## **How to Compile UDFs**

## How to Compile UDFs

Compiled method adopts more features of C than the interpreted method does, e.g. implementation of multidimensional array data.

NOTE: Do not include any SPACE in the name of the source file!

- 1. Put both the C source file and header files (if used) into the same folder as the case and data files.
  - a. If you're running from WorkBench, put these files into dp0\FFF\Fluent.
- 2. Open FLUENT.
- 3. Unload previously compiled UDFs first to avoid potential errors. (This step can be skipped if the compiled method in the case did not use a UDF.)
  - a. Define > Define UDF > Manager, unload previous compiled UDF library.
- 4. Define >Define UDF > User Defined scaler, User Defined memory. Give Fluent the correct number of UDS and UDMI if used. Just like what is done in interpreted method.
- 5. Build a UDF library.
  - a. Define-> Define UDF -> Compiled. Select the C source and header files, click "Build".
  - b. Fluent will then start compiling. Errors will be reported if there are any. Warnings can be ignored.
- 6. Click "Load". All the variables that are returned, executed, initialized, or etc. should be shown to be successfully loaded.
- 7. The remaining steps are identical to compiling functions from an interpreted method:
  - a. Choose the conditions you want to use the UDF. Select the right functions.
  - b. Go to Define > Define UDF > Function Hooks. Choose and hook the functions properly.
- 8. The remaining steps are identical to compiling functions from an interpreted method.

Note that functions from compiled UDF are distinguished by the appendage ::libudf. Functions from interpreted methods do not have that.

We would like to credit Shengzhi He for writing this guide.