**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 186.3 Draft 7

**ISSUE TITLE:** File Naming Rules

**REQUESTOR:**  Walter Katz, Signal Integrity Software, Inc.; Bob Ross, Teraspeed Labs

**DATE SUBMITTED:** November 29, 2016

**DATE REVISED:** February 16, 2017; April 26, 2017; TBD

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

IBIS AMI modeling and Interconnect modeling may result in a large number of supporting files. This BIRD removes the restriction that all files referenced by the .ibs file be restricted to the directory containing the IBIS file. Instead, files referenced by the .ibs file shall be either in the same directory as the .ibs file or in a directory structure below the directory containing the .ibs file.

In addition, this BIRD relaxes other constraints on file names, including allowing both upper and lower case letters and allowing longer file names.

The IBIS specification must meet these requirements:

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. Enable file names that are up to 256 characters in length. | IBIS in general will need to increase the number of characters permitted in each line. |
| 1. Enable files to be defined and located in a Hierarchical structure either in the same directory of the IBIS file or in a directory directly below it. | AMI supporting files and Interconnect model supporting files can be organized in a directory structure that makes it simple for IC vendors to organize and deliver their IBIS models to their customers. |
| 1. Enable file names with both lower and upper case characters. | IBIS should allow file naming rules similar to other modern tools and specifications. |
| 1. Enable file names to include a relative directory. |  |
| 1. Use consistent language to describe the parts of file names and paths. |  |

**SUMMARY OF PROPOSED CHANGES:**

For review purposes, the proposed changes are summarized as follows:

|  |  |  |
| --- | --- | --- |
| Specification Item | New/Modified/Other | Notes |
| Define terminology for the parts of file names in GENERAL SYNTAX RULES AND GUIDELINES. | New | Boost.org offers one source. Include an illustration. |
| Review 44 instances of “file name”, changing to “file” in some places and clarifying where relative paths are allowed. | Modified | Many places. |
| Change the file naming rules and IBIS File line length limits in GENERAL SYNTAX RULES AND GUIDELINES | Modified |  |

**PROPOSED CHANGES:**

*All page numbers refer to the IBIS version 6.1 Adobe PDF file.*

*On page 9 replace the colon at the end of the first sentence with a period.*

*On page 9 add the following subsection and additional subsection heading after the first sentence and ahead of the numbered list:*

## 3.1 FILE NAMING DEFINITIONS

The following terms related to file naming are defined here:

* **file name**: The name of a file without its location.
* **stem**: The portion of a file name before the last dot, or the full file name if no dot.
* **extension**: The portion of a file name after the last dot, if any.
* **directory**: A directory contains a list of files. Directories may include other directories, forming the basis for a hierarchical filesystem.
* **path**: A sequence of root directory (optional), directory elements and file name that identify the location of a file. A path may be absolute or relative.
* **absolute path**: A path that unambiguously identifies the location of a file without reference to an additional starting location.
* **relative path**: A path that is not absolute, and so only unambiguously identifies the location of a file when resolved relative to an implied starting location.
* **root name**: For operating systems supporting multiple filesystem roots, a name to identify the filesystem.
* **root directory**: A standard designation for the root of a filesystem.
* **file reference**: A reference to a file, expressed as either a simple file name or a relative path, which includes a simple file name.

Figure 1 shows an example of a file path with its parts delineated.



1. - Example of file naming definitions

## 3.2 SYNTAX RULES

*On page 9, replace:*

1. To facilitate portability between operating systems, file names used in a .ibs file must only have lower case characters. File names should have a basename of no more than forty (40) characters followed by a period (“.”), followed by a file name extension of no more than three characters. The file name and extension must use characters from the 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 \_ ^ $ ~ ! # % & - { } ) ( @ ‘ `

The file name and extension are recommended to be lower case on systems that support such names.

*with:*

1. File names should have a stem of no more than sixty (60) characters followed by a period (“.”), followed by a file name extension of no more than three characters. Files specified with a relative path should use no more than sixty-four (64) characters for the path name. The file name and extension must use characters from the 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 \_ ^ $ ~ ! # % & - { } ) ( @ ‘ `

The character sequence “./” is not permitted in any reference to a file, effectively restricting the naming of files to those in the same directory as the referring file or a subdirectory of that directory. Absolute paths, those beginning with a root name or root directory, are not permitted in any reference to a file.

*On page 9 replace:*

1. 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.

*with:*

1. A line of the file may have at most 1024 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.

*On page 10, replace:*

1. The use of tab characters is legal, but they should be avoided as much as possible. This is to eliminate possible complications that might arise in situations when tab characters are automatically converted to multiple spaces by text editing, file transferring and similar software. In cases like that, lines might become longer than 120 characters, which is illegal in .ibs files.

*with:*

1. The use of tab characters is legal, but they should be avoided as much as possible. This is to eliminate possible complications that might arise in situations when tab characters are automatically converted to multiple spaces by text editing, file transferring and similar software. In cases like that, lines might become longer than 1024 characters, which is illegal in .ibs files.

*On page 10, replace:*

14. Only ASCII characters, as defined in ANSI Standard X3.4-1986, may be used in IBIS file types. This includes files with file extensions .ibs, .pkg, .ebd, .ami and any other files used for passing parameter values.

*with:*

14. Only ASCII characters, as defined in ANSI Standard X3.4-1986, may be used in IBIS file types (including .ibs, .pkg, .ebd, .ami, .ims, and parameter value files) and in other un-compiled files referenced by IBIS.

*On page 11 renumber “3.1 KEYWORD HIERARCHY” to “3.3”.*

1. A line of the file may have at most 1024 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.

*On page 18, replace:*

*Keyword:* **[File Name]**

*Required:* Yes

*Description:* Specifies the name of the .ibs file.

*Usage Rules:* The file name must conform to the rules in paragraph 3 of Section 3, "GENERAL SYNTAX RULES AND GUIDELINES". In addition, the file name must use the extension “.ibs”, “.pkg”, or “.ebd”. The file name must be the actual name of the file.

*with:*

*Keyword:* **[File Name]**

*Required:* Yes

*Description:* Specifies the file name of the file containing this keyword

*Usage Rules:* The file name must conform to the rules in item 3 of Section 3.2, "SYNTAX RULES". In addition, the file name shall use the extension “ibs”, “pkg”, “ebd”, or ”ims”. The file name shall be the actual name of the file. The file name entry shall not include any path.

*On pages 212 and 213, replace:*

*Parameter:* **DLL\_ID**

*Required:* No, and illegal before AMI\_Version 6.0

*Direction:* Rx, Tx

*Descriptors*:

Usage: In

Type: String

Format: Value

Default: <string literal>

Description:<string>

*Definition:* The EDA tool is responsible for recognizing this parameter name and replacing the value declared in the .ami file with a string that contains a unique alphanumeric identifier. The algorithmic model is responsible for using DLL\_ID as the base name for any data files that the model creates, either for use as temporary storage or for recording output data. The use of DLL\_ID helps guarantee that multiple instances of the same model (or different models from the same vendor) do not mix up data as a result of collisions between temporary or permanent file names.

*with:*

*Parameter:* **DLL\_ID**

*Required:* No, and illegal before AMI\_Version 6.0

*Direction:* Rx, Tx

*Descriptors*:

Usage: In

Type: String

Format: Value

Default: <string literal>

Description:<string>

*Definition:* The EDA tool is responsible for recognizing this parameter name and replacing the value declared in the .ami file with a string that must conform to the rules in item 3 of Section 3.2, "SYNTAX RULES". The algorithmic model is responsible for using the DLL\_ID string as part of the **base name**???? for any data files that the model creates, either for use as temporary storage or for recording output data. The use of DLL\_ID helps guarantee that multiple instances of the same model (or different models from the same vendor) do not mix up data as a result of collisions between temporary or permanent file names.

*On pages 99 and 120, replace:*

Corner:

Three entries follow the Corner subparameter on each line:

corner\_name file\_name circuit\_name

The corner\_name entry is “Typ”, “Min”, or “Max”. The file\_name entry points to the referenced file in the same directory as the .ibs file.

*---*

No character limits, case-sensitivity limits or extension conventions are required or enforced for file\_name and circuit\_name entries. However, the total number of characters in each Corner line must comply with the rules in Section 3. Furthermore, lower-case file\_name entries are recommended to avoid possible conflicts with file naming conventions under different operating systems. Case differences between otherwise identical file\_name entries or circuit\_name entries should be avoided. External languages may not support case-sensitive distinctions.

*with:*

Corner:

Three entries follow the Corner subparameter on each line:

corner\_name file circuit\_name

The corner name entry is “Typ”, “Min”, or “Max”. The file entry points to the referenced file in the same directory as the .ibs file or in a relative path under this directory.

--

No character limits, case-sensitivity limits or extension conventions are required or enforced for file and circuit\_name entries. However, the total number of characters in each Corner line must comply with the rules in Section 3, “GENERAL SYNTAX RULES AND GUIDELINES”. Furthermore, lower-case file entries are recommended to avoid possible conflicts with file naming conventions under different operating systems. Case differences between otherwise identical file entries or circuit\_name entries should be avoided. External languages may not support case-sensitive distinctions.

*On pages 100, 101, 120, and 122, change:*

The reference must begin with a file name, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses.

*----*

The files referenced must be located in the same directory as the .ibs file containing the reference.

*to:*

The reference shall begin with a file name or relative path, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses.

*---*

The files referenced shall be located in the same directory as the .ibs file or in a relative path under this directory.

COMMENT – THE REFERENCE DIRECTORY FOR BOTH THE EXTERNAL MODEL AND ITS PARAMETERS ARE THE DIRECTORY FOR THE .IBS FILE. THE PARAMETERS ARE NOT RELATIVE TO THE DIRECTORY IN WHICH THE EXERNAL MODEL IS LOCATED. – DELETE THIS COMMENT

*On pages 100, 101-102, and 121-122, change (delete dot (.) for correct extension definition):*

The following rules apply to parameter trees located in parameter definition files whose file name extension is not “.ami”.

1. The parameter tree must not contain the Reserved\_Parameters branch.
2. The parameter tree must contain the Model Specific branch.
3. The parameter tree may only contain Usage Info parameters.

The following rules must be observed when [External Model] parameters or converter parameters reference parameters located in external parameter definition files.

1. Usage Info parameters may be referenced in any external parameter definition file with or without the “.ami” extension.
2. Usage In parameters may be referenced in any parameter definition file whose file name extension is “.ami”.
3. Usage Dep parameters may also be referenced in an AMI parameter definition file under the following conditions:

* the [External Model] keyword is located under a [Model] keyword which also contains an [Algorithmic Model] keyword,
* the [External Model]'s parameter and the [Algorithmic Model] keyword point to the same “.ami” file,
* the AMI parameter definition file contains the parameter AMI\_Resolve\_Exists with a value of True.

If all of these conditions are satisfied, the EDA tool must execute the AMI\_Resolve function in the executable model defined by the [Algorithmic Model] keyword to resolve the value of any Usage Dep parameter before passing its value to the [External Model] (see Section 10.2.3).

*to:*

The following rules apply to parameter trees located in parameter definition files whose file name extension is not “ami”.

1. The parameter tree must not contain the Reserved\_Parameters branch.
2. The parameter tree must contain the Model Specific branch.
3. The parameter tree may only contain Usage Info parameters.

The following rules must be observed when [External Model] parameters or converter parameters reference parameters located in external parameter definition files.

1. Usage Info parameters may be referenced in any external parameter definition file with or without the “ami” extension.
2. Usage In parameters may be referenced in any parameter definition file whose file name extension is “ami”.
3. Usage Dep parameters may also be referenced in an AMI parameter definition file under the following conditions:

* the [External Model] keyword is located under a [Model] keyword which also contains an [Algorithmic Model] keyword,
* the [External Model]'s parameter and the [Algorithmic Model] keyword point to the same “ami” file,
* the AMI parameter definition file contains the parameter AMI\_Resolve\_Exists with a value of True.

If all of these conditions are satisfied, the EDA tool shall execute the AMI\_Resolve function in the executable model defined by the [Algorithmic Model] keyword to resolve the value of any Usage Dep parameter before passing its value to the [External Model] (see Section 10.2.3).

---

EXAMPLE CHANGES (file\_name to file)

*On pages 125-129*

*Change* file\_name to file:

FOR IBIS-ISS and SPICE

| Corner corner\_name file\_name circuit\_name (.subcircuit name)

To

| Corner corner\_name file circuit\_name (.subcircuit name)

OR for VHDL-AMS

| Corner corner\_name file\_name entity(architecture)

To

| Corner corner\_name file entity(architecture)

OR for Verilog-AMS

| Corner corner\_name file\_name circuit\_name (module)

To

| Corner corner\_name file circuit\_name (module)

*On page 139, change:*

Note that the actual package models can be in a separate <package\_file\_name>.pkg file or can exist in the .ibs files between the [Define Package Model] ... [End Package Model] keywords for each package model that is defined.

*to:*

Note that the actual package models can be in a separate <stem>.pkg file or can exist in the .ibs files between the [Define Package Model] ... [End Package Model] keywords for each package model that is defined.

Page 140:

The <filename> provided must adhere to the rules given in Section **Error! Reference source not found.**, "GENERAL SYNTAX RULES AND GUIDELINES". Use the “.pkg” extension to identify files containing package models.

*to:*

The <stem> provided shall adhere to the rules given for the [File Name] keyword. Use the “pkg” extension to identify files containing package models.

*On page 154, change:*

A .ebd file is intended to be a stand-alone file, not referenced by or included in any .ibs or .pkg file. Electrical Board Descriptions are stored in a file whose name looks like <filename>.ebd, where <filename> must conform to the naming rules given in Section **Error! Reference source not found.** of this specification. The .ebd extension is mandatory.

*to:*

A .ebd file is intended to be a stand-alone file, not referenced by or included in any .ibs or .pkg file. Electrical Board Descriptions are stored in a file whose name looks like <stem>.ebd, where <stem> shall conform to the naming rules given in for the [File Name] keyword. The ebd extension is mandatory.

*On page 163, change (for* [Reference Designator Map]*)*:

By default, the .ibs or .ebd files are assumed to exist in the same directory as the calling .ebd file. It is legal for a reference designator to point to a component that is contained in the calling .ebd file.

---

| Ref Des File name Component name

*to:*

The referenced .ibs or .ebd files can exist in the same directory as the calling .ebd file or in a relative path under this directory.

---

| Ref Des File Component name

*On page 171, 172, change:*

Platform\_Compiler\_Bits File\_Name Parameter\_File

---

The File\_Name provides the name of the executable model file. The executable model file should be in the same directory as the .ibs file.

The Parameter\_File entry provides the name of the AMI parameter definition file, which shall have an extension of .ami. This must be an external file and should reside in the same directory as the .ibs file and the executable model file. See Section **Error! Reference source not found.** for details.

*to:*

Platform\_Compiler\_Bits Executable\_Model\_File AMI\_Parameter\_File

The Executable\_Model\_File provides the name of the executable model file. The executable model file shall reside in the same directory as the .ibs file or in a relative path under this directory. See Section 10.2 for details.

The AMI\_Parameter\_File entry provides the name of the AMI parameter definition file, which shall have an extension of ami. This must be an external file. The .ami file and the executable model files can reside in the same directory as the .ibs file or in a relative path under this directory. See Section 10.3 for details

*On page 210, correct file name to root name:*

(mySampleAMI | AMI parameter definition file name

*to*

(mySampleAMI | AMI parameter definition root name

*On page 211, change by deleting last sentence:*

The file names or directory names may be written with or without a path, but in either case, they must be expressed relative to the location of the .ami file in which the Supporting\_Files parameter is found~~. (The AMI executable models and the AMI parameter definition files are all required to be in the same directory as the .ibs file in which they are declared).~~

**BACKGROUND INFORMATION/HISTORY:**

Five drafts of this BIRD were discussed and revised in the Advanced Technology Modeling Task group. The group voted to submit the BIRD to the IBIS Open Forum November 29, 2016.

BIRD86.1:

1. The language in Version 6.1 states that all files must reside in the same directory. The statements are intentional and therefore need to be corrected in all locations.
2. Another change is to correctly list extensions WITHOUT the dot – as defined by “dot” extension. This issue has and inconsistency has been in all versions of IBIS.
3. Anticipate BIRD189 approval and add the .ims file type for Interconnect Model Set files.
4. Clarify that the extension after the “last dot” in a file name shall not contain a “/”.
5. Font size for Times New Roman set to 12.

BIRD186.2

[File Name] Description is changed. [File Name]: Usage Rules: may and must 🡪 shall.

BIRD186.3

Font corrections in four places: Courier to Times New Roman

Package Modeling Section 7: <package\_file\_name>.pkg and <filename>.pkg 🡪<base name>.pkg

Electrical Board Description Section 8: <filename>.ebd 🡪 <base name>.ebd

Note, “file name extension” is not changed because it describes a portion that is defined in a file name.

BIRD186.3

Boost.org terminology adopted plus some additions using file instead of file\_name for example columm headings. <base name> 🡪 <stem>, relative path used to describe files in directories under the referencing file.