An Introduction To GADD Scripting

There are 7 commands:

SET and CLEAR
SWEEP and ENDSWEEP
MEASURE and CHECK
WAIT

Every line (except for comments, described below) are going to begin with one of these 7 words. (And since 3 or 4 of those are pretty trivial your chances of typing the right command more or less randomly are actually pretty good).

A script file is just a text file with a .scrpt suffix that has the commands in it.
Here is an example script:

```
SET ac=1
SWEEP frequency, start=100, stop=1000, step=100
    MEASURE epsilon
ENDSWEEP
```

and here is another example:

```
SWEEP  temperature, start=25, stop=250, step=5
    SWEEP sample, start=1, stop=8, step=1
        SWEEP frequency, start=100, stop=1e6, multiplier=10
            MEASURE cd
        ENDSWEEP
        WAIT seconds=30
    ENDSWEEP
ENDSWEEP
```

A command may appear alone:

ENDSWEEP

with one parameter (which may have a value)

MEASURE epsilon

SET frequency=1000

or sometimes more than one parameter

SET frequency=1000, ac=2, dc=-1

SWEEP frequency, start=100, stop=1e6, multiplier=10

Some Things That Are Entirely Up to You

Except for the first blank space after the command, blank spaces are up to you:

SET frequency = 1000

SET frequency= 1000

SET frequency=1000

and

SWEEP ac, start=1, stop=10, step=1

SWEEP ac, start=1, stop= 10, step=1

do exactly the same thing. But

SETfrequency = 100

will fail.

Many long words can be abbreviated to a short form:

SET frequency=1000

and

SET freq=100 [frequency <=> freq]

do exactly the same thing. When there is a useful abbreviation for a command or parameter used in this tutorial it will appear in [<=>] near the term it abbreviates. The [<=>] stuff is not part of the command. There is an attached list of abbreviations.

Indenting with either blank spaces or tabs is up to you:

```
SWEEP ac, start=1, stop=10, step=1
        MEASURE cd
ENDSWEEP
```

and

```
SWEEP ac, start=1, stop=10, step=1
MEASURE cd
ENDSWEEP
```

do exactly the same thing, although the first one is MUCH easier to read when the scripts get longer. In this tutorial the script is indented from the rest of the text for readability.

Capitalization doesn't matter:

```
SET frequency= 1000          and          sEt FREQuenCY=1000
```

do exactly the same thing. In this tutorial commands will always be capitalized and other parameters in small letters for readability.

Any line that begins with a // is ignored, so you can use it to leave a comment in the file:

```
SWEEP frequency, start=100, stop=1000, step=100
        MEASURE epsilon
        WAIT seconds=30
ENDSWEEP
```

and

```
SWEEP frequency, start=100, stop=1000, step=100
        MEASURE epsilon
        WAIT seconds=30
 // do I really need to wait 30 seconds?
ENDSWEEP
```

do exactly the same thing. The //'s have to be at the very beginning of the line, so you can't indent comments, although you can have all the blank spaces or tabs after the // that you want.

Some Things That Aren't Up to You

You can't (at this time) have blank lines in a script file:

```
SWEEP frequency, start=100, stop=1000, step=100
    MEASURE epsilon

ENDSWEEP
```

will fail in some way.

Multiple parameters on the same line must be separated by commas:

```
SWEEP frequency start=100  stop=1000  step=100
```

will fail. And you can't use a comma after the command:

```
SWEEP, frequency, start=100, stop=1000, step=100
```

will fail.

All units are SI base units and ac quantities are rms. Numeric values can be either integers (3, 255, -7), floating (1.234, 0.0004) or scientific notation (1.2E5, 3.1e-10). The temperature units, Celsius or Kelvin, reflect the controller being used.

You must specify the hardware you want to use in your script using the SET command described below.

COMMANDS

WAIT (we're doing the easiest one first)

Pauses the execution of the script.

WAIT seconds= .5 [seconds <=> s]

WAIT show_me_this_message

WAIT

The first one pauses the program for 1/2 second then it continues automatically. The second one will pause the execution, pop up a window with "show_me_this_message" in it and wait for you to click ok. Messages containing blank spaces get truncated at the first blank (a strictly virtual exception to the "go ahead and use blanks rule") and the text always appears capitalized. The last example will do the same thing but with no specific message, just a popup window to click when you're ready. In a rare exception to the "SI base units only" rule, you can also:

WAIT minutes=60 [minutes <=> min]

SET

Sets some parameter. Things you can set, depending on the hardware being used:

Temperature, frequency, ac (level in rms), dc (bias), integration (integration time of LCR meter or dynamic reserve of lock-in), temperaturetolerance, temperatureholdtime, deg/min, sample, externalbits, harmonic, time, temperaturesensor. The temperature control settings are described in the temperature section. The most of the other ones are self-explanatory. Otherwise:

SET sample=3

will switch to the third sample when using a sample multiplexer (relays).

SET time=100

will cause the script to wait until 100 seconds have elapsed since the script was started. If the script had already been running for 100 seconds then it has no effect. Generally speaking, SWEEP time is more useful.

peak: peak voltage of pe measurement

temperaturesensor: RTD or KTCPL, used to tell the system which sensor you are using with a DMM to measure temperature [temperaturesensor <=> tempsensor]
externalbit0-3: SET externalbit0=1 will set the external digital line numbered 0 to the on (1) state. [externalbit <=> extbit]

You can stack up multiple SET's in one line using commas.
SET frequency=1000, ac=2, dc=-1
is the same as 3 individual SET commands.

Because the system can work with a wide variety of hardware you must use the SET command to specify the specific hardware you want to use in that script. Generally the first line or lines should look like:

SET acdevice=HP4284, dmm=FLUKE8840, pameter=HP4140

Please follow the guidance of the operator at your system to tell you which devices you can use. The GADD program will initialize the devices when it reaches that line in the script, which is why it should normally be the first line or lines.

CLEAR
Returns something to its default state.
Clear charge
zero's the charge converter (when used, obviously)
Clear plot
clears a real-time plot during the measurement
Clear heater
this will disable (turn heating and cooling off) for the heater/oven being used, assuming that oven has a disable feature.

SWEEP

Creates a loop to change a parameter over a range and repeatedly execute commands in the loop. The loop is always terminated by an ENDSWEEP. You can SWEEP pretty much any parameter that can be SET. The format of the SWEEP command is always:

```
SWEEP  something,  START= a,  STOP= b, how_you_get_there = c
        do_some_stuff
        ...
        last_of_the_stuff
ENDSWEEP
```

You can put a sweep inside another sweep, and that one inside another sweep, up to 8 levels. I wouldn't put a SET or a SWEEP of some parameter, say ac voltage, inside another sweep of the same parameter, but it might work, I don't really know. You can also have more than one sweep sequentially in a script, for instance you could make a measurement on heating then on cooling, although you should read the measurement section dealing with multiple measurements for a detail about that. The START and STOP are pretty self-explanatory but how_you_get_there has some different forms.

SWEEP something, START=1, STOP=10, STEP=1
will take "something" from 1 to 10 by 1's; 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

SWEEP something, START=1, STOP=10, STEP=2
will take "something" from 1 to 10 by 2's; 1, 3, 5, 7, 9. 10 won't get executed because 11 is next and it's too big.

SWEEP something, START=10, STOP=0, STEP=2
will take "something" from 10 to 0 by 2's; 10, 8, 6, 4, 2, 0.

SWEEP something, START=1, STOP=10, MULTIPLIER=2
will take "something" from 1 to 10, doubling at each point; 1, 2, 4, 8. 16 won't happen because it's too big.

SWEEP something, START=1, STOP=1e5, MULTIPLIER=10
will take "something" from 1 to 100,000 by decades; 1, 10, 100, 1000, 10000, 100000.

SWEEP something, START=1e5, STOP=1, MULTIPLIER=10
works the same way but in reverse.

There are two special sequences: SEQUENCE=125 and SEQUENCE=123468. Both produce nominally logarithmically spaced points, like MULTIPLIER, but they have different rounding. [SEQUENCE <=> SEQ]

SWEEP something, START=1, STOP=1e4, SEQ=125
will take "something" from 1 to 10,000 like this: 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, which is essentially a multiplier of 2 but with decimal-based rounding.

SWEEP something, START=1, STOP=100, SEQUENCE=123468
will take "something" from 1 to 100 like this: 1, 1.4, 2, 3, 4, 6, 8, 10, 14, 20, 30, 40, 60, 80, 100 which is essentially a multiplier of the square root of two with decimal-based rounding for cases when more points than a factor of two are needed. You should be careful when using logarithmic sweeps (multiplier or sequences 125 and 123468) because, being logarithmic, they will fail if the start or stop value is 0 or the sweep tries to pass through 0.

Just because you can program a sweep doesn't mean it can be done. For instance you can ask for values outside a machine's range, in which case the behaviour is undefined, I don't know what happens. Also, the older LCR meters only deal with 1,2,5 frequencies, they can't be set to 6kHz, for instance. So the only non-trivial sweeps that work are multiply by 10 and 125 sequences. Even the newer LCR meters don't always have the frequency resolution you might ask for.

MEASURE

Measure something. Some measurements are real numbers (temperature, DC voltage), others are complex numbers (C&D, epsilon), others are lists (pe). Depending on how deeply the measurement is nested in SWEEP loops it can have from one to eight dimensions, each SWEEP loop adds a dimension. This sweep command is what allows this system to automatically make measurements in a day over many different parameters that might take weeks otherwise, even though from the point of view of the user writing a script or setting up the measurement the dimensionality isn't a concern. The measurement is made, saved, and passed to VIZ transparently regardless of type or dimension of the measurement. As of this writing the available measurements are:

LCR meter (impedance):	CD, GB, or RX
DMM:	VDC, VAC, ADC (dc current), AAC (ac current), RDC (resistance), TEMPERATURE (see temperaturesensor in SET section above)
GADD hardware:	PE, PREISACH, EPSILON, nA (dc current), ACGAIN
pA meter:	pA (dc current)
Lock-in	ACV, ACA (ac microamps)
Sweep Devices	Gain, R&X, C&D, G&B as functions of frequency (the Agilent 4194, 4195, 4294, and others)

One rule that might not be expected is that you can't measure a particular thing at more than one place in a script file:

```
SWEEP frequency, start=100, stop=1000, step=100
        MEASURE cd
        WAIT seconds=30
        MEASURE cd
ENDSWEEP
```

will fail in some way. As would

```
SWEEP  temperature, start=25, stop=250, step=5
        MEASURE CD
ENDSWEEP
SWEEP  temperature, start=250, stop=25, step=5
        MEASURE CD
ENDSWEEP
```

But you can make multiple measurements by using a number:

```
SWEEP frequency, start=100, stop=1000, step=100
        MEASURE CD1
        WAIT seconds=30
        MEASURE CD2
ENDSWEEP
```

and

```
SWEEP  temperature, start=25, stop=250, step=5
        MEASURE CD0
ENDSWEEP
SWEEP  temperature, start=250, stop=25, step=5
        MEASURE CD1
ENDSWEEP
```

will work perfectly well. When you look at the data in VIZ, the two CD's will appear as separate pieces of data to be viewed. And you can make different measurements freely:

```
SWEEP  temperature, start=25, stop=250, step=5   [temperature <=> temp]
        SWEEP frequency, start=100, stop=1e6, multiplier=10
                measure cd
        ENDSWEEP
        MEASURE temperature
ENDSWEEP
```

is perfectly fine. In this case the cd data will be two dimensional and the temperature data will be one dimensional.
You can use the numbers 0-9. No number on a measurement implies a 0, so

MEASURE epsilon

and

MEASURE epsilon0

mean the same thing, which means you can't have MEASURE epsilon and MEASURE epsilon0 in the same script.

CHECK

Check something. This is very much like MEASURE but for simpler things:
CHECK ac
will use the LCR meter's monitor function to check the ac voltage level actually applied to a sample.
CHECK aca
will use the LCR meter's monitor function to check the ac current actually running through a sample.

CHECK temperature
will ask the temperature controller what the oven temperature is using its own sensor. This is different from MEASURE temperature which will use the defined temperature-sensor and the system DMM (when used, obviously) to make an independent temperature measurement.

CHECK time
will get the elapsed time of the script

CONTROLLING THE TEMPERATURE

Because of reality interfering, setting the temperature is different than setting, say, the frequency of an AC device since the temperature has to have some form of feedback control and it takes significant time to settle. Because of this, besides the SET temperature or SWEEP temperature commands, you will also need to SET some other parameters. There are two temperature control modes; step-and-hold, and ramp. Because the point of this system is to allow people to measure something as a function of a lot of other things, and measuring those things takes time during which the temperature should remain constant, the step-and-hold method should be preferred. Even though, depending on the number of points programmed, it can take longer.

Step and Hold

You need to SET two parameters: temperaturetolerance and temperatureholdtime.
[temperaturetolerance <=> temptol] [temperatureholdtime <=> temphold]
If you set the tolerance and hold time then when you do a temperature loop the system will wait at each temperature point until the actual temperature (read from the controller) has been within the tolerance for the hold time before doing anything in the loop. For example:

```
SET temptol=1, temphold=30
SWEEP temperature, start=25, stop=50, step=5
        MEASURE cd
ENDSWEEP
```

will first set the temperature to 25, check the temperature (once a second) and wait until the oven temperature has been within 1 degree of 25 for 30 seconds. Then it will make the cd measurement, go on to 30 degrees, and wait again. If the temperature goes outside below 24 or above 26 the 30 second clock starts again. The same rules apply for temperature which has been SET.

Ramp

You also need to SET two parameters: rate and temperaturetolerance. The tolerance is set as above. The rate is SET using one of the following keywords:
DEG/MIN, DEGREE/MIN, DEG/MINUTE, DEGREE/MINUTE,
DEGREE/S, DEGREE/SEC, DEGREE/SECOND, DEG/S,DEG/SEC, DEG/SECOND
Basically the word "degree" in any reasonable form and either "minute" or "second" in any reasonable form.

```
SET temptol=1, deg/min=3
SWEEP temperature, start=25, stop=50, step=5
        MEASURE cd
ENDSWEEP
```

In this mode the system will again first set the temperature to 25 and wait until the temperature is between 24 and 26. But when that happens it will immediately start a temperature ramp at 3 degrees per minute. It will make the cd measurement (at 25) and then wait until the temperature (read from the controller) is equal to or above 30, at which time it will make another cd measurement and wait again, continuing the process to 50. In both of these cases the oven will remain at temperature when the script is done. You can use a SET command to return it to, say, room temperature after the loop.

PLOTTING MEASUREMENTS ON THE FLY

Within limits, measurements can be plotted on screen during the measurement process. This is not the same thing as using VIZ. VIZ is flexible and interactive and used on the .diel file after the measurement is done. Plotting during the measurement is just to help you keep an eye on what is going on while it's happening. The plot always appears as a 3D surface even if the data is only 1 dimensional, in which case it will just appear as a line floating in a cube. To plot a measurement you assign one sweep parameter to the x-axis, another sweep parameter (if used) to the y-axis, and one measurement as the item to be plotted, e.g.:

```
SWEEP temperature, start=25, stop=100, step=5, xaxis
        SWEEP frequency, start=100, stop=1e6, mult=10, yaxis
```

```
            MEASURE cd, plot
        ENDSWEEP
ENDSWEEP
```

will measure C and D as a function of frequency and temperature and plot the results on screen during the measurement. There are a number of limitations at this point; plotting more than one measurement isn't supported, switching between scripts with complex measurements and real measurements isn't well defined.

Summaries

Commands:
SET, CLEAR, WAIT, MEASURE, CHECK, SWEEP, ENDSWEEP

MEASUREMENTs using:

LCR meter:	CD, GB, RX
pA meter:	PA
DMM:	TEMPERATURE, RDC, VDC, VAC, ADC, AAC
Lock-in:	ACV, ACA
Sweep Devices:	Device dependent, includes GAIN, R&X, G&B, C&D
Arbitrary Charge Hardware:	EPSILON, PE, PREISACH

CHECKs using:

Anything:	TIME
Heater (oven):	TEMPERATURE
LCR meter:	ACV, ACA, FREQ

SETs: (not sorted, too confusing)
SAMPLE, EXTERNALBITn
TEMPERATURE, TIME, AC, DC, FREQUENCY
TEMPSENSOR, TEMPHOLD, TEMPTOL, INTEG

Abbreviations

TEMP	TEMPERATURE	
EXTBITn	EXTERNALBITn	e.g. EXTBIT0
FREQ	FREQUENCY	
S	SECOND	
MIN	MINUTES	(used only with WAIT and rate setting commands)
DEG	DEGREE	
TEMPTOL	TEMPERATURETOLERANCE	
TEMPHOLD	TEMPERATUREHOLDTIME	
TEMPSENSOR	TEMPERATURESENSOR	
HARM	HARMONIC	
MEAS	MEASURE	
INTEG	INTEGRATIONTIME	