**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**Draft 7**

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

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

**DATE SUBMITTED:** *September 24, 2012*

**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:* Contains a list of the files and directories that the model requires in addition to the DLL or shared object file. The table shall contain one column and one or more rows. The string in each row is the relative path from the .ibs file directory to one supporting file or directory. The IBIS 5.1 specification already requires that the DLL and .ami file reside in the same directory as the .ibs file. Files and directories required by the DLL shall reside in this same directory. When copying a .ibs file to a project or library directory, the EDA tool or library utility should also copy the DLL and .ami files referenced by the .ibs file, and the supporting files and directories specified in **Supporting Files**.

*Usage Rules:*

*Other Notes:*

*Examples:*

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

 (Description

 "Additional files and directories required by this model")

 (Table

 ( "my\_stuff\_dir" )

 ( "m1.s4p" )

 ( "m2.s4p" )

 ( "m3.s4p" )

 )

)

*Parameter:* **DLL\_Path**

*Required:* No

*Descriptors*:

Usage: In

Type: String

Format: Value

Default: illegal

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 path to the directory where the DLL and .ami file reside. In this string, the path separator is the forward slash "/" . “\” are not allowed in the string. The model is responsible for making any OS-specific adjustments (for example, replacing forward slashes "/" with backslashes "\" if necessary). 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 simulation.

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 “/” to “\” on operating systems that require a “\” as a path delimiter, append a “/” or “\” as appropriate to the operating systems, and then append the name of the file. If the EDA tool chose to pass a relative path and if the CWD is where the DLL resides then DLL\_Path should be “.”.

*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 is 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.

*Examples:*

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

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

*Parameter:* **DLL\_ID**

*Required:* No

*Descriptors*:

Usage: In

Type: String

Format: Value

Default: illegal

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:*

*Examples:*

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

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