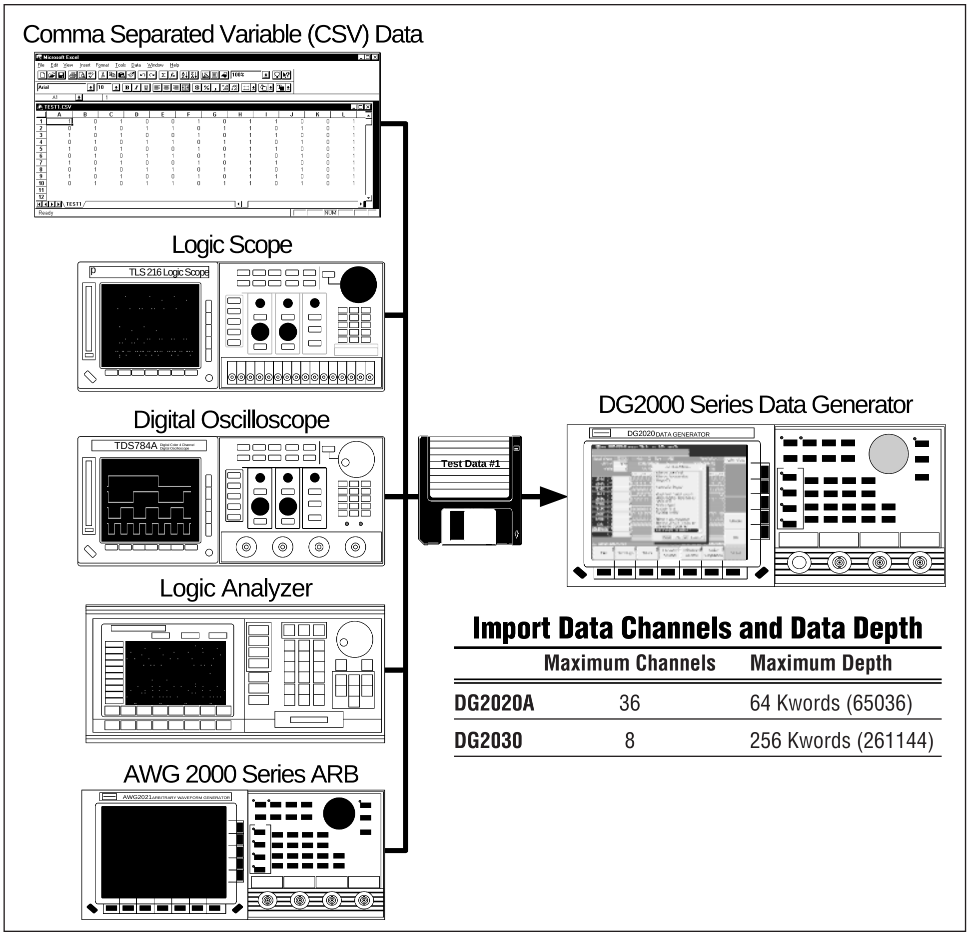
Tektronix

Application Note

Importing Data into the DG2000 Series



Import Data Channels and Data Depth

	Maximum Channels	Maximum Depth
DG2020A	36	64 Kwords (65036)
DG2030	8	256 Kwords (261144)

Figure 1. There are many different ways to import data into the DG2000 Series.

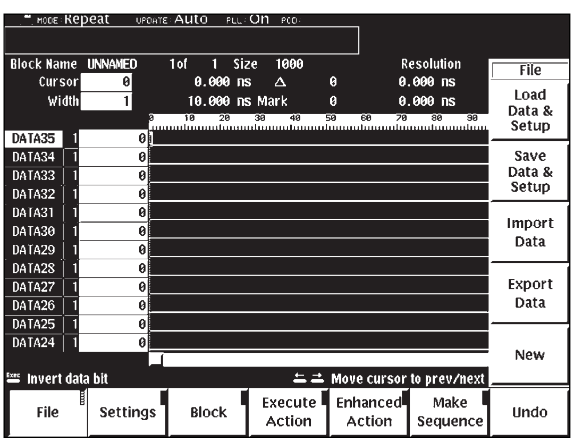


Figure 2. DG2020A Edit menu. The left-most selection on the bottom menu accesses the right side File menu which contains the Import Data and Export Data selections. Pressing either of these buttons prompts the user for a source or destination filename on the 3.5-inch floppy disk.

Introduction

The Tektronix DG2000 Series Data Generators can import data patterns or vectors from other sources to supplement their built-in pattern creation and editing functions. Waveform data can be imported from another instrument (e.g., logic analyzer, oscilloscope, etc.) that has captured actual signals. Another source of "raw" data is a software package (e.g., spreadsheet, simulator, etc.) that synthesizes the desired stimulus pattern.

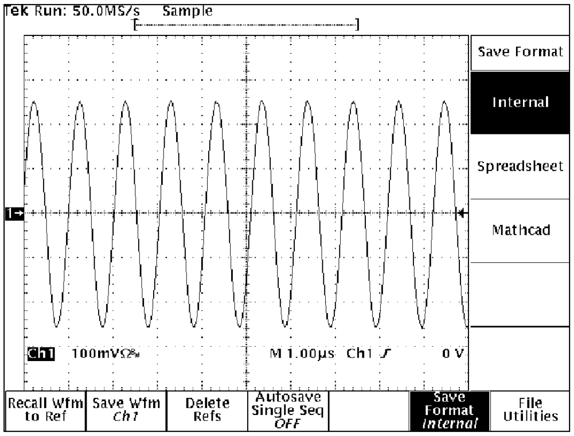
The DG2000 Series can directly read waveform files created by a number of Tektronix instruments including the TLS 216 logic scope, TDS oscilloscopes, GPX logic analyzers, and AWG 2000 arbitrary waveform generators. The DG2000 Series also reads a simple Comma Separated Variable (CSV) format which provides the link to virtually any source of data patterns. In each case, a 3.5-inch PC compatible floppy disk is used as the transport medium. Once the data is imported, you can generate the pattern "as-is" or you can modify the pattern using the DG2000 Series editing capabilities.

DG2000 Series Data Import

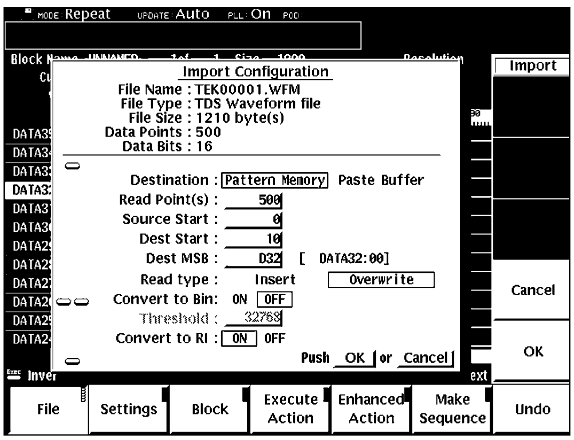
The data import function is accessed through the EDIT button which displays the EDIT menu (see Figure 2). When the File menu item is selected, a side menu appears with Import Data as one selection. Note that the side menu also has the Export Data selection which saves pattern data to the floppy disk for transfer to other applications. When the Import Data menu item is selected, the DG2000 Series lists the files available on the floppy disk. After a file is selected for import, the file is examined to determine the file format can be recognized.

The following sections step through examples of importing various file formats. Note that the import and export of data only transfers the data pattern itself. It doesn't transfer instrument setup parameters such as the physical voltage levels of the generated or captured signals.

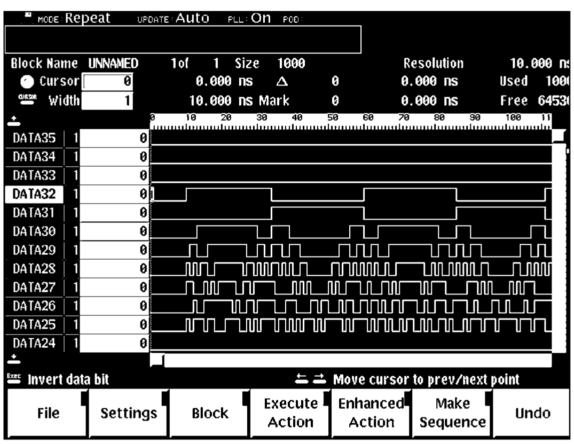
Data Import from a TDS Oscilloscope



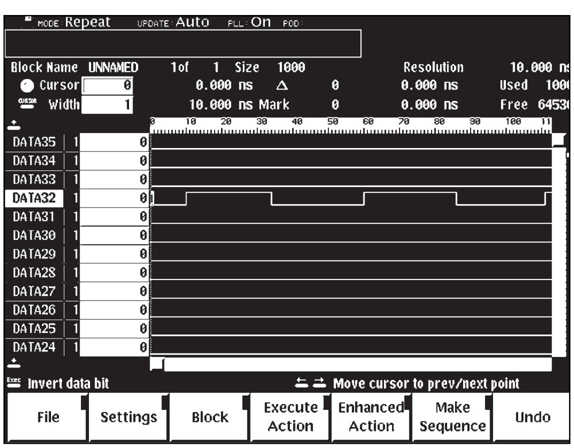
Step 1 – TDS data export. The TDS front-panel SAVE/RECALL WAVEFORM button accesses this menu. The bottom Save Wfm button is used to access the To File destination (right button) selection (not shown here). Save Format (bottom button) accesses the right menu shown here. Internal (right button) selects the Internal format for data transfer to the DG2000 Series. In this case the Channel 1 waveform is a 500-point record. The TDS saves each of the 500 points as a 16-bit value. The number of left-justified bits that is actually used depends on the TDS acquisition mode. In the Sample mode shown here eight of the 16-bit positions are used; the Averaging and Hi-Res acquisition modes have additional bits of resolution. The TDS stores files using the .WFM filename extension (TEK00001.WFM was used in this example).



Step 2 – DG2000 Series data import. After the file TEK00001.WFM is selected for data import, the DG2000 Series displays this Import Configuration menu. The DG2000 Series identifies the File Type as a waveform file and reads the 500 data points. All 500 points are selected for import directly into pattern memory. A smaller number of points can be selected. A starting point of 10 is selected in the DG2000 Series' pattern memory. The most significant bit of the imported data is selected to start at D32 within the 36-bit pattern memory range. The conversion to binary function is disabled in this case (its function is explained below).

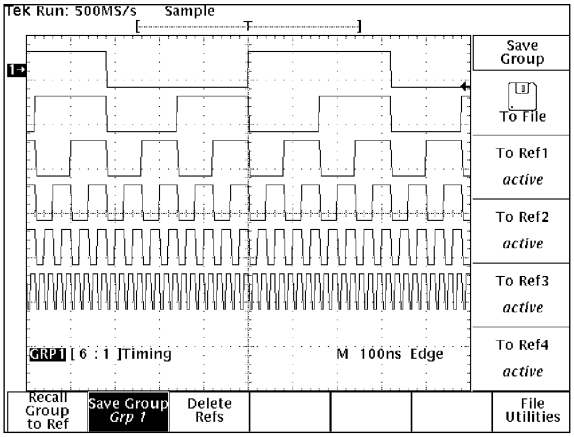


Step 3 – Result. The eight-bit waveform from the TDS is deposited into the DG2000 Series pattern memory. As specified in the Import Configuration above, the 500-point pattern starts at location 10 and the most-significant bit starts with DATA32; the least-significant bit is placed in DATA25. Note that the eight-bit pattern represent a quantized sinewave in this example. If these eight channels drove a digital-to-analog converter, the output should match the display on the TDS.

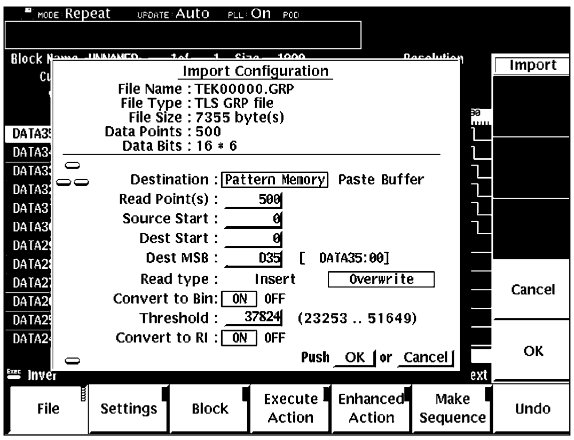


Step 4 – Alternative result. The waveform from the TDS can be converted to a single-bit binary result. If the Convert to Bin function is enabled in Step 2, the specified threshold value would be used to quantize the imported values to one bit of resolution. In this case, the threshold was specified at 32768 (half of the 16 bit full scale input range) so the result is essentially a squarewave at the same frequency as the original sinewave.

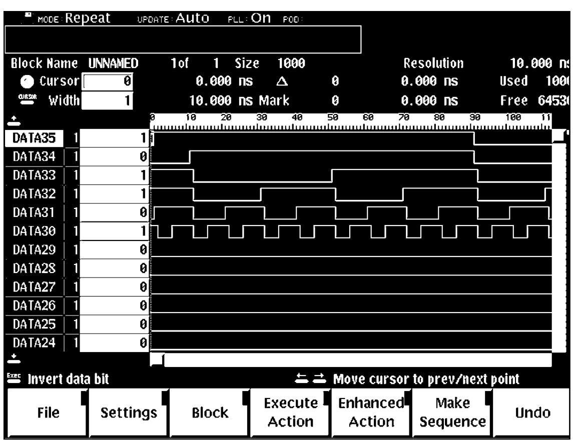
Data Import from a TLS Logic Scope



Step 1 – TLS data export. The TLS segments input channels into user-defined groups. In this case, Group 1 was defined to contain six channels which captured the output of a binary counter into a 500-word record. The TLS SAVE/RECALL WAVEFORM button accesses this screen. If the To File (right menu) item is selected, the data pattern is stored to the disk drive. The TLS saves the entire record using the .GRP filename extension. (TEK00000.GRP was used in this example).

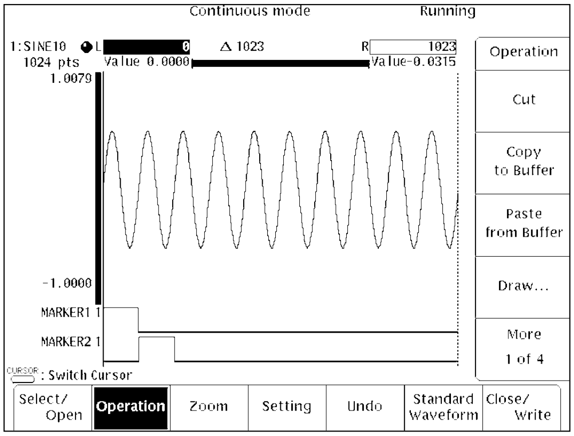


Step 2 – DG2000 Series data import. After the file TEK00000.GRP is selected for data import, the DG2000 Series displays this Import Configuration menu. The DG2000 Series identifies the File Type as a TLS GRP file and reads the 500 data points. The TLS saves each data point as a 16-bit word but the actual stored resolution depends on the acquisition mode. Since there were six saved channels, each point in time represents 16 x 6 bits. In this case, we specify conversion to binary with a threshold of 37824 within the total 16-bit range of 0 to 65535. The DG2000 Series displays the range of values encountered in the imported data stream. In this case the range was 23253 to 51649. Note that we must use the Convert to Bin function since the original raw data is 96 bits wide which exceeds the width of the DG2000 Series pattern memory.

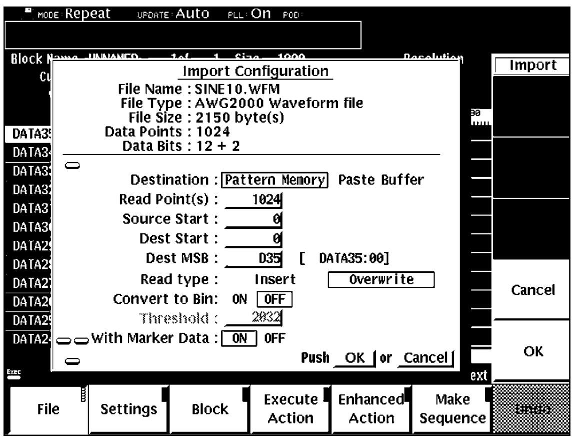


Step 3 – Result. The six imported channels are converted to binary format and are overwritten into the first six channels of the DG2000 Series pattern memory starting with DATA35. Approximately the first 100 locations of pattern memory are displayed in the timing display.

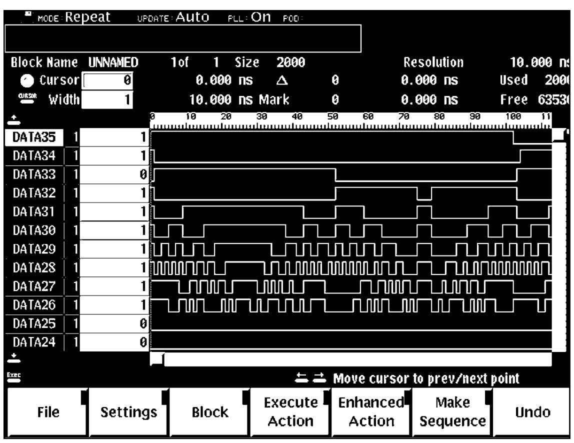
Data Import from an AWG 2000 Waveform Generator



Step 1 – AWG 2000 data export. Since the AWG 2000 is itself a signal generator, it has its own digital pattern memory. In this case, the 1024-point waveform SINE10 is displayed. In addition, the AWG can generate two independent binary marker signals (MARKER1 and MARKER2) that are generally used as trigger signals. Note that the two markers were defined to coincide with the first two cycles of the signal. An AWG waveform is transferred from the pattern memory to disk using the LOAD/SAVE button. In this case, the previously created waveform file SINE10 was transferred to the floppy disk as SINE10.WFM. The AWG saves its waveforms as a 12-bit value with two additional bits for the marker data.

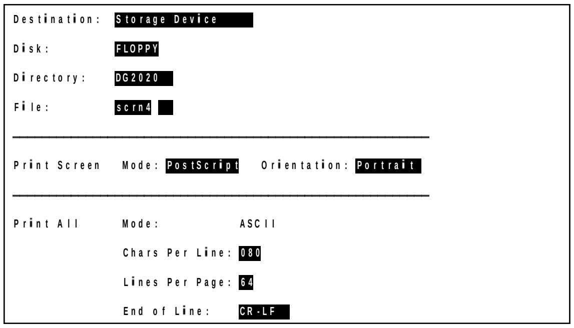


Step 2 – DG2000 Series data import. After the file SINE10.WFM is selected for data import, the DG2000 Series displays this Import Configuration menu. The DG2000 Series identifies the File Type as an AWG 2000 Waveform file and reads the 1024 data points. In this case, the Convert to Bin function is disabled so that the quantized representation of the sinewave is transferred. Note that the most-significant bit is specified to be D35 and that we import the Marker Data.

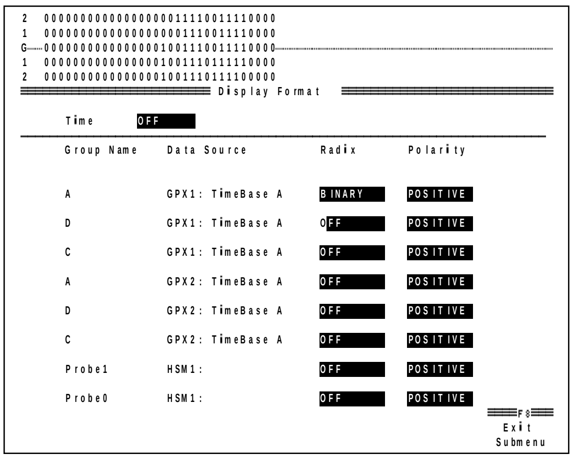


Step 3 – Result. The AWG waveform and the time-correlated marker data are imported. The two marker bits are written into the most-significant two bits (starting at D35 as specified). The data bits representing the sinewave start with D33. The Convert to Bin function was disabled so the quantized waveform was transferred. Note that the two marker signals still coincide with the first and second cycles of the 1024-point AWG sinewave pattern. In this screen, only the first ~100 points of the pattern are shown so only the first cycle of the quantized sinewave is completely shown.

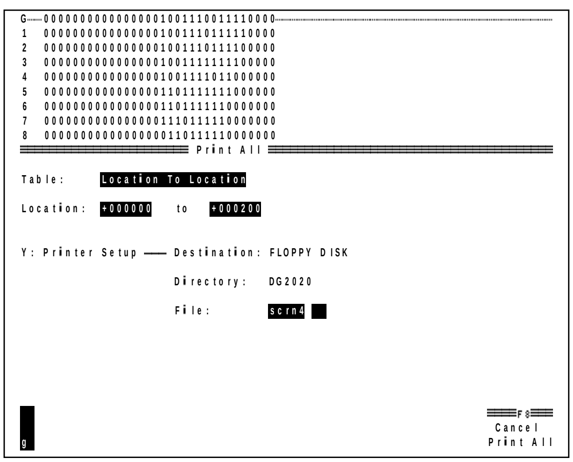
Data Import from a GPX Logic Analyzer



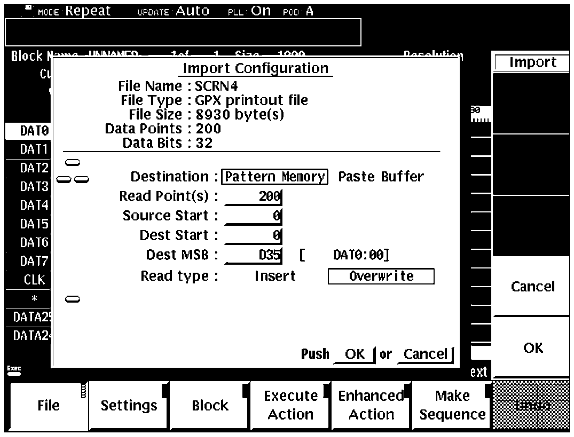
Step 1A – GPX data export. The data export procedure uses the ASCII print out function of the GPX. The UTIL button accesses the Utility menus which include the PRINTER SETUP menu. The parameters in the PRINTER SETUP menu should be set as shown. The destination is the FLOPPY Storage Device. Note that the destination file (scrn4 in this example) must reside in a directory on the diskette (DG2000 Series in this example). The Print Screen selection items are not relevant for data export. The Print All selections should be set to the ASCII mode as shown.



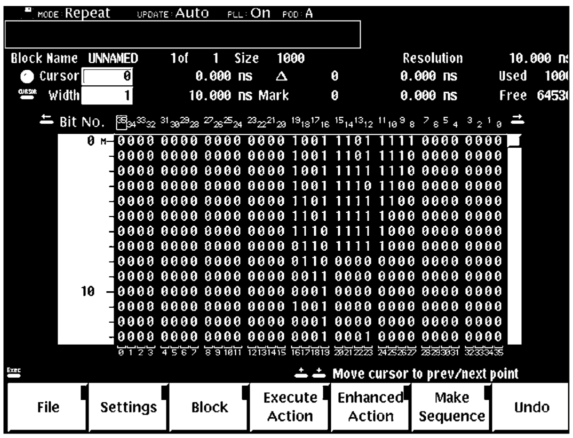
Step 1B – GPX data export. The DISP button accesses the Display menus which include the Display Format submenu shown here. The Time parameter should be disabled. The radix of the data group to be exported should be set to BINARY and the radix of other data groups should be turned OFF. Multiple data groups can be selected for export but note that the DG2000 Series pattern memory is only 36 bits wide for the DG2020A or 8 bits wide for the DG2030. In this case, the 32-bit GPX Group A is selected for export.



Step 1C – GPX data export. When the desired data has been captured, the data is “printed” to the export file. The Shift+Print key accesses the Print All menu shown here. The State Table is set to print from Location to Location and the data points are selected through the Print Location parameter. The floppy filename parameters are displayed as set in the previous menu. Note that the destination and directory parameters cannot be changed from this menu. To change these selections, use the PRINTER SETUP menu shown in Step 1A. The F1 Start Printing key initiates the actual data transfer to the floppy disk. In this case, the new filename is scrn4.



Step 2 – DG2000 Series data import. After the file scrn4 is selected for data import, the DG2000 Series displays this Import Configuration menu. The DG2000 Series identifies the File Type as a GPX printout file and reads the 200 data points. Recall that the GPX data pattern was 32 bits wide.



Step 3 – Result. The 32-bit pattern is displayed using the DG2000 Series binary display format. The lower rightmost four bits were not overwritten since the imported data was only 32 bits wide.

Data Import from a CSV File

The DG2000 Series can import data patterns from its built-in floppy drive using a simple ASCII comma-separated variable (CSV) text format (see example at right). This approach provides a universal transport format for data pattern sources from simulators, spreadsheets, test vector generators, and other instruments. In some cases, such as a spreadsheet program, the CSV format may be directly available as an output format option. The file can be saved to floppy disk and the data can be directly imported into the DG2000 Series. In other cases, a custom program must be written to convert a proprietary file format to the DG2000 Series CSV format. However, the DG2000 Series CSV format is very simple to understand and implement.

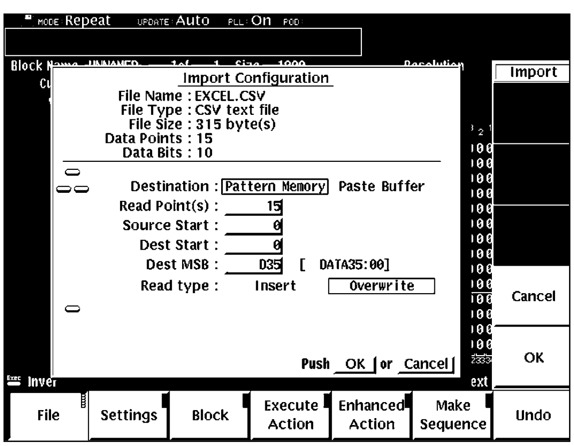
Step 1 – Create the CSV file. This short text file and the following notes define the DG2000 Series CSV format:

- The number of lines defines the number of words. The number of bits in a line defines the word width. In this case there are 15 words that are each 10-bits wide. The first bit in each line is the most-significant bit (MSB).
- Each bit is defined by the character 0 or 1. A comma separates each bit within a line.
- Each line is terminated with a carriage return <CR>, a linefeed <LF>, or both <CR><LF>.
- The file is DOS-compatible with a .CSV file extension. Each DG2000 Series bit occupies two ASCII characters or bytes in the file (one byte for the bit definition character and one byte for the comma or line terminator character).

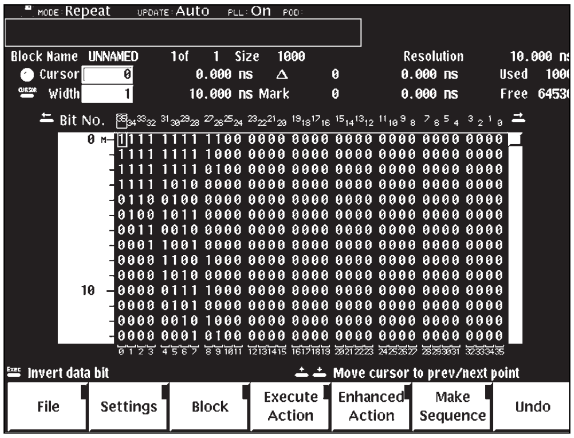
Example CSV Text Format

```
1,1,1,1,1,1,1,1,1,1<CR>
1,1,1,1,1,1,1,1,1,0<CR>
1,1,1,1,1,1,1,1,0,1<CR>
1,1,1,1,1,0,1,0,0,0<CR>
0,1,1,0,0,1,0,0,0,0<CR>
0,1,0,0,1,0,1,1,0,0<CR>
0,0,1,1,0,0,1,0,0,0<CR>
0,0,0,1,1,0,0,1,0,0<CR>
0,0,0,0,1,1,0,0,1,0<CR>
0,0,0,0,0,1,0,1,0,0<CR>
0,0,0,0,0,0,1,0,1,0<CR>
0,0,0,0,0,0,0,1,0,1<CR>
0,0,0,0,0,0,0,0,1,0<CR>
0,0,0,0,0,0,0,0,0,1<CR>
0,0,0,0,0,0,0,0,0,0<CR>
```

The disk space requirement for a DG2000 Series CSV pattern is twice the number of words multiplied by the number of bits per word. In this example, the file size is 300 bytes (2x15x10). The file size would be 315 bytes for the <CR><LF> termination which adds one byte per line. A large data pattern in the DG2000 Series CSV format can exceed the capacity of a single 3.5-inch floppy disk. For example, a 64 Kword pattern with 36 bits per word for the DG2020A or a 256 Kword pattern with 8 bits per word for the DG2030 requires over 4 Mbytes of disk space. The solution is to segment the pattern into smaller files. Since you can specify the starting location in DG2000 Series pattern memory for each import operation, the complete pattern can be imported by writing the smaller files into contiguous sections of pattern memory. Note that once a large pattern has been imported through the CSV format, the DG2000 Series can then save the entire data pattern and instrument setup parameters to disk using an internal and more compact binary file format.



Step 2 – DG2000 Series data import. In this case, the imported CSV filename is excel.csv and contains the pattern used above to illustrate the DG2000 Series CSV format. The DG2000 Series identifies the File Type as a CSV text file and indicates the correct number of points (15) and bits per word (10). It's important to verify these parameters when importing a CSV file. In particular, if the DG2000 Series cannot correctly interpret the CSV format, a File Type of "plain text" will be indicated. In addition, the point and data bit values will be incorrect. You can examine the CSV file with virtually any text editor to insure that each line conforms to the DG2000 Series CSV format. Note that your text editor may automatically place a line terminator on the last line of the file; this means that you don't need to manually insert a <CR> after the last line.



Step 3 – Result. The 10-bit pattern is displayed using the DG2000 Series binary display format. The data is left justified with D35 as the most-significant bit.

Using a Spreadsheet to Create a CSV Pattern

Most spreadsheets can create a CSV text file that separates columns of cells with a comma and rows of cells with the line terminator (<CR> and/or <LF>). For example, the CSV file excel.csv used in this section was created by saving the matrix of data between cells C3 and L17 using the CSV text format.

Note that the typical reason for using a spreadsheet is to generate a complex data pattern that represents more than a random string of 1's and 0's. For example, the DG2000 Series can be used to provide a quantized test signal to a digital signal processor. The test signal may represent a sinewave or some other numerically calculated signal burst. One spreadsheet task is to expand the numerical values into columns of 0's and 1's to match the DG2000 Series CSV format. In the example, the 15 words represent the 15 integer values in column A (A3:A17). Each value was converted to a 10-bit binary value by expansion into 10 columns (C through L). One approach to this decimal to binary conversion is shown here. The maximum binary value (1024 in this case) corresponding to the number of bits is placed in cell B1. The bit weights (512 to 1) for the 10 columns are then placed in the same row (C1:L1). The following equation is placed in Cell C3:

```
=IF(MOD($A3,B$1)=MOD($A3,C$1),0,
  1)
```

This equation is copied into all the other output cells with the spreadsheet managing the relative addressing. For example, cell D3 will contain the equation:

```
=IF(MOD($A3,C$1)=MOD($A3,D$1),0,
  1)
```

Cell C4 will contain the equation:

```
=IF(MOD($A4,B$1)=MOD($A4,C$1),0,
  1)
```

Finally, since we only need to save the 1's and 0's pattern, we must copy the matrix C3 to L17 into a temporary worksheet before saving it into a CSV text file.

For further information, contact Tektronix:

World Wide Web: http://www.tek.com; **ASEAN Countries** (65) 356-3900; **Australia & New Zealand** 61 (2) 888-7066; **Austria, Eastern Europe, & Middle East** 43 (1) 701 77-261; **Belgium** 32 (2) 725-96-10; **Brazil and South America** 55 (11) 3741 8360; **Canada** 1 (800) 661-5625; **Denmark** 45 (44) 850700; **Finland** 358 (9) 4783 400; **France & North Africa** 33 (1) 69 86 81 08; **Germany** 49 (221) 94 77-400; **Hong Kong** (852) 2585-6688; **India** 91 (80) 2275577; **Italy** 39 (2) 250861; **Japan** (Sony/Tektronix Corporation) 81 (3) 3448-4611; **Mexico, Central America, & Caribbean** 52 (5) 666-6333; **The Netherlands** 31 23 56 95555; **Norway** 47 (22) 070700; **People's Republic of China** (86) 10-62351230; **Republic of Korea** 82 (2) 528-5299; **Spain & Portugal** 34 (1) 372 6000; **Sweden** 46 (8) 629 6500; **Switzerland** 41 (41) 7119192; **Taiwan** 886 (2) 722-9622; **United Kingdom & Eire** 44 (1628) 403300; **USA** 1 (800) 426-2200

From other areas, contact: Tektronix, Inc. Export Sales, P.O. Box 500, M/S 50-255, Beaverton, Oregon 97077-0001, USA (503) 627-1916

Tektronix

10/97 TD/XBS 76W–10857–1