

Tests for Dev-C++

Johan Mes

Tuesday 10th March, 2015

Contents

1	Manual tests	2
1.1	Explorer	2
1.2	Parser	2
1.3	Project	3
1.4	Compilers	3
1.5	Versions	3
2	Automatic testing	4
2.1	actRunTests	4
3	Compare	5

1 Manual tests

1.1 Explorer

Perform the following tests using the files provided in the Explorer folder:

- Open file using explorer file association.
- Open project file using explorer file association.
- Open multiple files using explorer file association.
- Open file using File, Open.
- Open project file using File, Open.
- Open multiple files using File, Open.
- Open file using dragging onto page control.
- Open project file using dragging onto page control.
- Open multiple files using dragging onto page control.
- Open file using dragging onto devcpp.exe.
- Open project file using dragging onto devcpp.exe.
- Open multiple files using dragging onto devcpp.exe.
- Open file using dragging onto devcppPortable.exe.
- Open project file using dragging onto devcppPortable.exe.
- Open multiple files using dragging onto devcppPortable.exe.
- Open file using dragging onto devcppPortable.exe from other disk/folder.
- Open project file using dragging onto devcppPortable.exe from other disk/folder.
- Open multiple files using dragging onto devcppPortable.exe from other disk/folder.

1.2 Parser

Perform the following tests using the files provided in the Parser folder. Check if the class browser contains the correct information and hover above all identifiers to check if the parser provides the correct information.

- Correctly parse `basic_string.cpp`.
- Correctly parse `Code.cpp`. Check if a, b, c and d are parsed correctly.
- Correctly parse `float3.cpp`. Check if all synonyms of `D3DXVECTOR3` are equivalent. Check if `float3` correctly inherits from `D3DXVECTOR3`.
- Correctly parse `Function.cpp`.
- Correctly parse `Include.cpp`. Hover over all include lines and check if the tooltip displays correct information.
- Correctly parse `OperatorTest.cpp`.
- Correctly parse `Preprocessor.cpp`. Check if all variables are called `CorrectXX`.
- Correctly parse `SimpleClass.cpp`.

- Correctly parse all C++ containers:
 - Correctly parse `basic_string.h`.
 - Correctly parse `stl_deque.h`.
 - Correctly parse `stl_list.h`.
 - Correctly parse `stl_map.h`.
 - Correctly parse `stl_queue.h`.
 - Correctly parse `stl_stack.h`.
 - Correctly parse `stl_vector.h`.
- Correctly parse `string.cpp`. Check if `basic_string` and `string` code completion works
- Correctly parse `Struct.cpp`.
- Correctly parse `Typedef.cpp`. Check if all synonyms are parsed correctly.
- Correctly parse `Variable.cpp`.

1.3 Project

- Open Project, `TestProject.dev`.
- Add folder, and folders inside folder. Save, close, reopen. Check if both are still visible in their correct place.
- Add file to deep folder. Save, close, reopen. Check if it is still visible in their correct place.
- Rename top most folder. Save, close, reopen. Check if content is still displayed as it was before.

1.4 Compilers

- Remove all compiler sets. Reopen Dev-C++. Check if settings are remembered.
- Auto-add compiler sets. Use the new current set to compile a single `cpp` file. Reopen Dev-C++. Check if settings are remembered.
- Add empty set and configure manually. Use it to compile a single `cpp` file. Reopen Dev-C++. Check if settings are remembered.

1.5 Versions

- Perform first-time configuration for the version without a compiler using `devcpp.exe`.
- Perform first-time configuration for the version with a compiler using `devcpp.exe`.
- Perform first-time configuration for the version without a compiler using `devcppPortable.exe`.
- Perform first-time configuration for the version with a compiler using `devcppPortable.exe`.

2 Automatic testing

2.1 actRunTests

Run the automatic tests provided in Source\Tests.pas by executing actRunTests. If all goes well, no assertion errors or memory leaks should be brought up during or after these tests have been run. Make sure to disable action actRunTests in release builds.

3 Compare

Before committing any code, please compare your source code to a previous commit. :)