11/12

A Better Way to Write High-Frequency Data to 16 GB and Smaller CF Cards

A new method for writing high-frequency time-series data to CompactFlash (CF) cards is now available for Campbell Scientific CR1000, CR3000, and CR5000 dataloggers. It supports 16 GB or smaller CF cards. It improves the user interface by allowing smaller, user determined file sizes. This may be the most suitable method for writing to CF cards, especially in high speed measurement applications.

TableFile() with Option 64

Option 64 has been added as a format option for the CRBasic instruction **TableFile()**. It combines the speed and efficiency of the **CardOut()** instruction with the flexibility of the **TableFile()** instruction. CF cards up to 16 GB are supported. **TableFile()** with **Option 64, TOB3** is now available in the following datalogger operating systems:

- CR1000, OS 25 or greater
- CR3000, OS 25 or greater
- CR5000, OS 7 or greater (OS 7 available in Spring of 2013)

TableFile() is a CRBasic instruction that creates a file from a data table in datalogger CPU memory. **Option 64** directs that the file be written in TOB3 format exclusively to the CRD: drive¹.

Syntax for the **TableFile()** instruction is as follows:

```
TableFile(FileName, Option, MaxFiles, NumRecs/  
TimeIntoInterval, Interval, Units, OutStat, LastFileName)
```

where **Option** is given the argument of **64**. Refer to CRBasic Editor Help² for a detailed description of each parameter.

NOTE

The CRD: drive (the drive designation for the optional CF card) is the only drive that is allowed for use with **Option 64**.

¹The CRD: drive is a memory drive created when a CF card is connected to the datalogger through the appropriate peripheral device. CR5000 dataloggers are supplied with an integrated PCMCIA card slot. This slot is adapted for CF card use by insertion of the CF1 module. The CR1000 and CR3000 dataloggers are adapted for CF use by addition of the NL115 or CFM100 modules. CF1, NL115, and CFM100 modules are available at additional cost from Campbell Scientific.

²CRBasic Editor is included in Campbell Scientific datalogger support software suites LoggerNet, PC400, and RTDAQ.

CAUTION

Larger cards are primarily intended for storing high-frequency data in applications wherein storage cards are changed out frequently. Large cards may also be used in applications where the site cannot be accessed for extended periods. However, large CF cards do not eliminate the risk of data loss.

TableFile() with Option 64 Replaces CardOut()

TableFile() with *Option 64* has several advantages over **CardOut()** when used in most applications. These include:

- Allowing multiple small files to be written from the same data table so that storage for a single table can exceed 2 GB. **TableFile()** controls the size of its output files through the *NumRecs*, *TimeIntoInterval*, and *Interval* parameters.
- Faster compile times when small file sizes are specified.
- Easy retrieval of closed files via File Control utility, FTP, or E-mail.

TableFile() with Option 64 Programming

As shown in the following CRBasic code segment, the **TableFile()** instruction must be placed inside a **DataTable()** / **EndTable** declaration. The **TableFile()** instruction writes data to the CF card based on user-specified parameters that determine the file size based on number of records to store, or an interval over which to store data. The resulting file is saved with a suffix of **X.dat**, where **X** is a number that is incremented each time a new file is written.

```
DataTable(TableName,TriggerVariable,Size)
  TableFile(FileName, Option, MaxFiles, NumRecs/TimeIntoInterval, Interval,
            Units, OutStat, LastFileName)

  'Output processing instructions go here

EndTable
```

For example, in micrometeorological applications, **TableFile()** with *Option 64* is used to create a new high-frequency data file once per day. The size of the file created is a function of the datalogger scan frequency and the number of variables saved to the data table. For a typical eddy-covariance station, this daily file is about 50 MB large (10 Hz scan frequency and 15 IEEE4 data points).

The following CRBasic program shows how the instruction is used in micrometeorology eddy-covariance programs. The file naming scheme used in **TableFile()** in this example is customized using variables, constants, and text.

```
Public sensor(10)
DataTable(ts_data,TRUE,-1)
  'TableFile("filename",Option,MaxFiles,NumRec/TimeIntoInterval,Interval,Units,
            OutStat,LastFileName)
  TableFile("&CRD:&Status.SerialNumber(1,1)&".ts_data_",64,-1,0,1,Day,0,0)
  Sample(10,sensor(1),IEEE4)
EndTable

BeginProg
  Scan(100,mSec,100,0)

  'Measurement instructions go here.

  'Processing instructions go here.

  CallTable ts_data
NextScan
EndProg
```

Converting TOB3 Files with CardConvert

The TOB3 format that is used to write data to CF cards saves disk space. However, the resulting binary files must be converted to another format to be read or used by other programs. The CardConvert software, included in Campbell Scientific datalogger support software (LoggerNet, PC400, RTDAQ and PC200W), will convert data files from one format to another. CardConvert online Help has more details.

FAQs –TableFile() with Option 64

How does Option 64 differ from other TableFile() options?

Pre-allocation of memory combines with TOB3 data format to give *Option 64* two principal advantages over other **TableFile()** options. These are:

- increased runtime write performance
- short card eject times

Option 64 is unique among table file options in that it pre-allocates enough memory on the CF card to store an interval's amount of data¹. Pre-allocation allows data to be continuously and more quickly written to the card in ≈ 1 kB blocks. TOB3 binary format copies data directly from CPU memory to the CF card without format conversion, lending additional speed and efficiency to the data storage process.

¹Other options of **TableFile()** do not pre-allocate memory, so they should be avoided when collecting high-frequency time-series data. More information is available in CRBasic Editor Help.

NOTE

Pre-allocation of CF card files significantly increases run time write performance. It also reduces the risk of file corruption that can occur as a result of power loss or incorrect card removal.

CAUTION

To avoid data corruption and loss, CF card removal must always be initiated by pressing the initiate card removal button on the face of the NL115 or CFM100 modules (CR1000, CR3000) or through the keyboard and display on the CR5000. The card must only be ejected after the status light shows a solid green (NL115 or CFM100) or when instructed to do so from the display of the CR5000.

Why are individual files limited to 2 GB?

In common with many other systems, the datalogger natively supports signed-4-byte integers. This data type can represent a number as large as 2^{31} , or in terms of bytes, roughly 2 GB. This is the maximum file length that can be represented in the datalogger directory table.

Why does a large card cause long program compile times?

Program compile times increase with card and file sizes. As the datalogger boots up, the card must be searched to determine space available for data storage. In addition, for tables that are created by **TableFile()** with **Option 64**, an empty file that is large enough to hold all of the specified records must be created (i.e., memory is pre-allocated). When using **TableFile()** with **Option 64**, program compile times can be lessened by reducing the number of records or data interval that will be included in each file. For example, if the maximum file size specified is 2 GB, the datalogger must scan through and pre-allocate 2 GB of CF card memory. However, if smaller files are specified, then compile times are reduced because the datalogger is only required to scan through enough memory to pre-allocate memory for the smaller file.

Why does a freshly formatted card cause long compile times?

Program compile times take longer with freshly formatted cards because the cards use a FAT32 system (File Allocation Table with 32 table element bits) to be compatible with PCs. FAT32 uses an “info sector” to store the free cluster information. This info sector prevents the need to repeatedly traverse the FAT for the bytes free information. After a card is formatted by a PC, the info sector is not automatically updated. Therefore, when the datalogger boots up, it must determine the bytes available on the card prior to loading the Status table. Traversing the entire FAT of a 16 GB card can take 30 minutes or more when using a CR1000 datalogger (less with the CR3000 datalogger). However, subsequent compile times are much shorter because the info sector is used to update the bytes free information. To avoid long compile times on a freshly formatted card, the user can format the card on a PC, then copy a small file to the card, and then delete the file (while in the PC). Copying the file to the freshly formatted card forces the PC to update the info sector. The PC is several orders of magnitude faster than the datalogger at updating the info sector.

Which CF memory card should I use?

Campbell Scientific recommends and supports only the use of FMJ brand CF cards. These CF cards are industrial-grade and have passed Campbell Scientific hardware testing. Following are listed advantages FMJ brand CF cards have over less expensive commercial-grade cards:

- less susceptible to failure and data loss
- match the datalogger operating temperature range
- faster read/write times
- better vibration and shock resistance
- longer life spans (more read/write cycles)

NOTE More CF card recommendations are presented in the application note, *CF Card Information*, which is available at www.campbellsci.com.

Why not use SD cards?

CF cards offer advantages over Secure Digital (SD) cards, including ruggedness, ease of handling, and connection reliability.

Can closed files be retrieved remotely?

Yes. Closed files can be retrieved using the Retrieve function in LoggerNet, PC400, or RTDAQ File Control utility, FTP, HTTP, or e-mail. Although open files will appear in the CRD: drive directory, do not attempt to retrieve open files. Doing so may corrupt the file and result in data loss. Smaller files typically transmit more quickly and more reliably than large files.

Can data be accessed?

Yes. Data in the open or most recent file can be collected using the **Collect** or **Custom Collect** utilities in LoggerNet, PC400, or RTDAQ. Data can also be viewed using datalogger support software or accessed through the datalogger using data table access syntax such as **TableName.FieldName** (see CRBasic Editor Help). Once a file is closed, data can be accessed only by first retrieving the file, as discussed previously, and processing the file using CardConvert software.

What happens when a card is inserted?

When a card is inserted, whether it is a new card or the previously used card, a new file is always created.

What does a power cycle or program restart do?

Each time the program starts, whether by user control, power cycle, or a watchdog, **TableFile()** with **Option 64** will create a new file.

What happens when a card is filled?

If the CF card fills, new data are written over oldest data.

A card must be exchanged before it fills, or the oldest data will be overwritten by incoming new records and lost. During the card exchange, once the old card is removed, the new card must be inserted before the data table in datalogger CPU memory rings¹, or data will be overwritten and lost. For example, consider an application wherein the data table in datalogger CPU memory has a capacity for about 45 minutes of data². The exchange must take place anytime before the 45 minutes expire. If the exchange is delayed by an additional 5 minutes, 5 minutes of data at the beginning of the last 45 minute interval (since it is the oldest data) will be overwritten in CPU memory before transfer to the new card and lost.

¹“rings”: the datalogger has a ring memory. In other words, once filled, rather than stopping when full, oldest data are overwritten by new data. In this context, “rings” designates when new data begins to overwrite the oldest data.

²CPU data table fill times can be confirmed in the datalogger Status table.

Campbell Scientific Companies

Campbell Scientific, Inc. (CSI)

815 West 1800 North
Logan, Utah 84321
UNITED STATES

www.campbellsci.com • info@campbellsci.com

Campbell Scientific Africa Pty. Ltd. (CSAf)

PO Box 2450
Somerset West 7129
SOUTH AFRICA

www.csafrica.co.za • cleroux@csafrica.co.za

Campbell Scientific Australia Pty. Ltd. (CSA)

PO Box 8108
Garbutt Post Shop QLD 4814
AUSTRALIA

www.campbellsci.com.au • info@campbellsci.com.au

Campbell Scientific do Brazil Ltda. (CSB)

Rua Luisa Crapsi Orsi, 15 Butantã
CEP: 005543-000 São Paulo SP BRAZIL

www.campbellsci.com.br • suporte@campbellsci.com.br

Campbell Scientific Canada Corp. (CSC)

11564 - 149th Street NW
Edmonton, Alberta T5M 1W7
CANADA

www.campbellsci.ca • dataloggers@campbellsci.ca

Campbell Scientific Centro Caribe S.A. (CSCC)

300 N Cementerio, Edificio Breller
Santo Domingo, Heredia 40305
COSTA RICA

www.campbellsci.cc • info@campbellsci.cc

Campbell Scientific Ltd. (CSL)

Campbell Park
80 Hathern Road
Shepshed, Loughborough LE12 9GX
UNITED KINGDOM

www.campbellsci.co.uk • sales@campbellsci.co.uk

Campbell Scientific Ltd. (France)

3 Avenue de la Division Leclerc
92160 ANTONY
FRANCE

www.campbellsci.fr • info@campbellsci.fr

Campbell Scientific Spain, S. L.

Avda. Pompeu Fabra 7-9, local 1
08024 Barcelona
SPAIN

www.campbellsci.es • info@campbellsci.es

Please visit www.campbellsci.com to obtain contact information for your local US or international representative.