

# **vtags-2.0 change**

Vtags-2.0 almost completely refactored vtags-1.x all the features and implementation, both in function, speed and user experience have a qualitative leap, the main changes are reflected in:

1. The new version greatly expanded support for verilog multiple syntaxes.
2. The new version of the implementation of a complete and fast database dynamic update mechanism, vtags will change with the design of dynamic with the new.
3. The new version uses a new recognition algorithm, which greatly improves the accuracy and speed of recognition.
4. The new version uses a decentralized and lazy loading technology, which greatly saves the resources that vtags runs.
5. The new version provides a set of functions and APIs to facilitate the use of vtags and the database when the user does not open vim.
6. The new version optimized vtags's import, so that the vim open non-vtags support file is almost no delay.

## **1 vtags unrecognized verilog syntax**

(1) The multy line comment form `"/*...*/"` can not be recognized.

(2) can not identify the compiler options such as: `'ifdef xxx ...' else ... 'endif`

## 2 command line function set

Vtags-2.0 provided a set of function(display topology, display module instance trace, open the module ...) used in command line to access vtags database (placed in the vtags.db folder) without open vim.

The main function is shown in the following table :

function	description
open_module_file( module_name )	Use gvim to open the file where module is located and jump the cursor to the module definition location.
print_module_filelist( module_name )	Recursive display input module and all sub-module file path, Can be used to extract the RTL code of a module separately.
print_module_io(module_name)	Format the print input module's io information.
print_module_topo( module_name, depth, mask)	Used to print the input module topology, depth used to specify the print depth, mask used to shield the number of instances of too many basic modules (such as dff, etc.), when the number of times greater than the mask when printing topo only show the number does not show specific Instance name (only the number of modules specified by default for the basemodule in the specified vtags db)
print_module_trace(to_module, from_module)	In the whole rtl find to_module call relationship, show all from the top-level module to to_module the entire call track. Such as: Execute: vtags -func -db ... 'print_module_trace (lru)' Display results: cpu -> core -> l2 -> lru Said lru a call from the cpu to the core of the last line to lru

## 3 vtags python api

Vtags-2.0 provides a vtag python api for users to facilitate the use of vtags data

structures and functions to achieve their own characteristics of the application script.

Provide the api in the vtags installation directory "vtags\_custom\_api.py"

see vtags installation directory's " custom\_api\_example.py" as example