
Artichoke Documentation

Release 0.1

Jean-Christophe Fillion-Robin

Oct 13, 2018

Contents

1	Introduction	3
2	Features	5
3	Examples	7
4	Usage	9
5	Integrating CMake Module in your project	11
6	API	13
6.1	ExternalProjectDependency	13
7	Test	17
8	Indices and tables	19

CHAPTER 1

Introduction

You will find here a `CMake` module allowing to easily create a build system on top of `ExternalProject` module.

- Intuitive and easy way to “pass” CMake variables to external project. By default, variables are associated with `CMAKE_CACHE_ARGS`.
- Support association of CMake variable with labels.
- Automatically associate variables with `CMAKE_ARGS` if a multi-configuration CMake generator is used and `${CMAKE_CFG_INTDIR}` is found in the value.
- Automatically set external options `LIST_SEPARATOR`, `CMAKE_GENERATOR`, `CMAKE_GENERATOR_PLATFORM` and `CMAKE_GENERATOR_TOOLSET`.
- Display a well-formatted and user friendly summary of the included external projects. For example, see [here](#)
- Support Ninja generator:
 - If supported, `USES_TERMINAL_*` options are always set to dependent projects.
 - The following variables are automatically propagated to all projects:
 - * `CMAKE_EXPORT_COMPILE_COMMANDS`
 - * `CMAKE_JOB_POOL_COMPILE`
 - * `CMAKE_JOB_POOL_LINK`
 - * `CMAKE_JOB_POOLS`
- Intuitive handling of `USE_SYSTEM` options.
- Recipe to write external project file that can be:
 - downloaded/configured/built/installed
 - and/or found on system using `find_package(<projectname> ...)` calls
 - and/or explicitly specified using `-D<projectname>_DIR:PATH=... variables`.

CHAPTER 3

Examples

Note: *To be documented*

CHAPTER 4

Usage

Note: *To be documented*

Integrating CMake Module in your project

Two possible approaches:

- Use `git submodule` to reference this repository.
- Copy *ExternalProjectDependency.cmake* into your source tree making sure you reference the SHA in the associated commit message.

6.1 ExternalProjectDependency

6.1.1 Global Variables

EXTERNAL_PROJECT_DIR

This variable describes the directory in which external project files matching `<EXTERNAL_PROJECT_FILE_PREFIX><projectname>.cmake` expression are globbed.

EXTERNAL_PROJECT_ADDITIONAL_DIR

If set, this variable represents an other directory in which external project files are searched for if not already found in `EXTERNAL_PROJECT_DIR`.

EXTERNAL_PROJECT_ADDITIONAL_DIRS

If set, this variable represents additional directories in which external project files are searched for if not already found in `EXTERNAL_PROJECT_DIR` and `EXTERNAL_PROJECT_ADDITIONAL_DIR`.

EXTERNAL_PROJECT_FILE_PREFIX

This variable describes the prefix of the external project files looked up in `EXTERNAL_PROJECT_DIR`. It defaults to `External_`.

SUPERBUILD_TOPLEVEL_PROJECT

This variable can be set to explicitly identify the name of the top-level project. If not set, it default to the value of `CMAKE_PROJECT_NAME`.

EP_LIST_SEPARATOR

This variable is used to separate list items when passed in various external project `..._COMMAND` options.

If defaults to `^^`.

EP_GIT_PROTOCOL

The value of this variable is controlled by the option `<SUPERBUILD_TOPLEVEL_PROJECT>_USE_GIT_PROTOCOL` automatically defined by including this CMake module. Setting this option allows to update the value of `EP_GIT_PROTOCOL` variable.

If enabled, the variable `EP_GIT_PROTOCOL` is set to `git`. Otherwise, it is set to `https`. The option is enabled by default.

The variable `EP_GIT_PROTOCOL` can be used when adding external project. For example:

```
ExternalProject_Add(${proj}
  ${${proj}_EP_ARGS}
  GIT_REPOSITORY "${EP_GIT_PROTOCOL}://github.com/Foo/Foo.git"
  [...]
)
```

6.1.2 Functions

mark_as_superbuild

```
mark_as_superbuild(<varname1>[:<vartype1>] [<varname2>[:<vartype2>] [...]])
```

```
mark_as_superbuild(
  VARS <varname1>[:<vartype1>] [<varname2>[:<vartype2>] [...]]
  [PROJECTS <projectname> [<projectname> [...]] | ALL_PROJECTS]
  [LABELS <label1> [<label2> [...]]]
)
```

`PROJECTS` corresponds to a list of `<projectname>` that will be added using `ExternalProject_Add` function.
↳ If not specified and called within a project file, it defaults to the value of `'SUPERBUILD_TOPLEVEL_PROJECT'`.
↳ If instead `'ALL_PROJECTS'` is specified, the variables and labels will be passed to all projects.

`VARS` is an expected list of variables specified as `<varname>:<vartype>` to pass to `<projectname>`

`LABELS` is an optional list of label to associate with the variable names specified using `'VARS'` and passed to the `<projectname>` as CMake `CACHE` args of the form:
-D<projectname>_EP_LABEL_<label1>=<varname1>;<varname2>[...]
-D<projectname>_EP_LABEL_<label2>=<varname1>;<varname2>[...]

ExternalProject_DeclareLabels

```
ExternalProject_DeclareLabels(
  [PROJECTS <projectname> [<projectname> [...]] | ALL_PROJECTS]
  LABELS <label1> [<label2> [...]]
)
```

`PROJECTS` corresponds to a list of `<projectname>` that will be added using `ExternalProject_Add` function.
↳ If not specified and called within a project file, it defaults to the value of `'SUPERBUILD_TOPLEVEL_PROJECT'`.
↳ If instead `'ALL_PROJECTS'` is specified, the variables and labels will be passed to all projects. (continues on next page)

(continued from previous page)

LABELS is a list of label to pass to the <projectname> as CMake CACHE args of the form `-D<projectname>_EP_LABEL_<label>=` unless specific variables have been associated with the labels using `mark_as_superbuild`.

ExternalProject_Message

```
ExternalProject_Message(<project_name> <msg> [condition])
```

ExternalProject_Add_Dependencies

```
ExternalProject_Add_Dependencies(<project_name>
  DEPENDS <dep1> [<dep2> [...]]
)
```

DEPENDS List of additional dependencies to associat with `<project_name>`.

ExternalProject_Include_Dependencies

```
ExternalProject_Include_Dependencies(<project_name>
  [PROJECT_VAR <project_var>]
  [EP_ARGS_VAR <external_project_args_var>]
  [DEPENDS_VAR <depends_var>]
  [USE_SYSTEM_VAR <use_system_var>]
  [SUPERBUILD_VAR <superbuild_var>]
  [CMAKE_GENERATOR <cmake_generator>]
  [CMAKE_GENERATOR_PLATFORM <cmake_generator_platform>]
  [CMAKE_GENERATOR_TOOLSET <cmake_generator_toolset>]
)
```

PROJECT_VAR Name of the variable containing the name of the included project.
By default, it is `proj` and it is set to `<project_name>`.

EP_ARGS_VAR Name of the variable listing arguments to pass to ExternalProject.
If not specified, variable name default to `<project_name>_EP_ARGS`.

DEPENDS_VAR Name of the variable containing the dependency of the included project.
By default, it is `<project_name>_DEPENDS`.

USE_SYSTEM_VAR Name of the variable indicating if the system version of <project_name> should be looked up. Lookup of the project is left to the developer,
↳ implementing
the external project file.
By default, it is `<SUPERBUILD_TOPLEVEL_PROJECT>_USE_SYSTEM_<project_
↳name>`.

SUPERBUILD_VAR Name of the variable indicating if the top-level or inner project is,
↳ being built.
By default, it is `<SUPERBUILD_TOPLEVEL_PROJECT>_SUPERBUILD`.

CMAKE_GENERATOR

CMAKE_GENERATOR_PLATFORM

CMAKE_GENERATOR_TOOLSET These three options allow to overwrite the values set in the,
↳ top-level project that

(continues on next page)

(continued from previous page)

```
↳projects.           would otherwise automatically be propagated to dependent_
```

ExternalProject_Add_Empty

```
ExternalProject_Add_Empty(<project_name>  
    DEPENDS <depends>  
)
```

ExternalProject_Install_CMake

Install an external CMake-based project as part of the `install` target.

```
ExternalProject_Install_CMake(<project_name>)
```

This causes building the main project's `install` target to also execute the CMake install script for the specified external project. The project must be previously declared with **ExternalProject_Add**.

ExternalProject_SetIfNotDefined

Set a variable to its default value if not already defined.

```
ExternalProject_SetIfNotDefined(<var> <defaultvalue> [OBFUSCATE] [QUIET])
```

If *NOT* already defined, the variable `<var>` is set with:

- (1) the value of the environment variable `<var>`, if defined.
- (2) the value of the local variable `<var>`, if defined.
- (3) if none of the above is defined, the `<defaultvalue>` passed as a parameter.

Passing the optional parameter 'OBFUSCATE' will display 'OBFUSCATED' instead of the real value. Passing the optional parameter 'QUIET' will not display any message.

For convenience, the value of the cache variable named `<var>` will be displayed if it was set and if QUIET has not been passed.

ExternalProject_AlwaysConfigure

Add an external project step named *forceconfigure* to *project_name* ensuring the project will always be reconfigured.

```
ExternalProject_AlwaysConfigure(<project_name>)
```

CHAPTER 7

Test

```
sudo apt-get install cmake
git clone git://github.com/commontk/Artichoke
mkdir ArtichokeTest-build
cd ArtichokeTest-build
cmake ../Artichoke/Tests
ctest
```


CHAPTER 8

Indices and tables

- `genindex`
- `search`

E

EP_GIT_PROTOCOL
variable, [13](#)

EP_LIST_SEPARATOR
variable, [13](#)

EXTERNAL_PROJECT_ADDITIONAL_DIR
variable, [13](#)

EXTERNAL_PROJECT_ADDITIONAL_DIRS
variable, [13](#)

EXTERNAL_PROJECT_DIR
variable, [13](#)

EXTERNAL_PROJECT_FILE_PREFIX
variable, [13](#)

ExternalProject_Add_Dependencies
function, [15](#)

ExternalProject_Add_Empty
function, [16](#)

ExternalProject_AlwaysConfigure
function, [16](#)

ExternalProject_DeclareLabels
function, [14](#)

ExternalProject_Include_Dependencies
function, [15](#)

ExternalProject_Install_CMake
function, [16](#)

ExternalProject_Message
function, [15](#)

ExternalProject_SetIfNotDefined
function, [16](#)

F

function

- ExternalProject_Add_Dependencies, [15](#)
- ExternalProject_Add_Empty, [16](#)
- ExternalProject_AlwaysConfigure, [16](#)
- ExternalProject_DeclareLabels, [14](#)
- ExternalProject_Include_Dependencies, [15](#)
- ExternalProject_Install_CMake, [16](#)
- ExternalProject_Message, [15](#)

ExternalProject_SetIfNotDefined, [16](#)

mark_as_superbuild, [14](#)

M

mark_as_superbuild
function, [14](#)

S

SUPERBUILD_TOPLEVEL_PROJECT
variable, [13](#)

V

variable

- EP_GIT_PROTOCOL, [13](#)
- EP_LIST_SEPARATOR, [13](#)
- EXTERNAL_PROJECT_ADDITIONAL_DIR, [13](#)
- EXTERNAL_PROJECT_ADDITIONAL_DIRS, [13](#)
- EXTERNAL_PROJECT_DIR, [13](#)
- EXTERNAL_PROJECT_FILE_PREFIX, [13](#)
- SUPERBUILD_TOPLEVEL_PROJECT, [13](#)