

Update on IBISCHK6 V6.1.3 and Executable Model File Checking

Bob Ross, Teraspeed Labs
bob@teraspeedlabs.com

DesignCon IBIS Summit
Santa Clara, California
February 3, 2017

(Updated from November 8, 11, 18, 2016 presentations)



Goals and Contents

- **ibischk6 V6.1.3 executables and User Guide uploaded:**
www.ibis.org/ibischk6/
- **Fixes 7 bugs**
- **Differences (new ibischk6 file names)**
- **New – executable model file checking per BUG179 enhancement for [Algorithmic Model] Executable lines**
- **Source code quality assurance tests**
- **Some limitations**



New BUGs 174-180 Fixed

www.ibis.com/bugs/ibischk/

180	Error with Legal List Tip in Reserved Parameters with Format List	Mike LaBonte, Signal Integrity Software (SiSoft)
179	DLL SO Checking and Functional Existence and Platform Information	Mike LaBonte, SiSoft; Bob Ross, Teraspeed Labs; Lance Wang, IO Methodology
178	Error with Same Platform Compiler Bits for Rx and Tx of I/O* in IBIS-AMI	Michael Schaefer, Zuken; Bob Ross, Teraspeed Labs
177	Empty [Node Declarations] Stops Parser Completion	Arpad Muranyi, Mentor Graphics; and Bob Ross, Teraspeed Labs
176	[External Model] Error Not Issued for Ports List With Undeclared Port	Arpad Muranyi, Mentor Graphics; and Bob Ross, Teraspeed Labs
175	Incorrect Model References Through [Model Selector] Not Reported	Walter Katz, SiSoft
174	File Not Found Line Printed Under Some Operating Systems	Mike LaBonte, SiSoft and Bob Ross, Teraspeed Labs



New Names for ibischk6

- **Command line operation:**
- **Windows:**
 - **ibischk6_32.exe** (versus **ibischk6.exe**)
 - **ibischk6_64.exe** (versus **ibischk6.exe**)
- **Linux (Ubuntu):**
 - **ibischk6_32** (versus **ibischk6**)
 - **ibischk6_64** (versus **ibischk6**)
- **Used for source code quality assurance tests per BUG179**
- **Macintosh osx_32, osx_64 (also uploaded), but not part of source code release**



New Command Line Based on ibischk6 Name

Example: `ibischk6_32.exe` → "`ibischk6_32`" command

IBISCHK6 V6.1.3

IBISfile validation:

This program has been provided free to the electrical engineering community by the IBIS Open Forum. The purpose of this program is to validate that the contents of ASCII device data in a file specified conform to the IBIS specification.

Usage: `ibischk6_32` <IBS filename>
: `ibischk6_32` -ebd <EBD filename>
: `ibischk6_32` -pkg <PKG filename>
: `ibischk6_32` -ami <AMI filename>
Usage: `ibischk6_32` -caution -numbered <IBS filename>
: `ibischk6_32` -caution -numbered -ebd <EBD filename>
: `ibischk6_32` -caution -numbered -pkg <PKG filename>
: `ibischk6_32` -caution -numbered -ami <AMI filename>

The flags prior to the file name can be in any order, and the -caution and/or -numbered flags are optional.



Can Rename ibischk6 File

(Examples Below)

- **ibischk6.exe** as in earlier versions for “**ibischk6**” command
- **ibischk613_32.exe** showing version and bits detail for “**ibischk613_32**” command
- Etc. - the new ibischk6 file name shows up in the Usage: lines, as shown on previous slide



Syntax and Notation

Related to BUG179

- Under [Algorithmic Model] Executable, Executable_Rx and Executable_Tx lines have this information:

Platform_Compiler_Bits File_Name Parameter_File

- Platform is operating system such as Linux, Windows
- Bits is 32 or 64 for common platforms
- File_Name is “executable model file” name such as abc.dll or abc.so
- Parameter_File is the .ami file



Testing and Executable Combinations

- 32-bit and 64-bit operating system, ibischk6_* and executable model files are done consistent with the platform operating system
 - ibischk6_32 will work on 64-bit platforms, but will load and test 32 bit executable model files



bug179-1.ami

Full Function Existence Checking

(bug179

 (Reserved_Parameters

 (AMI_Version (Usage Info) (Type String) (Value "6.1"))

 (Init_Returns_Impulse (Usage Info) (Type Boolean) (Value True))

 (GetWave_Exists (Usage Info) (Type Boolean) (Value True))

 (Resolve_Exists (Usage Info) (Type Boolean) (Value True))

)

)



Function Existence Testing

Each DLL/SO file must contain code symbols for exported functions in one of four possible combinations:

- A. Case 1: Executable model file has AMI_Init, AMI_GetWave and AMI_Close. (IBIS 5.0 and above)
- B. Case 2: Executable model file has AMI_Init and AMI_Close. (IBIS 5.0 and above)
- C. Case 3: Executable model file has AMI_Resolve, AMI_Resolve_Close, AMI_Init, AMI_GetWave and AMI_Close. (IBIS 6.0 and above)
- D. Case 4: Executable model file has AMI_Resolve, AMI_Resolve_Close, AMI_Init and AMI_Close. (IBIS 6.0 and above)

Test file names:

: noicgr No “C” combinations above and below

A: icg AMI_Init, AMI_Close, AMI_GetWave

B: ic AMI_Init, AMI_Close

C: icgr AMI_Init, AMI_Close, AMI_GetWave, AMI_Resolve, AMI_Resolve_Close

D: icr AMI_Init, AMI_Close, AMI_Resolve, AMI_Resolve_Close



Some Tests for ibischk6_32.exe when Checked with bug179-1.ami

| Normal case: GetWave and ResolveExists return true

|

Executable	Windows_1_32	noicgr_32.dll	bug179-1.ami	← 5 Errors
Executable	Windows_2_32	ic_32.dll	bug179-1.ami	← 3 Errors
Executable	Windows_3_32	icg_32.dll	bug179-1.ami	← 2 Errors
Executable	Windows_4_32	icgr_32.dll	bug179-1.ami	← Good
Executable	Windows_5_32	icr_32.dll	bug179-1.ami	← 1 Error

Executable	Windows_6_64	noicgr_64.dll	bug179-1.ami	← Executable lines for 64-bit Windows lines not checked
Executable	Windows_7_64	ic_64.dll	bug179-1.ami	
Executable	Windows_8_64	icg_64.dll	bug179-1.ami	
Executable	Windows_9_64	icgr_64.dll	bug179-1.ami	
Executable	Windows_10_64	icr_64.dll	bug179-1.ami	

|

Lines 112 to 121



Partial Report for ibischk6_32.exe with New Error Codes

E4702 (line 112) - Code file noicgr_32.dll does not contain required AMI_Init() function
E4703 (line 112) - Code file noicgr_32.dll does not contain required AMI_Close() function
E4704 (line 112) - Code file noicgr_32.dll does not contain AMI_GetWave() function, required because GetWave_Exists=True in AMI file bug179-1.ami
E4705 (line 112) - Code file noicgr_32.dll does not contain AMI_Resolve() function, required because Resolve_Exists=True in AMI file bug179-1.ami
E4706 (line 112) - Code file noicgr_32.dll does not contain AMI_Resolve_Close() function, required because Resolve_Exists=True in AMI file bug179-1.ami

E4704 (line 113) - Code file ic_32.dll does not contain AMI_GetWave() function, required because GetWave_Exists=True in AMI file bug179-1.ami
E4705 (line 113) - Code file ic_32.dll does not contain AMI_Resolve() function, required because Resolve_Exists=True in AMI file bug179-1.ami
E4706 (line 113) - Code file ic_32.dll does not contain AMI_Resolve_Close() function, required because Resolve_Exists=True in AMI file bug179-1.ami

E4705 (line 114) - Code file icg_32.dll does not contain AMI_Resolve() function, required because Resolve_Exists=True in AMI file bug179-1.ami
E4706 (line 114) - Code file icg_32.dll does not contain AMI_Resolve_Close() function, required because Resolve_Exists=True in AMI file bug179-1.ami

E4704 (line 116) - Code file icr_32.dll does not contain AMI_GetWave() function, required because GetWave_Exists=True in AMI file bug179-1.ami

Line 115 is Good (0 Errors):

Executable Windows_4_32 icgr_32.dll bug179-1.ami



Summary for Platforms/Bits Test for Full Test Case

... Status of [Algorithmic Model] Executables for Windows 32:

icr_64.dll:	Windows 64:	Not Checked
icgr_64.dll:	Windows 64:	Not Checked
icg_64.dll:	Windows 64:	Not Checked
ic_64.dll:	Windows 64:	Not Checked
icr_32.dll:	Windows 32:	Checked
icgr_32.dll:	Windows 32:	Checked
icg_32.dll:	Windows 32:	Checked
ic_32.dll:	Windows 32:	Checked
noicgr_64.dll:	Windows 64:	Not Checked
icr_32.dll:	Windows 64:	Not Checked
icgr_32.dll:	Windows 64:	Not Checked
icg_32.dll:	Windows 64:	Not Checked
ic_32.dll:	Windows 64:	Not Checked
noicgr_32.dll:	Windows 64:	Not Checked
icr_64.dll:	Windows 32:	Checked, has platform issue
icgr_64.dll:	Windows 32:	Checked, has platform issue
icg_64.dll:	Windows 32:	Checked, has platform issue
ic_64.dll:	Windows 32:	Checked, has platform issue
noicgr_64.dll:	Windows 32:	Checked, has platform issue
noicgr_32.dll:	Windows 32:	Checked



Summary for Platforms/Bits Test for Full Test Case (Continued)

icr_64.so:	Linux 64:	Not Checked
icgr_64.so:	Linux 64:	Not Checked
icg_64.so:	Linux 64:	Not Checked
ic_64.so:	Linux 64:	Not Checked
icr_32.so:	Linux 32:	Not Checked
icgr_32.so:	Linux 32:	Not Checked
icg_32.so:	Linux 32:	Not Checked
ic_32.so:	Linux 32:	Not Checked
noicgr_64.so:	Linux 64:	Not Checked
icr_32.so:	Linux 64:	Not Checked
icgr_32.so:	Linux 64:	Not Checked
icg_32.so:	Linux 64:	Not Checked
ic_32.so:	Linux 64:	Not Checked
noicgr_32.so:	Linux 64:	Not Checked
icr_64.so:	Linux 32:	Not Checked
icgr_64.so:	Linux 32:	Not Checked
icg_64.so:	Linux 32:	Not Checked
ic_64.so:	Linux 32:	Not Checked
noicgr_64.so:	Linux 32:	Not Checked
noicgr_32.so:	Linux 32:	Not Checked
... This IBISCHK6 executable supports Windows 32 bit only		

Errors : 23
Warnings: 5

File Failed

ibischk6_32.exe is used in the test



Status Report Lines

- Checked
- Not checked
- Checked, has platform issue
 - Platform bits different than operating system platform “Bits” entry for Executable line
 - Executable model file does not have dll or so extension
 - Ibischk6_32 cannot load 64-bit executable model files and vice versa



Recommended Extensions

- Windows: dll
- Linux: so
- Other messages may suggest .dll for Windows or .so for Linux)



Limits and Issues

- **ibischk6_32 works on 64-bit platforms if 32-bit executable model files load and work on the platform**
- **Platform name examples shown in the IBIS Specification – some may be specified in the future**
 - **Case insensitive Windows**
 - **Case insensitive Linux**
- **For unknown platform names, ibischk6 will try to load executable model file and run the functional existence test**



Conclusion

- **ibischk6 V6.1.3 adds [Algorithmic Model] Executable line function existence testing of Reserved_Parameters in .ami and associated .dll/.so files**
- **IBIS User Guide updated with new messages**
- **No reference IBIS-AMI waveform checking**
 - **Offered by some commercial vendors**
 - **Free utilities available**
- **Thanks to Mike LaBonte for extra work on BUG179 and the User Guide**

