

U++ - Feature #1993

Ide/Builders: U++ builder for clang-tidy

10/15/2019 01:36 PM - Sender Ghost

Status:	Patch ready	Start date:	10/15/2019
Priority:	Normal	Due date:	
Assignee:	Miroslav Fidler	% Done:	0%
Category:	IDE	Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
<p>There is clang-tidy tool available as part of LLVM/Clang.</p> <p>The clang-tidy has following usage format (for advanced use): clang-tidy [options] <source0> [... <sourceN>] [-- [compiler options]]</p> <p>Where "options" is clang-tidy options, "source0" is some C/C++ source code file(s), "--" is some delimiter between clang-tidy and compiler options, such as includes (-I) and defines (-D).</p> <p>For example: clang-tidy --quiet -header-filter="*" -checks="*",modernize-use-override" test.cpp</p> <p>may use modernize-use-override checker for test.cpp file.</p> <p>Possible to apply available fixes with using "-fix" clang-tidy option: clang-tidy --quiet -header-filter="*" -checks="*",modernize-use-override" -fix test.cpp</p> <p>Some command to get list of available checkers: clang-tidy -list-checks -checks="**"</p> <p>Possible to use SCRIPT builder for some build method for clang-tidy. But also possible to implement this as some kind of U++ builder (which also may support BLITZ build).</p> <p>Therefore, I propose to create U++ builder for clang-tidy.</p> <p>Some patch attached. Based on GccBuilder::BuildPackage method.</p> <p>The clang-tidy options possible to use as part of "Compiler name" for build method. So, I also extended width of EditString for compiler variable in BuildMethodsDefaultBuilderSetupLayout layout for uppsrc/ide/ide.lay file.</p>			

History

#1 - 10/15/2019 01:59 PM - Sender Ghost

I noticed, that for proper usage there is also a need to specify default include compiler path(s) for "INCLUDE directories" or as "-isystem path" for "Common options" in build method.

Possible to get list of default compiler paths with using following command (replace /dev/null for other empty file on Windows):

```
clang++ -v -fsyntax-only -x c++ /dev/null
```

and search between "#include <...> search starts here:" and "End of search list."

Maybe this is some kind of feature, if use compiler options after "--" delimiter.

Overall, this may be simple to configure, based on error(s) output.

#2 - 10/15/2019 02:13 PM - Sender Ghost

- File *modernize-use-override_output.txt* added
- File *modernize-use-override_fixes.diff* added

Attached some example for clang-tidy's [modernize-use-override](#) checker and applied fixes for uppsrc/Core directory (at [13636](#) revision).

Used following "Compiler name" for this build method:
clang-tidy --quiet -header-filter=".*" -checks="*,modernize-use-override" -fix

#3 - 10/15/2019 02:52 PM - Sender Ghost

Also need to note, that I used 1 HYDRA thread for IDE settings, when using "-fix" option for clang-tidy command.

There is possibility to use --export-fixes=<filename> option for clang-tidy command, which may be used for clang-apply-replacements tool. But path for <filename> should be unique, otherwise its contents may be overwritten after new fixes (e.g. if use "Build package" for some U++ package, instead of "Compile" for some file).

#4 - 10/15/2019 02:52 PM - Sender Ghost

- Status changed from New to Patch ready

#5 - 02/06/2020 01:26 PM - Sender Ghost

- File deleted (*uppsrc_ide.diff*)

#6 - 02/06/2020 01:27 PM - Sender Ghost

- File *uppsrc_ide_r13969.diff* added

Updated patch after [13968](#) revision changes.

#7 - 02/24/2020 09:17 PM - Sender Ghost

- File deleted (*uppsrc_ide_r13969.diff*)

#8 - 02/24/2020 09:18 PM - Sender Ghost

- File *uppsrc_ide_r14075.diff* added

Updated patch after 14075 revision changes. Removed previously proposed changes for uppsrc/ide/ide.lay file.

Files			
modernize-use-override_output.txt	120 KB	10/15/2019	Sender Ghost
modernize-use-override_fixes.diff	33.7 KB	10/15/2019	Sender Ghost
uppsrc_ide_r14075.diff	6.7 KB	02/24/2020	Sender Ghost