9 Understanding the CITIfile Data Format

Using This Chapter

The descriptions and examples shown in this chapter demonstrate how CITIfile may be used to store and transfer both measurement information and data. The use of a single, common format will allow data to be more easily moved between instruments and computers.

This chapter contains the following sections:

- CITIfile Data Format
  - Description and Overview
  - Definition Of CITIfile Terms
  - CITIfile Examples
- CITIfile Keywords
- Useful Calculations

NOTE For many data processing applications, the S2P file (filename.S1 and filename.S2) may provide a more convenient format.

The CITIfile Data Format

Description and Overview

CITIfile is a standardized data format, used for exchanging data between different computers and instruments. CITIfile is an abbreviation for "Common Instrumentation Transfer and Interchange file". This standard has been a group effort between instrument designers and designers of computer-aided design programs. As much as possible, CITIfile meets current needs for data transfer, and it was designed to be expandable so it can meet future needs.

CITIfile defines how the data inside an ASCII package is formatted. Since it is not tied to any particular disk or transfer format, it can be used with any operating system (BASIC, DOS, UNIX, etc.), with any disk format (LIF, DOS, HFS, etc.), or with any transfer mechanism (disk, LAN, GPIB, etc.). By careful implementation of the standard, instruments and software packages using CITIfile are able to load and work with data created on another instrument or computer. It is possible, for example, for a network analyzer to directly load and display data measured on a scalar analyzer, or for a software package running on a computer to read data measured on the network analyzer.

Data Formats

There are two main types of data formats: binary and ASCII. CITIfile uses the ASCII text format. While this format does take up more bytes of space than a binary format, ASCII data is a transportable, standard type of format which is supported by all operating systems. In addition, the ASCII format is accepted by most text editors. This allows files to be created, examined, and edited easily, making CITIfile easier to test and debug.

File and Operating System Formats

CITIfile was designed to be independent of the data storage mechanism, and therefore may be implemented for any file system. However transfer between file systems may sometimes be necessary. Any commercially available software that has the ability to transfer ASCII files between systems may be used to transfer CITIfile data.

Definition of CITIfile Terms

This section will define the following terms:

- package
- header
- data array
- keyword

A CITIfile Package

A typical package is divided into two parts: The first part, the header, is made up of keywords and setup information. The second part, the data, usually consists of one or more arrays of data. Example 1 shows the basic structure of a CITIfile package:

Example 1, A CITIfile Package

The "header" part	CITIFILE A.01.00
	NAME MEMORY
	VAR FREQ MAG 3
	DATA S RI
The "data" part	BEGIN
	-3.54545E-2, -1.38601E-3
	0.23491E-3, -1.39883QE-3
	2.00382E-3, -1.40022E-3
	END

When stored in a disk file there may be more than one CITIfile package. With the 8510 network analyzer, for example, storing a "memory all" will save all eight of the memories held in the instrument. This results in a single file which contains eight CITIfile packages.

The CITIfile Header

The header section contains information about the data that will follow. It may also include information about the setup of the instrument that measured the data. For example, the header may include information such as:

- CITIfile version number
- Network analyzer model number
- Firmware revision currently installed in the analyzer
- Type of Data
- Data Format
- Measurement parameters
- Start and stop frequencies
- Number of sample points

The CITIfile header shown in Example 1 has just the bare minimum of information necessary; no instrument setup information was included.

An Array of Data

An array is numeric data that is arranged with one data element per line. In the Smith chart and polar formats, the data is in real and imaginary pairs. In all other formats, the data is still in pairs, but the second term of the pair is 0E0. All information is true formatted data in the same format as on the analyzer display (dB, SWR, etc.).

A CITIfile package may contain more than one array of data. Arrays of data start after the BEGIN keyword, and the END keyword will follow the last data element in an array. A CITIfile package does not necessarily need to include data arrays; for instance, CITIfile could be used to store the current state of an instrument. In that case the keywords VAR, DATA, BEGIN, and END would not be required.

CITIfile Keyword

Keywords are always the first word on a new line. They are always one continuous word without embedded spaces. A listing of all the keywords used in the latest A.01.01 version of CITIfile is shown in "CITIfile Keywords." When reading a CITIfile, unrecognized keywords should be ignored. This allows new keywords to be added, without affecting an older program or instrument that might not use the new keywords. The older instrument or program can still use the rest of the data in the CITIfile as it did before. Ignoring unknown keywords allows backwards compatibility to be maintained.

CITIfile Examples

Example 2, An 8510 Display Memory File

Example 2 shows a simple file that contains no frequency information. Some instruments do not keep frequency information for display memory data, so this information is not included in the CITIfile package. Note that instrument-specific information (#NA= Network Analyzer information) is also stored in this file. This convention allows the designer to define keywords that are particular to his or her particular implementation.

Example:

```
CITIFILE A.01.00
#NA VERSION HP8510B.05.00
NAME MEMORY
#NA REGISTER 1
VAR FREQ MAG 5
DATA S RI
BEGIN
-1.31189E-3,-1.47980E-3
-3.67867E-3,-0.67782E-3
-3.43990E-3,0.58746E-3
-2.70664E-4,-9.76175E-4
0.65892E-4,-9.61571E-4
END
```

Example 3, 8510 Data file

Example 3 shows a CITIfile package created from the data register of an 8510 Network Analyzer. In this case 10 points of real and imaginary data was stored, and frequency information was recorded in a segment list table.

Example:

```
CITIFILE A.01.00
#NA VERSION HP8510B.05.00
NAME DATA
#NA REGISTER 1
VAR FREQ MAG 10
DATA S[1,1] RI
SEG_LIST_BEGIN
SEG 1000000000 4000000000 10
SEG_LIST_END
BEGIN
0.86303E-1,-8.98651E-1
8.97491E-1,3.06915E-1
-4.96887E-1,7.87323E-1
-5.65338E-1,-7.05291E-1
8.94287E-1,-4.25537E-1
1.77551E-1,8.96606E-1
-9.35028E-1,-1.10504E-1
3.69079E-1,-9.13787E-1
7.80120E-1,5.37841E-1
-7.78350E-1,5.72082E-1
END
```

Example 4, 8510 3-Term Frequency List Cal Set File

Example 4 shows how CITIfile may be used to store instrument setup information. In the case of an 8510 Cal Set, a limited instrument state is needed in order to return the instrument to the same state that it was in when the calibration was done. Three arrays of error correction data are defined by using three DATA statements. Some instruments require these arrays to be in the proper order, from E1 to E3. In general, CITIfile implementations should strive to handle data arrays that are arranged in any order.

Example:

```
CITIFILE A.01.00
#NA VERSION HP8510B.05.00
NAME CAL_SET
#NA REGISTER 1
VAR FREQ MAG 4
DATA E[1] RI
DATA E[2] RI
DATA E[3] RI
#NA SWEEP_TIME 9.999987E-2
#NA POWER1 1.0E1
#NA POWER2 1.0E1
#NA PARAMS 2
#NA CAL_TYPE 3
#NA POWER_SLOPE 0.0E0
#NA SLOPE_MODE 0
#NA TRIM_SWEEP 0
#NA SWEEP_MODE 4
#NA LOWPASS_FLAG -1
#NA FREQ_INFO 1
#NA SPAN 1000000000 3000000000 4
#NA DUPLICATES 0
#NA ARB_SEG 1000000000 1000000000 1
#NA ARB_SEG 2000000000 3000000000 3
VAR_LIST_BEGIN
1000000000
2000000000
2500000000
3000000000
VAR_LIST_END
BEGIN
1.12134E-3,1.73103E-3
4.23145E-3,-5.36775E-3
-0.56815E-3,5.32650E-3
-1.85942E-3,-4.07981E-3
END
BEGIN
2.03895E-2,-0.82674E-2
-4.21371E-2,-0.24871E-2
0.21038E-2,-3.06778E-2
1.20315E-2,5.99861E-2
```

```
END
BEGIN
4.45404E-1,4.31518E-1
8.34777E-1,-1.33056E-1
-7.09137E-1,5.58410E-1
4.84252E-1,-8.07098E-1
END
```

When an instrument's frequency list mode is used, as it was in Example 4, a list of frequencies is stored in the file after the VAR_LIST_BEGIN statement. The unsorted frequency list segments used by this instrument to create the VAR_LIST_BEGIN data are defined in the #NA ARB_SEG statements.

CITIfile Keywords

Keyword	**Explanation and Examples**
CITIFILE	CITIFILE A.01.01 identifies the file as a CITIfile, and indicates the revision level of the file. The CITIfile keyword and revision code must precede any other keywords. The CITIfile keyword at the beginning of the package assures the device reading the file that the data that follows is in the CITIfile format. The revision number allows for future extensions of the CITIfile standard. The revision code shown here following the CITIfile keyword indicates that the machine writing this file is using the A.01.01 version of CITIfile as defined here. Any future extensions of CITIfile will increment the revision code.
NAME	NAME CAL_SET allows the current CITIfile "package" to be named. The name of the package should be a single word with no embedded spaces. A list of standard package names follows:
Label	Definition.
RAW_DATA	Uncorrected data.
DATA	Data that has been error corrected.When only a single data array exists, it should be named DATA.
FORMATTED	Corrected and formatted data.
MEMORY	Data trace stored for comparison purposes.
CAL_SET	Coefficients used for error correction.
CAL_KIT	Description of the standards used.
DELAY _TABLE	Delay coefficients for calibration.
VAR	VAR FREQ MAG 201 defines the name of the independent variable (FREQ), the format of values in a VAR_LIST_BEGIN table (MAG, if used), and the number of data points (201).Typical names for the independent variable are FREQ (in Hz), TIME (in seconds), and POWER (in dBm). For the VAR_LIST_BEGIN table, only the "MAG" format is supported at this point. # #NA POWER1 1.0E1 allows variables specific to a particular type of device to be defined. The pound sign (#) tells the device reading the file that the following variable is for a particular device. The "NA" shown here indicates that the information is for a Network Analyzer. This convention allows new devices to be defined without fear of conflict with keywords for previously defined devices. The device identifier (i.e. NA) may be any number of characters.

SEG_LIST_BEGIN SEG_LIST_BEGIN indicates that a list of segments for the independent variable follow. Format for the segments is: [segment type] [start] [stop] [number of points]. The current implementation only supports a single segment. If there is more than one segment, the VAR_LIST_BEGIN construct is used. CITIfile revision A.01.00 supports only the SEG (linear segment) segment type.

SEG_LIST_END SEG_LIST_END defines the end of a list of independent variable segments.

VAR_LIST_BEGIN VAR_LIST_BEGIN indicates that a list of the values for the independent variable (declared in the VAR statement) follow. Only the MAG format is supported in revision A.01.00.

VAR_LIST_END VAR_LIST_END defines the end of a list of values for the independent variable.

DATA DATA S[1,1] RI defines the name of an array of data that will be read later in the current CITIfile package, and the format that the data will be in. Multiple arrays of data are supported by using standard array indexing. Versions A.01.00 and A.01.01 of CITIfile only support the RI (real and imaginary) format, and a maximum of two array indexes. Commonly used array names include the following: "S" for "S parameter" Example: S[2,1] "E" for "Error term" Example: E[1] "USER" for "User parameter" Example: USER[1] "VOLTAGE" Example: VOLTAGE[1] "VOLTAGE_RATIO" for a ratio of Example: VOLTAGE_RATIO[1,0] two voltages (A/R).

CONSTANT CONSTANT [name] [value] allows for the recording of values which don't change when the independent variable changes.

CONSTANTs are part of the main CITIfile definition. Users must not define their own CONSTANTs. Use the #KEYWORD device specification to create your own KEYWORD instead. The #NA device specification is an example of this. No constants were defined for revision A.01.00 of CITIfile. CITIfile revision A.01.01 defined the following constant:

CONSTANTs are part of the main CITIfile definition. Users must not define their own CONSTANTs. Use the #KEYWORD device specification to create your own KEYWORD instead. The #NA device specification is an example of this. No constants were defined for revision A.01.00 of CITIfile. CITIfile revision A.01.01 defined the following constant:

CONSTANT TIME [year] [month] [day] [hour] [min] [secs] Example:

COMMENT	YEAR	MONTH	DAY	HOUR	MINUTE	SECONDS
CONSTANT TIME	1999	02	26	17	33	53.25

- The COMMENT statement is not absolutely required, but is highly recommended to aid readability.
- The year should always be the full four digits ("1999" is correct, but "99" is not). This is to avoid problems with the year 2000, when the shortened version of the year will be "00."
- The hour value should be in 24-hour "military" time.
- When writing a CITIfile and the fractional seconds value is zero, then the "seconds" value may be printed either with or without a decimal point: either "47.0" or "47" would be acceptable. When reading a CITIfile, the seconds value should always be read as if it were a floating point number.

Useful Calculations

This section contains information on computing frequency points and expressing CITIfile data in other data formats.

Computing Frequency Points

In CITIfile, the frequency data is not listed point by point, only the start and stop values are given. If you are using a spreadsheet program, you can create a new frequency column to the left of the data pairs. Use the following formula to obtain each frequency point:

Fn = Fstart + {(n-1) * [(Fstop - Fstart) / (# of points - 1)]}

where:

Fstart = Start Frequency
Fstop = Stop Frequency
Fn = Frequency point with n being an integer
# of points = number of sample points per sweep

Here is an example of how this formula may be entered:

F1 = 30E3 + {(1-1) * [(6E9 - 30E3) / (201 - 1)]} = 30E3 = 30 kHz
F2 = 30E3 + {(2-1) * [(6E9 - 30E3) / (201 - 1)]} = 30E6 = 30 MHz
F201 = 30E3 + {(201 - 1) * [(6E9 - 30E3) / (201 - 1)]} = 6E9 = 6 GHz

Once these cells are entered, copy the formula to the remaining data points, and the frequency will be indicated for each row.

Expressing CITIfile Data in Other Data Formats

CITIfile data is represented in real and imaginary pairs. Equations can be used to express this information in logarithmic magnitude, phase, polar, and Smith chart formats. Refer to the following table for these equations.

Desired Format	Mathematical Equation[a]	Microsoft Excel Command[b]
Log Magnitude	20*Log10((Re2 + Im2)) 1/2	=20*LOG10(SQRT((SUMSQ(ReCell 1,Im Cell 1)))) (dB)
Phase	tan-1(Im/Re) or arctan (Im/Re)	ATAN2(ReCell 1, ImCell 1)*180/PI() (Degree)
Polar	Magnitude = ((Re2 + Im2)) 1/2 Phase = tan-1(Im/Re) or arctan (Im/Re)	Magnitude = (SQRT((SUMSQ(ReCell 1,Im Cell 1))) Phase = ATAN2(ReCell 1, ImCell 1)*180/PI()
Smith Chart (Marker)	Resistance = (1 - Re2-Im2) / ((1 -Re) 2+Im2)) * Z0 Reactance = (2*Im) / ((1 - Re) 2+Im2)) * Z0	Resistance = ((1-POWER(ReCell 1,2)-POWER(ImCell 1,2)) / (POWER((1- ReCell 1),2)+POWER(ImCell 1,2))) * Z Cell 1 Reactance = (2*ImCell 1) / (POWER((1- ReCell 1),2)+POWER(ImCell 1,2)))*Z Cell 1

a. Re = real. Im = imaginary.
b. The references to ReCell 1, ImCell 1, and Z Cell 1 refer to the real and imaginary data pair numeric values that have been entered into specific cells in the Microsoft Excel spread sheet.

Example Data

This example shows how the following CITIfile data for a three-point trace can be expressed in other data formats.

```
CITIFILE A.01.00
#NA VERSION HP8753E.07.12
NAME DATA
VAR FREQ MAG 3.0000
DATA S[11] RI
SEG_LIST_BEGIN
SEG 1550000000 1570000000 3.0000
SEG_LIST_END
BEGIN
```

Table 9-1 Data Values

				Calculated Smith Chart Readings		Calculated Polar Readings	
Real Value	**Imag Value**	**Calculated LogMag**	**Calculated Phase**	**Resistance**	**Reactance**	**Magnitude**	**Phase**
4.43E–02	–4.52E–01	–6.8593	–84.4025	35.5204	–40.4294	0.4539	-84.4025
–6.32E–02	–4.47E–01	–6.9150	–98.0545	29.9477	–33.5840	0.4510	-98.0545
–1.66E–01	–4.38E–01	–6.5847	–110.7272	25.1562	–28.2510	0.4685	-110.727

Table 9-2 Marker Reading Values

Log Mag (Marker)	Phase (Marker)	S11 Smith Chart Resistance (Marker)	S11 Smith Chart Reactance (Marker)	Polar Magnitude (Marker)
–6.859	–84.403	35.520	–40.429	454.98mU
–6.915	–98.055	29.948	–33.584	451.07mU
–6.585	–110.737	25.156	–28.251	468.56mU