==========

I B I S C O N N E C T O R S P E C I F I C A T I O N

Last Updated: 02-26-2002, AP

- Added S. Peters Pin Map update
- Updated Cn Sparameter: Remove CITI references
- Original [Path Description] marked with “OLD”.
- Arpad’s adjusted path description added and marked with a “NEW”
- Adjusted Arpad’s latest path description to include M or L functionality
- S. Peters updated derivation method.

ISSUES TO ADDRESS

~~~~~~~~~~~

* The [File Rev] keywords require a Text String. Wouldn't it be better to require specific types of field formats, so that this information could be processed automatically?

* Still need to address “What information needs to stay with the specific connector mode. . .” especially if the connector model is to be eventually included in a larger file.

* Dual purpose use of the [Source] keyword. It is used to record the manufacturer's name, and the place and method from whence the model was created. This mixes two pieces of information that are both useful. I would propose another keyword [Part Manufacturer] which can be used to contain the name of the manufacturer of the part. This will make it very easy to search for manufacturers in a database of connector models.

[Source] then becomes the source of the information for extraction, which might also be some person or organization other than the connector manufacturer. (For example, SiQual or Mentor.)

==========

Specification Revision level 0.988

==========

Table of Contents

==========

I B I S C O N N E C T O R S P E C I F I C A T I O N 1
Table of Contents 1
REVISION HISTORY 2
1.0 GENERAL INTRODUCTION 2
2.0 STATEMENT OF INTENT 3
3.0 GENERAL SYNTAX RULES AND GUIDELINES 3
4.0 KEYWORD TREE DIAGRAM 5
5.0 HEADER KEYWORDS 6
Keyword: [Begin Header] 6
Keyword: [IBIS Cn Model Ver] 6
Keyword: [File Name] 6
Keyword: [File Rev] 7
Keyword: [Date] 7
Keyword: [Source] 7
Keyword: [Notes] 7
Keyword: [Disclaimer] 8
Keyword: [Copyright] 8
~~~~~~~~~~~

Keyword: [Support] 8
Keyword: [Redistribution] 9
Keyword: [Redistribution Text] 9
Keyword: [End Header] 9
6.0 GLOBAL KEYWORDS 10
Keyword: [Comment Char] 10
7.0 CONNECTOR MODEL FAMILY KEYWORDS 10
Keywords: [Begin Cn Model Family] 12
Keyword: [Manufacturer] 12
Keywords: [Cn Family Description] 12
Keywords: [Cn Model List] 12
Keywords: [Begin Cn Model] 14
OLD Keyword: [Path Description] 16
NEW Keyword: [Path Description] 20
Keywords: [Cn Model Description] 27
Keyword: [Cn Number of Conductors] 28
Keywords: [Cn Swath Parameters] 28
Keywords: [Cn Swath Pin Numbers] 32
Keywords: [Cn Pin Map] 34
Keywords: [End Cn Model] 36
Keywords: [End Cn Model Family] 36
8.0 MATRIX KEYWORDS 36
Keyword: [Begin Cn Section] 36
Keyword: [Derivation Method] 37
Keyword: [Resistance Matrix], [Inductance Matrix], [Capacitance Matrix], [Conductance Matrix] 37
Matrix Descriptions 39
Resistance Matrix Description 40
Inductance Matrix Description 40
Loop-Inductance Matrix Description (SLMs ONLY): 40
Partial Self Inductance Matrix Description (MLMs ONLY): 42
Capacitance Matrix Description 43
Conductance Matrix Description 44
Loss Matrix Description 44
Keyword: [Bandwidth] 45
Keyword: [Row] 45
Keyword: [Cn Sparameter] 46
Keyword: [End Cn Section] 47
Matrix Examples 47
9.0 End Keyword 49
Keyword: [End] 49

==========

REVISION HISTORY

xx-xx-200x, v1.0 Specification

==========

==========

1.0 GENERAL INTRODUCTION

==========

This section gives a general overview of the remainder of this document.

Sections 2 to 5 contain general information about the IBIS versions and the general rules and guidelines.

The remaining sections describe the connector keywords and usage rules.

2.0 STATEMENT OF INTENT

In order to enable an industry standard method to electronically transport IBIS connector modeling data between connector vendors, simulation vendors, and end customers, this template is proposed. The intention of this specification is to specify a consistent format that can be parsed by software, allowing simulation vendors to derive models compatible with their own products.

One goal of this template is to represent the current state of IBIS data, while allowing a growth path to more complex models / methods (when deemed appropriate). This would be accomplished by a revision of the base template, and possibly the addition of new keywords or categories.

Another goal of this template is to ensure that it is simple enough for connector vendors and customers to use and modify, while ensuring that it is rigid enough for simulation vendors to write reliable parsers.

This template is meant to contain a complete description of the elements on an entire connector or family of connectors.

This specification is intended to provide a basis for future versions which are expected to be backward compatible with this initial version.

3.0 GENERAL SYNTAX RULES AND GUIDELINES

1) Only ASCII characters, as defined in ANSI Standard X3.4-1986, may be used in an IBIS-Connector file. The use of characters with codes greater than hexadecimal 07E is not allowed. Also, ASCII control characters (those numerically less than hexadecimal 20) are not allowed, except for TAB characters and line termination sequences.
NOTE: the use of TAB characters is strongly discouraged.

2) Except for keywords (words enclosed by a square bracket []), the content of an Ibis Connector Model (ICM) file is case sensitive.

3) Keywords must be enclosed in square brackets, [], and must start in column 1 of the line. No space is allowed immediately after the opening bracket '[' or immediately before the closing bracket ']'. If used, only one space (' ') or underscore ('_') character separates the parts of a multi-word keyword. Spaces and underscores are equivalent within square bracket. Some keywords may be followed by an argument. Keyword arguments are of three types: either a text string, a file name, or a text block. Text string and file name arguments start on the same line as the keyword and are terminated with a line termination sequence. A text block starts on the same line as the keyword, may extend over multiple lines, and is terminated by the occurrence of the next keyword.

4) Keywords and sub-parameters must begin with a letter, and shall consist only of alphanumeric characters and the underscore (_). Spaces are not allowed in sub-parameter names.

5) Each IBIS Connector data file is divided into sections. Each section is delimited by keywords of the form [Begin "section name"] and [End "section

name"]. Within each section only specific keywords are legal (note, however, a keyword may be legal in more than one section). Unless otherwise noted, a section's keywords can appear in any order.

6) To facilitate portability between operating systems, file names used in the IBIS Connector file must only have lower case characters. File names should have a basename followed by a period ('.'), followed by a file name extension of no more than three characters. There is no length restriction on the basename. The basename and extension must use characters from the following set (space, ' ', 0x20 is not included):

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 _ -
```

7) A line of the file may have at most 120 characters, followed by a line termination sequence. The line termination sequence must be one of the following two sequences: a linefeed character, or a carriage return followed by linefeed character.

8) Anything following the comment character is ignored and considered a comment on that line. The default "|" (pipe) character can be changed by the keyword [Comment Char] to any other character. The [Comment Char] keyword can be used throughout the file as desired.

9) Valid scaling factors are:

T = tera	k = kilo	n = nano
G = giga	m = milli	p = pico
M = mega	u = micro	f = femto

When no scaling factors are specified, the appropriate base units are assumed. (These are volts, amperes, ohms, farads, henries, and seconds.) The parser looks at only one alphabetic character after a numerical entry; therefore it is enough to use only the prefixes to scale the parameters. However, for clarity, it is allowed to use full abbreviations for the units, (e.g., pF, nH, mA, mOhm). In addition, scientific notation IS allowed (e.g., 1.2345e-12).

10) All temperatures are represented in degrees Celsius.

11) Multiple definitions of a specific symbol are illegal. Specifically, multiple definitions (either redundant or alternate) of an object type of the same object class are illegal. Refer to section 8 for more information on object classes and object types.

12) All lines that occur before the [Begin Header] keyword or after the [End] keyword shall be treated as comments and ignored by the parser. The intent of this rule is to facilitate the inclusion of HTML tags, revision control headers, etc. at the beginning of the file.

13) The following words are reserved words and must not be used for any other purposes in the document:

```
    PWRGND- reserved model name, used with power or ground pins,
    POWER - reserved model name, used with power supply pins,
    GND   - reserved model name, used with ground pins,
    RET   - reserved word indicating either power or ground used
            as a high-frequency return path.
    NC    - reserved model name, used with no-connect pins,
    NA    - used where data not available.
```

==========

4.0 KEYWORD TREE DIAGRAM

```
================================================================================

/-- Start of File
|      -----------------
|  /-- [Begin Header]
|  |    |-- [Comment Char]
|  |    |-- [IBIS Cn Model Ver]
|  |    |-- [File Name]
|  |    |-- [File Rev]
|  |    |-- [Date]
|  |    |-- [Source]
|  |    |-- [Notes]
|  |    |-- [Disclaimer]
|  |    |-- [Copyright]
|  |    |-- [Support]
|  |    |-- [Redistribution]
|  |    |-- [Redistribution Text]
|  \-- [End Header]
|
|  /-- [Begin Cn Model Family] (Section 7)
|  |  |-- [Manufacturer]
|  |  |-- [Cn Family Description]
|  |  |-- [Cn Model List]
|  |
|  |--/ [Begin Cn Model]
|  |  |         Cn_Model_Type
|  |  |
|  |  | -- [Path Description]
|  |  |         Model_PinMap
|  |  |         Cn_Fork, Cn_EndFork
|  |  |         Cn_Section
|  |  |
|  |  | -- [Cn Model Description]
|  |  | -- [Cn Number of Conductors]
|  |  | -- [Cn Swath Parameters]  (optional for Swath)
|  |  | -- [Cn Swath Pin Numbers] (optional for Swath)
|  |  |
|  |  |
|  |  \ -- [End Cn Model]
|  |
|  | -- [Cn Pin Map]  |  Pins to Physical
|  \ -- [End Cn Model Family]
|
|
| -- /-- [Begin Pkg Model Family](PROPOSED ONLY Chapter n (TEXT DOES NOT EXIST)
|    |  | --------  sub-params: Pkg_Model_Type
|    |  |
|    |  |-- [Pkg Family Description]
|    |  |-- [Manufacturer(s)]
|    |  |-- [Pkg Model List]
|    |
|    |--/ [Begin Pkg Model]
|    |  | -- [Pkg Model Description]
|    |  | -- [Pkg Number of Conductors]
|    |  | -- [Pkg Swath Parameters]  (optional for Swath)
|    |  | -- [Pkg Swath Pin Numbers] (optional for Swath)
|    |  |
|    |  \ -- [End Pkg Model]
|    |
|    | -- [Pkg Pin Map]  |  Pins to Physical (is this needed for a pkg?)
|    |
```

```
|    \-- [End Pkg Model Family]
|
|
|--/-- [Begin Cn Section] | (Section 8)  Define Matrices
|  |-- [Derivation Method]
|  |-- [Resistance Matrix]
|  |        /-- [Bandwidth]
|  |        \-- [Row]
|  |-- [Inductance Matrix]
|  |         /-- [Bandwidth]
|  |         \-- [Row]
|  |-- [Capacitance Matrix]
|  |         /-- [Bandwidth]
|  |         \-- [Row]
|  |
|  |-- [Conductance Matrix]
|  |         /-- [Bandwidth]
|  |         \-- [Row]
|  |
|  |-- [Cn Sparameter]
|  |
|  \-- [End Cn Section]
|
|
\-- [End]
```

==========

5.0 HEADER KEYWORDS

==========

```
     Keyword: [Begin Header]
    Required: Yes
    Argument: None
 Description: This keyword denotes the beginning of the file header section.
 Usage Rules: [Begin Header] must be the first keyword in any ICM file.
              This keyword may be proceeded by lines of general text and/or
              comments which should be treated as comments by the parser.
------------------------------------------------------------------------------
[Begin Header]
```

==========

```
     Keyword: [IBIS Cn Model Ver]
    Required: Yes
    Argument: Text String
 Description: This keyword allows electronic parsers to immediately determine
              that this file contains an IBIS connector model.
              The version number is used to inform the parser what keywords
              are valid for this model and to allow backward support as new
              keywords are added.
 Usage Rules: This keyword must appear directly after the [Begin Header]
               keyword.

------------------------------------------------------------------------------
[IBIS Cn Model Ver]  1.0
```

==========

```
    Keyword: [File Name]
   Required:  Yes
```

```
   Argument:  Text String (file name)
Description:  Specifies the name of the IBIS Connector Model (ICM) file.
Usage Rules:  The purpose of this keyword is to document a single filename
              the file uses even if the file is transferred from one
              computer to another.  The file name must conform to the rules
              given in section 4, "General Syntax Rules and Guidelines".
-------------------------------------------------------------------------
[File Name]     iconm_hdi_202.icm


=========================================================================
     Keyword: [File Rev]
    Required: Yes
    Argument: Text String
 Description: Tracks the revision level of a particular .icm file.
 Usage Rules: Revision level is set at the discretion of the engineer
              defining the file.  The following guidelines are suggested:

                 0.x     connector and/or file in development
                 1.x     pre-manufacturing connector and/or model
                 2.x     mature connector and or model, no more changes likely

-------------------------------------------------------------------------
[File Rev]     1.0                    Used for .icm file versions


=========================================================================
     Keyword: [Date]
    Required: No
    Argument: Text String (Date)
 Description: Date this file was last modified.
 Usage Rules: This keyword is provided to insure the last changed date for
              this file is not lost if the file is transmitted between
              computer systems. The format is fixed.  The pattern to be used
              is “Month Day, Year”.  Month is fully written out.  Day uses two
              digits.  Year uses four digits.  See example.

 -------------------------------------------------------------------------
[Date]          July 1, 2000         | The latest file revision date


=========================================================================
     Keyword: [Source]
    Required:  No
    Argument:  Text Block
 Description:  Records the originating source of model data.
 Usage Rules:  Use this keyword to record how the model information was
               obtained (physical measurement of device, simulations, data
               book, etc.).  This field contain the name of the company who
               created the model.

               NOTE: It is recommended that the argument to the [Source]
               keyword be limited to a maximum of 24 lines of text.
-------------------------------------------------------------------------
[Source] Put originator and the source of information here. For example:
         Results from field simulation

=========================================================================
     Keyword: [Notes]
    Required: No
    Argument: Text Block
 Description: Optional notes regarding the file.
 Usage Rules: The keyword provides a place for the model maker to record
```

```
              important notes about the file or model data that are not
              included elsewhere.  Such information may include notes on
              validation level, model limits, usage assumptions, etc.

              NOTE: It is recommended that the argument to the [Notes]
              keyword be limited to a maximum of 24 lines of text.

              The [Notes] keyword can only be used once.

-------------------------------------------------------------------------------
[Notes]       Use this section for any special notes related to the file.


===============================================================================
     Keyword: [Disclaimer]
    Required: No
    Argument: Text Block
 Description: Legal disclaimer and copyright information
 Usage Rules: This keyword provides a place for the user to add a legal
              disclaimer.

              NOTE: It is recommended that the argument to the [Disclaimer]
              keyword be limited to a maximum of 96 lines of text.

-------------------------------------------------------------------------------
[Disclaimer]  This information is for modeling purposes only, and is not
              guaranteed.


===============================================================================
     Keyword: [Copyright]
    Required: No
    Argument: Text Block
 Description: Legal copyright information
 Usage Rules: Because model writers may consider the information in
              these keywords essential to users, and sometimes legally
              required, design automation tools should make this information
              available. Any text following the [Copyright] keyword must be
              included in any derivative models verbatim.

              NOTE: It is recommended that the arguments to the [Copyright]
                    keywords be limited to a maximum of 96 lines of text.

-------------------------------------------------------------------------------
[Copyright]   Copyright 2000, XYZ Corp., All Rights Reserved


===============================================================================
     Keyword: [Support]
   Required:  No
   Argument:  Text Block (URL path name)
Description:  Specifies a web site that can be visited to get the latest
              version of the  file.
Usage Rules:  Following the [Support] keyword is the URL of a web site the
              user may visit for more information on the model or model(s).
              The entire link, including any file extension, is required. The
              [Support] keyword may appear only once between the [Begin
              Header]/[End Header] keyword pair.

              NOTE:  It is recommended that a separate IBIS model web site
              (not page) be maintained to prevent accidental changes of page
              name from breaking this link.
```

```
-------------------------------------------------------------------------------
[Support] http://www.VendorNameIbisModels.com     |example ibis web site root
```

```
===============================================================================
     Keyword: [Redistribution]
   Required:  Yes
   Argument:  Text String (Yes, No, Specific)
Description:  Indicates to EDA tool companies and model users who may use and
              redistributed this file.
Usage Rules:  Following the [Redistribution] keyword is one of three
              arguments: "Yes", "No" and "Specific".

              An argument value of "Yes" means that a EDA tool vendor or end
              user may freely distribute the model as long as no fee is
              charged.  A fee may be charged if authorized by the model
              creator.

              An argument value of "No" means that the model may not be
              redistributed or retransmitted in any form.

              An argument value of "Specific" means that specific license
              information is contained in the [Redistribution Text] field.

              Use of the "Specific" or “No” argument prevents automated
              redistribution.

-------------------------------------------------------------------------------
[Redistribution]  Yes
```

```
===============================================================================
     Keyword: [Redistribution Text]
    Required: Yes, ONLY if [Redistribution] value is “Specific”, otherwise it
              is optional
    Argument: Text block
 Description: Allows model creator to supply specific redistribution
              information.
 Usage Rules: If the argument to the [Redistribution] keyword is 'Specific',
              then the user must include additional information on licensing
              details, or where to find them. Any text following the
              [Redistribution Text] keyword must be included in any derivative
              models verbatim.


              Note: It is recommended that the argument to the [Redistribution
                    Text] keyword be limited to a maximum of 24 lines of text.


-------------------------------------------------------------------------------
[Redistribution Text]  Your text goes here.
```

```
===============================================================================
     Keyword: [End Header]
    Required:  Yes
    Argument:  None
 Description:  Marks the end of an IBIS Connector header section.
 Usage Rules:  [End Header] must be the last keyword in any IBIS Connector
               header section.

-------------------------------------------------------------------------------
[End Header]
```

==========

6.0 GLOBAL KEYWORDS

==========

```
     Keyword: [Comment Char]
    Required: No
    Argument: Text String
 Description: Defines a new comment character to replace the default
              "|" (pipe) character, if desired.
 Usage Rules: The new comment character to be defined must be followed by
              the underscore character and the letters "char".  For example:
              "|_char" redundantly redefines the comment character to be
              the pipe character.  The new comment character is in effect
              only following the [Comment Char] keyword.  The following
              characters MAY be used:

                 ! " # $ % & ' ( ) * , : ; < > ? @ \ ^ ` { | } ~

              If used, the [Comment Char] keyword can only appear after the
              Begin Header] and [IBIS-X Ver] keywords. A file may contain
              multiple [Comment Char] keywords
-------------------------------------------------------------------------------
[Comment Char]  |_char
```

==========

7.0 CONNECTOR MODEL FAMILY KEYWORDS

==========

The Connector Model Family keywords in this section document several ways that connector models will be formatted. Some keywords work with or require only certain other keywords. The hierarchtical scope and intent of certain groups of keywords are as follows:

[Begin Cn Model Family] Contains all electrical information in the file.

[Manufacturer], [Cn Family Description], - Informational documentation

[Cn Model List] - Describes each connector that is in the file:

- Model_name - This names the model described by [Begin Cn Model] below
  A unique name is required for each electrical variation, even if it is for the same physical connector.

- Mated (for through path model) or Unmated_* (disconnected electrical termination)

- Min_slew_time - Documents the electrical simulation limit of the model. EDA tools are expected to issue a Warning if this value is exceeded during simulation. The same connector may have models with different Min_slew_time values for user selection.

- Image - Supplemental and optional physical support documentation supplied by the connector manufacturer

[Begin Cn Model] Each Model_name requires a separate block.

```
      Description sub-parameters:

        Cn_Model_Type  Specifies certain model types: MLM, SLM_General,
                       SLM_Quiescent, SLM_Even, SLM_Odd
        Cn_SGR         Other electrical descriptive information that
        Ref_Impedance  may be used for checking.

      [Path Description]  Specific electrical path
        Model_PinMap    Pin Map naming information, if applicable
        Cn_Section      Multiple occurrences of Cn_Section provides names
        Cn_Section      for electrical descriptions.  Each unique name
                        requires a separate [Begin Cn Section] keyword
                         block below.
        ....
      [Cn Pin Map]      Describes a pin map, if applicable referred to
                        by Model_PinMap.  Each Model_PinMap name requires
                        [Cn Pin Map] keyword
      ....
    [End Cn Model]

    [Begin Cn Section]  Each Cn_Section named under [Path Description]
     ...                is documented in this block, as discussed in
    [End Cn Section]    Matrix Keywords section.

  [End Cn Model Family] End of technical information
```

Two distinct modes are available to provide connector information under each [Begin Cn Model] keyword. They are defined and briefly described below followed by the related keywords.

A. Full Pinout, Full Matrix:
All connector Pins are listed, and the Section matrices document ALL pins

A restriction exists that SLM_Quiescent, SLM_Even, and SLM_Odd require fixed, predetermined return paths and are only meaningful for this mode.

This is the only mode supporting non-rectangular pin arrangements such as those with a different number of pins in each row.

B. Full Pinout, Partial Matrix:
All connector Pins are listed, but the Section matrices document just a "Swath" or rectangular portion of the pins.

This is a common format when connectors can contain hundreds of pins, but the accurate electrical information for Sections is often limited to a smaller rectangular block of pins within the connector.

The list of keywords in this section that are required, optional or prohibited for each mode are listed below for an overall summary.

A. Full Pinout, Full Matrix
(to be filled in)

B. Full Pinout, Partial Matrix:
(to be filled in)

Each keyword described in this section also documents to which mode or mode it applies.

Keyword descriptions follow:

```
================================================================================


================================================================================
    Keywords: [Begin Cn Model Family]
    Required: Yes
    Argument: Text String
 Description: This keyword marks the beginning of the connector family
              description section.  A connector “family” consists of one or
              more similar connectors.  Note that each individual connector
              in a family will have a separate connector model.
 Usage Rules: Following the keyword is the name of the connector family
              Only one [Begin Cn Model Family]/[End Cn Model Family]
              keyword is allowed per file.

              Connector family descriptions are terminated by the occurrence
              of the [End Cn Model Family] keyword.

--------------------------------------------------------------------------------
[Begin Cn Model Family] High_Speed_Connector

================================================================================
     Keyword: [Manufacturer]
    Required: Yes
    Argument: Text String
 Description: This keyword gives the name of the manufacturer of the
              connector family being described in this file.
 Usage Rules: Following the keyword is the manufactures name.  Blank
              characters are allowed.

                    NOTE: It is highly recommended that each model maker
                          uses the connector manufacture name in
                          all .icm files.
--------------------------------------------------------------------------------
[Manufacturer]  XYZ Incorporated


================================================================================
    Keywords: [Cn Family Description]
    Required: Yes
    Argument: Text Block
 Description: Provides a human-readable description of this connector family.
 Usage Rules: Only a single occurrence of [Cn Family Description] is allowed
              in the file, and this keyword must be placed between the
              [Begin Cn Model Family] and [End Cn Model Family] keywords.

              Note: It is recommended that the family description be limited
              to a maximum of 4 lines.

--------------------------------------------------------------------------------
[Cn Family Description]
High Density 0.1 center square pin connector designed for use on IEEE 99999
buses.


================================================================================
    Keywords: [Cn Model List]
    Required: Yes
    Argument: None
 Description: This keyword lists the model name, mating conditions, minimum
              slew time and picture link for each individual connector model
              described in this file.
 Usage Rules: Following the keyword are four columns of data, as shown in
```

the example below (the column headings are shown as comments and are not required). The ‘Name’, ‘Mating’ and ‘Min_Slew_Time’ columns are required, and all columns must be in the left to right order shown.

The ‘name’ column lists the model name for each connector model contained in the file. The model name must match one of the model names given as an argument to the [Begin Cn Model] keyword(s).

The ‘mating’ column specifies the mating condition under which each model’s data is valid. Valid mating choices are “Mated”, “Unmated_Side_A” and “Unmated_Side_B”. These choices are defined as follows:

Mated: Both halves of a two piece connector are mated together for the model.
UnMated_Side_A: A model of non-mated connector. Typically, this would be the connector half that is attached to a baseboard
UnMated_Side_B: A model of non-mated connector. Typically, this would be the connector half that is attached to a daughter card, cable, or edgecard.

Note that using two unmated models in series DOES NOT provide a correct model of a mated connector.

The Min_Slew_Time column specifies the minimum slew time (i.e. fastest edge rate) for which each model is valid. Slew time is defined as the time it takes a signal to transition from 20% to 80% of its final value. This term does not indicate the bandwidth of the actual connector. Rather, this term is used to define a usage limit of each model.

Note: It is strongly recommended that an EDA tool issue a warning if a signal with a lesser slew time (i.e. faster edge rate) is applied to the model during a simulation.

The optional image column specifies an associated file that contains an image of each connector. The image file shall be either a .jpg or .txt (ASCII) formatted picture. The connector model MUST be considered valid by EDA tools even if an image file is specified but is missing.

If present, images should be maintained in the same directory as the connector model file or stored in the same compressed file with the connector to insure they files remained paired.

The same image file may be repeated for all model types if desired.

Note: it is suggested that EDA tools provide a viewer for the .jpg format and that model builders provide at least one picture for a connector.

Suggested .jpg image attributes:
- size = 160 horizontal by 120 vertical (pixels)
- color = 24 bit
- Image quality = 85%

Larger or smaller images may be used but are not recommended. A standard image size will allow EDA tools to

provide uniform look and feel for images from all vendors. The size was chosen to be a small disk file size and be useful as either a thumb-nail in a model selection window or scaled up to give a larger picture. The 160x120 was chosen as a perfect 4:1 scale in both axis of a 640x480 picture.

```
|-------------------------------------------------------------------------
|
[Begin Cn Model List]
|        Name             Mating      Min_Slew_Time   Image
|-------------------------------------------------------------------------
HDI_TEST_202              Mated          100ps       HDI_TEST_202_Mated.jpg
HDI_TEST_202_UnMatedA     UnMated_Side_A 100ps       HDI_TEST_202_UnMatedA.jpg
HDI_TEST_202_UnMatedB     UnMated_Side_B 100ps
HDI_TEST_202_ThruHole_to_Cable Mated     25ps        HDI_TEST_202_Mated.jpg
HDI_TEST_202_SMT_to_Cable    Mated       25ps        HDI_TEST_202_Mated.jpg
HDI_TEST_202_SMT_to_ThruHole Mated       25ps        HDI_TEST_202_Mated.jpg
```

==========

Keywords: [Begin Cn Model]
Required: Yes
Argument: Text String (Model name)
Description: This keyword marks the beginning of each individual connector model.
Sub-params: Cn_Model_Type, Cn_SGR, Ref_impedance
Usage Rules: Following the keyword is a single argument specifying the name of this particular connector model. The name must be one of the model names listed under the ‘model name’ column of the preceding [Cn Model List] keyword. Note that for every model name listed by the [Cn Model List] keyword there must be a corresponding model as defined by a [Begin Cn Model]/[End Cn Model] keyword pair.

Sub-Parameters:

Cn_Model_Type

Following the keyword line is a required sub-parameter Cn_Model_Type. This sub-parameter details the conditions and usage assumptions under which the connector electrical (L,R,C) data was gathered.

The legal values for this sub-parameter are “SLM_General” “SLM_Quiescent”, “SLM_EvenMode”, “SLM_OddMode”, “MLM”, and “S-parameter”.

The sub-parameter and its value are separated by a single whitespace. Only one Cn_model_Type is allowed per connector model.

A Cn_Model_Type sub-parameter value of “SLM_Quiescent” indicates that the model data was gathered with all other neighboring conductors non-switching and terminated by a reference impedance as given by the optional Ref_Impedance sub-parameter.

A value of “SLM_EvenMode” indicates that all other neighboring conductors are transitioning at the same time and in the same direction as the conductor being modeled (i.e. under even mode switching conditions). A value of “SLM_OddMode”

indicates that all other neighboring conductors are transitioning at the same time but in the opposite direction as the conductor being modeled (i.e. under odd mode switching conditions).

A value of “SLM_General” indicates that the model data was gathered with all other neighboring conductors non-switching and the ground to signal ratio is specified by Cn_GSR and the return path is unknown.

Note that the use of an SLM_* value indicates that this is a “single line model” (SLM) and no electrical data is supplied to model the coupling between pins. Specifically, the corresponding R/L/C matrix must contain only a single diagonal term. SLMs can be used to evaluate for Prop Delay, bandwidth, and impedance where ideal ground can be assumed. SLMs should not be used when lossy return paths contributions are required for simulations.

“SLM_Quiescent”, “SLM_EvenMode” and “SLM_OddMode” have a defined reference or return path. The connection pins used as the return path should be specified as “GND” in the matrix. SLM_General does not have specified Return Path given in the matrix.

A Cn_Model_Type value of “MLM” indicates that this model is a “multi line model” and the electrical data suitable for modeling pin to pin coupling is supplied. Any combination of Diagonal_matrix, Sparse_matrix, Banded_matrix or Full_matix may Be used in an MLM model. In general, MLM’s should not be used in simulations with perfect ground planes on both sides of the connector model.

A Cn_Model_Type value of “S-parameter” indicates that this model is a “multi line model” and the electrical data suitable for modeling pin to pin coupling is supplied in a frequency domain format. Specifically, the model is a secondary file in “Touchstone” format.

Example: Cn_Model_type SLM_Quiescent

Cn_SGR

If the Cn_Model_Type sub-parameter is equal to SLM_General, then this Cn_GSR sub-parameter is required. This sub-parameter specifies the signal to ground ratio used when generating the SLM_General model. The signal to ground ratio is indicated as "number of signal pins per ground pin", followed by a colon (:), followed by the number one (1). For example, if the data for an 18 pin connector was taken with 6 of the pins grounded, the signal to ground ration is 18/6 or 3 to 1. The Cn_SGR sub-parameter argument would then be listed as 3:1.

This sub-parameter is only valid when the Cn_Model_Type sub-parameter has a value of "SLM_General".

If this sub-parameter is present and the Cn_Model_Type sub-parameter is NOT "SLM_General" a warning should be issued.

Ref_impedance

An optional sub-parameter that can be used to specify the impedance of the system used to extract SLM values. If not given the Ref_Impedance value is assumed to be 50 ohms. The reference impedance for a “S-parameter” model is located in the

model file.

==========

OLD Keyword: [Path Description]

Required:	Yes
Argument:	None
Description:	This keyword describes the connection path between the pin or pins on one side of the connector and the corresponding set of pins on the other side. This pin to pin interconnect is divided into one or more "sections", where each section consists of a matrix of L,R and C and/or s-parameter data. Sections are generally cascaded in series, but the Cn_fork and Cn_endfork sub-parameters allows sections to branch off the main interconnect path as a stub or tee.
Sub-params:	Model_PinMap, Cn_Section Cn_Fork, Cn_EndFork
Usage Rules:	A single [Path Description] is required between a [Begin Cn Model] and [End Cn Model] keyword pair. Following the [Path Description] keyword line are the sub-parameters that describe the topology and electrical characteristics of the pin to pin connections through the connector.

A path description begins with the sub-parameter Model_PinMap, which points to the [Cn Pin Map] used for one end of the connector. This is followed by one or more Cn_section sub-parameters. The electrical properties of a connector are contained in the [Cn Section] matrix(s) pointed to by the Cn_Section sub-parameter. Branches or "T" connections are indicated by the "Cn_Fork" and "Cn_EndFork" sub-parameters. Finally, a path description is terminated by a Model_PinMap sub-parameter.

These sub-parameters are described in detail below.

Model_PinMap

A path description begins with the required sub-parameter Model_PinMap. This sub-parameter is followed by the name of a "pin map" which names the pins on that end of the connector. A second Model_PinMap is used at the end of the path description to reference the pin map used for the other end of the connector. The Model_PinMap sub-parameter is also used to reference a pin map for the pins at the end of any Cn_fork. Note that the pin map name given as an argument to a Model_PinMap sub-parameter must match the name of a pin map defined by a subsequent [Cn Pin Map] keyword.

The argument to the Model_PinMap sub-parameter and the sub-Parameter itself are separated by whitespace. There can be as many Model_PinMap sub-parameters as required to represent a connector, but each must be on a single line.

In general, a connector will have a single pin map that is referenced at the beginning and end of a path description. However, two different pin maps will used if the connector contains cross over pins.

Cn_Section

Following the opening Model_PinMap sub-parameter is one or more instances of the required sub-parameter Cn_Section. This sub-

parameter defines which [Cn Section] matrix is part of this connector model. Matrix sections are serially interconnected and may be re-used to form a complete connector model.

Arguments "multiplier" and "section name" must follow the Cn_Section parameter.

The multiplier is used describe the number of times the same section should be serially connected. The multiplier may be any positive value greater than zero.

The "section name" is the name of the section as specified under the [Begin Cn Section] keyword.

Cn_Section may be used as many times as required to describe the connector. Each scale factor and section name MUST be separated by at least one space.

Named sections are NOT specified in per-unit-length. They are specified as unit-less total R, L, G, and C values and/or s-parameters for the given section.
Cn_Section must appear after the keyword [Begin Cn Model] and before the keyword [End Cn Model].

Cn_Fork, Cn_EndFork

The Cn_Fork and Cn_EndFork sub-parameters are used if there are forks or 'stubs' in the connector being modeled. Cn_Fork marks the beginning of the stub while Cn_EndFork marks the end of the stub. These sub-parameters do not have any arguments.

At a fork, a "tee" section is created. Matrix sections are connected to the previous section and interconnected to Zero or more sections between Cn_Fork and Cn_Endfork. If no pin map name is placed between a Cn_Fork and Cn_Endfork block, then the fork constitutes a stub. However, if a pin map name is placed at the end of the Cn_Fork and Cn_Endfork block, an externally available port is created on the connector.

Any number of Cn_Fork sub-parameters may be used per path description, however only one sub-parameter is allowed per line.

Example: Cn_Fork
Cn_EndFork

==========

"Cn Model" Examples

==========

Example 1: (a simple single line model with one section)

```
[Begin Cn Model] MyModelExample1
  Cn_Model_Type SLM_General
  Cn_SGR 3:1
  Ref_Impedance 50

[Path Description]
  Model_PinMap  Example1_pinmap
```

```
  Cn_Section 1.0 Diagonal_matrix1
  Model_PinMap  Example1_pinmap
  .
  .
  .
[End Cn Model]
```

Example 2: (2 ports per pin, 5 sections straight through)

```
        SectA       SectB      SectC     SectD      SectE
Port1 >--------< >--------< >-------< >--------< >----------< Port2
```

```
[Begin Cn Model] MyModelExample2
  Cn_Model_Type SLM_Quiescent

[Path Description]
  Model_PinMap MyModelPinMapA
  Cn_Section 1.0 SectA
  Cn_Section 1.0 SectB
  Cn_Section 1.0 SectC
  Cn_Section 1.0 SectD
  Cn_Section 1.0 SectE
  Model_PinMap MyModelPinMapB
  .
  .
  .
[End Cn Model]
```

Example 3: (2 ports per pin, 2 series sections with one stub)

```
        SectionA          SectionB
Port1 >------------< >--------------< Port2
                  |
             StubSection1
```

```
[Begin Cn Model] MyModelExample3
  Cn_Model_Type MLM
[Path Description]
  Model_PinMap MyModelPinMapA
  Cn_Section 1.0 SectionA
  Cn_Fork
    Cn_Section 1.0 StubSection1
  Cn_EndFork
  Cn_Section 1.0 SectionB
  Model_PinMap MyModelPinMapB
  .
  .
  .
[End_Cn_Model]
```

Example 4: (2 ports per pin, 2 stub sections, one stub matrix is re-used)

```
           A            B            C
Port1 >----------< >-----------< >----------< Port2
                  |stub1        |stub4
                  |stub2
                  |stub3
```

```
[Begin Cn Model] MyModelExample4
  Cn_Model_Type MLM
```

```
[Path Description]
  Model_PinMap MyModelPinMapA
  Cn_Section 1.0 A
  Cn_Fork
    Cn_Section 1.0 stub1
    Cn_Section 1.0 stub2
    Cn_Section 1.0 stub3
  Cn_EndFork
  Cn_Section 1.0 B
  Cn_Fork
    Cn_Section 1.0 stub4
  End_Fork
  Cn_Section 1.0 C
  Model_PinMap MyModelPinMapB
  .
  .
  .
[End_Cn_Model]
```

Example 5: (2 ports per pin, 3 stubs)

```
         A        B        C         D
Port1 >------< >------< >------< >-------< Port2
              |Stub1   |Stub2   |Stub1
                       |Stub3
                       |Stub4
```

```
[Begin Cn Model] MyModel4
  Cn_Model_Type MLM
[Path Description]
  Model_PinMap MyModelPinMapA
  Cn_Section 1.0 A
  Cn_Fork
    Cn_Section 1.0 Stub1
  Cn_Endfork
    Cn_Section 1.0 B
  Cn_Fork
    Cn_Section 1.0 Stub2
    Cn_Section 1.0 Stub3
    Cn_Section 1.0 Stub4
  Cn_Endfork
    Cn_Section 1.0 C
  Cn_Fork
    Cn_Section 1.0 Stub1
  Cn_EndFork
    Cn_Section 1.0 D
  Model_PinMap MyModelPinMapB
  .
  .
  .
[End Cn Model]
```

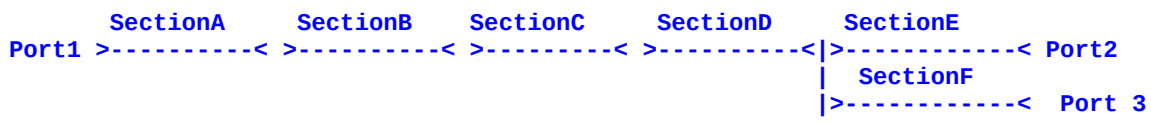
Example 6: (3 ports per pin, 2 stubs)

```
         A        B        C         D
Port1 >------< >------< >------< >-------< Port 2
              |Stub1   |Stub2   |Stub1
                       |Stub3
                       |Stub4
                       |Port3
```

```
[Begin Cn Model] MyModelExample6
```

```
Cn_Model_Type MLM
[Path Description]
  Model_PinMap MyModelPinMapA
  Cn_Section 1.0 A
  Cn_Fork
    Cn_Section 1.0 Stub1
  Cn_EndFork
  Cn_Section 1.0 B
  Cn_Fork
    Cn_Section 1.0 Stub2
    Cn_Section 1.0 Stub3
    Cn_Section 1.0 Stub4
    Model_PinMap MyModelPinMapC
  Cn_EndFork
  Cn_Section 1.0 C
  Cn_Fork
    Cn_Section 1.0 Stub1
  Cn_EndFork
  Cn_Section 1.0 D
  Model_PinMap MyModelPinMapB
  .
  .
  .
[End Cn Model]
```

Example 7: (3 ports per pin, 5 sections straight through)




```
[Begin Cn Model] MyModelExample7
  Cn_Model_Type SLM_Quiescent
[Path Description]
  Model_PinMap = MyModelPinMapA
  Cn_Section 1.0 SectionA
  Cn_Section 1.0 SectionB
  Cn_Section 1.0 SectionC
  Cn_Section 1.0 SectionD
  Cn_Fork
    Cn_Section  1.0 Cn_SectionF
    Model_PinMap MyModelPinMapB
  Cn_EndFork
  Cn_Section 1.0 SectionE
  Model_PinMap MyModelPinMapC
  .
  .
  .
[End Cn Model]
```

```
================================================================================
NEW  Keyword: [Path Description]
    Required: Yes
    Argument: Text String (name_of_a_path)???
 Description: This keyword marks the beginning of a path description
              section.  This section contains information about how
              connections are made between any two or more nodes.  The nodes
              can represent the pins on either side of connectors, packages,
              or sockets, as well as the pads or bumps of a die (where
              connections to the package or printed circuit board are made),
              or the ends of either side of interconnect sections of printed
              circuit boards.  An interconnect path may have one or more
```

Sections where each Section is described by RLGC, or S-parameter matrices. Multiple Sections within a path description may be connected in any arbitrary manner by using the usual node naming techniques, however, note that special rules and restrictions apply when the model makes use of the swath feature.

Sub-params: Section, Model_NodeMap

Usage Rules: Following the keyword is the name of the path description.(???) Multiple [Begin Path Description]/[End Path Description] keywords are allowed per file, but at least one is required.

Path description sections are terminated by the occurrence of the [End Path Description] keyword.

This keyword pair must be placed outside the [Begin Cn??? Model Family] / [End Cn??? Model Family] keywords to allow it to be reused.(???)

The [Begin Path Description] keyword line is followed by at least one Section sub-parameter which describes the topology and electrical characteristics of the interconnect. The electrical properties of an interconnect section are described in the matrices under the [Begin Cn??? Section] keyword which is referenced by the Section sub-parameter discussed below.

Connections between the nodes of the various Sections, branching or T-ing are achieved using the usual node naming techniques.

* * The sub-parameter Model_NodeMap is required only when swath matrices are used.

The order in which the Section and Model_NodeMap sub-parameters are listed in a path description is irrelevant. However, for visual clarity it is a good practice to follow the physical layout of the interconnect with the order in which they are listed.

Detailed descriptions of these sub-parameters follow below.

Section

The Section sub-parameter is used to describe the connectivity and electrical characteristics of an interconnect segment in the path description.

A path description must have at least one occurrence of the required sub-parameter Section, but can have as many Sections as needed to adequately describe an interconnect, but each must be on a single line. (???Long node list???)

The sub-parameter name Section is followed by three arguments, separated by at least one white space.

The first argument is a list of nodes and must be enclosed in parentheses. The node list must always contain an even number of nodes, which must be twice the number of rows in the corresponding RLGC matrix data, or must match the number of ports in the corresponding S-parameter data.

The node names within the parentheses are separated by at least one white space and/or a comma (semicolon???) which may be used for better readability. (One may want to use a comma in the middle of the node list to visually separate the first and second half of the nodes representing the nodes at the beginning and the end of a Section, respectively).

The order of the nodes must match the order of the rows in the corresponding RLGC matrix data, or the order of the nodes must be the same as the port number in the corresponding S-parameter data.

Matching node names inside a path description section are assumed to be connected with an ideal short. Nodes having the same names in different path description sections are assumed to be not connected.

The second argument contains either:

*** a) “M” which is a single value multiplier, which is either a scale factor determining how many times the section should be repeated (cascaded). The multiplier may be any positive value greater than one.

*** b) “L” which is used to define a per unit length value, which is a factor that defines the current path length of the model. The length may be any positive value greater than zero and is given in meters.

???The multiplier allows a matrix section to be re-used as a scaled value. The scale factor is applied to each parameter of all matrices resulting in scaled R,L, G, and C matrices.???

???Named Sections are NOT specified in per-unit-length. They are specified as unit-less total R, L, G, C values for the given Section.???

The third argument contains the name of the Section which must match the Section name defined by a subsequent [Begin Cn??? Section] keyword. The electrical characteristics of Sections are described in the form of RLGC or S-parameter matrices under the [Begin Cn??? Section] keywords. A [Begin Cn??? Section] keyword may be referenced by more than one Sections sub-parameters even if they are in different path descriptions.

Model_NodeMap

The optional sub-parameter Model_NodeMap is used to provide a mapping between the node names and the rows of swath matrices. This sub-parameter is also useful when the data book pin names are identical on either side of a connector, or when only a signal name is available for a conductor on package or BCP, to provide a clear mapping of the node names to a particular end or side of the connector or interconnect (without shorting the pins of the two sides). Typically there would be as many Model_NodeMap sub-parameters in a path description as many ends or sides a connector or an interconnect has.

If used, the Model_NodeMap is followed by two arguments.

The first argument is the name of the Model_NodeMap

sub-parameter, and is separated from the sub-parameter by at least one white space. The name should be a meaningful word which describes which end or side of the interconnect the subsequent mapping table refers to.

* The second argument is a three or four-column table. The
* first column contains the data book pin names of a
* connector, or the signal names of conductors on a package
* or PCB. The second and third columns provide the physical row
* and physical column numbers of that pin, respectively. The
* fourth column contains the corresponding node names that are
* used in the path description in which the sub-parameter
* appears. This column may or may not have an entry depending
* on whether the swath matrix includes this pin or not.

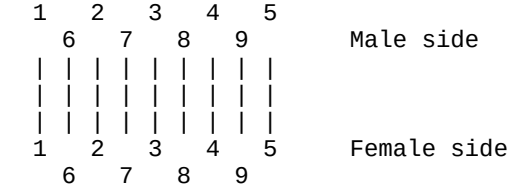
There can be as many Model_NodeMap sub-parameters in a path description as needed to provide pin or signal name to node name maps of each end or side of a connector, package, or PCB interconnect. Note that even though the first columns of different Model_NodeMaps may be identical pin or signal name lists, this does not indicate a shorted connection between them as it is with node names. Also, nodes listed under the Model_NodeMap sub-parameter are not required to be nodes having a single connection only.

If the model uses the swath feature, the number of unique node names listed in each Model_NodeMap sub-parameter must be the same, and every Section in a path must have the same number of conductors (i.e. matrix size). These restrictions do not apply if the swath feature is not used.

Please note that the following examples can be easily applied to connectors as well as cable assemblies, packages and printed circuit boards.

Example 1:

A simple model for a DB9 male to female adapter with a straight through wiring modeled as one section. The pin numbering increases left to right when facing the male side, and right to left when facing the female side. The optional Model_NodeMap sub-parameter is included for completeness.




```
[Begin Cn Model] DB9M_F
  Cn_Model_Type MLM
[Begin Path Description] Path_DB9M_F
  Model_NodeMap Male_side
|       physical
|  pin  row  col  node
    1    1    1    M1
    2    1    3    M2
    3    1    5    M3
    4    1    7    M4
    5    1    9    M5
    6    2    2    M6
```

```
    7    2    4    M7
    8    2    6    M8
    9    2    8    M9
  Section (M1 M2 M3 M4 M5 M6 M7 M8 M9,
           F1 F2 F3 F4 F5 F6 F7 F8 F9) M=1.0 RLGC_matrix_9x9
  Model_NodeMap Female_side
|       physical
|  pin  row  col  node
    1    1    1    F1
    2    1    3    F2
    3    1    5    F3
    4    1    7    F4
    5    1    9    F5
    6    2    2    F6
    7    2    4    F7
    8    2    6    F8
    9    2    8    F9
[End Path Description]
[End Cn Model]
-------------------------------------------------------------------------------
Example 1 using a 6x6 swath matrix:

Note that the Mode_NodeMap sub-parameter is required for this example because
a swath matrix is used.

  1   2   3   4   5
    6   7   8   9       Male side
  | | | | | | | | |
  | | | | | | | | |
  | | | | | | | | |
  1   2   3   4   5     Female side
    6   7   8   9

[Begin Cn Model] DB9M_F
  Cn_Model_Type MLM
[Begin Path Description] Path_DB9M_F
  Model_NodeMap Male_side
|       physical
|  pin  row  col  node
    1    1    1    M1
    2    1    3    M2
    3    1    5
    4    1    7    M4
    5    1    9    M5
    6    2    2    M6
    7    2    4
    8    2    6
    9    2    8    M9
  Section (M1 M2 M4 M5 M6 M9,
           F1 F2 F4 F5 F6 F9) M=1.0 RLGC_matrix_6x6
  Model_NodeMap Female_side
|       physical
|  pin  row  col  node
    1    1    1    F1
    2    1    3    F2
    3    1    5
    4    1    7    F4
    5    1    9    F5
    6    2    2    F6
    7    2    4
    8    2    6
    9    2    8    F9
[End Path Description]
```

```
[End Cn Model]
```

Example 2:

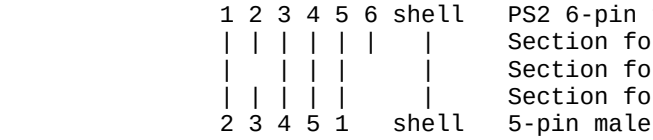
This is a model of the same adapter as in example 1, except that both sides are male which results in a crossed pinout. The optional Model_NodeMap sub-parameter is not included for brevity.

```
  1   2   3   4   5
    6   7   8   9       Male side A
  | | | | | | | | |
  | | | | | | | | |
  | | | | | | | | |
  5   4   3   2   1     Male side B
    9   8   7   6

[Begin Cn Model] DB9M_M
  Cn_Model_Type MLM
[Begin Path Description] Path_DB9M_M
  Section (A1 A2 A3 A4 A5 A6 A7 A8 A9,
           B5 B4 B3 B2 B1 B9 B8 B7 B6) M=1.0 RLGC_matrix_9x9
[End Path Description]
[End Cn Model]
```

Example 3:

This is a model of a PS2 6-pin female mini DIN to 5-pin male DIN adapter. The optional Model_NodeMap sub-parameter is included for completeness. The model consist of three sections to provide a more detailed model. The drawing of this adapter was found on the following web site: http://www.grand-tek.com/SPEC-7/GM06F-R05MM0-04.pdf




```
[Begin Cn Model] PS2_6_DIN_5
  Cn_Model_Type MLM
[Begin Path Description] Path_PS2_6_DIN_5
  Model_NodeMap PS2_side
|       physical
|  pin  row  col  node
    1    1    1   PS1
    2    1    2   PS2
    3    1    3   PS3
    4    1    4   PS4
    5    1    5   PS5
    6    1    6   PS6
  shell  1    7   PS7
  Section ( PS1  PS2  PS3  PS4  PS5  PS6  PS7,
            n11  n12  n13  n14  n15  n16  n17) M=1.0 RLGC_matrix_7x7
  Section ( n11       n13  n14  n15       n17,
            n21       n23  n24  n25       n27) M=1.0 RLGC_matrix_5x5
  Section ( n21  n22  n23  n24  n25       n27,
           DIN2 DIN3 DIN4 DIN5 DIN1      DIN7) M=1.0 RLGC_matrix_6x6
  Model_NodeMap DIN_side
|       physical
|  pin  row  col  node
    1    1    1   DIN1
    2    1    2   DIN2
    3    1    3   DIN3
```

```
    4    1    4   DIN4
    5    1    5   DIN5
  shell  1    6   DIN7
[End Path Description]
[End Cn Model]
```

Example 4:

This is a model of a PS2 keyboard Mouse Y-splitter adapter. The optional Model_NodeMap sub-parameter is included for completeness. The model consist of only one section for simplicity, but includes connections between the shells (not shown on the drawing of the web site). The specifications for this adapter were found on the following web site: http://www.hardwarebook.net/adapter/userinput/ps2keyboardygateway.html

```
    1 2  3   4  5 6 shell       6-pin male mini DIN (computer)
    | | +-+ +-+ | |   +---+
    | | | | | | | |   |   |     Section
    2 | 3 | 4 | 6 | shell |     6-pin female DIN (keyboard)
      2   3   4   6     shell   6-pin female DIN (mouse)

[Begin Cn Model] PS2_splitter
  Cn_Model_Type MLM
[Begin Path Description] Path_PS2_splitter
  Model_NodeMap Computer
|       physical
|  pin  row  col  node
    1    1    1    C1
    2    1    2    C2
    3    1    3    C3
    4    1    4    C4
    5    1    5    C5
    6    1    6    C6
  shell  1    7    Cs
  Section (C1 C3 C4 C5 Cs C2 C3 C4 C6 Cs,
           K2 K3 K4 K6 Ks M2 M3 M4 M6 Ms) M=1.0 RLGC_matrix_10x10
  Model_NodeMap Keyboard
|       physical
|  pin  row  col  node
    2    1    2    K2
    3    1    3    K3
    4    1    4    K4
    6    1    6    K6
  shell  1    7    Ks
  Model_NodeMap Mouse
|       physical
|  pin  row  col  node
    2    1    2    M2
    3    1    3    M3
    4    1    4    M4
    6    1    6    M6
  shell  1    7    Ms
[End Path Description]
[End Cn Model]
```

Example 5:

2 ports per pin, 5 sections straight through

```
         SectA     SectB     SectC     SectD     SectE
Port_A >-------< >-------< >-------< >-------< >-------< Port_B

[Begin Cn Model] MyModelExample5
```

```
  Cn_Model_Type SLM_Quiescent
[Begin Path Description] name5
  Section (A1 A2 A3 A4, 11 12 13 14) L=1.0 SectA
  Section (11 12 13 14, 21 22 23 24) L=1.0 SectB
  Section (21 22 23 24, 31 32 33 34) L=1.0 SectC
  Section (31 32 33 34, 41 42 43 44) L=1.0 SectD
  Section (41 42 43 44, B1 B2 B3 B4) L=1.0 SectE
[End Path Description]
[End Cn Model]
```

Example 6:

2 ports per pin, 2 series sections with one stub that taps into line_2 and line_4

```
        SectionA          SectionB
Port_A1 >----------<-----<----------< Port_B1
Port_A2 >----------<---o-<----------< Port_B2
Port_A3 >----------<---+-<----------< Port_B3
Port_A4 >----------<-o-+-<----------< Port_B4
Port_A5 >----------<-+-+-<----------< Port_B5
                    | |
                    | |
               StubSection1
```

```
[Begin Cn Model] MyModelExample6
  Cn_Model_Type MLM
[Begin Path Description] name6
  Section   (A1 A2 A3 A4 A5, 11 12 13 14 15) M=1.0 SectionA
    Section (   12    14   ,    x2    x4   ) M=1.0 StubSection1
  Section   (11 12 13 14 15, B1 B2 B3 B4 B5) M=1.0 SectionB
[End Path Description]
[End Cn Model]
```

Example 7:

2 ports per pin, 2 stub sections, one stub matrix is re-used
??? what is reused here???

```
            A           B           C
Port_A >---------< >---------< >---------< Port_B
                  |stub1      |stub4
                  |stub2
                  |stub3
                  |Port_C
```

```
[Begin Cn Model] MyModelExample7
  Cn_Model_Type MLM
[Begin Path Description] name7
  Section   (A1 A2 A3 A4 A5, 11 12 13 14 15) M=1.0 A
    Section (11 12 13 14 15, s1 s2 s3 s4 s5) M=1.0 stub1
    Section (s1 s2 s3 s4 s5, t1 t2 t3 t4 t5) M=1.0 stub2
    Section (t1 t2 t3 t4 t5, C1 C2 C3 C4 C5) M=1.0 stub3
  Section   (11 12 13 14 15, 21 22 23 24 25) M=1.0 B
    Section (21 22 23 24 25, x1 x2 x3 x4 x5) M=1.0 stub4
  Section   (21 22 23 24 25, B1 B2 B3 B4 B5) M=1.0 C
[End_Cn_Model]
```

Keywords: [Cn Model Description]
Required: No
Argument: Text Block

```
 Description:  Provides a concise yet easily human-readable description of
               this connector model

 Usage Rules:  This keyword may appear multiple times in the file however
               it may appear only once between each pair of
               [Begin_Cn_Model]/[End_Cn_Model] keywords.

-------------------------------------------------------------------------------
[Begin Cn Model Description]
High Density 0.1 center square pin with PCB effects


===============================================================================
     Keyword:  [Cn Number of Conductors]
    Required:  If [Cn Pin Map] is present, this keyword is required, if
               [Cn Pin Map] is not present, this keyword cannot be used.
    Argument:  Numeric Value
 Description:  Documents the number of current carrying conductors for the
               physical connector as defined by the matrix.
 Usage Rules:  A single positive numeric value from 1 to 100,000 is
               recommended. Larger values shall create warnings by the parser.
               This number also represents the maximum swath size for a given
               connector model.

               This keyword MUST BE USED BETWEEN KEYWORDS
               [Begin Cn Model] and [End Cn Model].

Fixed sized connector example:
-------------------------------------------------------------------------------
[Cn Number of Conductors]  8


===============================================================================
    Keywords:  [Cn Swath Parameters]
    Required:  No, unless a connector is described by a swath.
    Argument:  None
  Sub-Params:  LeftEdge RightEdge TopEdge BottomEdge
 Description:  Describes a smaller matrix which can be used to
               represent connectors of equal or larger size.  To visualize
               this, consider the swath matrix being a smaller version of the
               connector.  The edges of the matrix represent the same edge
               effects that exists in the full size connector.  The center area
               of the matrix represents the same effects as the center of the
               full size connector.

               It is the responsibility of the EDA tool to use the matrix and
               the required keyword information as a “mini” connector that is
               centered about the electrical paths of interest and/or to
               expand this information into a larger sized swath up to the
               full sized connector itself.  For example a 2x8 matrix might be
               expanded to represent a connector that is 2x100.
 Usage Rules:  Only rectangular connectors without missing pins may be
               represented by a swath matrix.  If the connector is non-
               rectangular or the pin field has missing pins then the matrix
               must describe the full connector and this keyword cannot be
               used.  This keyword MUST be present if number of entries in the
               matrix does not equal the number of conductors as given by the
               [Number of Conductors] keyword.

               This keyword must appear between the [Begin Cn Model]/[End
               Cn Model] keyword pair but after the [Cn Number of Conductors]
               keyword.
```

MUST be followed by 4 required sub-parameters:

1) LeftEdge: Columns in the swath matrix used to represent left edge
2) RightEdge: Columns in the swath matrix used to represent right edge
3) TopEdge: Rows in the swath matrix used to represent top edge
4) BottomEdge: Rows in the swath matrix used to represent bottom edge

These sub-parameters use the physical representation of the pins to define the minimum coupling bandwidth. This keyword is only required when using only a part of the matrix or when expanding the matrix.

The model maker decides on the minimum number of columns (for a column swath) or rows (for a row swath) that should be used to define the banded matrix, then assigns values to the appropriate sub-parameter that will, by subtraction of the columns or rows, create the appropriate maximum matrix size for the swath.

The basic goal is to define how far the edge effects extend into the coupling fields.

None of the values of these numbers may exceed the number of rows or column of pins that are defined by the matrices of a given model. The swath is applicable to any model supported by the specification

At this time, there must be at least two sub-params with a value not equal to zero.

Please see the example below to see how these are used with the keyword

The edge effects describe the edge rows or columns that do not contain all of the coupling effects (from both sides) that the centermost rows or columns. This information is useful for “swath expansion” when determining what larger size to use.

When the swath represents a SLM or else documents all of the pins in a physical connector, the off diagonal entries are 0 by convention. If the swath describes a section with fewer columns, but the same number of rows as the physical connector, then the row edge effects are 0 by convention. Similarly, if the swath describes a section with fewer rows, but the same number of columns as the physical connector, then the column edge effects are 0 by convention.

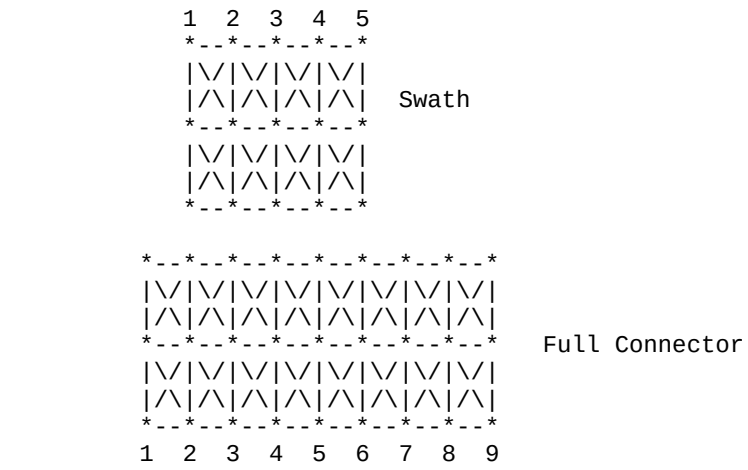
In a fully coupled swath where all pins couple to all others, all of the rows or columns except for the one designated as the centermost row or column can have edge effect simplification. So the number of rows or columns can extend to all but the center row or column. On the other hand, a coupling pattern that extends over a few columns may result in a smaller number of edges designated. Below is an example of a 3 row by 5 column swath, where the lines in a rectangular connector represent the electrical coupling:

Notice that columns 2, 3, and 4 have the maximum amount of coupling from each side. In this case the edge effect columns 1 and 5 do not have coupling from both sides.

Application of Swath Matrices:

For the purposes of this discussion a 3 row by 5 column swath is used and the full sized connector is 3 rows by 9 columns. The swath matrix data can be organized by [Cn Swath Parameters]. Three methods can be used to apply the Swath matrix information:

(1) Centering the Swath around the pins of interest



The Swath above is centered about column 4. The paths of interest including those of columns 3 and 5 are also described by swath matrix data that has coupling from both sides. However, because of edge effects, columns 2 and 6 are less accurately approximated.

Centering is a way to map the swath matrix pins directly into the corresponding pins of the actual connector.

When the signals of interest are at the left or right edges, the swath would normally be positioned just to the edges. For example, column 1 of the swath matrix would line up with column 1 of the connector.

(2) Expansion and Centering: Expanding the Swath matrix into a larger sized Swath matrix and centering it about the paths of interest.

This approach is the same as centering. However, the signal pins of interest might have more columns than the non-edge section of the swath. One solution is to expand the swath to a larger swath and then use the larger swath

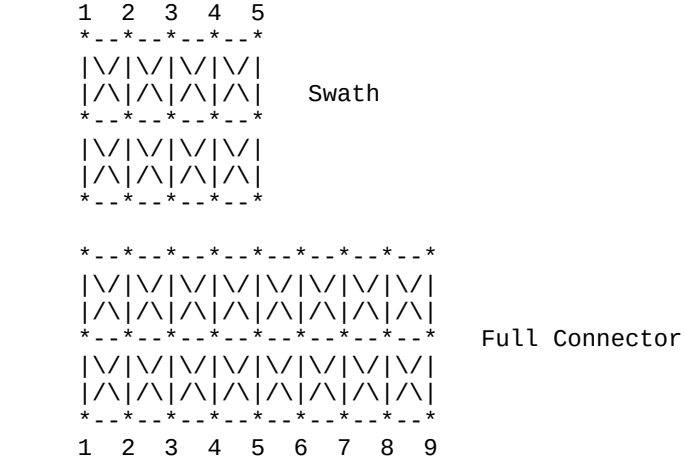
for centering. This is a subset of expanding the swath into the full sized connector.

A larger swath might be constructed to span the number of columns of interest plus the edge effect columns to assure that the most accurate coupling information from both sides is used for the columns of interest.

(3) Expansion to Full Sized Connector: Expanding the swath matrix to the full sized connector matrix.

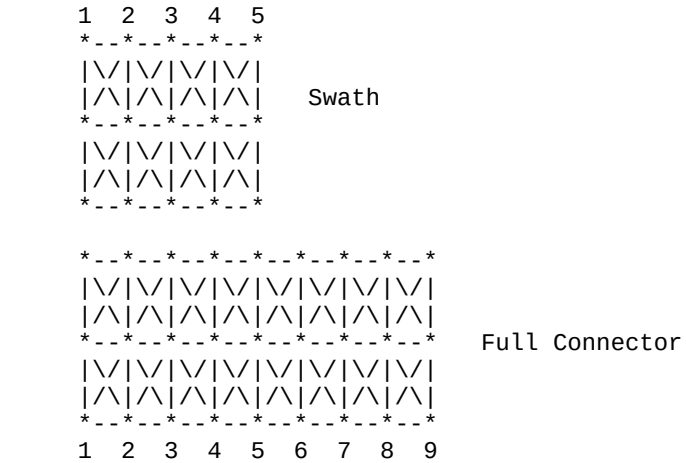
There are three steps to this procedure (a, b, and c below). This is illustrated using the 3 row, 5 column swath and showing how it is expanded into a 3 row, 9 column full sized connector

(a) Copy the left column pattern



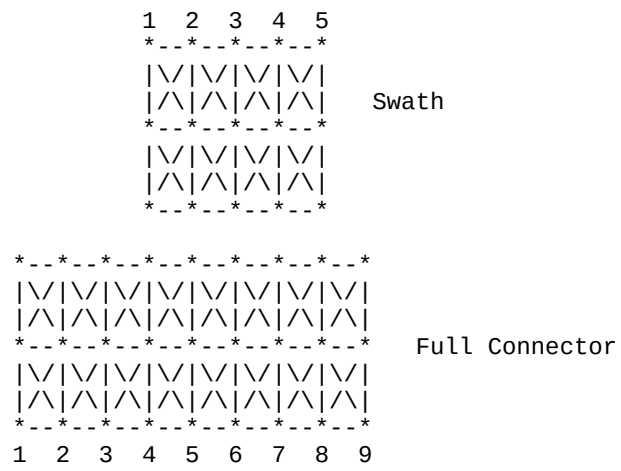
The matrix date for swath columns 1 and 2 (including forward references) copied into the new full connector matrix.

(b) repeat the center pattern,



Starting at the center of the swath, the coupling pattern of just the center column (column 3 in this example) is copied for the connector matrix. In this case, the coupling pattern extends forward to the next column. The pattern is copied by mapping the swath pin numbers into the connector pin numbers and copying the center column forward referenced data. In this example, the operation starts at the full connector column 3 and ends upon completion at the full connector column 7.

(c) Copy the right column patterns



The final step is similar to Step (a), but concluded using the right side of the swath. When the swath is positioned against the right edge as shown, then the coupling patterns for the swath columns are duplicated for the final columns of the full sized connector. So the patterns of swath columns 4 and then 5 are duplicated for the full sized connector columns 8 and 9.

A similar process could be created for expanding the number of rows. Some details are left to the EDA tool to implement. For example, if the swath contains an even number of columns, then either the left of center or the right of center column may be used for center section expansion.

```
[Cn Swath Parameters]
    LeftEdge = 1
    RightEdge = 1
    TopEdge = 0
    BottomEdge = 0
```

==========

Keywords: [Cn Swath Pin Numbers]

Required: Yes, if [Cn Swath Parameters] is present, else it must not be used.

Arguments: None

Description: Lists the pins of the matrix in the exact order that the swath matrix rows are entered and gives the corresponding physical location of the pin.

Usage Rules: The [Cn Swath Pin Numbers] keyword is followed by a list of entries that apply for all of the sections for which the [Path Description] references.

The number of entries in the list is equal to the product of the rows and columns of the [Cn Swath Pin Numbers] keyword.

Each line must contain three columns documenting the row names and physical row and column locations for each entry of the matrices under the corresponding [Begin Cn Section] keyword. The entries consist of [Row] name designated row_name, physical row number designated physical_row, and physical column number designated physical_column. All three columns are required and must be separated by at least one white space.

The physical row number consists of an integer value from 1 to the number of rows documented in the [Cn Swath Pin Numbers] keyword. Similarly, the physical column number consists of an integer value from 1 the number of columns documented in the [Cn Swath Pin Numbers] keyword.

These physical row and column numbers give the relative physical location of each of the swath matrix elements as if they were the pin numbers of the connector itself. Only one unique physical row and physical column number pair are allowed in the list. In other words, no two or more row_name entries shall refer to the same physical_row, physical_column pair.

Other Notes: It is required that all of the sections that are referenced by the [Begin Cn Section] keyword have swath matrix entries that are ordered in the same manner. This would be expected from the field solver working on a single connector swath, but issuing its results as separate sections

The EDA tool that processes this keyword has enough information to reorder the swath matrix, if needed, to position its electrical contents among the pins of interest in the actual connector.

Consider a 2 row by 3 column Swath matrix that was extracted for the pins as shown below:

A B C
D E F

The field extractor may have produced matrices ordered: C E B F A D

[Swath Pin Numbers]

row_name	physical_row	physical_column
C	1	3
E	2	2
B	1	2
F	2	3
A	1	1
D	2	1

==========

Keywords: [Cn Pin Map]
Required: Yes
Argument: Text String (Model_PinMap name)
Sub-params: pin_order, num_of_columns, num_of_rows
Description: This keyword gives the names of the pins on one side of a connector, then defines a pin ordering such that pins can be associated with rows in the RLGC or S-parameter section matrixes. This keyword also defines if the pins are listed in column ordered or row ordered fashion.

Usage Rules: Following the keyword is the name of this pin map. The pin map name must match the pin map name used as an argument to one of the Model_PinMap subparameters in a models [Path Definition] keyword.

Following the [Cn Pin Map] line are three subparameters: pin_order, num_of_rows, num_of_columns. Each subparameter is placed on a seperate line, and when used they must appear in the above order. Following the subparameters is a list of pin names that are covered by this pin map.

Subparameters and the pin list are described in detail below.

pin_order
The pin_order subparameter is required. Legal arguments to the pin_order subparameter are Row_ordered, Column_ordered or Un-ordered. The subparameter name and its argument are seperated by one or more white spaces.

Row_ordered and Column_ordered refers to the order in which the pins in the accompanyng pin list are listed. Row_ordered means that all the pins in the top row of a connector are listed in order, then the second row of pins is listed, and so on. Likewise, Column_ordered means that all the pins in the first column of a connector are listed, then the second column, as so forth.

If the Row_ordered or Column_ordered argument is specified then the num_of_rows and num_of_columns subparameters are also required.

If the un-ordered argument is specified, the pins are not listed in row or column order and the num_of_rows and num_of_columns subparameters must not be used. Un-ordered also implies that the connector is not swathable.

num_of_columns
This optional subparameter specifies the number of columns that are in the physical connector. The subparameter name is followed by a single positive integer.

num_of_rows
This optional sub-parameter specifies the number of rows that are in the physical connector. The subparameter name is followed by a single positive integer.

Note that the pin order and number information are required in order for EDA software to properly position and expand a swath matrix around a given pin.

Pin List
Following the subparameters is a list of the pin names covered by this pin map. The pin list is required. Each pin name is listed on a separate line on the file. The pin names must be organized as specified by the Pin_order subparameter.

IMPORTANT NOTE: The intent of the [Cn Pin Map] keyword pin order information is to give an EDA tool enough information to map pin names to the rows of a connectors' section matrix. In the case of a full-matrix/full-pinout description of a connector, it is required that the rows of a section matrix appear in the same row or column order as the pin names. For example, suppose the pins names of a six pin connector were listed as follows:

A1
A2
A3
B1
B2
B3

A six pin connector can be described by a 6X6 matrix. It is required that the first row of the matrix contain the data for pin A1, the second row contain the data for pin A2, and so on.

Other Notes: More than one [Cn Pin Map] keyword is allowed in a file.

This keyword must occur outside the [Begin Cn Model]/[End Cn Model]
keyword pair. This is to allow a [Cn Pin Map] to be referenced by multiple connector models.

The examples below use the connector represented by:
A1 A2 A3 A4
B1 B2 B3 B4

```
[Cn Pin Map] Baseboard_side
pin_order = row_ordered
num_of_columns = 4
num_or_rows = 2
|Pin
  A1
  A2
  A3
  A4
  B1
  B2
  B3
  B4

[Cn Pin Map] Baseboard_side2
pin_order = column_ordered
num_of_columns = 4
num_or_rows = 2
|Pin
  A1
  B1
  A2
```

```
  B2
  A3
  B3
  A4
  B4

[Cn Pin Map] Daughtecard_side
pin_order = Un_ordered
|Pin
  A1
  A2
  A3
  A4
  B1
  B2
  B3
  B4
```

```
================================================================================
Keywords: [End Cn Model]
    Required: Yes
    Argument: None
 Description: Defines the end of the connector models for the connector family.
--------------------------------------------------------------------------------
[End Cn Model]


================================================================================
Keywords: [End Cn Model Family]
    Required: Yes
    Argument: None
 Description: Defines the end of the connector model family.
 Usage Rules: These commands wrap the entire model in this file.

--------------------------------------------------------------------------------
[End Cn Model Family]


================================================================================
```

8.0 MATRIX KEYWORDS

```
================================================================================
     Keyword:  [Begin Cn Section]
    Required:  Yes
    Argument:  Text String
 Description:  Indicates the beginning and the name of a set of R, L, G, C
               matrix data.
 Usage Rules:  This keyword is followed by the name of a section.  The section
               must be used in one of the models in the document.

               Each matrix used for given connector model MUST have exactly
               the same number conductors.

               If the swath method is used the matrix may have fewer
               conductors then the connector model.

               If the swath method is not used the matrix MUST have exactly
               the same number of conductors as the connector has pins.

               The matrices used for the R, L, G, C description can be defined
               using any of the matrix description methods allowed in the
               specification. The matrix description methods can also be
```

mixed within a given section. For example the resistance matrix might use the Diagonal_matrix and the inductance matrix might use a Full matrix.

MUST be placed outside the context of keywords [Begin Cn Model] and [End Cn Model] as sections may be reused by more than one connector model.

MUST be placed outside the context of the [Begin Cn Model Family] and End Cn Model Family]

For an example of [Begin Cn Section] see matrix examples that follow.

==========

Keyword: [Derivation Method]
Required: Yes
Argument: Text String (Lumped or Distributed)
Description: Describes how the data in the RLGC or S-parameter matrixes should be interpreted. This keyword also places rules on what individual matrixes must be present in a section.
Usage Rules: Following the keyword is one of two arguments: "Lumped" or "Distributed".

If 'Lumped' is specified then the RLGC matrix data represents discrete values suitable for creating a circuit of interconnected inductor, capacitor and resistor elements. A section may be replicated, but the values in each matrix are fixed. For sections described using lumped data any single, any pair, or all of the following keywords could be used to describe a section:
[Inductance Matrix], [Capacitance Matrix], [Conductance Matrix] and [Resistance Matrix]

If 'Distributed' is specified then the RLGC matrix data represents per-unit-length values suitable for creating a transmission line representation of the section. A multiplier value representing the physical length of a section may be applied to the data in the RLGC matrixes. For sections described using distributed data both the [Inductance Matrix] and [Capacitance Matrix] keywords are required. The [Conductance Matrix] and [Resistance Matrix] keywords are optional.

For a section described using S-parameter data the argument to the [Derivation Method] keyword shall be 'lumped'.

All matrices within the same section must share the same derivation method.

[Derivation Method] Lumped

==========

Keyword: [Resistance Matrix], [Inductance Matrix], [Capacitance Matrix], [Conductance Matrix]
Required: Yes, If a simple single line model is defined then the following matrices are required:
case 1: [Resistance Matrix] only (Not to be used with distributed models
OR

case 2: [Inductance Matrix] and [Capacitance Matrix] only for which the [Resistance Matrix] is assumed to be zero ohms

OR

case 3: [Resistance Matrix] and [Inductance Matrix] and [Conductance Matrix] and [Capacitance Matrix]

NOTE: In case 1,2,3 the matrices must be Diagonal_matrix.

If a coupled model is defined then the following matrices are required:

case 4: [Inductance Matrix] and [Capacitance Matrix] only for which the [Resistance Matrix] is assumed to be zero ohms

OR

case 5: [Resistance Matrix] and [Inductance Matrix] and [Capacitance Matrix]

case 6: [Resistance Matrix] and [Inductance Matrix] and [Capacitance Matrix] and [Conductance Matrix]

NOTE: In case 4,5, and 6 the matrices shall be any combination of matrix types (Diagonal, Banded, Sparse, Full).

Argument: Matrix Type
Description: The sub-parameters mark the beginning of a matrix, and specify how the matrix data is formatted.
Usage Rules: For each matrix keyword, only one of the sub-parameter of each type is allowed (i.e. one inductance, one capacitance, one conductance, and one resistance matrix is allowed per section.)

After each of these sub-parameters, insert the matrix data in the appropriate format. (These formats are described in detail below.)

MUST BE USED between keywords [Begin Cn Section] and [End Cn Section].

If the model type in SLM, then only the diagonal matrix is allowed.

Matrix Type Options:
Diagonal_matrix, Banded_matrix, Sparse_matrix, or Full_matrix

Diagonal_matrix
Required: No
Argument: None
Description: The Diagonal_matrix is used to specify uncoupled models.

This sub-parameter is exactly the same as using the Banded_matrix described below with a Bandwidth=0. It has the added benefit of requiring half as many lines.

A Diagonal_matrix is one whose entries are guaranteed to be zero if they are not on the main diagonal.

Specify the Diagonal_matrix one row at a time, starting with row 1 and working up to higher

rows. Diagonal elements are entered one per line. MUST be used between keywords [Begin Cn Section] and [End Cn Section].

Banded_matrix
Required: No
Argument: None
Description: The Banded_matrix is used to specify the coupling effects up to “X” pins on either side for a given pin.

A Banded_matrix is one whose entries are guaranteed to be zero if they are farther away from the main diagonal than a certain distance, known as the "bandwidth."

The bandwidth for a Banded_matrix must be specified using the [Bandwidth] keyword.

Specify the banded matrix one row at a time, starting with row 1 and working up to higher rows. Mark each row with the [Row] keyword.

Symmetry is exploited. Entries other than the main diagonal are not required.

Sparse_matrix
Required: No
Argument: None
Description: A Sparse_matrix is expected to consist mostly of zero-valued entries, except for a few nonzero. Unlike the Banded_matrix, there is no restriction on where the nonzero entries can occur.

Symmetry can be exploited to reduce the amount of data by eliminating from the matrix any entries below the main diagonal.

An N x N Sparse_matrix is specified one row at a time, starting with row 1 and continuing down to row N. Each new row is marked with [Row] keyword, as in the other matrix formats.

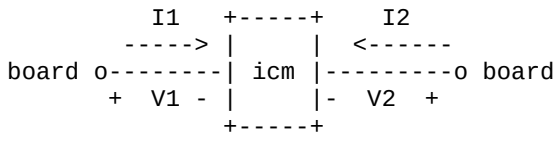
Full_matrix
Required: No
Argument: None
Description: When the Full_matrix format is used, the couplings between every Pair of elements is specified explicitly. Assume that the Matrix has N rows and N columns.

The Full_matrix is specified one row at a time, starting with Row 1 and continuing down to Row N. Each new row is identified with the [Row] keyword.

Matrix Descriptions

The resistance, inductance, conductance, and capacitance matrices

may also be referred to as "RLGC matrices" within this specification. When measuring the entries of the RLGC matrices, either with laboratory equipment or field-solver software, currents are defined as ENTERING the pins of the package from the board. The corresponding voltage drops are to be measured with the current pointing "in" to the "+" sign and "out" of the "-" sign.



It is important to observe this convention in order to get the correct signs for the mutual inductance and resistances.

For each [Resistance Matrix], [Inductance Matrix], [Conductance Matrix] or [Capacitance Matrix] a different matrix type can be used for the data.

Resistance Matrix Description

UNITS: OHMS
DESCRIPTION: The resistance matrix represents the total DC resistance between the two ends of this section of the connector.

NOTES:
The resistance matrix is optional. Simulation tools shall use 0.0 ohms for all diagonal entries and “infinite” for all off diagonal entries if this matrix is not present.

DO NOT USE ohms/meter as this is the total value and includes the length of this section.

Values should all appear in the diagonal terms of the matrix. For example: value 1,1 is the resistance for pin 1 and value 2,2 is the resistance for pin 2.

A "Diagonal_matrix" matrix type is the best choice for describing the resistance matrix. Diagonal elements may be dominant. Comparably, the off diagonal elements may be negligible thus leading to off diagonal element values of zero leading to the [Diagonal_matrix] format

Inductance Matrix Description

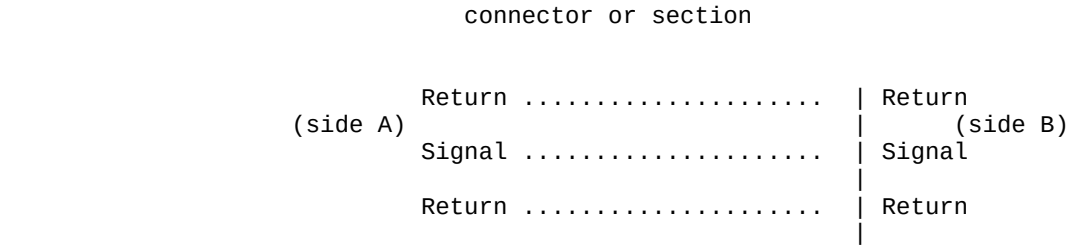
UNITS: HENRIES
DESCRIPTION: Two inductance matrix methods are supported:
Loop-Inductance Matrix for SLMs ONLY (non-coupled)
Partial-Inductance Matrix for MLMs ONLY (coupled)

If ModelType is SLM then all sections must use loop-inductance and capacitance matrices.

If ModelType is MLM then all sections must use partial-inductance and capacitance matrices.

Loop-Inductance Matrix Description (SLMs ONLY):

DESCRIPTION: To define inductance requires a loop. The loop consists of a closed path including a signal and all possible return paths. The inductance is a coefficient between the emf induced in the loop and the change of magnetic flux through the loop.



The return path must include all pins that are power or ground. In a loop inductance matrix, the diagonal terms represent the loop inductance of a specific pin with return currents assumed returned through all pins defined to be power or ground.

NOTES (Single-Line Loop-Inductance Matrix):

This method must be used if the ModelType is SLM.

Loop inductance can be experimentally measured.

Loop inductance matrices can be created using field solvers.

An SGR value is normally specified for an SLM model that indicates the ratio of signal pins to return path pins (i.e. power or ground) pins.

SLM models make have zero values for power or ground R, L, G, C values which prevents there use in analyzing connector ground bounce effects.

Typical uses for SLM model might include for example a 2x17 header connector where the bottom row is all grounded. The SGR would be specified in this case as 1:1. A model would be developed with these assumptions made. Then all pins would be filled with average values. To further expand this example see the figure below:

Pin out:

```
1  ................. 17
18 ................. 34
```

Measurement or Field solver connection:

```
Sig Sig Sig ... Sig Sig
Gnd Gnd Gnd ... Gnd Gnd
```

Measured or Computed values:

```
3nH 4nH 4nH ... 4nH 3nH
```

Matrix data:

```
3nH 4nH 4nH ... 4nH 3nH
3nH 4nH 4nH ... 4nH 3nH
```

This allows the user to specify any of the following power/ground connections and still use this model:

Simulator Connection A:

```
                Sig Sig Sig ... Sig Sig
                Gnd Gnd Gnd ... Gnd Gnd

        Simulator Connection B:
                Gnd Gnd Gnd ... Gnd Gnd
                Sig Sig Sig ... Sig Sig

        Simulator Connection C:
                Sig +5V Sig ... Gnd Sig
                Gnd Sig Gnd ... Sig +5V
```

Given the following connection a simulation should generate a warning to the user indicate the ground-signal-ratio has been violated.

```
        Simulator Connection D:
                Sig Sig Sig ... Sig Sig
                Gnd Sig Sig ... Sig Sig
```

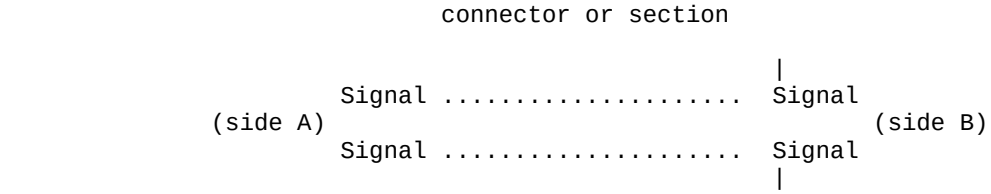
All connector pins must be specified with non-zero L and C values in at least one of the sections. The simulator will zero out portions of all sections corresponding to the ground or power pins.

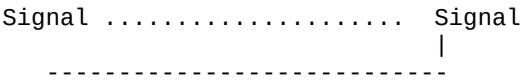
SLM models are most useful where coupling analysis is not critical and where the rise time of the signal is long compared to the electrical delay through the connector.

Partial Self Inductance Matrix Description (MLMs ONLY):

DESCRIPTION: A partial self inductance is an approximation of the loop inductance in situations where the current return path cannot be explicitly stated in the field extraction stages. Partial self inductance can be numerically calculated by introducing a current in a circuit element that originates on one end and terminates on the other end. There is no physical loop per se, instead charges will build up on each side of the circuit element. In practice, this situation cannot exist more than instantaneously. For the purposes of simulation the condition does not affect the resultant loop inductance values. Then from the currents found the magnetic fields and inductance's can be calculated. In this way, the self and mutual inductances of any section of a closed circuit can be calculated.

This representation of inductance in terms of partial self and mutual inductance allows the analysis of any combination of current paths. The signal and return current paths can be defined in circuit after the model is established. Assuming that the self inductance is included in the return path, the circuit under simulation can account for the impact on signal-integrity, cross-talk and ground-bounce from currents passing through any combination of pins.





There are many ways to create this information using either 2D or 3D field solvers. It may also be possible to create this information using other methods which are left to the model developer.

For an MLM model it is required that at least one section include partial inductance and capacitance matrices.

Capacitance Matrix Description

UNITS: FARAD

DESCRIPTION: A capacitance is an approximation of the capacitance in situations where the return path cannot be explicitly stated in the field extraction stages.

See "Multi-Line Capacitance Matrix" section above for more information on Multi-Line Matrices.

USAGE RULES: Two capacitance matrix methods are supported:

SLM, Single-Line, Capacitance Matrix (non-coupled)
MLM, Multi-Line, Maxwell Capacitance Matrix (coupled)

If ModelType is SLM then all sections must only contain a single capacitance value per line.

If ModelType is MLM then all sections must use a Maxwell Capacitance as stated below.

Single-Line Capacitance Matrix Description:

To define capacitance requires conductors separated by a dielectric. The conductors are those found in a loop consisting of a closed path including a signal and all possible return paths. The Maxwell Capacitance matrix relates the total charges on the conductors to the voltages applied to them. [Q]=[Cm][V] Where [Cm] is the Maxwell matrix, [Q] is the vector of total conductor charges, and [V] is a vector of applied voltages. The negative values of the off-diagonal entries can be explained on physical grounds: if you put +1 Volt on a single conductor and 0 Volts on the others, then you induce negative charges on those other conductors to terminate the field lines emanating from the conductor at 1 Volt.

In the Maxwell Capacitance matrix, the diagonal term is the maximum capacitance that the conductor encounters, meaning the capacitance to ground plus the capacitances to all the other conductors.

In the Maxwell Capacitance matrix, the OFF-diagonal term is the capacitance that the conductor encounters relative to a second conductor. This capacitance value is affected by other nearby conductors during the field extraction phase. It should not normally be used by itself to determine the capacitance of a random pair of conductors in a given structure.

Conductance Matrix Description

Units: Siemens (mhos)
Description: The Conductance maxtrix represents the energy transfer between the conductor and its surroundings, including the return path. Conductance includes dielectric and radiation losses, as well as any DC conductance.

Notes: The conductance matrix is optional. Simulation tools shall use 0.0 Siemens for all entries if this matrix is not present.

Do NOT use Siemens/meter as this is the total value and includes the length of this section.

Loss Matrix Description

This release supports S-Parameters for frequency dependent loss.

Uncoupled Models:

Single-Line-Models or SLM models are used for un-coupled models. These models are useful for most simulations where coupled effects can be ignored safely. These models are most accurate if closely spaced ground pins or a ground plane is present.

A "Diagonal_matrix" matrix type is the best choice for describing all matrices R, L, G, and C for SLM models. (see matrix descriptions below).

Several methods can be used:
Field solvers with return path pins defined. Only the diagonal matrix data is used. The coupling information is ignored. The R, L, G, C matrices should represent the electrical parameters for a specific section.

2D field solver with a specific return path or ground. Only the diagonal matrix data is needed. No coupling data is present in the final matrix

Closed form round wire inductance equation.

Experimental lab data may be obtained from TDR, VNA or other means

Coupled Models:

Multi-Line-Models or MLM models are used for coupled models.

A "Full_matrix" matrix type is the best choice for describing all matrices R, L, G, and C for MLMs. (see matrix descriptions below)

Several methods can be used:
3D field solver can be used to create partial inductance and capacitance matrices that allow accurate simulations without the need to pre-define the ground pins.

Field solvers without specific return path pins. A return plane May be specified as a reference only and is placed far field.

Experimental lab data may be obtained from multichannel measurements that incorporate TDR, VNAs or other means as long as

the resulting inductance matrix contains partial inductance values.

Common to Coupled and Uncoupled Models

One common aspect of all the different formats is that they exploit the symmetry of the matrices they describe. This means that the entries below the main diagonal of the matrix are identical to the corresponding entries above the main diagonal. Therefore, only roughly one-half of the matrix needs to be described. By convention, the main diagonal and the UPPER half of the matrix are provided.

In the following text, we use the notation [I, J] to refer to the entry in row I and column J of the matrix. Note that I and J are allowed to be alphanumeric strings as well as integers.

The ordering of these strings is determined by the row/column ordering convention given by the [Cn Row Swath] or [Cn Column Swath] convention, or else matches exactly the [Cn Pin Map] order when the pin map contains the same number of pins as the Swath.

In the following text, "Row 1", means the row corresponding to the first pin.

Also note that the numeric entries of the RLGC matrices are standard IBIS floating point numbers. As such, it is permissible to use metric "suffix" notation. Thus, an entry of the C matrix could be given as 1.23e-12 or as 1.23p or 1.23pF.

Each new row is identified with the Row keyword.

Keyword:	[Bandwidth]
Required:	Yes, if a Banded_matrix matrix is being defined.
Argument:	Numeric Value
Description:	Indicates the size of the width of the coupling matrix. This does not have anything in common with electrical bandwidth. The bandwidth field must be a nonnegative integer. This is the number of off-diagonal matrix rows for which data is provided.
Usage Rules:	MUST be used between keywords [Begin Cn Section] and [End Cn Section]. MUST follow any usage of one of the following keyword: Banded_matrix

---

See example "Matrix Example" lower in this document.

Keyword:	[Row]
Required:	Yes if a matrix is being defined.
Argument:	None
Description:	Indicates the beginning of a new row of the matrix.
Usage Rules:	The argument must be a sequential number beginning at 1 and continuing to the last row for the matrix Following a [Row] keyword is a block of numbers that represent the entries for that row. Suppose that the current row is number M. Then the first number listed is the diagonal entry, [M,M]. Following this number are the entries of the upper half of the matrix that belong to row M: [M, M+1], [M, M+2], ... up

to [M,N].

For even a modest-sized matrix, this data will not all fit on one line. You can break the data up with new-line characters so that this limit is observed.

Example: Suppose the matrix has 40 conductors and that we are currently working on Row 19. There is 1 diagonal entry, plus 40 - 19 = 21 entries in the upper half of the matrix to be specified, for 22 entries total.

Row 1 always has the most entries, and that each successive row has one fewer entry than the last; the last row always has just a single entry.

Usage Context:
MUST be used between keywords [Begin Cn Section] and [End Cn Section].

MUST follow any usage of one of the following keywords: Banded_matrix, Sparse_matrix, Full_matrix

---

See example "Matrix Example" lower in this document.

Keyword:	[Cn Sparameter]
Required:	No, unless Cn_Model_Type sub-parameter of [Begin Cn Model] is equal to “S-parameter”.
Sub-Params:	file_type, file_name, port_map
Description:	Indicates a reference to an external S-parameter file
Usage Rules:	Can be used in addition to or in place of place of LRCG matrix.

file_type: Touchstone

file_name
file_name is to reference the name of the CITI or Touchstone file. It is required that the icm and the referenced file be in the same directory. One [Cn Sparameter] keywords is allowed for each [Begin Cn Section] / [End Cn Section].

port_map
“port_map” is to be used to relate the ports in the S-parameter file to the “fixed” pin map in the icm file. It is important that the “port_map” uses the pin names from [Cn PinMap]. No more or no less entries in the table are allowed.

There are three columns used in the port_map. The first column is a reference to a “Pin Name” as defined in the [Cn PinMap] keyword. The second column should reference the input port name as defined in the Touchstone file that is related to the “Pin Name”. The third column should reference the output port name as defined in the Touchstone file that is related to the “Pin Name”.

Example

```
[Cn Sparameter]
file_type touchstone
file_name anyname.any_extension
port_map
|Pin    Port A  Port B
```

```
  A1      1       2
  A2      3       4
  A3      5       6
  A4      7       8
  B1      9       10
  B2      11      12
  B3      13      14
  B4      15      16
```

```
================================================================================
     Keyword:  [End Cn Section]
    Required:  Yes
    Argument:  Text String
 Description:  Indicates the end of [Begin Cn Section]
```

Matrix Examples

```
================================================================================
 The resistance matrix for this package does not have coupling

[Begin Cn Section]      ExampleMatrix00
[Derivation Method]     Distributed
[Resistance Matrix]     Diagonal_matrix
10.0
15.0
15.0
10.0
10.0
15.0
15.0
10.0
[End Cn Section] ExampleMatrix00

[Begin Cn Section]       ExampleMatrix01

 The resistance matrix for this package does not have coupling
 and is exactly the same as the Diagonal_matrix shown above
 and is described using a trivial version of the Banded_matrix
 as an example only.  The Diagonal_matrix should be used for
 this example to conserve file size.

[Resistance Matrix]     Banded_matrix
[Bandwidth]             0
[Row]   1
10.0
[Row]   2
15.0
[Row]   3
15.0
[Row]   4
10.0
[Row]   5
10.0
[Row]   6
15.0
[Row]   7
15.0
[Row]   8
10.0
```

```
 The inductance matrix has coupling between pins

[Inductance Matrix]     Full_matrix
[Row]   1
3.04859e-07      4.73185e-08      1.3428e-08     6.12191e-09
1.74022e-07      7.35469e-08     2.73201e-08     1.33807e-08
[Row]   2
3.04859e-07      4.73185e-08      1.3428e-08     7.35469e-08
1.74022e-07      7.35469e-08     2.73201e-08
[Row]   3
3.04859e-07      4.73185e-08     2.73201e-08     7.35469e-08
1.74022e-07      7.35469e-08
[Row]   4
3.04859e-07      1.33807e-08     2.73201e-08     7.35469e-08
1.74022e-07
[Row]   5
4.70049e-07      1.43791e-07     5.75805e-08     2.95088e-08
[Row]   6
4.70049e-07      1.43791e-07     5.75805e-08
[Row]   7
4.70049e-07      1.43791e-07
[Row]   8
4.70049e-07

 The capacitance matrix has sparse coupling

[Capacitance Matrix]    Sparse_matrix
[Row]   1
1       2.48227e-10
2       -1.56651e-11
5       -9.54158e-11
6       -7.15684e-12
[Row]   2
2       2.51798e-10
3       -1.56552e-11
5       -6.85199e-12
6        -9.0486e-11
7       -6.82003e-12
[Row]   3
3       2.51798e-10
4       -1.56651e-11
6       -6.82003e-12
7        -9.0486e-11
8       -6.85199e-12
[Row]   4
4       2.48227e-10
7       -7.15684e-12
8       -9.54158e-11
[Row]   5
5       1.73542e-10
6       -3.38247e-11
[Row]   6
6       1.86833e-10
7       -3.27226e-11
[Row]   7
7       1.86833e-10
8       -3.38247e-11
[Row]   8
8       1.73542e-10

[End Cn Section] ExampleMatrix01
```

==========

9.0 End Keyword

==========

```
     Keyword:  [End]
    Required:  Yes
    Argument:  None
 Description:  Specifies end of file
-------------------------------------------------------------------------------
[End]
```