Codimension Brief Python Parser

Overview

Codimension needs to collect information about the content of each python module. In some cases, e.g. for the file system browser, a brief information is enough. In other cases, e.g. for the graphics python control flow editor a detailed information is required.

The standard python module pyclbr does not suite well codimension needs in brief parsing so codimension introduces its own implementation of it. The standard modules looses in speed and in the granularity of the provided information.

The codimension brief python parser is implemented as an extension module written in C with a bit of the python glue code. The C code uses [ANTLR v3](http://www.antlr.org/) tool to generate lexer and parser of the python code.

Comparison

The table below shows the comparison between the standard pyclbr module and the codimension’s cdmbriefparser module.

|  |  |  |
| --- | --- | --- |
| **feature** | **pyclbr** | **cdmbriefparser** |
| Extracting coding string | N | Y |
| Extracting module docstring | N | Y |
| Extracting global variables | N | Y |
| Extracting imports | N | Y |
| Extracting top level functions | Y | Y |
| Extracting nested functions | N | Y |
| Extracting functions arguments | N | Y |
| Extracting functions docstrings | N | Y |
| Extracting functions decorators | N | Y |
| Extracting classes | Y | Y |
| Extracting base classes | Y | Y |
| Extracting class attributes | N | Y |
| Extracting class instance attributes | N | Y |
| Extracting class methods | Y | Y |
| Extracting class methods arguments | N | Y |
| Extracting nested classes | N | Y |
| Extracting classes docstrings | N | Y |
| Extracting class methods docstrings | N | Y |
| Extracting classes decorators | N | Y |
| Extracting decorators arguments | N | Y |
| Keeping the hierarchy of the classes/functions of the arbitrary depth | N | Y |
| Ability to work with partially syntactically correct files | Y (silent) | Y (error messages are provided) |
| Ability to parse python code from a file | Y | Y |
| Ability to parse python code from memory | N | Y |
| Extracting classes and functions with the same names | N | Y |
| Supported python version | ANY | Up to 2.7 |
| Time to process 2189 python files (python 2.6 distribution and some third party packages) on Intel Atom 330 based system (Fedora Core 11 distribution). | 2 min 47 sec | 1 min 2 sec |