tsbin1.0.pdf – Binary Frequency and Data

Updated 4/14/2010

To be entered:
- Binary Encoding applies only for Touchstone 2.1 files and above - add statement
- Binary encoding is illegal when [Noise Data] exists (which is limited to 2-ports)
- The [End] statement is removed before the binary encoding is done
- In Touchstone 2.1 and above, [Binary] is positioned as the next line after [Network Data] – add statement

Questions - can we remove the [End] restriction? Can we remove the [Noise Data] restriction?

[Binary]

Rules for Version 1.0 Files:
The [Binary] keyword is not permitted in Version 1.0 files.

Rules for Version 2.0 and Greater Files:
The [Binary] keyword is not permitted in Version 2.0 files. The [Binary] keyword is optional for Version 2.1 and greater files. However, the [Noise Data] keyword and the [Binary] keyword shall not be used in the same file.

Each Touchstone Version 2.1 file may contain one and only one [Binary] keyword.

The [Binary] keyword indicates that network data is presented in binary format, for purposes of file size compression and faster file parsing.

The [Binary] keyword shall be the first keyword after the [Network Data] keyword and before any data (i.e., between the [Network Data] keyword and the network data itself, to inform parsers that the network data is in binary format).

[Binary] accepts three and only three character arguments. For explanatory purposes in this document only, these are designated T1, T2 and T3 below.

The first token, T1, indicates the numerical precision of the frequency information.

The second token, T2, indicates numerical precision of the data.

Both the T1 and T2 tokens use the same characters to designate precision:
F: single precision (floating point)
D: double precision

The third token, T3, indicates byte order:
B: big-endian (most significant byte first)
L: little-endian (least significant byte first)

Example #:

```
[Binary] DFB
```

The example above indicates double-precision frequency and float data in big-endian order.
The [Binary] line and its arguments may be followed by a line-termination sequence. Immediately following the line-termination sequence shall be a single 0 byte-length token to indicate that the information that follows will be in binary format.

Example #:

```
[Version] 2.1
# MHZ S RI R 5.00e+001
[Number of Ports] 4
[Number of Frequencies] 1
! FREQ S11 S12 S13 S14
! S21 S22 S23 S24
! S31 S32 S33 S34
! S41 S42 S43 S44
!
[Network Data]
[Binary] DFB
! Binary Encoded data
```

{insert binary frequency and data in hex format here, for example}

The hex data shown in the example above corresponds to the following ASCII network information:

```
1.000000e+001
2.063717e-002 -1.480975e-002 9.540607e-001 -1.925392e-001
-2.306818e-003 7.529011e-003 -5.623072e-003 -1.259668e-003
9.540620e-001 -1.925394e-001 2.063725e-002 -1.480983e-002
-5.622481e-003 -1.259875e-003 -2.307512e-003 7.529252e-003
-2.306700e-003 7.528990e-003 -5.622914e-003 -1.259719e-003
2.063738e-002 -1.480973e-002 9.540608e-001 -1.925388e-001
-5.622897e-003 -1.259744e-003 -2.307649e-003 7.529295e-003
9.540621e-001 -1.925393e-001 2.063837e-002 -1.481020e-002
```