IBIS Interconnect SPICE Subcircuits Specification
(IBIS-ISS)

Draft 1.10
~~January 26~~April 22, 2011

Style Definition: Heading 3: Font: 14 pt, Indent: Hanging: 0.06"

Formatted: Font: (Default) Helvetica

Contents

1 Overview 8
2 Goals and Scope 9
3 Conventions 10
4 Input Structure and Data Entry 11
4.1 Input File Guidelines 11
4.2 Statements and Arguments 11
4.3 Special Characters 14
4.4 First Character 17
4.5 Delimiters 18
4.6 Instance Names 18
4.7 Numbers 19
4.8 Parameters and Expressions 20
4.9 Node Name (or Node Identifier) Conventions 21
4.10 Element, Instance, and Subcircuit Naming Conventions 21
4.11 Line Continuations 22
4.12 IBIS-ISS Structure 23
5 Parameters 24
5.1 Using Parameters in Simulation (.PARAM) 24
Defining Parameters 24
Assigning Parameters 25
5.2 Using Algebraic Expressions 26
Built-In Functions and Variables 26
5.3 String Parameters 31
5.4 Parameter Scoping and Passing 31
6 File Includes 33
7 Comments 34
8 Model Definitions (.MODEL Statements) 35
9 Subcircuit Definitions 36
9.1 Subcircuit Scoping Rules 36
10 Subcircuit Definition Ending Statements 37
11 Elements 38
11.1 Subcircuits 38
11.2 Linear Resistor 39
11.3 Linear Capacitor 39

11.4 Voltage Source 40
11.5 Mutual Inductor 40
11.6 Linear Inductor 41
11.7 T-element (Ideal Transmission Line) 42
11.8 W-element (Coupled Transmission Line) 43
Format 1: RLGC Model 44
Format 2: Frequency-Dependent Tabular Specification 47
11.9 Frequency-Dependent Matrices 47
Small-Signal Parameter Data Frequency Table Model (SP Model) 48
11.10 S-element 53
S-Element Model Syntax 55
11.11 E-element (Voltage-Controlled Voltage Source) 57
Syntax (Linear Form) 57
Syntax (Laplace Transform) 57
Syntax (Pole-Zero Function) 57
Syntax (Foster Pole-Residue Form) 57
E-element Arguments 57
Laplace Transform Details 58
Pole-Zero Function Details 58
Foster Pole-Residue Details 59
11.12 F-element (Current-Controlled Current Source) 61
11.13 G-element (Voltage-Controlled Current Source) 62
Syntax (Laplace Transform) 62
Syntax (Pole-Zero Function) 62
Syntax (Foster Pole-Residue) 62
G-element Arguments 62
Laplace Transform Details 63
Pole-Zero Function Details 64
Foster Pole-Residue Form 64
11.14 H-element (Current-Controlled Voltage Source) 66
12 Best Practices 68
13 References 71
~~1 Overview ..... 7~~
~~2 Goals and Scope ..... 8~~

Formatted: Default Paragraph Font

Formatted: Default Paragraph Font

Formatted: Default Paragraph Font

Formatted: Default Paragraph Font

3 Conventions 9
4 Input Structure and Data Entry 10
4.1 Input File Guidelines 10
4.2 Statements and Arguments 10
4.3 Special Characters 12
4.4 First Character 15
4.5 Delimiters 16
4.6 Instance Names 16
4.7 Numbers 17
4.8 Parameters and Expressions 18
4.9 Node Name (or Node Identifier) Conventions 19
4.10 Element, Instance, and Subcircuit Naming Conventions 19
4.11 Line Continuations 19
5 Parameters 21
5.1 Using Parameters in Simulation (.PARAM) 21
Defining Parameters 21
Assigning Parameters 22
5.2 Using Algebraic Expressions 23
Built-In Functions and Variables 23
5.3 String Parameters 28
5.4 Parameter Scoping and Passing 28
6 File Includes 29
7 Comments 30
8 Model Definitions (.MODEL Statements) 31
9 Subcircuit Definitions 32
9.1 Subcircuit Scoping Rules 32
10 Subcircuit Definition Ending Statements 33
11 Elements 34
11.1 Subcircuits 34
11.2 Linear Resistor 34
11.3 Linear Capacitor 35
11.4 Voltage Source 36
11.5 Mutual Inductor 36
11.6 Linear Inductor 37

Formatted ... [1]
Formatted ... [2]
Formatted ... [3]
Formatted ... [4]
Formatted ... [5]
Formatted ... [6]
Formatted ... [7]
Formatted ... [8]
Formatted ... [9]
Formatted ... [10]
Formatted ... [11]
Formatted ... [12]
Formatted ... [13]
Formatted ... [14]
Formatted ... [15]
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted ... [16]
Formatted: Default Paragraph Font
Formatted ... [17]
Formatted ... [18]
Formatted ... [19]
Formatted ... [20]
Formatted ... [21]
Formatted ... [22]
Formatted ... [23]
Formatted ... [24]
Formatted ... [25]
Formatted ... [26]
Formatted ... [27]
Formatted ... [28]
Formatted ... [29]
Formatted ... [30]
Formatted ... [31]

11.7 T-element (Ideal Transmission Line) 37
11.8 W-element (Coupled Transmission Line) 39
Format 1: RLGC Model 40
Using RLGC Matrices 42
Format 2: Frequency-Dependent Tabular Specification 42
11.9 Frequency-Dependent Matrices 43
Small-Signal Parameter Data Frequency Table Model (SP Model) 44
11.10 S-element 48
S-Element Model Syntax 50
11.11 E-element (Voltage-Controlled Voltage Source) 52
Linear 52
Laplace Transform 52
Pole-Zero Function 52
Foster Pole-Residue Form 53
E-element Arguments 54
11.12 F-element (Current-Controlled Current Source) 55
11.13 G-element (Voltage-Controlled Current Source) 56
Linear 56
Laplace Transform 56
Pole-Zero Function 56
Foster Pole-Residue Form 57
G-element Arguments 57
11.14 H-element (Current-Controlled Voltage Source) 58
12 Best Practices 60
13 Theoretical Background 61
13.1 Introduction to the Complex Dielectric Loss Model 61
13.2 Frequency-Dependent Matrices 61
14 References 63

Formatted: Default Paragraph Font

Table 1: Document Conventions 10
Table 2: Prohibited Strings for Specific Elements 13
Table 3: IBIS-ISS Special Characters 14
Table 4: First Character Rules 17
Table 5: Element Identifiers 18
Table 6: Scale Factors 19
Table 7: .PARAM Statement Syntax and Examples 24
Table 8: IBIS-ISS Built-in Functions 26
Table 9: File Include Arguments 33
Table 10: Subcircuit Arguments 38
Table 11: Linear Resistor Arguments 39
Table 12: Linear Capacitor Arguments 39
Table 13: Voltage Source Arguments 40
Table 14: Mutual Inductor Arguments 41
Table 15: Linear Inductor Arguments 41
Table 16: T-element (Ideal Transmission Line) Arguments 42
Table 17: W-element (Coupled Transmission Line) Arguments 43
Table 18: W-element RLGC Model Arguments 45
Table 19: W-element SP Model Arguments 49
Table 20: W-element Model Definition Arguments 52
Table 21: S-element Arguments 53
Table 23: E-element Arguments 57
Table 24: F-element Arguments 61
Table 25: G-element Arguments 63
Table 26: H-element Arguments 66
Table 1: Document Conventions 10
Table 2: Prohibited Strings for Specific Elements 12
Table 3: IBIS-ISS Special Characters 13
Table 4: First Character Rules 16
Table 5: Element Identifiers 17
Table 6: Scale Factors 18
Table 7: .PARAM Statement Syntax and Examples 22
Table 8: Parameter Passing Order 23
Table 9: IBIS-ISS Built-in Functions 24
Table 10: IBIS-ISS Special Variables 27
Table 11: File Include Arguments 30
Table 12: Subcircuit Arguments 35
Table 13: Linear Resistor Arguments 36
Table 14: Linear Capacitor Arguments 36
Table 15: Voltage Source Arguments 37
Table 16: Mutual Inductor Arguments 37
Table 17: Linear Inductor Arguments 38
Table 18: T-element (Ideal Transmission Line) Arguments 39
Table 19: W-element (Coupled Transmission Line) Arguments 40
Table 20: W-element RLGC Model Arguments 42
Table 21: W-element SP Model Arguments 45

Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font

~~Table 22: W-element Model Definition Arguments ..... 48~~
~~Table 23: S-element Arguments ..... 49~~
~~Table 24: S-element Model Definition Arguments ..... 51~~
~~Table 25: E-element Arguments ..... 55~~
~~Table 26: F-element Arguments ..... 56~~
~~Table 27: G-element Arguments ..... 59~~
~~Table 28: H-element Arguments ..... 59~~

Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font
Formatted: Default Paragraph Font

1 Overview

The IBIS Open Forum, in order to enable easier data exchange between users of signal/power integrity simulation and physical layout/routing software tools, is issuing a generic circuit description~~netlist~~ format, to be called ~~"~~"IBIS Interconnect SPICE Subcircuits~~"~~" (IBIS-ISS).

This format is similar in structure and major functions to the SPICE (Simulation Program with Integrated Circuit Emphasis) nodal syntax developed at the University of California at Berkeley and since implemented in various forms by individual software tool vendors. IBIS-ISS is the first industry-wide attempt to standardize SPICE subcircuit representation.

This version of IBIS-ISS is based on a subset of HSPICE®, used with permission from Synopsys, Inc. HSPICE is a registered trademark of Synopsys, Inc.

Formatted: Font: (Default) Helvetica

2 Goals and Scope

The syntax of IBIS-ISS is intended to:

Formatted: Font: (Default) Helvetica

- describe interconnect structures (such as PCB traces, connectors, cables, etc.) electrically, for analysis in a signal integrity and/or power integrity context
- describe the arrangement or topology of interconnect structures, as they relate to each other and to active devices in a system

To these ends, IBIS-ISS includes support for:

- elementary circuit elements (resistors, capacitors, inductors)
- transmission line elements (lossless and lossy)
- frequency-domain network parameters (e.g., S-parameters)
- parameter/variable passing to elements and subcircuits
- dependent and limited independent sources
- string-based node naming
- user-defined comments
- abstraction through modular, user-defined subcircuit definitions

IBIS-ISS does NOT include or cover:

- descriptions of complete netlists intended for input "as-is" to simulation tools
- model formats or "process cards" for active devices (e.g., diodes, transistors)
- ~~independent sources~~
- controls or options for any simulation engine (e.g., precision, algorithm selection)
- simulation or analysis types (e.g., DC, transient)
- sweep or run control (e.g., Monte Carlo)
- geometrical descriptions for field solvers
- support for other kinds of data extraction/export (e.g., S-parameter generation)
- measurement, printing or probing
- encryption support

3 Conventions

The following typographical conventions are used in IBIS-ISS. Note that these may be combined (e.g., Courier font in bold type).

Table 1: Document Conventions

Convention	Description
`Courier`	Indicates statement syntax
Italic	Indicates a user-defined value, where a specific text string other than the italics shown will replace the ~~italics~~ italic text shown ~~in an actual IBIS ISS file~~ (e.g., R*xxxx* is a generic representation of a resistor element name, such as `Rname`)
Bold	Indicates verbatim text in syntax descriptions
[ ]	Denotes optional arguments
...	Indicates that arguments of the same type may be added as appropriate to the element structure: pin1 pin2 ... pin
|	Indicates a choice among defined alternatives, such as low | medium | high
+	Indicates a continuation of a statement across ~~line termination~~line-termination sequence(s). Note that continuation may only be used between arguments and shall not split any single non-quoted-string argument across ~~line~~line-termination sequences.~~s~~

4 Input Structure and Data Entry

This section describes the input file and structures for representing input data.

4.1 Input File Guidelines

An input file consists of a collection of statements describing a portion of a complete circuit. This input file is intended for inclusion in a larger netlist or description of a complete circuit, to be used by a simulation tool.

An input filename may be up to 1024 characters long. The input file shall be in ASCII format (as defined in ISO/IEC 8859-1:1998, with each character encoded in 8 bits). The input file shall not be in a binary, packed or compressed format.

Any individual input line (combination of characters separated by line-termination sequences) may be up to 1024 characters long.

Formatted: Font: (Default) Helvetica

4.2 Statements and Arguments

A statement in IBIS-ISS is a text string consisting of character sequence arguments and delimiters. An IBIS-ISS file may contain multiple statements (the number of statements is not limited by the IBIS-ISS definition, but may be limited by the computer architecture and/ or operating system used to process the file).

Statements may be split across line-termination sequences, provided a line continuation character or sequence (defined later) is used. Statements shall be separated by line-termination sequences.

~~Any individual input line (combination of characters separated by line-termination sequences) may be up to 1024 characters long.~~

Formatted: BulletNext, Indent: Left: 0"

Statements in an input file may appear in any order.

Any valid string of characters between two delimiters is an argument.

For the purpose of this specification, statements are grouped into the following types:

- Element instances
- Parameter definitions
- File includes
- Subcircuit definitions
- Model definitions
- Comments
- Subcircuit ending statements

Subcircuit ending statements, subcircuit definitions, model definitions, parameter definitions and file includes all begin with the ~~dot~~ period (.) character.

The specific syntax of the above statement types are described in the sections below.

- IBIS-ISS ignores differences between upper and lower case in input statements, except in quoted filenames.
- To continue a statement across multiple lines (i.e., across a line-termination sequence), the plus (+) sign shall be used as the first ~~non-alphanumeric,~~ non-blank character after the line-termination sequence. The + sign shall be used in addition to a delimiter and never to split up the character sequences between delimiters.
- Quoted strings with extended length (such as file names including paths~~paths~~ and expressions) may span a line-termination sequence using a double backslash (\\ ) sequence ~~at the end of the line containing the string to be continued on the following line~~, where the double backslash precedes the line-termination sequence. Note that the double backslash (\\ ) sequence may be preceded by whitespace when used within a quoted string. Individual arguments may be split across line-termination sequences using a double backslash, but leading whitespace shall not be used. For example:

```
*** string continuation ***
R6 4 0 R='res1-\\
res2'
R5 4 0 R='res1- \\
res2'
*** token continuation ***
R4 node1 no\\
de2 R= 'res1-res2'
```

- The following characters are reserved for special use and shall not be used as part of any parameter or node name:

Formatted: Font: (Default) Courier New

Formatted: Indent: Left: 1.5", Space After: 0 pt, No bullets or numbering

Formatted: Font: (Default) Courier New

Formatted: Font: (Default) Courier New

Formatted: Indent: Left: 0.5", Space After: 0 pt, No bullets or numbering

( ) = " '

Formatted: Indent: Left: 0"

- "Directional" or matched quotation marks, as represented by ISO 8859-1 (extended ASCII) characters 0x145d, 0x146d, 0x147d and 0x148d, are prohibited in IBIS-ISS files. ~~String expressions may be grouped by surrounding pairs of ‘ or “ characters.~~
- To prevent conflicts with proprietary SPICE variants, the following strings shall not be used as part of any parameter or node name ~~in the associated element~~: time, temper, hertz. Further, the strings listed in Table 2Table 2shall not be used as part of any parameter or node name in the associated element.

Table 2: Prohibited Strings for Specific Elements

Element	Prohibited String Sequence
Capacitor	POLY, TC, SENS
E-element G-element	AND, DELAY, FOSTER, LAPLACE, NAND, NPWL, NOR, VCCS, OPAMP, OR, POLE, POLY, PWL, SPUR, TRANSFORMER, VCR, VCCAP, VCVS, FREQ, ZTRANS, VMRF, NOISE, NOISEFILE, MNAME, PHASE, SCALE, MAX, PAR
F-element H-element	POLY, PWL, AND, NAND, OR, NOR, VMRF, CCCS, CCVS, DELAY
Inductor	POLY, TC, SENS, RELUCTANCE, TRANSFORMER_NT, FILE
Resistor	POLY,TC,SENS
S-element	ZO, Z0, MNAME
T-element	IC
W-element	RLGCFILE, PRINTZO, RLGCMODEL,TABLEMODEL, FSMODEL, UMODEL, SMODEL

4.3 Special Characters

The following table lists the special characters that may be used as part of node names, element parameter names, and element instance names. For detailed discussion, see the appropriate sections in this chapter.

Note:

To avoid unexpected results or error messages, do not use the following mathematical characters in a parameter name in IBIS-ISS: * - + ^ and /.

Table 3: IBIS-ISS Special Characters

Special Character*		Usage in Node Names	Usage in Element and Subcircuit Instance Names**	Usage in Parameter Names**	CommentsRemarks
~	Tilde	Legal anywhere	Included only	Included only	n/a
!	Exclamation point	Legal anywhere	Included only	Included only	n/a
@	At sign	Legal anywhere	Included only	Included only	n/a
#	Pound sign	Legal anywhere	Included only	Included only	n/a
$	Dollar sign	Included only (avoid if after a number in node name)	Included only	Included only	In-line comment character
%	Percent	Legal anywhere	Included only	Included only	n/a
^	Caret	Legal anywhere	Included only	Included only (avoid usage)	"To the power of", i.e., 2^5, two raised to the fifth power

Formatted Table

&	Ampersand	Legal anywhere	Included only	Included only	n/a
*****	Asterisk	Included only (avoid using in node names)	Included only	Included only (avoid using in parameter names)	Comment and wildcard character. Double asterisk (**) is "to the power of".
( )	Parentheses	Illegal	Illegal	Illegal	Delimiter
-	Minus	Included only	Included only	Illegal	n/a
_	Underscore	Legal anywhere	Included only	Included only	n/a
+	Plus sign	Included only	Included only	Included only (avoid usage)	Continues previous line, except for quoted strings
=	Equals	Illegal	Illegal	Optional in .PARAM statements	Delimiter
< >	Less/more than	Legal anywhere	Included only	Included only	n/a
?	Question mark	Legal anywhere	Included only	Included only	Wildcard character
/	Forward slash	Legal anywhere	Included only	Illegal	n/a
{ }	Curly braces	included only, converts to [ ]	Included only	Included only	Engine shall auto-convert to square brackets

					([ ])
[ ]	Square brackets	Included only	Included only	Included only	n/a
****	Double backslash	Included only	Illegal	Illegal	Continuation character sequence
|	Pipe	Legal anywhere	Included only	Included only	n/a
,	Comma	Illegal	Illegal	Illegal	Delimiter
.	Period	Illegal	~~Included only~~Illegal	I~~ncluded only~~legal	Used only as a ~~Statement~~ special statement identifier~~,~~ (i.e., .PARAM~~ETER , etc.~~).
:	Colon	Included only	Included only	Included only	Delimiter for element attributes
;	Semi-colon	Included only	Included only	Included only	n/a
" "	Double-quotes	Illegal	Illegal	Illegal	Expression and filename delimiter
' '	Single quotes	Illegal	Illegal	Illegal	Expression and filename delimiter

	Blank (whitespace)	May be used before \\ sequence for continuations	Illegal	Illegal	Delimiter

* ~~"Legal anywhere"=first character or~~ any position in string, including the first character

~~"Included only"=any position except first character~~

** cannot be the first character; element key letter only

"Legal anywhere" = any position in string, including the first character
"Included only"=any position except first character

4.4 First Character

The first non-blank character ~~in every line~~after a line-termination sequence specifies how IBIS-ISS interprets the remainder of the ~~line~~input.

Table 4: First Character Rules

If the First Character is...	Indicates
. (period)	Statement identifier (e.g., .PARAM)
c, C, e, E, f, F, g, G, h, H, k, K, l, L, r, R, s, S, t, T, v, V, w, W, x, X	Element instantiation
* (asterisk)	Comment ~~line~~
+ (plus)	Continues previous input~~line~~

4.5 Delimiters

Delimiters separate arguments in the input file. Input delimiters are: tab, blank, comma (,), equal sign (=), and parentheses ( ).

In addition, single (') or double quotes (") delimit and group expressions and filenames.

4.6 Instance Names

The names of element instances begin with the element key letter or identifier as listed below. Instance names may be up to 1024 characters long.

Table 5: Element Identifiers

Key Letter (First Char)	Element	Example Statement~~Line~~
C	Capacitor	`Cbypass 1 0 10pf`
E	Voltage-controlled voltage source	`Ea 1 2 3 4 1K`
F	Current-controlled current source	`Fsub n1 n2 vin 2.0`
G	Voltage-controlled current source	`G12 4 0 3 0 10`
H	Current-controlled voltage source	`H3 4 5 Vout 2.0`
K	Linear mutual inductor (general form)	`K1 L1 L2 1`
L	Linear inductor	`LX a b 1e-9`
R	Resistor	`R10 21 10 1000`
S	S-parameter element	`S1 nd1 nd2 s_model2`

T	Transmission Line	`Txxx in 0 out 0 Zo=50 TD=30n`
V	DC Voltage source~~)~~	`V1 8 0 DC=0`
W	Transmission Line	`W1 in1 0 out1 0 N=1 L=1`
X	Subcircuit instance	`X1 2 4 17 31 MULTI WN=100 LN=5`

4.7 Numbers

Numbers may be entered as integer, floating point, floating point with an integer exponent, or integer or floating point with one of the scale factors listed below.

Table 6: Scale Factors

Scale Factor	IEEE Standard Prefix	IEEE Standard Symbol	Multiplying Factor
T	~~Tera~~tera	T	1e+12
G	giga	G	1e+9
MEG	mega	M	1e+6
K	kilo	k	1e+3
MIL	n/a	mil	25.4e-6
M	milli	m	1e-3
U	micro	µ	1e-6
N	nano	n	1e-9

P	pico	p	1e-12
F	~~f~~Femto	f	1e-15
A	~~Atto~~atto	a	1e-18

Note:

Scale factor A is not a scale factor in a character string that contains amps. For example, IBIS-ISS-compliant tools shall interpret the string ~~"~~"20amps~~"~~" as 20 amperes of current, not as 20e-18mps.

- Numbers may use exponential format or engineering key letter format, but not both (1e-12 or 1p, but not 1e-6u).
- To designate exponents, use D or E.
- Trailing alphabetic characters are interpreted as units comments (for example, 1kV is interpreted as 1k, while 1w is interpreted as 1).
- Units comments are not checked.

4.8 Parameters and Expressions

- ~~Parameter names shall begin with an alphabetic character. Subsequent characters in the parameter name shall each be either a digit, or one of these following characters:~~

~~! # $ % [ ] _~~

- Parameter names shall begin with an alphabetic character. Subsequent characters in the parameter name shall be either alphanumeric characters or special characters as defined in Table 3: IBIS-ISS Special Characters
- If multiple definitions are given for the same parameter, IBIS-ISS uses the last parameter definition even if that definition occurs later in the input than a reference to the parameter.
- A parameter shall be defined before that parameter is used in a definition for another parameter.
- To delimit expressions, single quotes shall be used.
- Expressions shall not exceed 1024 characters.

Parameters are evaluated only once, at the time of parsing. Dynamic, recursive or iterative definitions of parameters are prohibited (e.g., parameters defined in terms of

Formatted: Font: (Default) Helvetica

Formatted: BulletNext, Space After: 0 pt, Tab stops: Not at 1.44" + 1.89" + 2.35" + 2.79" + 3.25" + 3.69" + 4.14" + 4.6" + 5.04" + 5.5"

the voltage at a node, where that voltage is evaluated at every time step in a transient analysis).

- Parameters are used in two contexts.
  - Parameters in parameter definition statements are strings, defining names that are assigned specific values by the statement. These values may themselves be interpreted as strings (using the ~~"~~"str()~~"~~" construction noted elsewhere), numeric values, an expression or equation, or strings matching parameters defined elsewhere.
  - Parameters may also appear in element instances, model definitions and subcircuit definitions. These parameters may be user-defined or may use names ~~pre-~~defined by the syntax of the element. Parameter names are input arguments. Delimiters shall precede and follow names.
- Parameter names may be up to 1024 characters long and are not case-sensitive.

4.9 Node Name (or Node Identifier) Conventions

Nodes are the points of connection between elements in the input ~~netlist~~circuit description. ~~Only alphanumeric strings shall be used to designate nodes.~~ If entirely numeric, node numbers shall be between 1 and 999999999999999 (1 to 1e16-1). A node number of 0 is permitted but is interpreted as ground. Letters that follow a leading number in a node name are ignored; this means that node strings such as ~~"~~3n5~~"~~ and ~~"~~3~~"~~ shall be interpreted as referring to the same node.

When the node name begins with a letter or a valid special character, the node name may contain a maximum of 1024 characters. See Table 3: IBIS-ISS Special Characters for a list of valid special characters.

To indicate the ground node, use either the number `0`, or the names `GND`, ~~or~~ `!GND`, ~~or~~ `GROUND`, or `GND!`. Every node shall have at least two connections, except for transmission line nodes (unterminated transmission lines are permitted).

4.10 Element, Instance, and Subcircuit Naming Conventions

Instances and subcircuits are elements and as such, follow the naming conventions for elements.

Element names begin with an identifying letter designating the element type, followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters). Element identifiers are R for resistor, C for capacitor and so on.

Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica

4.11 Line Continuations

~~Input~~ Statements may be continued across a line-termination sequence in one of three ways:

- Statements are continued across line-termination sequences using the + character as the first non-blank character in the continued input
- Quoted strings are continued across line-termination sequences using the \\ sequence as the last two characters before the line-termination sequence
- Individual arguments are continued using the \\ sequence, without leading whitespace

~~To continue a statement across multiple lines, the plus (+) sign shall be used as the first non-numeric, non-blank character of each continued line.~~ The + sign shall be used only between arguments and delimiters and never to split up the character sequence used for a single argument.

Here is an example of comments and line continuation in a ~~netlist~~ statement~~file~~:

```
*This shows continuation of a statement describing a
resistor
Rexample
+ n3 n4 R=30
```

To continue a statement with extended length across a line-termination sequence, an argument may be split using a double backslash (\\ ) sequence. Leading whitespace shall not be included before the double backslash sequence. A quoted string (such as an expression) may be continued across a line-termination sequence using a double backslash sequence with or without leading whitespace.

Here are examples of string and argument continuation.

```
*** string continuation ***
R6 4 0 R='res1-\\
res2'
R5 4 0 R='res1- \\
res2'
*** argument continuation ***
R4 node1 no\\
de2 R= 'res1-res2'
```

Formatted: Font: (Default) Helvetica

Note that whitespace leading the continued input will be interpreted as a delimiter. In the example above, whitespace before the string de2 will cause an error, as R4 will appear to have three nodes.

Formatted: Font: (Default) Courier New

Formatted: Font: (Default) Courier New

4.12 IBIS-ISS Structure

IBIS-ISS files shall include at least one subcircuit definition, aside from any included files. An IBIS-ISS file may contain multiple subcircuit definitions.

Formatted: Indent: Left: 0"

5 Parameters

Parameters are similar to the variables used in most programming languages. Parameters hold values assigned when the circuit design is created ~~or that are calculated based on circuit solution values~~. Parameters may store static values for a variety of quantities (resistance, source voltage, rise time, and so on). Parameters may also be alphabetic strings used with elements where string input is expected (for example, filenames or model names).

Formatted: Font: (Default) Helvetica

5.1 Using Parameters in Simulation (.PARAM)

Defining Parameters

Parameters may be defined using the methods shown below. Note that a .param statement without an assignment is not permitted.

Table 7: .PARAM Statement Syntax and Examples

Formatted Table

Usage	Description/Example
Simple Assignment	`.PARAM SimpleParam=1e-12`
Algebraic Definition	`.PARAM AlgebraicParam='SimpleParam*8.2'`
User-defined Function	`.PARAM MyFunc(x,y)='SQRT((x*x)+(y*y))'`
String Assignment	`.PARAM StringParam=str('mystring')`
Subcircuit Definition	`.SUBCKT SubName ParamDefName=Value`

Subcircuit Instance	`Xxxx nodename1 ... nodenamen`
	`+ SubName`
	`+ ParamDefName = Value | str('string')`

~~Predefined Analysis Function~~	~~`.PARAM f(a,b)=POW(a,2)+a*b g(d)=SQRT(d)`~~ ~~`+ h(e)=e*f(1,2) g(3)`~~

A parameter definition in IBIS-ISS always uses the last value found in the input ~~netlist~~statements. The definitions below assign a value of 3 to the `DupParam` parameter.

```
.PARAM DupParam=1
...
.PARAM DupParam=3
```

IBIS-ISS assigns `3` as the value for all instances of `DupParam`, including instances that are earlier in the input than the `.PARAM DupParam=3` statement.

Note that any ~~ordered~~ tail-truncated substring of "".parameters"" containing at least the characters "".para"" is also acceptable as an alternative to "".param"".

The parameter resolution order is:

1. Resolve all literal assignments.
2. Resolve all expressions.
3. Resolve all function calls.

~~**Table 8 shows the parameter passing order.**~~

~~**Table 8: Parameter Passing Order**~~

~~**.PARAM statement ()**~~	~~**.SUBCKT call (instance)**~~
~~**.SUBCKT call (instance)**~~	~~**.SUBCKT definition (symbol)**~~
~~**.SUBCKT definition (symbol)**~~	~~**.PARAM statement ()**~~

Formatted: Font: Not Bold
Formatted: Heading 3
Formatted: Heading 3, Don't keep with next
Formatted: Heading 3
Formatted: Heading 3
Formatted: Heading 3
Formatted: Heading 3

Assigning Parameters

The following types of values may be assigned to parameters:

- Constant real number
- Algebraic expression of real values

- ~~Predefined function~~
- Strings not for algebraic evaluation

Any complex expression to be evaluated shall be enclosed in single quotes.

A simple expression consists of one parameter name. Simple expressions shall not be enclosed in single or double quotes.

The parameter keeps the assigned value, unless a later definition changes its value.

Assignments of different values to the same parameter name in different areas of the IBIS-ISS file are resolved in the following order (in decreasing order of priority):

1. .SUBCKT call (where a parameter is assigned as part of a subcircuit instance)
2. .SUBCKT definition (where a parameter symbol is defined for passing in a value, and a default value is assigned)
3. .PARAM statement

See also Section 5.4 for additional rules and an example.

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Character scale: 100%

Formatted: Indent: Left: 0"

5.2 Using Algebraic Expressions

In IBIS-ISS, an algebraic expression, with quoted strings, may replace any parameter. Some uses of algebraic expressions are:

- Parameters:

```
.PARAM x='y+3'
```

- Inline ~~algebra~~ expressions in elements:

```
R1 1 0 r='27*3.14'
```

~~As noted above, the continuation character sequence in IBIS-ISS is a double backslash (\\). For quoted strings, the double backslash may be preceded by whitespace or used without whitespace.~~

Built-In Functions and Variables

In addition to simple arithmetic operations (+, -, *, /), the built-in functions and variables listed below may be used in IBIS-ISS expressions.

Table ~~8~~9: IBIS-ISS Built-in Functions

IBIS-ISS Form	Function	Class	Description
sin(x)	sine	trig	Returns the sine of x ~~(radians)~~

cos(x)	cosine	trig	Returns the cosine of x ~~(radians)~~
tan(x)	tangent	trig	Returns the tangent of x ~~(radians)~~
asin(x)	arc sine	trig	Returns the inverse sine of x ~~(radians)~~
acos(x)	arc cosine	trig	Returns the inverse cosine of x ~~(radians)~~
atan(x)	arc tangent	trig	Returns the inverse tangent of x ~~(radians)~~
sinh(x)	hyperbolic sine	trig	Returns the hyperbolic sine of x ~~(radians)~~
cosh(x)	hyperbolic cosine	trig	Returns the hyperbolic cosine of x ~~(radians)~~
tanh(x)	hyperbolic tangent	trig	Returns the hyperbolic tangent of x ~~(radians)~~
abs(x)	absolute value	math	Returns the absolute value of x: |x|
sqrt(x)	square root	math	Returns the square root of the absolute value of x: sqrt(-x)=-sqrt(|x|)
pow(x,y)	absolute power	math	Returns the value of x raised to the integer part of y: $x^{(\text{integer part of y})}$
pwr(x,y)	signed power	math	Returns the absolute value of x, raised to the y power, with the sign of x: (sign of x)$\|x\|^y$

x**y	power		If x<0, returns the value of x raised to the integer part of y. If x=0, returns 0. If x>0, returns the value of x raised to the y power.
log(x)	natural logarithm	math	Returns the natural logarithm of the absolute value of x, with the sign of x: (sign of x)log(|x|)
log10(x)	base 10 logarithm	math	Returns the base 10 logarithm of the absolute value of x, with the sign of x: (sign of x)$\log_{10}(\|x\|)$
exp(x)	exponential	math	Returns e, raised to the power x: e^x
db(x)	decibels	math	Returns the base 10 logarithm of the absolute value of x, multiplied by 20, with the sign of x: (sign of x)$20\log_{10}(\|x\|)$
int(x)	integer	math	Returns the integer portion of x. The fractional portion of the number is lost.
nint(x)	integer	math	Rounds x up or down, to the nearest integer.
sgn(x)	return sign	math	Returns -1 if x is less than 0. Returns 0 if x is equal to 0. Returns 1 if x is greater than 0
sign(x,y)	transfer sign	math	Returns the absolute value of x, with the sign of y: (sign of y)|x|
def(x)	parameter defined	control	Returns 1 if parameter x is defined. Returns 0 if parameter x is not defined.

min(x,y)	smaller of two args	control	Returns the numeric minimum of x and y
max(x,y)	larger of two args	control	Returns the numeric maximum of x and y
[cond] ?x : y	ternary operator		Returns *x* if *cond* is not zero. Otherwise, returns *y*. .param z='condition ? x:y'
<	relational operator (less than)		Returns 1 if the left operand is less than the right operand. Otherwise, returns 0. .param x=y<z (y less than z)
<=	relational operator (less than or equal)		Returns 1 if the left operand is less than or equal to the right operand. Otherwise, returns 0. .param x=y<=z (y less than or equal to z)
>	relational operator (greater than)		Returns 1 if the left operand is greater than the right operand. Otherwise, returns 0. .param x=y>z (y greater than z)
>=	relational operator (greater than or equal)		Returns 1 if the left operand is greater than or equal to the right operand. Otherwise, returns 0. .param x=y>=z (y greater than or equal to z)
==	equality		Returns 1 if the operands are equal. Otherwise, returns 0. .param x=y==z (y equal to z)
!=	inequality		Returns 1 if the operands are not equal. Otherwise, returns 0. .param x=y!=z (y not equal to z)

&&	Logical AND		Returns 1 if neither operand is zero. Otherwise, returns 0. .param x=y&&z (y AND z)
||	Logical OR		Returns 1 if either or both operands are not zero. Returns 0 only if both operands are zero. .param x=y||z (y OR z)

~~Table 10: IBIS-ISS Special Variables~~

~~IBIS-ISS Form~~	~~Function~~	~~Class~~	~~Description~~
~~Time~~	~~current simulation time~~	~~control~~	~~Uses parameters to define the current simulation time, during transient analysis.~~
~~Temper~~	~~current circuit temperature~~	~~control~~	~~Uses parameters to define the current simulation temperature, during transient/temperature analysis.~~
~~Hertz~~	~~current simulation frequency~~	~~control~~	~~Uses parameters to define the frequency, during AC analysis.~~

5.3 String Parameters

Parameters may be defined and instantiated using strings. String parameters ~~must~~ use special syntax; characters such as single quotes (‘’), double quotes (“”), or curly brackets ({}) alone are not sufficient for string parameter definition or instantiation.

When defining a parameter that is a character string, the combination `str('`*`string`*`')` ~~shall~~ may be used to define the parameter, where *`string`* is ~~be~~ the string to be used as the parameter value. When ~~an instance of~~ the parameter is used, the parameter name is called as `str(`*`parameter_name`*`)`. Note that quotation marks are not used in the call `str(`*`parameter_name`*`).`

IBIS-ISS supports string parameter definition and instantiation for the following ~~netlist components~~:

- `.PARAM` statements
- `.SUBCKT` statements
- `TSTONEFILE` keyword in the S-element
- ~~`FILE` and `MODEL` arguments~~
- W-element arguments `RLGCMODEL` and `TABLEMODEL`

Note that the `str(`*`parameter_name`*`)`construction is not required for arguments that take string inputs (for example, RLGCMODEL and TABLEMODEL assignments). The `str(`*`parameter_name`*`)` construction is used when

Formatted: Font: (Default) Helvetica

Formatted: Font: (Default) Helvetica

5.4 Parameter Scoping and Passing

A parameter is defined either by a `.param` statement (local to the ~~that~~ subcircuit in which it is defined), or may be passed into a subcircuit, or may be ~~defined on~~assigned as part of a .subckt definition ~~line~~.

All parameters defined within a subcircuit are local to that subcircuit under the defined name. Parameter values may be passed between subcircuits, but the values ~~must~~shall be passed explicitly, with the parameter present both in the subcircuit definition and in the instance where it is used. IBIS-ISS does not support global parameters.

```
.subckt def 1 2
.param x=1
x1 1 2 abc x=2
.subckt abc 1 2 x=3
.param x=34
r1 1 2 R=x
.ends abc
.ends def
```

In the example above, the value of r1 is 2 ohms. The .SUBCKT definition x=3 and the local .PARAM assignments x=1 and x=4 are overridden by the .SUBCKT instance assignment x=2.

Formatted: Indent: Left: 0"

6 File Includes

The include statement inserts another file's contents in the current file at evaluation. Note that any ordered tail-truncated substring of ".include" containing at least the characters ".inc" is also acceptable as an alternative to ".include".

An include file may can contain nested .INCLUDE calls to another include file.

Syntax

.INCLUDE '*file_path file_path_and_name*'

.inc '*file_path file and_name*'

Table 9 11: File Include Arguments

Argument	Description
file_path_and_name	Path File name, optionally preceded by a file path, of a file for computer operating systems that support tree-structured directories. Any strings, including directory separator characters, valid under the computer's operating system may be used. An include file can contain nested .INCLUDE calls to another include file.
file_name	Name of a file to include in the data file. Any name valid under the computer's operating system may be used.

Formatted Table

Use this statement to include the contents of another netlist file in the current circuit description. The file path and file name shall be enclosed in single or double quotation marks. Relative paths are strongly suggested.

```
.INCLUDE '~/myhome/subcircuits/circuit'
```

7 Comments

Comments require an asterisk (*) as the first non-blank character ~~in a line~~after a line-termination sequence or a dollar sign ($) directly in front of the comment ~~anywhere on the line~~ text.

Syntax

** <comment_on_a_line_by_itself>*

or

<IBIS-ISS statement> $ <comment following input>

Comment statements may appear anywhere in the circuit description. The dollar sign ($) shall be used for comments that do *not* begin in the first character position ~~on~~ after a line-termination sequence (for example, for comments that follow simulator input ~~on the same line~~without a separating line-termination sequence). If it is not the first nonblank character, then the dollar sign shall be preceded by either:

- Whitespace
- Comma (,)
- Valid numeric expression

The dollar sign may also be used within node or element names. For example:

```
* RF=1K GAIN SHOULD BE 100
$ CIRCUIT EXAMPLE
VIN 1 0 PL 0 0 5V 5NS $ 10v 50ns
R12 1 0 1MEG $ FEED BACK
.PARAM a=1w$comment a=1, w treated as a space and ignored
.PARAM a=1k$comment a=1e3, k is a scale factor
```

In the example above, the first parameter statement definition assigns a value of 1 to the parameter a, as training alphabetic characters are interpreted as units comments and w is not a valid scale factor. The second parameter statement definition uses k, a valid scale factor, which changes the assigned value of a to 1000 (see Section 4.7).

A dollar sign is the preferred way to indicate comments, because of the flexibility of its placement within the code.

Comments shall not appear after the double-slash (\\) ~~line continuation~~ sequence ~~on the same line (~~before a ~~line termination~~line-termination sequence~~)~~.

Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica
Formatted: Font: (Default) Helvetica

8 Model Definitions (.MODEL Statements)

Model definitions are used to specify the electrical parameters for W-element and S-element instances. ~~They can be considered a special form of subcircuit definition, in which the defined subcircuit is only available to W- and S-elements.~~

The specific syntax for W-element and S-element .MODEL definitions is ~~are~~ detailed below, as part of the W-element and S-element portions of the IBIS-ISS specification. ~~Note that .MODEL statements are hierarchically at the same level as element instances.~~

9 Subcircuit Definitions

A subcircuit definition groups a set of statements into a module that may be reused an unlimited number of times. The definition is instantiated using the X-element. At least one node is required. Parameters may be optionally assigned to a value as part of the .subckt definition. The parameter value is a default and may be overridden in instantiations of the subcircuit by assignments using the same parameter name.

Syntax

```
.subckt name n1 [n2 n3 …] [parnam=val]
statement
statement
statement
…
.ends
```

Formatted: Font: Bold
Formatted: Font: Not Italic
Formatted: Font: Not Italic
Formatted: Font: Bold
Formatted: Font: Not Italic
Formatted: Font: Bold

9.1 Subcircuit Scoping Rules

A .subckt or .model definition shall occur in the subcircuit in which the subcircuit or model is referenced, or in a calling subcircuit at any level above.

Formatted: Font: (Default) Helvetica

10 Subcircuit Definition Ending Statements

Subcircuit definitions shall end with the .ends statement. See Subcircuit Definitions~~Subcircuit Definitions~~ above for syntax and examples.

Formatted: Font: (Default) Helvetica

11 Elements

The sections below describe the individual circuit elements that may appear in an IBIS-ISS file.

11.1 Subcircuits

Reusable cells are the key to saving labor in any CAD system. To create and simulate a reusable circuit, construct it as a subcircuit. Parameters are used to expand the utility of a subcircuit.

X~~<*subcircuit_name*xxx>~~ creates an instance of a ~~a~~ subcircuit. ~~.~~The subcircuit shall have already been defined elsewhere in the IBIS-ISS file using a `.SUBCKT` statement.

Syntax

X*xxx* *n1* [*n2 n3* …] *subnam*
[*parnam* = *val*] [**M** = *val*]

The name of an X-element instance shall begin with the character ~~"~~"X~~"~~", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters)~~.~~.

Table 10~~12~~: Subcircuit Arguments

Argument	Definition
n1 …	Node names for external reference.
subnam	Subcircuit model reference name.
parnam=*val*	A parameter name (*parnam*) set to a value (*val*) for use only in the subcircuit. It overrides a parameter value in the subcircuit definition~~, but is overridden by a value set in a .PARAM statement~~.
M=*val*	Multiplier value *val* instantiates *val* number of subcircuits connected in parallel to the named nodes. The multiplier *val* shall be a positive, non-zero~~n~~ integer.

Formatted: Font: (Default) Helvetica

Formatted: Font: Bold

Formatted: Font: (Default) Courier New

Formatted: Space After: 0 pt

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Formatted Table

11.2 Linear Resistor

A linear resistor is a basic electrical circuit element for impeding current flow.

Syntax

R*xxx n~~ode~~1 n~~ode~~2* **[R =]** *value*

The value of a linear resistor may be a constant, or an expression of parameters.

The name of an R-element instance shall begin with the character "R", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Table 11~~13~~: Linear Resistor Arguments

Argument	Description
n~~ode~~1, *n~~ode~~2*	Names ~~or numbers~~ of ~~the~~ connecting nodes.
R=*value*	Resistance value, in ohms. The text "**R=**" is optional.

11.3 Linear Capacitor

A linear capacitor is a basic electrical circuit element for charge storage.

Syntax

C*xxx n~~ode~~1 n~~ode~~2* **[C =]** *value*

The value of a linear capacitor may be a constant, or an expression of parameters.

The name of a C-element instance shall begin with the character "C", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Table 12~~14~~: Linear Capacitor Arguments

Argument	Description

node1, *node2*	Names of connecting nodes.
C=*value*	Capacitance value, in farads. The text "**C=**" is optional.

11.4 Voltage Source

A voltage source establishes a DC potential difference between two nodes. A zero-volt source may be used to create a short between two nodes.

Syntax

V*xxx node1 node2* **[DC =]** *val*

The name of a V-element instance shall begin with the character "V", followed by up to 1023 alphanumeric characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Table 13: Voltage Source Arguments

Argument	Description
node1, *node2*	Nodes between which the source is placednames of connecting nodes.
DC=*val*	Sets *val* as the DC voltage between the source nodes. The text "**DC=**" is optional.

11.5 Mutual Inductor

A mutual inductor describes inductive coupling between two defined inductors.

Syntax

K*xxx* **L***yyy* **L***zzz* **[K =]** *coupling*

The name of a K-element instance shall begin with the character "K", followed by up to 1023 alphanumeric characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Table 14~~16~~: Mutual Inductor Arguments

Argument	Description
L*yyy*	Name of the first of two coupled inductors. This inductor shall be defined elsewhere in the file.
L*zzz*	Name of the second of two coupled inductors. This inductor shall be defined elsewhere in the file.
K=*coupling*	Sets *coupling* as the coefficient of mutual coupling. This is a positive non-zero unitless real number~~, with magnitude > 0~~. If the coupling coefficient is negative, the direction of coupling reverses. This is equivalent to reversing the polarity of either of the coupled inductors. Use the K=*xxx* syntax when defining the coupling coefficient using a parameter name or an equation. The text "K=" is optional.

11.6 Linear Inductor

Syntax

L*xxx n~~ode~~1 n~~ode~~2* [**L =**] *inductance*

The name of an L-element instance shall begin with the character "L", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Table 15~~17~~: Linear Inductor Arguments

Argument	Description
n~~ode~~1, n~~ode~~2	Names ~~or numbers~~ of ~~the~~ connecting nodes.
L=*inductance*	Sets *inductance* as the inductance value, in henries. The text "L=" is optional.

11.7 T-element (Ideal Transmission Line)

Syntax

T*xxx in refin out refout* **Zo=***val* **TD=***val* [**L=***val*]

Formatted: Font: Not Italic

Formatted: Font: Not Italic

The name of a T-element instance shall begin with the character ""T"", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Table 16~~18~~: T-element (Ideal Transmission Line) Arguments

Argument	Description
i~~l~~n	Signal input node.
refin	Ground reference for the input signal.
out	Signal output node.
refout	Ground reference for the output signal.
Zo=*val*	Sets *val* as the characteristic impedance of the transmission line in ohms. Note that the beginning character sequence may be either zo or z0.
TD=*val*	Sets *val* as the propagation time delay of the transmission line, in seconds. If physical length (L) is specified, then units for TD are considered in seconds per meter.
L=*val*	Sets *val* as the physical length of the transmission line, in meters. Default=1.

11.8 W-element (Coupled Transmission Line)

~~This section describes how to use basic transmission line simulation equations and an optional method for computing the parameters of transmission line equations.~~

The W-element is a versatile transmission line model that may be used to describe a variety of transmission line structures, from a simple lossless line to complex frequency-dependent lossy-coupled lines.

Syntax

W*xxx i1 i2 ... iN iR o1 o2 ... oN oR* **N=***val* **L=***val*
\+ ~~[~~**RLGCMODEL=***name* | **TABLEMODEL=***name*~~]~~ [**FGD=***val*]

The name of a W-element instance shall begin with the character ~~"~~"W~~"~~", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Table 17~~19~~: W-element (Coupled Transmission Line) Arguments

Argument	Description
N=*val*	Sets positive non-zero integer *val* as the number of signal conductors (excluding the reference conductor).
i1...iN	Node names for the near-end signal-conductor terminals
iR	Node name for the near-end reference-conductor terminal
o1... oN	Node names for the far-end signal-conductor terminals
oR	Node name for the far-end reference-conductor terminal
L=*val*	Defines *val* as the length of the transmission line, in meters
RLGCMODEL=*name*	Defines string *name* as the name of the RLGC model

Formatted: Font: Bold

Formatted: Font: Italic

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

TABLEMODEL=*name*	Defines string *name* as the name of the frequency-dependent tabular model
FGD=*val*	Specifies the cut-off frequency of dielectric loss, in hertz, as real number *val*.

The W-element supports two formats to specify transmission line properties:

- Format 1: RLGC specification, specified in a .MODEL statement
  - ~~Internally specified in a .MODEL statement~~
  - ~~Externally specified in a different file~~
- Format 2: Frequency-dependent tabular specification, specified in a .MODEL statement

The **RLGCMODEL** and **TABLEMODEL** arguments associate a W-element definition with a .MODEL using either Format 1 or Format 2, respectively. A W-element shall use either **RLGCMODEL** or **TABLEMODEL**, but not both.

Parameters in the W-element element declaration may be declared in any order. Specify the number of signal conductors, N, after the list of nodes. ~~The nodes and parameters in the W-element element declaration may be interspersed.~~

Format 1: RLGC Model

The inputs of the W-element are given in per unit length matrices: R_o (DC resistance), L, G, C, R_s (skin effect), and G_d (dielectric loss)

The W-element does not limit ~~any of the following parameters:~~the number of

- ~~Number of~~ coupled conductors~~.~~.
- ~~Shape of the matrices.~~
- ~~Line loss.~~
- ~~Length or amount of frequency dependence.~~

The RLGC text file contains frequency-dependent RLGC matrices per unit length. The W-element also handles frequency-independent RLGC, and lossless (LC) lines. It does not support RC lines.

Because RLGC matrices are symmetrical, the RLGC model specifies only the lower triangular parts of the matrices. The syntax of the RLGC model for the W-element is:

```
.MODEL name W MODELTYPE=RLGC N=val
+ Lo=matrix_entries
```

Formatted: Font: Not Bold, Italic
Formatted: Font: Italic
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Body, No bullets or numbering

+ **Co=***matrix_entries* [**Ro=***matrix_entries*] [**Go=***matrix_entries*]
+ [**Rs=***matrix_entries*] [**~~wp=val~~ Gd=***matrix_entries*] [**Rogndval**]
+ [**Rsgnd=***val*] [**Lgnd=***val*]

Table 18~~20~~: W-element RLGC Model Arguments

Argument	Description	Units
N=*val*	Number of conductors (same as in the element card).	
Lo=*matrix_entries*	DC inductance matrix, per unit length $\left[\frac{H}{m}\right]$.	H/m
Co=*matrix_entries*	DC capacitance matrix, per unit length $\left[\frac{F}{m}\right]$.	F/m
Ro=*matrix_entries*	DC resistance matrix, per unit length $\left[\frac{\Omega}{m}\right]$.	Ω/m
Go=*matrix_entries*	DC shunt conductance matrix, per unit length $\left[\frac{S}{m}\right]$.	S/m
Rs=*matrix_entries*	Skin effect resistance matrix, per unit length $\left[\frac{\Omega}{m\sqrt{Hz}}\right]$.	Ω/(m•√Hz)
Gd=*matrix_entries*	Dielectric loss conductance matrix, per unit length $\left[\frac{S}{m \cdot Hz}\right]$.	S/m•Hz
~~Wp~~	~~Angular frequency of the polarization constant [radian/sec] (see Introduction to the Complex Dielectric Loss Model). When the wp value is specified, the unit of Gd~~	

Formatted: Font: 11 pt

Formatted Table

	~~becomes [S/m].~~	
Lgnd=*val*	Defines *val* as the DC inductance value, per unit length for ground~~s~~ $\left[\frac{H}{m}\right]$ (reference line).	H/m
Rognd=*val*	Defines *val* as the DC resistance value, per unit length for ground $\left[\frac{\Omega}{m}\right]$ (reference line).	Ω/m
Rsgnd=*val*	Defines *val* as the skin effect resistance value, per unit length for ground (reference line) $\left[\frac{\Omega}{m\sqrt{Hz}}\right]$.	Ω/(m•√Hz)

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

The following ~~input netlist file~~example shows RLGC input for the W-element:

```
* W-Element example, four-conductor line
W1 N=3 1 3 5 0 2 4 6 0 RLGCMODEL=example_rlc l=0.97


* RLGC matrices for a four-conductor lossy
.MODEL example_rlc W MODELTYPE=RLGC N=3
+ Lo=
+ 2.311e-6
+ 4.14e-7 2.988e-6
+ 8.42e-8 5.27e-7 2.813e-6
+ Co=
+ 2.392e-11
+ -5.41e-12 2.123e-11
+ -1.08e-12 -5.72e-12 2.447e-11
+ Ro=
+ 42.5
+ 0 41.0 + 0 0 33.5
+ Go=
+ 0.000609
+ -0.0001419 0.000599
+ -0.00002323 -0.00009 0.000502
+ Rs=
+ 0.00135
```

```
+ 0 0.001303
+ 0 0 0.001064
+ Gd=
+ 5.242e-13
+ -1.221e-13 5.164e-13
+ -1.999e-14 -7.747e-14 4.321e-13
```

~~Using RLGC Matrices~~

RLGC matrices in the RLGC model of the W-element are in the Maxwellian format.

Format 2: Frequency-Dependent Tabular Specification

The tabular RLGC model may be used as an extension of the analytical RLGC model to model any arbitrary frequency-dependent behavior of transmission lines (this model does not support RC lines).

The W-element syntax supports tables of data (use a `.MODEL` statement of type w). To accomplish this, the `.MODEL` statement refers to .MODEL statements where the ~~"~~type~~"~~ is SP (described in Small-Signal Parameter Data Frequency Table Model (SP Model)~~Small-Signal Parameter Data Frequency Table Model (SP Model)~~), which contain the actual table data for the RLGC matrices.

~~To ensure accuracy, t~~The W-element tabular model requires the following:

- R and G tables shall include zero-frequency data points.
- L and C tables shall include infinite-frequency data points as well as zero-frequency data points.

To specify a zero-frequency point, either the DC argument shall be used or, alternatively, the f parameter in the DATA field of the SP model may be set to a value of 0. To specify an infinity frequency point, use the INFINITY argument.

See also, Small-Signal Parameter Data Frequency Table Model (SP Model)~~Small-Signal Parameter Data Frequency Table Model (SP Model)~~.

11.9 Frequency-Dependent Matrices

The static (constant) L and C matrices are accurate for a wide range of frequencies. In contrast, the static (DC) R matrix applies to only a limited frequency range, mainly due to the skin effect. A good approximate expression of the R resistance matrix with the skin effect, is:

Equation 1 $$\boldsymbol{R(f) \cong R_o + \sqrt{f}(1+j)R_s}$$

where:

- R_o is the DC resistance matrix.

- R_s is the skin effect matrix.

The imaginary term depicts the correct frequency response at high frequency; however, it might cause significant errors for low-frequency applications. In the W-element, this imaginary term may be excluded:

```
Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val INCLUDERSIMAG=NO
```

Formatted: Indent: Left: 0"

In contrast, the G (loss) conductance matrix is often approximated as:

Equation 2

$$G(f) \cong G_o + \frac{f}{\sqrt{1+\left(\frac{f}{f_{gd}}\right)^2}} G_d$$

Where,

- G_o models the shunt current due to free electrons in imperfect dielectrics.
- G_d models the power loss due to the rotation of dipoles under the alternating field.
- f_{gd} is a cut-off frequency.

If f_{gd} is not set, or if f_{gd} is set to 0, then G(f) keeps a linear dependency on the frequency. In the W-element, the default f_{gd} is zero (that is, G(f) does not use the f_{gd} value).

An alternate value may be included in the W-element statement:

```
Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val fgd=val
```

To use the previous linear dependency, set f_{gd} to 0.

Formatted: Font: (Default) Helvetica

Note: fgd shall be used used to estimate frequency dependent shunt loss conductance for the RLGC model when INCLUDEGDIMAG=yes has not been set.

When INCLUDEGDIMAG=yes, the RLGC model shall estimate frequency dependent shunt (C and G) parameters described as Equation 2 and the fgd value shall be ignored.

Formatted: Font: (Default) Helvetica

Field Code Changed

Both of these are ways to fit the RLGC model fit with actual measurements.
If measured or computationally extracted data is used for a tabular RLGC model, it should be more accurate if parameter extraction is accurately done.

Small-Signal Parameter Data Frequency Table Model (SP Model)

The small-signal parameter data frequency table model (SP model) is a generic model that describes frequency-varying behavior.

Syntax

```
.MODEL name sp [N=val [FSTART=val] [FSTOP=val] [NI=val]
+ [SPACING=spacing] MATRIX=matrix VALTYPE=valtype REAL
+ [INFINITY=matrixval]
+ [INTERPOLATION=val] [EXTRAPOLATION=val]
+
+ [DATA=(npts ...) | DATAFILE=filename]
```

Table 19: W-element SP Model Arguments

Argument	Description
N=*val*	Sets positive, non-zero integer *val* as the matrix dimension (number of signal terminals). Default is 1.
FSTART=*val*	Sets *val* as the starting frequency point for data. Default=0.
FSTOP=*val*	Sets *val* as the final frequency point for data. Use this parameter only for the **LINEAR** and **LOG** spacing formats.
NI=*val*	Number of frequency points per interval. Use this parameter only for the **DEC** and **OCT** spacing formats. Default=10.
SPACING=*spacing*	Sets *spacing* as one of the following data sample spacing formats: • **LIN** (LINEAR): uniform spacing with frequency step of (**FSTOP**-**FSTART**)/(*npts*-1). The dDefault. The entry **LINEAR** is also a legal alternative. • **OCT**: octave variation with **FSTART** as the starting frequency and NI points per octave. *npts* sets the final frequency. • **DEC**: decade variation with **FSTART** as the starting frequency and NI points per decade. *npts* sets the final frequency.

Formatted: Font: Not Italic
Formatted: Font: Not Italic
Formatted: Font: Bold, Not Italic
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Italic
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Italic
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Italic

	• **LOG**: logarithmic spacing. **FSTART** and **FSTOP** are the starting and final frequencies. • **POI**: non-uniform spacing. Pairs data ~~(NONUNIFORM)~~ points with frequency points. **NONUNIFORM** is an acceptable alternative entry.
~~MATRIX=~~*~~matrix~~*	~~Sets *matrix* as one of the following matrix (data point) formats:~~ ~~• SYMMETRIC: symmetric matrix. Specifies only lower half triangle of a matrix (default).~~ ~~• HERMITIAN: similar to SYMMETRIC; off-diagonal terms are complex conjugates of each other.~~ ~~• NONSYMMETRIC: non-symmetric (full) matrix.~~
VALTYPE=~~valtype~~REAL	Sets ~~*valtype* as one of following data types for matrix elements:~~ ~~•~~ REAL: ~~real entry.~~data formatting to real values only (required for compatibility with existing SPICE implementations). ~~• CARTESIAN: complex number in real/imaginary format (default).~~ ~~• POLAR: complex number in polar format. Specify angles in radians.~~
INFINITY=~~*matrix*~~*val*	Sets ~~*maxtrix*~~*val* as the data point at infinity. Typically real-valued. This data format shall be consistent with the **MATRIX** and **VALTYPE** specifications. ~~*Npts*~~ *npts* does not count this point.
INTERPOLATION=*val*	Sets *val* as the interpolation scheme: • **STEP**: piecewise step (default). • **LINEAR**: piecewise linear. • **SPLINE**: b-spline curve fit. •

Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold, Not Italic
Formatted: Normal, No bullets or numbering
Formatted: List Paragraph
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Italic
Formatted: Font: Italic
Formatted Table
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: (Default) Helvetica, 11 pt, Font color: Auto
Formatted: Don't adjust right indent when grid is defined, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5", Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

EXTRAPOLATION=*val*	Sets *val* as one of the following the extrapolation schemes to be used for the model data during simulation: • **NONE**: no extrapolation is allowed. Simulation terminates if a required data point is outside of the specified range. • **STEP**: uses the last boundary point. The default. • **LINEAR**: linear extrapolation by using the last two boundary points. If the data point at infinity is specified, then extrapolation is not used.
~~*npts*~~	~~Number of data points.~~

DATA= ~~*filename*~~*npts …*	Data ~~points~~, where *npts* is the number of data points defined. The syntax to be used depends on spacing:
	• Syntax for **LIN** spacing:
	`.MODEL name sp SPACING=LIN [N=dim]`
	`+ FSTART=f0`
	`+ DF=f1 DATA=npts d1 d2 ...`
	• Syntax for **OCT** or **DEC** spacing:
	`.MODEL name sp SPACING=DEC or OCT`
	`[N=dim]`
	`+`
	`+ FSTART=f0 NI=n_per_intval`
	`+ DATA=npts d1 d2 ...`
	`.MODEL name sp SPACING=OCT [N=dim]`
	`+ FSTART=f0 NI=n_per_intval`
	`+ DATA=npts d1 d2 ...`
	• Syntax for **POI**/**NONUNIFORM** spacing:
	`.MODEL name sp SPACING=NONUNIFORM`
	`+ [N=dim]`
	`+ DATA=npts f1 d1 f2 d2 ...`

DATAFILE=*filename*	Sets filename as the name of an external file containing the data points.

An SP model definition shall use either **DATA** or **DATAFILE**, but not both.

If an external file is used for the SP model data points, this file ~~must~~shall contain only raw numbers without any suffixes, comments or continuation characters. The first number in the file ~~must~~shall be an integer value to indicate the number of sampling points in the file (*npts*). Then, sampling data ~~must~~shall follow. The order of sampling data ~~must~~shall be the same as in the DATA statement. This data file has no limitation on line length.

Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Italic
Formatted: Font: Bold
Formatted ... [32]
Formatted ... [33]
Formatted ... [34]
Formatted ... [35]
Formatted ... [36]
Formatted ... [37]
Formatted ... [38]
Formatted ... [39]
Formatted ... [40]
Formatted ... [41]
Formatted ... [42]
Formatted ... [43]
Formatted ... [44]
Formatted: Font: 11 pt

Note that, while STEP is the default setting for INTERPOLATION, due to differences between implementations, the LINEAR setting is strongly recommended.

W-element Model Definition Syntax

```
.MODEL name W MODELTYPE=TABLE [FITGC=0|1] N=val
+ LMODEL=l_freq_model CMODEL=c_freq_model
+ [RMODEL=r_freq_model] [GMODEL=g_freq_model]
```

Table 2022: W-element Model Definition Arguments

Argument	Description
FITGC=0|1	A value of 1 instructs the tool to run a causality check on the data. A value of 0 turns any causality checking off (default)
N=*val*	Sets *val* as the integer, positive, non-zero integer number of signal conductors (excluding the reference conductor).
LMODEL=*l_freq_model*	Sets *l_freq_model* as the SP model name for the inductance matrix array.
CMODEL= *c_freq_model*	Sets *c_freq_model* as the SP model name for the capacitance matrix array.
RLMODEL=*r_freq_model*	Sets *r_freq_model* as the SP model name for the resistance matrix array. By default, it is zero.
GMODEL=*g_freq_model*	Sets *g_freq_model* as the SP model name for the conductance matrix array. By default, it is zero.

11.10 S-element

An S-element is a frequency-domain set of network data, described using scattering parameters.

Syntax

S*xxx nd1 nd2 ... ndN* [*ndRef*]
+ **MNAME=***Smodel_name*
+ [**FBASE =** *base_frequency*] [**FMAX =** *maximum_frequency*]

The name of an S-element instance shall begin with the character "S", followed by up to 1023 alphanumeric characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Table 2123: S-element Arguments

Argument	Description
nd1 nd2...ndN	Nodes of an S-element. Three kinds of definitions are permitted: ▪ With no reference node ndRef, the default reference node is GND. Each node ndi (i=1~N) and GND construct one of the N ports of the S-element. ▪ With one reference node, ndRef is defined. Each node ndi (i=1~N) and the ndRef construct one of the N ports of the S-element. ▪ With an N reference nodes, each port has its own reference node. The node definition may be written more clearly: *nd1+ nd1- nd2+ nd2- ... ndN+ ndN-* Each pair of the nodes (*ndi+* and *ndi-*, *i*=1~N) constructs one of the N ports of the S-element.
ndRef	Reference node
MNAME=*Smodel_name*	Sets *Smodel_name* as the name of the S model. Note that string parameters are supported in calling an **MNAME**.

Formatted: Normal, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Formatted: Font: Bold

FBASE=*base_frequency*	Sets *base_frequency* as the base frequency to use for transient analysis. This value becomes the base frequency point for Inverse Fast Fourier Transformation (IFFT). ■ If this value is not set, the base frequency is a reciprocal value of the transient period. ■ If a frequency is set that is smaller than the reciprocal value of the transient, then transient analysis performs circular convolution, and uses the reciprocal value of FBASE as its base period.
FMAX=*maximum_frequency*	Sets *maximum_frequency* as the maximum frequency used in transient analysis. Used as the maximum frequency point for Inverse Fast Fourier Transformation (IFFT).

The nodes of the S-element shall be placed immediately after the identifier string. All optional parameters in both the S-element and S-model statements may be defined by the user, except for MNAME argument.

The optional arguments may be entered in any order, and the parameters specified in the element statement have a higher priority.

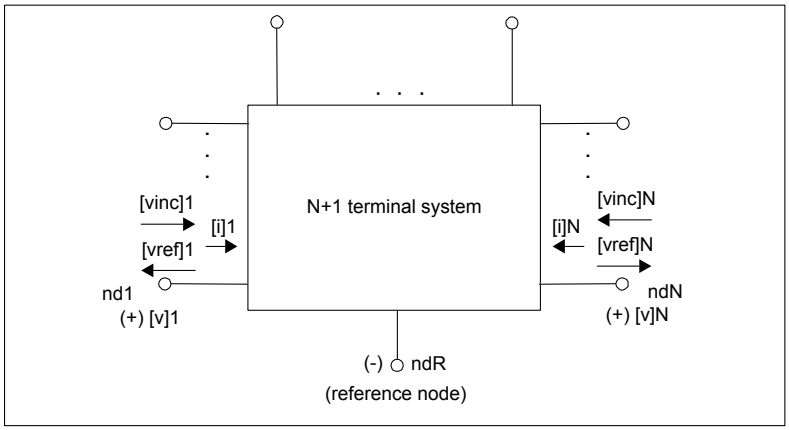

Figure 1: Terminal Node Notation

The following example illustrates the *nd1 nd2... ndN* no reference, single reference, and multi-reference parameters.

```
**S-parameter example
```

```
* no reference
S_no_ref n1 n2 mname=s_model

* single reference
S_one_ref n1 n3 gnd mname=s_model

*multi-reference
S_multi_ref n1 gnd n4 gnd mname=s_model
```

~~The S-element must have a call to one of the supported S-parameter file formats (~~IBIS-ISS ~~gets the number of ports from the S-parameter file~~will match the S-element instance node definitions to the number of ports 'n' specified in the S-element .MODEL definition, as follows~~ The number of ports, 'n', may be specified explicitly as N=n~~.

- For n ~~terminals~~nodes, the S-element assumes no reference node.
- For n+1 ~~terminals~~nodes, the S-element assumes one reference node.
- For 2n ~~terminals~~nodes, the S-element assumes n signal nodes and n reference nodes. Each pair of nodes ~~is~~ comprises a signal and a reference node.

S-Element Model Syntax

Use the following syntax to describe specific S-element models:

.MODEL *Smodel_name* **S** ~~[~~**N=***dimension*~~]~~
~~+ [~~**TSTONEFILE=***filename*
~~+ [**FBASE=***base_frequency*~~] [**FMAX=***maximum_frequency*~~]~~

Formatted: Indent: Left: 0", First line: 0"

Table 2224: S-element Model Definition Arguments

Argument	Description
Smodel_name	Name of the S model.
S	Specifies that the model type is an S model.
N=~~*dimension*~~*val*	Sets positive, non-zero integer ~~*dimension*~~ *val* as the ~~S-model dimension (the number of terminals of the S-element to be defined), excluding the reference node.~~number of ports for the S-model. This value must match the number of ports defined in

	the associated Touchstone file.

TSTONEFILE=*filename*	Sets the string *filename* as the name of a Touchstone file. Data contains frequency-dependent array of matrixes. Touchstone files must follow the .s#p file extension rule, where # represents the dimension of the network.
	Note that string parameters are supported for TSTONEFILE
	Example:
	`.subckt sparam n1 n2 tsfile=str('ss_ts.s2p')`
	`S1 n1 n2 0 mname=s_model`
	`.model s_model S TSTONEFILE=str(tsfile)`
	`.ends`
	`x1 A B sparam tsfile=str('ss_ts.s2p')`
	…
	For details, see *Touchstone® File Format Specification* by the IBIS Open Forum (http://www.eda.org/ibis/).

FBASE=*base_frequency*	Sets *base_frequency* as the expected base frequency to be used for transient analysis involving the associated S-element model. IBIS-ISS uses this value as the base frequency point for Fast Inverse Fourier Transformation (IFFT). ▪ If FBASE is not set, the simulator shall use a reciprocal of the transient period as the base frequency. ▪ If FBASE is set smaller than the reciprocal value of transient period, transient analysis performs circular convolution by using the reciprocal value of FBASE as a base period.
FMAX=*maximum_frequency*	Sets *maximum_frequency* as the expected maximum frequency to be used for transient analysis involving the associated S-element model. Used as the maximum frequency point for Inverse Fast Fourier Transform (IFFT).

Formatted Table

The TSTONEFILE parameter describes the frequency-varying behavior of a network.

11.11 E-element (Voltage-Controlled Voltage Source)

This section explains the E-element syntax and parameters.

Syntax (Linear Form)

E*xxx n+ n-* [**VCVS**] *in+ in- gain*

Syntax (Laplace Transform)

~~Voltage Gain H(s):~~

E*xxx n+ n-* **LAPLACE** *in+ in- k0, k1, ..., kn / d0, d1, ..., dm*

Syntax (Pole-Zero Function)

E*xxx n+ n-* **POLE** *in+ in- a αz1, fz1, ..., αzn, fzn / b, + αp1, fp1, ..., αpm, fpm*

Syntax (Foster Pole-Residue Form)

E*xxx n+ n-* **FOSTER** *in+ in- k0 k1*
+ *(Re{A1}, Im{A1})/ (Re{p1}, Im{p1})*
+ *(Re{A2}, Im{A2})/ (Re{p2}, Im{p2})*
+ *(Re{A3}, Im{A3})/ (Re{p3}, Im{p3})*
+ *...*

E-element Arguments

The E-element arguments are described in the following list. Note that the element name shall begin with the character "E" and be followed by up to 1023 characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Formatted: Normal

Formatted: Font: (Default) Helvetica

Table 23: E-element Arguments

Argument	Description
gain	Voltage gain

in+/-	Positive or negative controlling nodes. Specify one pair for each dimension
n+/-	Positive or negative node of a controlled element
VCVS	Identifier argument for voltage-controlled voltage source. VCVS is a reserved word; do not use it as a node or parameter name

Laplace Transform Details

H(s) is a rational function, with parameters used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).

~~H(s) is a rational function, with parameters used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).~~

Pole-Zero Function Details

~~Voltage Gain H(s):~~

~~**E***xxx n+ n- **POLE** in+ in- a az1, fz1, ..., azn, fzn / b, + ap1, fp1, ..., apm, fpm*~~

The following equation defines H(s) in terms of poles and zeros:

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1}) \ldots (s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1}) \ldots (s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

Equation 3

The complex poles or zeros are in conjugate pairs. The element description specifies only one of them, and the program includes the conjugate. Parameters may be used to specify the a, b, α, and f values.

```
Elow_pass out 0 POLE in 0 1.0 / 1.0, 1.0,0.0 0.5,0.1379
```

The `Elow_pass` statement describes a low-pass filter, with the transfer function:

$$H(s) = \frac{1.0}{1.0 \cdot (s + 1)(s + 0.5 + j2\pi \cdot 0.1379)(s + 0.5 - (j2\pi \cdot 0.1379))}$$

Foster Pole-Residue ~~Form~~Details

~~Gain E(s) form~~

~~*Exxx n+ n- **FOSTER** in+ in- k0 k1*~~
~~*+ (Re{A1}, Im{A1})/ (Re{p1}, Im{p1})*~~
~~*+ (Re{A2}, Im{A2})/ (Re{p2}, Im{p2})*~~
~~*+ (Re{A3}, Im{A3})/ (Re{p3}, Im{p3})*~~
~~*+ ...*~~

In the ~~above~~ Foster pole-residue syntax, parentheses, commas, and slashes are separators—they have the same meaning as ~~a~~ whitespace. A pole-residue pair is represented by four numbers (real and imaginary part of the residue, then real and imaginary part of the pole).

For convergence, the *Re{p~i~}* ~~must~~shall be less than zero. ~~For an N-port admittance matrix Y, *Re{Y}* should be positive definite to ensure passivity of the model.~~

~~For real poles, half the residue value is entered because it is applied twice. In the above example, the first pole-residue pair is real, but is written as "A1/(s-p1)+A1/(s-p1)"; therefore, 0.0004 is entered rather than 0.0008.~~For example, to represent a E(s) equation in the form

$$E(s) = 0.001 + 1\times10^{-12}s + \frac{0.0008}{s+1\times10^{10}} + \frac{(0.001 - j0.006)}{s-(-1\times10^{8}+j1.8\times10^{10})} + \frac{(0.001 + j0.006)}{s-(-1\times10^{8}-j1.8\times10^{10})}$$

The IBIS-ISS syntax would be:

```
E1 1 0 FOSTER 2 0 0.001 1e-12
+ (0.0004, 0)/(-1e10, 0) (0.001, -0.006)/(-1e8, 1.8e10)
```

Note:

For real poles, half the residue value is entered because it is applied twice. In the above example, the first pole-residue pair is real, but is written as "A1/(s-p1)+A1/(s-p1)"; therefore, 0.0004 is entered rather than 0.0008.

Formatted: Font: Italic
Formatted: Font: Italic
Formatted: Font: Italic, Subscript
Formatted: Font: Italic
Formatted: Body
Formatted: Indent: Left: 0"

H(s) is a rational function, with parameters used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).

Formatted: Font: Not Italic, Character scale: 0%

Formatted: Figure

E-element Arguments

The E-element arguments are described in the following list. Note that the element name must begin with the character "E" and be followed by up to 1023 alphanumeric characters.

Table 25: E-element Arguments

Argument	Description
gain	Voltage gain
in+/-	Positive or negative controlling nodes. Specify one pair for each dimension
n+/-	Positive or negative node of a controlled element
VCVS	VCVS is a reserved word; do not use it as a node name

11.12 F-element (Current-Controlled Current Source)

This section explains the F-element syntax and parameters.

~~**Note:**~~

~~G-elements with algebraic expressions may be used to duplicate the functions of an F-element.~~

Syntax

F*xxx n+ n-* [**CCCS**] *vn1 gain*

F-element Arguments

The F-element parameters are described in the following list. Note that the F-element name ~~must~~shall begin with the character ~~"~~"F~~"~~", followed by up to 1023 ~~alphanumeric~~ characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Table 24~~26~~: F-element Arguments

Argument	Description
CCCS	~~Pre-defined~~Identifier argument for current-controlled current source. CCCS is an IBIS-ISS reserved word; do not use it as a node or parameter name.
Gain	Current gain.
n+/-	Connecting nodes for a positive or negative controlled source.
vn1	Name~~s~~ of the voltage source~~,~~ through which the controlling current flows.

Note that the controlling current flows from the positive terminal of source *vn1* to the negative terminal of source *vn1*, per the definition of vn1 elsewhere in the circuit.

Formatted: Body, No bullets or numbering

Formatted: Normal

Formatted: Font: (Default) Helvetica

Formatted: Font: (Default) Helvetica

Formatted: Font: (Default) Helvetica, Italic

Formatted: Font: (Default) Helvetica

Formatted: Font: (Default) Helvetica, Italic

Formatted: Font: (Default) Helvetica

11.13 G-element (Voltage-Controlled Current Source)

This section explains G-element syntax statements, and their parameters.

Syntax (Linear Form)

G*xxx n+ n-* [**VCCS**] *in+ in- transconductance*

Syntax (Laplace Transform)

G*xxx n+ n-* **LAPLACE** *in+ in- k0, k1, ..., kn / d0, d1, ..., dm*

~~H(s) is a rational function, in the following form:~~

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

~~Equation 4~~

~~Parameters may be used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).~~

Syntax (Pole-Zero Function)

G*xxx n+ n-* **POLE** *in+ in- a* ~~*a*~~*z1, fz1, ..., ~~a~~zn, fzn / b, + ~~a~~p1, fp1, ..., ~~a~~pm, fpm*

Syntax (Foster Pole-Residue)

G*xxx n+ n-* **FOSTER** *in+ in- k0 k1*
+ (Re{A1}, Im{A1})/ (Re{p1}, Im{p1})
+ (Re{A2}, Im{A2})/ (Re{p2}, Im{p2})
+ (Re{A3}, Im{A3})/ (Re{p3}, Im{p3})
+ ...

G-element Arguments

Formatted: Indent: Left: 0.19", First line: 0.5"

Formatted: Indent: Left: 0.5", First line: 0.5"

Formatted: Indent: Left: 1"

The G-element arguments are described in the following list. Note that the G-element name shall begin with the character "G", followed by up to 1023 characters (see Table 3: IBIS-ISS Special Characters for a list of valid special characters).

Table 25: G-element Arguments

Argument	Description
in+/-	Positive or negative controlling nodes. Specify one pair for each dimension.
n+/-	Positive or negative node of the controlled element. Current flows from the positive to the negative node.
transconductance	Voltage-to-current conversion factor.
VCCS	Identifier argument for the voltage-controlled current source. VCCS is a reserved IBIS-ISS word; do not use it as a node or parameter name.

~~**Linear**~~

~~**Gxxx n+ n- [VCCS] in+ in- transconductance**~~

~~**For a description of the G-element parameters, see Table VCCS Parameters.**~~

~~**Laplace Transform**~~

Laplace Transform Details

For the Laplace transform, the transconductance, H(s), is a rational function, in the following form:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

Equation 4

Parameters may be used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).

Formatted: Normal

Formatted: Font: (Default) Helvetica

Formatted: Font: Italic

Formatted: Heading 3, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Formatted: Heading 3

Formatted: Indent: Left: 0", First line: 0"

Formatted: Indent: Left: 0"

Transconductance H(s):

G*xxx n+ n- **LAPLACE** in+ in- k0, k1, ..., kn / d0, d1, ..., dm*

H(s) is a rational function, in the following form:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

Equation 4

Parameters may be used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).

Pole-Zero Function Details

Transconductance H(s):

G*xxx n+ n- **POLE** in+ in- a az1, fz1, ..., azn, fzn / b, + ap1, fp1, ..., apm, fpm*

For the pole-zero function, The the following equation defines H(s) in terms of poles and zeros:

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1}) \ldots (s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1}) \ldots (s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

Equation 5

The complex poles or zeros are in conjugate pairs. The element description specifies only one of them, and the simulation program will include the conjugate automatically. Parameters may be used to specify the a, b, α, and f values. An example is shown below.

```
Ghigh_pass 0 out POLE in 0 1.0 0.0,0.0 / 1.0 0.001,0.0
```

The `Ghigh_pass` statement describes a high-pass filter, with the transfer function:

$$H(s) = \frac{1.0 \cdot (s + 0.0 + j \cdot 0.0)}{1.0 \cdot (s + 0.001 + j \cdot 0.0)}$$

Foster Pole-Residue Form

~~Transconductance G(s) form~~

```
~~Gxxx n+ n- FOSTER in+ in- k0 k1~~
~~+ (Re{A1}, Im{A1})/ (Re{p1}, Im{p1})~~
~~+ (Re{A2}, Im{A2})/ (Re{p2}, Im{p2})~~
~~+ (Re{A3}, Im{A3})/ (Re{p3}, Im{p3})~~
~~+ ...~~
```

In the ~~above~~ Foster pole-residue syntax, parentheses, commas, and slashes are separators—they have the same meaning as a space. A pole-residue pair is represented by four numbers (real and imaginary part of the residue, then real and imaginary part of the pole).

For convergence, the *Re{p~i~}* shall be less than zero.

~~For an N-port admittance matrix Y, Re{Y} should be positive-definite to ensure passivity of the model.~~

~~For a description of the G-element parameters, see Table 27: G-element Arguments.~~

For example, to represent a G(s) in the form,

$$G(s) = 0.001 + 1 \times 10^{-12} s + \frac{0.0008}{s + 1 \times 10^{10}} + \frac{(0.001 - j0.006)}{s - (-1 \times 10^{8} + j1.8 \times 10^{10})} + \frac{(0.001 + j0.006)}{s - (-1 \times 10^{8} - j1.8 \times 10^{10})}$$

The IBIS-ISS syntax would be:

```
G1 1 0 FOSTER 2 0 0.001 1e-12
+(0.0004, 0)/(-1e10, 0) (0.001, -0.006)/(-1e8, 1.8e10)
```

Note:

For real poles, half the residue value is entered because it is applied twice. In the above example, the first pole-residue pair is real, but is written as "A1/(s-p1)+A1/(s-p1)"; therefore, 0.0004 is entered rather than 0.0008.

~~G-element Arguments~~

~~The G-element arguments are described in the following list. Note that the G-element name must begin with the character "G", followed by up to 1023 alphanumeric characters.~~

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic, Subscript

Formatted: Font: Italic

Table 27: G-element Arguments

Argument	Description
n+/-	Positive or negative controlling nodes. Specify one pair for each dimension.
n+/-	Positive or negative node of the controlled element.
transconductance	Voltage-to-current conversion factor.
VCCS	Pre-defined argument for the voltage-controlled current source. VCCS is a reserved IBIS-ISS word; do not use it as a node name.

11.14 H-element (Current-Controlled Voltage Source)

This section explains H-element syntax statements, and defines their parameters.

Note:

The E-element with algebraic expressions may be used to duplicate the function of the H-element.

Syntax

H*xxx n+ n-* [**CCVS**] *vn1 transresistance*

Table 2628: H-element Arguments

Argument	Description
CCVS	Pre-definedIdentifier argument for the current-controlled voltage source. CCVS is an IBIS-ISS reserved word; do not use it as a node or parameter name.
n+/-	Connecting nodes for positive or negative controlled source.

Formatted Table

transresistance	Current-to-voltage conversion factor.
vn1 ~~…~~	Name~~s~~ of voltage source~~s,~~ through which controlling current flows. ~~One name for each dimension shall be specified.~~

Formatted Table

Note that the controlling current flows from the positive terminal of source *vn1* to the negative terminal of source *vn1*, per the definition of vn1 elsewhere in the circuit.

12 Best Practices

This section details syntax recommendations for ensuring maximum compatibility with existing proprietary SPICE variants. While not requirements for IBIS-ISS, following these practices will help maintain the portability of IBIS-ISS files.

- Exponent ranges should be limited to between e-60 and e+60.
- The use of a single backslash character (\) as a line continuation character is not supported in IBIS-ISS.
- The use of an "X" character as the "Meg" scale factor is not supported in IBIS-ISS.
- For maximum compatibility, in-line comments may be preceded by the * (asterisk) character instead of the $ (dollar) character. However, while not illegal, this practice should be avoided.
- Instance names shall begin with the appropriate identifying character. The remaining characters of a instance name should be limited to upper- and lower-case alphabetic characters, numeric characters 0-9 and the characters:
    - ~!@#%&_<>?[]|:;
- Parameter names should begin with a-z or A-Z, and the remaining characters should be limited to upper- and lower-case alphabetic characters, numeric characters 0-9 and the characters:
    - ! # $ % [ ]
- While a parameter may be defined in more than one .param statement within a subcircuit, this practice is best avoided.
- Node names should either be all numeric (0-9), or follow the rules of instance names stated above.

13 Theoretical Background

13.1 Introduction to the Complex Dielectric Loss Model

When INCLUDEGDIMAG=yes and there is no wp input, the W-element algorithm shall regard the Gd matrix as the conventional model. The tool shall then automatically extract constants for the complex dielectric model.

In conventional use, the W-element RLGC model, frequency-dependent conductance is approximated as Equation 2.

Where, Equation 2 represents the increase of shunt conductance due to dielectric loss. These pure real non-constant functions of frequency violate causality (1). As system operating frequency becomes significantly high even for PCB systems which use high polymer dielectric materials like FR4, the appearance of the dielectric loss becomes significant and significant non-causality of Equation 2 appears.

13.2 Frequency-Dependent Matrices

The frequency-dependent loss of the shunt conductance in the dielectric is mainly due to dielectric polarization. This polarization loss leads to a complex permittivity $\varepsilon(\omega)$ for the dielectric material (2).

Equation 6 $$\varepsilon(\omega) = \varepsilon'(\omega) - j\varepsilon''(\omega)$$

The loss tangent of the dielectric material can be specified as the ratio of imaginary part of $\varepsilon(\omega)$ to the real part,

Equation 7 $$\tan\delta(\omega) = \frac{\varepsilon''(\omega)}{\varepsilon'(\omega)}$$

For a single dielectric dipolar moment, complex electric permittivity can be written as,

Equation 8 $$\varepsilon(\omega) = \varepsilon_{\infty} + \omega_p \frac{\varepsilon_{dc} - \varepsilon_{\infty}}{j\omega + \omega_p}$$

Where ε_{dc}, and $\varepsilon_{\infty}$are low and high frequency limits of dielectric permittivity which are real numbers. And ω_p is the angular frequency that corresponds to the polarization time constant of the dielectric material. From Equation 5, frequency-dependent complex shunt loss conductance can be expressed as (2)

Equation 9 $$G(\omega) = Go + Gd\frac{j\omega}{j\omega + \omega_p}$$

Where, the imaginary part of the conductance contributes reactively. In cases of multiple dielectric materials surrounding the system, the complex loss conductance can be extended as linear combinations of multiple dipole moments as,

Equation 10 $$G(\omega) = Go + \sum_k Gd_k \frac{j\omega}{j\omega + \omega_{pk}}$$

Since Equation 7 satisfies the Kramers-Kronig condition, we can ensure the passivity/causality of the system. Note that when this new model is activated, the definition of Gd changes from conventional [S/m*Hz] to [S/m].

~~14~~13 References

1. Luca Daniel and Joel Philips, "Model Order Reduction for Strictly Passive and Causal Distributed Systems", ~~Daniel, Luca and Philips, Joel. 2002.~~ Proceeding of DAC, 2002.
2. Colin Gordon, ~~Colin, Blazeck,~~ Thomas Blazeck, and Raj Mittra, ~~Raj.~~ "Time-Domain Simulation of Multiconductor Transmission Lines with Frequency-Dependent Losses," ~~Gordon, Colin, Blazeck, Thomas and Mittra, Raj. 1992.~~ *IEEE Transactions on Computing-Aided Design*, ~~V~~, vol~~s~~. II, No. 11, pp. 1372-1387, 1992.
3. Omar Wing, ~~Omar~~ and Yu, Qingjian. ~~1994~~ Yu, "Computational Models of Transmission Lines with Skin Effects and Dielectric Loss," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, ~~Wing, Omar and Yu, Qingjian. 1994.~~ Vols. 41, No. 2, pp. 107-119, 1994.
4. Bjorn Gustavsen and Adam Semlyen, "Rational Approximation of Frequency Domain Responses by Vector Fitting. ~~Gustavsen, Bjorn and Semlyen, Adam. 1999~~", *IEEE Transaction on Power Delivery*, Vols. 14, No. 3, pp. 1052-1061, 1999.
5. E. Hammerstad, E. and O. Jensen, ~~O.~~ "Accurate models for microstrip computer aided design," ~~Hammerstad, E. and Jensen, O. 1980.~~ *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 407-409, 1980.
6. C.A. Balanis, ~~C.A.~~ *Advanced Engineering Electromagnetics.* New York : Wiley, 1989.
7. Eric Bogatin, ~~Eric,~~ *Signal Integrity Simplified.* Upper Saddle River : Prentice Hall, 2003.
8. Ravi Kollipara, ~~Ravi~~ et al. "Practical Design Considerations for 10 to 25 Gbps Copper Backplane Serial Links.", ~~Kollipara, Ravi et al. Santa Clara : IEC, 2006.~~ DesignCon, 2006.
9. *HSPICE® User Guide: Simulation and Analysis E-2010.12*, Mountain View, CA : Synopsys, Inc., 2010.
10. *HSPICE® User Guide: Signal Integrity E-2010.12*, Mountain View, CA : Synopsys, Inc., 2010.

Formatted ... [45]
Formatted ... [46]
Formatted ... [47]
Formatted ... [48]
Formatted ... [49]
Formatted ... [50]
Formatted ... [51]
Formatted ... [52]
Formatted ... [53]
Formatted ... [54]
Formatted ... [55]
Formatted ... [56]
Formatted ... [57]
Formatted ... [58]
Formatted ... [59]
Formatted ... [60]
Formatted ... [61]
Formatted ... [62]
Formatted ... [63]
Formatted ... [64]
Formatted ... [65]
Formatted ... [66]
Formatted ... [67]
Formatted ... [68]
Formatted ... [69]
Formatted ... [70]
Formatted ... [71]
Formatted ... [72]
Formatted ... [73]
Formatted ... [74]
Formatted ... [75]
Formatted ... [76]
Formatted ... [77]
Formatted ... [78]
Formatted ... [79]
Formatted ... [80]
Formatted ... [81]
Formatted ... [82]
Formatted ... [83]
Formatted ... [84]
Formatted ... [85]
Formatted ... [86]
Formatted ... [87]
Formatted ... [88]
Formatted ... [89]
Formatted ... [90]
Formatted ... [91]
Formatted ... [92]

Page 4: [1] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [1] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [2] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [2] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [3] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [3] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [4] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [4] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [5] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [5] Formatted Michael Mirmak

Default Paragraph Font

Page 4: [6] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [6] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [7] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [7] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [8] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [8] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [9] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [9] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [10] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [10] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [11] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [11] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [12] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [12] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [13] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [13] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [14] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [14] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [15] Formatted Michael Mirmak

Default Paragraph Font

Page 4: [15] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [16] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [16] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [17] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [17] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [18] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [18] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [19] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [19] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [20] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [20] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [21] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [21] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [22] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [22] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [23] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [23] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [24] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [24] Formatted Michael Mirmak

Default Paragraph Font

Page 4: [25] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [25] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [26] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [26] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [27] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [27] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [28] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [28] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [29] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [29] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [30] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [30] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [31] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 4: [31] Formatted Michael Mirmak
4/26/2011 11:45:00 AM

Default Paragraph Font

Page 51: [32] Formatted Michael Mirmak
4/26/2011 1:52:00 PM

Font: (Default) Courier New

Page 51: [32] Formatted Michael Mirmak
4/26/2011 1:52:00 PM

Font: (Default) Courier New

Page 51: [32] Formatted Michael Mirmak
4/26/2011 1:52:00 PM

Font: (Default) Courier New

Page 51: [32] Formatted Michael Mirmak
4/26/2011 1:52:00 PM

Font: (Default) Courier New

Page 51: [32] Formatted Michael Mirmak

Font: (Default) Courier New

Page 51: [33] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New

Page 51: [33] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New

Page 51: [33] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New

Page 51: [34] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New, 11 pt

Page 51: [34] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New, 11 pt

Page 51: [34] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New, 11 pt

Page 51: [34] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New, 11 pt

Page 51: [34] Formatted Michael Mirmak
4/26/2011 1:57:00 PM

Font: (Default) Courier New, 11 pt

Page 51: [35] Formatted Michael Mirmak
4/26/2011 9:37:00 AM

Font: Bold

Page 51: [35] Formatted Michael Mirmak
4/26/2011 9:37:00 AM

Font: Bold

Page 51: [36] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [36] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [36] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [36] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [36] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [37] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [37] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [37] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [37] Formatted Michael Mirmak

Font: (Default) Courier New

Page 51: [37] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [38] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [38] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [38] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [38] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [38] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [39] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New, Italic

Page 51: [39] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New, Italic

Page 51: [39] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New, Italic

Page 51: [39] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New, Italic

Page 51: [40] Formatted Michael Mirmak
4/26/2011 1:56:00 PM

Font: (Default) Courier New, Italic

Page 51: [40] Formatted Michael Mirmak
4/26/2011 1:56:00 PM

Font: (Default) Courier New, Italic

Page 51: [41] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [41] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [41] Formatted Michael Mirmak
4/26/2011 1:55:00 PM

Font: (Default) Courier New

Page 51: [42] Formatted Michael Mirmak
4/26/2011 9:37:00 AM

Font: Bold

Page 51: [42] Formatted Michael Mirmak
4/26/2011 9:37:00 AM

Font: Bold

Page 51: [43] Formatted Michael Mirmak
4/26/2011 1:56:00 PM

Font: (Default) Courier New

Page 51: [43] Formatted Michael Mirmak

Font: (Default) Courier New

Page 51: [43] Formatted Michael Mirmak
4/26/2011 1:56:00 PM

Font: (Default) Courier New

Page 51: [44] Formatted Michael Mirmak
4/26/2011 1:56:00 PM

Font: (Default) Courier New, Italic

Page 51: [44] Formatted Michael Mirmak
4/26/2011 1:56:00 PM

Font: (Default) Courier New, Italic

Page 71: [45] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [46] Formatted Michael Mirmak
4/26/2011 6:21:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [47] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [48] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [49] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [50] Formatted Michael Mirmak
4/26/2011 6:18:00 PM

Font: Not Bold

Page 71: [51] Formatted Michael Mirmak
4/26/2011 6:18:00 PM

Font: Not Bold

Page 71: [52] Formatted Michael Mirmak
4/26/2011 6:18:00 PM

Font: Not Bold

Page 71: [53] Formatted Michael Mirmak
4/26/2011 6:17:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [54] Formatted Michael Mirmak
4/26/2011 6:17:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [55] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [56] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [57] Formatted Michael Mirmak
4/26/2011 6:17:00 PM

Font: (Default) Helvetica, Italic

Page 71: [58] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [59] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [60] Formatted Michael Mirmak

Font: (Default) Helvetica

Page 71: [61] Formatted Michael Mirmak
4/26/2011 6:22:00 PM

Font: Not Bold

Page 71: [62] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [63] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [64] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [65] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: Not Bold

Page 71: [66] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: Not Bold

Page 71: [67] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [68] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: (Default) Helvetica

Page 71: [69] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: (Default) Helvetica

Page 71: [70] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: (Default) Helvetica, Italic

Page 71: [71] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [72] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [73] Formatted Michael Mirmak
4/26/2011 6:25:00 PM

Font: Not Bold

Page 71: [74] Formatted Michael Mirmak
4/26/2011 6:25:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [75] Formatted Michael Mirmak
4/26/2011 6:25:00 PM

Font: Not Italic

Page 71: [76] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [77] Formatted Michael Mirmak
4/26/2011 6:24:00 PM

Font: (Default) Helvetica, Italic

Page 71: [78] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [79] Formatted Michael Mirmak

Font: (Default) Helvetica

Page 71: [80] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: (Default) Helvetica, Not Bold

Page 71: [81] Formatted Michael Mirmak
4/26/2011 6:23:00 PM

Font: (Default) Helvetica

Page 71: [82] Formatted Michael Mirmak
4/26/2011 6:24:00 PM

Font: (Default) Helvetica, Not Bold

Page 71: [83] Formatted Michael Mirmak
4/26/2011 6:24:00 PM

Font: (Default) Helvetica, Not Bold

Page 71: [84] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [85] Formatted Michael Mirmak
4/26/2011 6:26:00 PM

Font: Not Bold

Page 71: [86] Formatted Michael Mirmak
4/26/2011 6:26:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [87] Formatted Michael Mirmak
4/26/2011 6:26:00 PM

Font: (Default) Helvetica, Not Italic

Page 71: [88] Formatted Michael Mirmak
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [89] Formatted **Michael Mirmak**
4/26/2011 6:13:00 PM

Font: (Default) Helvetica

Page 71: [90] Formatted **Michael Mirmak**
4/26/2011 6:12:00 PM

Normal

Page 71: [91] Formatted **Michael Mirmak**
4/26/2011 6:14:00 PM

Font: 12 pt, Italic

Page 71: [92] Formatted **Michael Mirmak**
4/26/2011 6:14:00 PM

Font: (Default) Helvetica, Check spelling and grammar