**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**ISSUE TITLE:** *IBIS-AMI New Reserved Parameters for Data Management*

**REQUESTOR:**  *Walter Katz, Mike Steinberger, Todd Westerhoff, SiSoft*

**DATE SUBMITTED:** *October 20, 2010*

DATE REVISED: *June 1, 2011; November 20, 2012*

DATE ACCEPTED BY THE OPEN FORUM: *January 11, 2013*

**ANALYSIS PATH/DATA THAT LED TO SPECIFICATION:**

Model developers and EDA vendors building IBIS-AMI models using the IBIS 5.0 specification have come across a number of modeling issues that are not addressed in IBIS 5.0. In order to deliver models and EDA tools that meet end-user demands for model accuracy and functionality, EDA vendors have defined "extensions" to add new capabilities to IBIS-AMI models. Unfortunately, EDA vendors have had to use proprietary (and different) syntax to add these capabilities to models, limiting model portability between different EDA tools.

This BIRD proposes new syntax for the .ami control file that improves model functionality and accuracy. Including this syntax in the IBIS standard will allow creation of accurate, compliant IBIS-AMI models that are readily portable between commercial EDA simulators.

The parameters defined in this document are to be added in Section ?? of the IBIS

5.1 specification as new Reserved\_Parameters:

Data Management & Simulation Control

Supporting\_Files, DLL\_Path, DLL\_ID

**ANY OTHER BACKGROUND INFORMATION:**

This BIRD is being requested by the following IBIS users and model developers, in conjunction with the authors:

Cisco Systems: Upen Reddy, Doug White

Ericsson: Anders Ekholm

Broadcom: Yunong Gan

IBM: Adge Hawes

TI: Alfred Chong, Srikanth Sundaram

## Parameter DEFINITIONs

This section defines the structure and parameters used with required and optional functions.

*Parameter:* **Supporting\_Files**

*Required:* No

*Descriptors*:

Usage: Info

Type: String

Format: Table

Default: illegal

Description:<string literal>

*Definition:* **Supporting\_Files** contains strings of file names and/or directory names to point to files and/or directories which are used by the IBIS-AMI executable model directly or by the EDA tool (for example to generate the channel impulse response) to function properly. **Supporting\_Files** is organized as a table containing a single column and one or more rows, in which each file name or directory name entry must be placed into a separate row. 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 parameter files are all required to be in the same directory as the .ibs file in which they are declared). Path separators in the entires of **Supporting\_Files** must be forward slashes "/". Back slashes “\” are not allowed. The EDA tool is responsible for making any OS-specific adjustments (for example, replacing forward slashes "/" with backslashes "\") if necessary. The last character of this string shall not be a forward slash “/”. A **Supporting\_Files** entry may not be an empty string “”, or a string containing a period alone “.”.

*Usage Rules:* The purpose of the **Supporting\_Files** parameter is to enumerate all of the supporting files of an AMI model. This is important in situations when the EDA tool needs to know about the supporting files of an AMI model, for example to copy the original model files into its own simulation model library. For this reason, all supporting files of an AMI model must be listed in the **Supporting\_Files** parameter, either using individual file names, or using directory names. When directory names are used in this parameter, it is implied that all of the files and subdirectories in that directory are needed by the AMI model. A file definition is legal but redundant if the directory in which it is located is also defined in a **Supporting\_Files** entry.

*Other Notes:* The EDA tool is not expected to make wildcard expansions (globbing) for any characters in the string.

*Example:*

(Supporting\_Files (Usage Info)(Type String)

(Description

"Additional files and directories required by this model")

(Table

("my\_stuff\_dir")

("my\_deeper\_stuff\_dir/here")

("m1.s4p")

("my\_special\_dir/m2.s4p")

)

)

*Parameter:* **DLL\_Path**

*Required:* No

*Descriptors*:

Usage: In

Type: String

Format: Value

Default: <string literal>

Description:<string literal>

*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 the path to the directory where the DLL and .ami files reside. The Value specified in the .ami file shall be ignored by the EDA tool. The value of DLL\_Path passed to the DLL can either be an absolute path, or a path relative to the current working directory of the process running the DLL. In this string, the path separator is the forward slash "/". Back slashes “\” are not allowed. The model is responsible for making any OS-specific adjustments (for example, replacing forward slashes "/" with backslashes "\") if necessary.

The last character of the value passed to the DLL shall not be a forward slash “/”. To access a supporting file, the DLL should create a file name by creating a string consisting of the value of the DLL Path, convert forward slashes “/” to backslashes “\” on operating systems that require a backslash “\” as a path separator, append a forward slash “/” or backslash “\” as appropriate to the operating systems, and then append the name of the file. If the EDA tool choses to pass a relative path and if the current working directory (CWD) is where the DLL resides then DLL\_Path should be a period “.”.

*Usage Rules:*

*Other Notes:* A DLL should not rely on the current working directory (CWD) set by the EDA tool or simulator to determine the locations of files. If DLL\_Path is a relative path name then the DLL shall assume that it is a relative path from the CWD, and the EDA tool is responsible for setting the CWD to ensure that the relative DLL\_Path is correct. The DLL shall not change the CWD. The EDA tool is not expected to make wildcard expansions (globbing) for any characters in the string.

*Example:*

(DLL\_Path (Usage In)(Type String)(Value "placeholder")

(Description "Path to where the DLL is located"))

*Parameter:* **DLL\_ID**

*Required:* No

*Descriptors*:

Usage: In

Type: String

Format: Value

Default: <string literal>

Description:<string literal>

*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.

*Usage Rules:*

*Other Notes:*

*Example:*

DLL\_ID (Usage In)(Type String)(Value "placeholder")

(Description "Unique base name for each AMI model instance and run"))