Accelerating IBIS & Touchstone Specification Feature Development with ANTLR

Michael Mirmak

DesignCon Hybrid IBIS Summit

Santa Clara, CA

February 27, 2026

Legal Disclaimer

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Intel technologies may require enabled hardware, software or service activation.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Performance varies by use, configuration, and other factors. Learn more on the Performance Index site.

Your costs and results may vary.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not “commercial” names and not intended to function as trademarks.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Results have been estimated or simulated.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visiting the Resource and Documentation Center.

Personal Disclaimers

- I don’t speak for Intel Corporation, SAE ITC, the IBIS Open Forum, or ANTLR.org.

- All opinions and errors here are entirely my own.

Agenda

- IBIS Archeology: 1994 IBIS 2.0 BNF
- Issues with BNFs
- ANTLR (ANother Tool for Language Recognition)
  - Structure and Capabilities
- Using LLMs to Generate ANTLR Input Files
- ANTLR Outputs and Findings
- The Most Useful Applications
- Summary
- References

IBIS Archeology

- September, 1994
  - Bob Ross and Steffen Rochel developed a BNF (Backus-Naur Form) for IBIS 2.0 but…
    - *No BNF tool was available to check their work*
    - *The BNF did not support key IBIS assumptions (e.g., keyword ordering rules)*
- A similar discussion re-emerged in 2010
  - Katz, Ross, Steinberger, Warwick, and Westerhoff worked on a .ami BNF
  - Similar challenges on formatting and parsing encountered: lexx/yacc were the suggested approaches at the time

```
//*** Changes in "keywords ... and added next line:

// Remarks
// comments starts with "//"
// keywords are enclosed by '[' '...' ']'. An underbar "_"
// may be substituted for any internal single space and a
// single space may be substituted for an internal
// underbar. No space may exist between the '[' or ']' and
// the internal keyword.
// defined characters and verbatium strings are enclosed
// by '...'
// a set is specified by { }, options are enclosed by
// [ ... ]
// "|" defines an alternative

ibisfile ::= ibisfileheader sections '[' 'end' ']'

packagefile ::= packagefileheader
packagedefinitionsections '[' 'end' ']'

ibisfileheader ::= ibisversion ibisfilename fileversion
fileheaderitems
```

Objective: a truly universal and unambiguous IBIS grammar

Modern BNFs & ANTLR

BNF Format and ANTLR

- BNF (Backus-Naur Form) is a theoretical way to define computer language and data file grammars
- Hurdles…
  - Original BNF, EBNF, etc. concepts were for human readers – they were not themselves standardized to support automation
  - A BNF is not lexer/parser code *per se*; a lexer or parser has to be written around the BNF (e.g., *lex* and *yacc* help this process)
  - Code generation is usually language-limited
- Enter ANTLR: ANother Tool for Language Recognition

- **The BNF notation was proposed in 1959 as part of ALGOL 58 development.**
- **Additional refinements published in 1963 for ALGOL 60**
- **Extended BNF added some regular expressions**
- **Lex & Flex/Flex++: C/C++ UNIX-based lexical scanners dating from the 1970s**
- **Yacc & Bison: C-based parser generators also from the 1970s**

Fast ANTLR Basics

- https://antlr.org
- Branch-and-leaf (terminator) structure using common regular expression rules
- Strict distinctions between lexers and parsers
  - Lexer: matches strings as *tokens*
  - Parser: checks arrangement of tokens per *grammar* rules
- An ANTLR advantage: automated target code and tree generation
  - C#, Java, JavaScript, Python (installation requires Java & Python)
- An enormous number of ANTLR 4 grammar definitions (.g4 input files) already exist
  - C/C#, Fortran, Haskell, LISP, MATLAB, Rust, Verilog, VHDL…
  - https://github.com/antlr/grammars-v4

```
grammar Expr;
prog:    (expr NEWLINE)* ;
expr:    expr ('*'|'/') expr
     |   expr ('+'|'-') expr
     |   INT
     |   '(' expr ')'
     ;
NEWLINE : [\r\n]+ ;
INT     : [0-9]+ ;
```

A canonical ANTLR4 example combining lexing (uppercase) and parsing functions (lowercase) (from antlr.org)

Can We Port the IBIS 2.0 BNF to ANTLR?

- … using LLMs to create proper syntax?
  - OpenAI* ChatGPT 5.2 (public)
  - On-premises Anthropic* Claude Sonnet 4 derivative (proprietary)

- The prompt…

```
The file below is a Backus-Naur Format definition for a data
format called IBIS. Please create an ANTLR 4 parsing and lexing
input file based on this definition. Ensure that the parser and
lexer sections are separate.
```

Only the prompt and the original 1994 BNF file were provided.

Initial Results

- OpenAI* ChatGPT
  - Created separate lexer and parser .g4 (input) files, as requested
  - Confused comment character definition with actual comments
  - Did not support in-line or whole-line comments rules correctly
  - Created extensive, case-insensitive keyword definitions (though without _ support)
  - Recognized and implemented most postfix units (not including temperature

- Anthropic* Claude Sonnet 4
  - Created a single .g4 input file
  - Correctly captured comment_char and default (“|”) comment operation
  - Did not support in-line or whole-line comments rules correctly
  - Implemented field character length rules
  - Recognized and implemented most postfix units (not including temperature

Issues appear to be from original BNF definition; hand correction needed.

Some behaviors may be due to outside training data.

ANTLR Flow

- Write the .g4 input file(s)
  - Can separate the lexer and parser functions into two files, or combine them
- Use ANTLR to test the rules of the .g4 file itself
  - For Java, *antlr4 ibis2p0_igpt.g4*; can add *–Dlanguage=Python3* or other options
  - This builds lexer and parser code files, token tables and more
  - STDOUT lists warnings & errors
- *ANTLR-parse* reports parse tree and token assignments from any levels
  - Can be text or visual
- Code for parsing the trees of any file is generated automatically

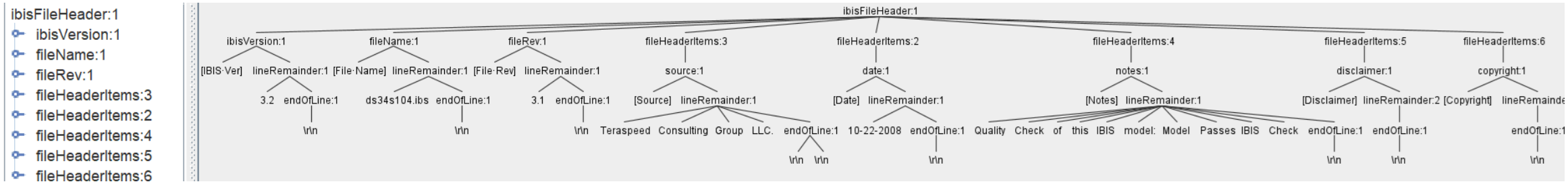
Test Model Results

- Avoiding validation files in ibischk7 source code
- Using Analog Devices public model repository
  - https://analog.com/en/resources/simulation-models/ibis-models.html
  - DS34S104 device model, built by Teraspeed
- ANTLR inputs from OpenAI* ChatGPT used for testing
  - Minor edits required to get partial tree generation ([e.g., Comment Char] is challenging)

STDOUT

```
line 287:41 mismatched input '1.940E-02' expecting TEXT_SIGNAL_NAME
line 288:5 missing TEXT_SIGNAL_NAME at 'ACVDD1'
line 288:41 mismatched input '1.890E-02' expecting TEXT_SIGNAL_NAME
line 289:5 missing TEXT_SIGNAL_NAME at 'CLK_SYS_S'
line 289:41 mismatched input '1.890E-02' expecting TEXT_SIGNAL_NAME
line 290:5 missing TEXT_SIGNAL_NAME at 'JTCLK'
line 290:41 mismatched input '1.940E-02' expecting TEXT_SIGNAL_NAME
```

-gui tree (Java)



What Can We Do With This?

- This is NOT intended to replace ibischk
  - The ibischk executable is a syntax checker
  - The ibischk source code defines a lexer+parser
  - Our syntax checker includes cross-checking and user feedback that simple ANTLR BNFs will not automatically generate

- Two major applications for the future
  - Testing new data format structures, including examples, for consistency
    - Find SPIM, Touchstone 3.0, and IBIS-ISS definition errors BEFORE approval of text and parser creation
  - Streamlining existing specification rules for efficiency
    - IBIS field lengths, unit definitions, inconsistent string definitions, etc. create very complex file rules

In Summary

- Comprehensive IBIS BNFs have been a goal for over 30 years
- ANTLR addresses the BNF standardization problem
  - In regular use across industry, and supports a variety of common languages
  - No actual code-writing is needed to test structure definitions and input files
- Available LLMs can considerably simplify ANTLR input generation
  - LLM outputs are highly variable – consider combining multiple outputs
- SPIM, Touchstone, and IBIS-ISS (reduced SPICE) definitions needed

This is just a very high-level summary of what is possible.

Go play!

References

- ANTLR references
  - https://antlr.org
  - https://tomassetti.me/antlr-mega-tutorial/
  - https://github.com/antlr/grammars-v4

- ANTLR .g4 (Claude-derivative) and BNF source to be published at the IBIS GitHub
  - IBIS-Open-Forum
  - Initial inputs may not work for all .ibs files – please test and update!

Thank you!