**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 186.1

**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

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

IBIS AMI modeling, and Interconnect Modeling may result in a large number of supporting files. This BIRD replaces the restriction that all files referenced by the .ibs file be restricted to the directory containing the IBIS file, to 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 such as 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 its rule on the number of characters 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 similar file naming rules as modern day tools and specifications. |
| 1. Enable file names do define a directory
 |  |
|  |  |
|  |  |

**SUMMARY OF PROPOSED CHANGES:**

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

|  |  |  |
| --- | --- | --- |
| Specification Item | New/Modified/Other | Notes |
| Change the file naming rules in paragraph 3 and the IBIS File line length limits in Paragraph 4 of the GENERAL SYNTAX RULES AND GUIDELINES |  |  |

**PROPOSED CHANGES:**

Replace the following two paragraphs on page 9 of the IBIS specification

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.

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.

To:

3.   File names should be no more than two hundred and fifty six (256) characters. The file name 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

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 \_ ^ $ ~ ! # % & - { } ) ( @ ‘ ` . /

Note that files names in the Windows operating system are case insensitive and case preserving, while file names in Linux operating systems are case sensitive. When a [computer](https://en.wikipedia.org/wiki/Computer%22%20%5Co%20%22Computer) [file system](https://en.wikipedia.org/wiki/File_system%22%20%5Co%20%22File%20system) stores file names, the computer may keep or discard [case](https://en.wikipedia.org/wiki/Letter_case%22%20%5Co%20%22Letter%20case) information. When the case is stored, it is called **case preservation**. File names in IBIS files should be case sensitive so that the IBIS file and the files it references will work properly on both Windows and Linux operating systems. The “/” character is used to delineate directories in a file name. The character sequence “../” is not permitted, except that it is permitted if generated by the EDA tool. Absolute or relative paths can be generated by the EDA tool in applications as needed. The EDA tool is responsible for making any operating system-specific adjustments (for example, replacing forward slashes "/" with backslashes "\") if necessary.

The characters after the last “.”  are considered the file name extension, where the extension shall not contain a “/”. There are places in this document that specify the file name extension for specific files types (e.g. .ibs, .pkg, and .ami). These extensions are case sensitive.

“file name” shall mean the name of the file, including the path relative to the directory containing the file that references the “file name”.  A “file name” may not be a directory.

Directory (path) names follow the same rules as file names, except that they shall be the name of a directory.

Example file names defined inside of IBIS file xyz.ibs:

xyz/ami/Tx/Tx.ami

xyz/ami/Tx/Tx.dll

xyz/interconnect/xyz\_uncoupled/xyz\_uncoupled.ims

xyz\_coupled xyz/interconnect/xyz\_coupled/xyz\_coupled.ims

Example file names defined inside of file xyz/interconnect/xyz\_uncoupled/xyz\_uncoupled.ims

File\_TS snp/DQ1.s2p

If xyz.ibs is in the directory IBIS, then these files and directories will be in the following directories:

IBIS

xyz.ibs

xyz

IBIS/xyz

ami

interconnect

IBIS/xyz/ami

Tx.ami

Tx.dll

IBIS/xyz/interconnect

xyz\_uncoupled

xyz\_coupled

IBIS/xyz/interconnect/xyz\_uncoupled

xyx\_uncoupled.ims

snp

IBIS/xyz/interconnect/xyz\_uncoupled/snp

DQ1.s2p

IBIS/xyz/interconnect/xyz\_coupled

xyx\_coupled.ims

Note that when a “file name” is referenced inside of a “source file” the location of that “file name” shall always be relative to the location of the “source file”.

The space character remains an illegal file name character. When an IBIS file is delivered all supporting files shall either be in the same directory as the IBIS file, or in directories below this directory. Absolute files names (e.g. that begin with // or C:) are not permitted. Users and EDA tools may choose to move supporting files into other directories that are not located below the location of the “source file”. These files may be put on a tool specific search path or have added symbolic links. The IBIS parser may report such non-standard file names as either errors or warnings.

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 should be avoided.

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.

Change to  *Keyword:* **[File Name]**

Currently

*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.

Change to

*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”, “ebd”, or “ims”.
* The file name must be the actual name of the file.
* There may not be a “/” in the file name.

Change  *Parameter:* **DLL\_ID**

From

*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.

To:

*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 paragraph 3 of Section 3, "GENERAL SYNTAX RULES AND GUIDELINES. 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.

------

Update the language in the main document to explicitly allow directories in the file name and to deal properly with extensions (changes or highlights in red, and underline means the area and possibly the description of the change):

Section 3(14)

Change

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.

[Generalize the last sentence because the rule applies throughout IBIS for any file type such as .s2p, .ts, .iss, .spi, .iss, .abc .cir, .v, .vhl, .dll, .so, .xyz, etc. and any other file referenced by IBIS). We may not check the contents (only file names and exposed nodes, the rule should still apply to all files used in IBIS.]

To:

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 files referenced by IBIS.

Section 6.3 – Multi-lingual – “Same directory clarifications problem in red

Change (pg. 99, 120)

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.

To (pg. 99, 120)

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 or in a directory under the .ibs file as determined by the directory path.

Change (pg. 100, 101, 120, 122)

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

To (pg. 100, 101, 120, 122)

The files referenced must be located in the same directory as the .ibs file or in a directory under the .ibs file as determined by the directory path.

Section 6.3 – Multi-lingual Parameter Passing – Remove the “.” In the extensions shown in red (pages 100, 101-102, 121-122)

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).

Also pages 100, 122 relate to the “.”. Also, filename only or filename with dot are permitted:

name, and a matching set of closing parentheses. Spaces are allowed in the reference following the file name. The file reference may point to any file which contains one or more parameter trees. The files referenced must be located in the same directory as the .ibs file containing the reference. The file names of parameter definition files must follow the rules for file names given in Section 3, “GENERAL SYNTAX RULES AND GUIDELINES”. In addition, files with no extensions (e.g, xyz) or with just a dot (e.g., xyz.) are permitted. IBIS file formats except .ami (e.g., .ibs, .pkg, and .ebd) do not contain parameter trees and are not permitted as parameter definition files. Parameter definition files may only contain parameter trees using the tree syntax described in IBIS in Section 10.3 with the following exceptions and additions:

Change:

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

To

The files referenced must be located in the same directory as the .ibs file containing the reference or in a directory under the .ibs file as determined by the directory path.

---

Section 7, Define Package Model page 139:

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. .

Suggested change:

Note that the actual package models can be in a separate <filename>.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 3, "GENERAL SYNTAX RULES AND GUIDELINES". Use the “.pkg” extension to identify files containing package models.

Suggested change:

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

Section 8: Electrical Board Description

Page 154:

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 3 of this specification. The .ebd extension is mandatory.

Suggest changing 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 <filename>.ebd, where <filename> must conform to the naming rules given in the [File Name] keyword. The ebd extension is mandatory.

Under [Reference Designator Map], pg. 163

Change:

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.

To

The referenced .ibs or .ebd files can exist in the same directory as the calling .ebd file or in a directory under the .ebd file as determined by the directory path. ~~It is legal for a reference designator to point to a component that is contained in the calling .ebd file.~~

Section 10.1 under [Algorithmic Model], page 172 (remove the “.”) and remove the stated same-directory restriction:

Change

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 10.3 for details.

To:

The File\_Name provides the name of the executable model file. The executable model file shall be in the same directory as the .ibs file or in a directory below the .ibs file as determined by the directory path.

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. The .ami file and the executable model files can reside in the same directory as the .ibs file or in a directory below the .ibs file as determined by the directory path. See Section 10.3 concerning the Parameter\_File details.

Section 10.4 under Supporting\_Files, page 211 (remove the sentence shown):

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.