

NAME

cmake-instrumentation – CMake Instrumentation

Added in version 4.0.

INTRODUCTION

NOTE:

This feature is only available when experimental support for instrumentation has been enabled by the **CMAKE_EXPERIMENTAL_INSTRUMENTATION** gate.

The CMake Instrumentation API allows for the collection of timing data, target information and system diagnostic information during the configure, generate, build, test and install steps for a CMake project.

This feature is only available for projects using the *Makefile Generators* or the *Ninja Generators*.

All interactions with the CMake instrumentation API must specify both an API version and a Data version. At this time, there is only one version for each of these: the *API v1* and *Data v1*.

Data Collection

Whenever a command is executed with instrumentation enabled, a *v1 Snippet File* is created in the project build tree with data specific to that command. These files remain until after *Indexing* occurs.

CMake sets the *RULE_LAUNCH_COMPILE*, *RULE_LAUNCH_LINK* and *RULE_LAUNCH_CUSTOM* global properties to use the **ctest --instrument** launcher in order to capture details of each compile, link and custom command respectively. If the project has been configured with *CTestUseLaunchers*, **ctest --instrument** will also include the behavior usually performed by **ctest --launch**.

Indexing

Indexing is the process of collating generated instrumentation data. Indexing occurs at specific intervals called hooks, such as after every build. These hooks are configured as part of the *v1 Query Files*. Whenever a hook is triggered, an index file is generated containing a list of snippet files newer than the previous indexing.

Indexing can also be performed by manually invoking **ctest --collect-instrumentation <build>**.

Callbacks

As part of the *v1 Query Files*, users can provide a list of callbacks intended to handle data collected by this feature.

Whenever *Indexing* occurs, each provided callback is executed, passing the path to the generated index file as an argument.

These callbacks, defined either at the user-level or project-level should read the instrumentation data and perform any desired handling of it. The index file and its listed snippets are automatically deleted by CMake once all callbacks have completed. Note that a callback should never move or delete these data files manually as they may be needed by other callbacks.

ENABLING INSTRUMENTATION

Instrumentation can be enabled either for an individual CMake project, or for all CMake projects configured and built by a user. For both cases, see the *v1 Query Files* for details on configuring this feature.

Enabling Instrumentation at the Project-Level

Project code can contain instrumentation queries with the *cmake_instrumentation()* command.

In addition, query files can be placed manually under **<build>/.cmake/instrumentation/<version>/query/** at the top of a build tree. This version of CMake supports only one version schema, *API v1*.

Enabling Instrumentation at the User-Level

Instrumentation can be configured at the user-level by placing query files in the `CMAKE_CONFIG_DIR` under `<config_dir>/instrumentation/<version>/query/`.

Enabling Instrumentation for CDash Submissions

You can enable instrumentation when using CTest in *Dashboard Client* mode by setting the `CTEST_USE_INSTRUMENTATION` environment variable to the current UUID for the **CMAKE_EXPERIMENTAL_INSTRUMENTATION** feature. Doing so automatically enables the **dynamicSystemInformation** query.

The following table shows how each type of instrumented command gets mapped to a corresponding type of CTest XML file.

<i>Snippet Role</i>	CTest XML File
configure	Configure.xml
generate	Configure.xml
compile	Build.xml
link	Build.xml
custom	Build.xml
build	unused!
cmakeBuild	Build.xml
cmakeInstall	Build.xml
install	Build.xml
ctest	Build.xml
test	Test.xml

By default the command line reported to CDash is truncated at the first space. You can instead choose to report the full command line (including arguments) by setting `CTEST_USE_VERBOSE_INSTRUMENTATION` to 1.

API V1

The API version specifies both the subdirectory layout of the instrumentation data, and the format of the query files.

The Instrumentation API v1 is housed in the **instrumentation/v1/** directory under either **<build>/cmake/** for output data and project-level queries, or **<config_dir>/** for user-level queries. The **v1** component of this directory is what signifies the API version. It has the following subdirectories:

query/ Holds query files written by users or clients. Any file with the **.json** file extension will be recognized as a query file. These files are owned by whichever client or user creates them.

query/generated/

Holds query files generated by a CMake project with the `cmake_instrumentation()` command. These files are owned by CMake and are deleted and regenerated automatically during the CMake configure step.

data/ Holds instrumentation data collected on the project. CMake owns all data files, they should never be removed by other processes. Data collected here remains until after *Indexing* occurs and all *Callbacks* are executed.

cdash/ Holds temporary files used internally to generate XML content to be submitted to CDash.

v1 Query Files

Any file with the **.json** extension under the **instrumentation/v1/query/** directory is recognized as a query for instrumentation data.

These files must contain a JSON object with the following keys. The **version** key is required, but all other fields are optional.

version

The Data version of snippet file to generate, an integer. Currently the only supported version is **1**.

callbacks

A list of command-line strings for *Callbacks* to handle collected instrumentation data. Whenever these callbacks are executed, the full path to a *v1 Index File* is appended to the arguments included in the string.

hooks A list of strings specifying when *Indexing* should occur automatically. These are the intervals when instrumentation data should be collated and user *Callbacks* should be invoked to handle the data. Elements in this list should be one of the following:

- **postGenerate**
- **preBuild** (called when **ninja** or **make** is invoked; unavailable on Windows)
- **postBuild** (called when **ninja** or **make** completes; unavailable on Windows)
- **preCMakeBuild** (called when **cmake --build** is invoked)
- **postCMakeBuild** (called when **cmake --build** completes)
- **postInstall**
- **postTest**

queries

A list of strings specifying additional optional data to collect during instrumentation. Elements in this list should be one of the following:

staticSystemInformation

Enables collection of the static information about the host machine CMake is being run from. This data is collected during *Indexing* and is included in the generated *v1 Index File*.

dynamicSystemInformation

Enables collection of the dynamic information about the host machine CMake is being run from. Data is collected for every *v1 Snippet File* generated by CMake, and includes information from immediately before and after the command is executed.

The **callbacks** listed will be invoked during the specified hooks *at a minimum*. When there are multiple query files, the **callbacks**, **hooks** and **queries** between them will be merged. Therefore, if any query file includes any **hooks**, every **callback** across all query files will be executed at every **hook** across all query files. Additionally, if any query file includes any optional **queries**, the optional query data will be present in all data files.

Example:

```
{
  "version": 1,
  "callbacks": [
    "/usr/bin/python callback.py",
    "/usr/bin/cmake -P callback.cmake arg",
  ],
  "hooks": [
    "postCMakeBuild",
    "postInstall"
  ],
  "queries": [
    "staticSystemInformation",
    "dynamicSystemInformation"
  ]
}
```

```
}
```

In this example, after every **cmake --build** or **cmake --install** invocation, an index file **index-`<hash>`.json** will be generated in `<build>/cmake/instrumentation/v1/data` containing a list of data snippet files created since the previous indexing. The commands `/usr/bin/python callback.py index-<hash>.json` and `/usr/bin/cmake -P callback.cmake arg index-<hash>.json` will be executed in that order. The index file will contain the **staticSystemInformation** data and each snippet file listed in the index will contain the **dynamicSystemInformation** data. Once both callbacks have completed, the index file and all snippet files listed by it will be deleted from the project build tree.

DATA V1

Data version specifies the contents of the output files generated by the CMake instrumentation API as part of the *Data Collection* and *Indexing*. There are two types of data files generated: the *v1 Snippet File* and *v1 Index File*. When using the API v1, these files live in `<build>/cmake/instrumentation/v1/data/` under the project build tree.

v1 Snippet File

Snippet files are generated for every compile, link and custom command invoked as part of the CMake build or install step and contain instrumentation data about the command executed. Additionally, snippet files are created for the following:

- The CMake configure step
- The CMake generate step
- Entire build step (executed with **cmake --build**)
- Entire install step (executed with **cmake --install**)
- Each **ctest** invocation
- Each individual test executed by **ctest**.

These files remain in the build tree until after *Indexing* occurs and any user-specified *Callbacks* are executed.

Snippet files have a filename with the syntax `<role>-<timestamp>-<hash>.json` and contain the following data:

version

The Data version of the snippet file, an integer. Currently the version is always **1**.

command

The full command executed. Excluded when **role** is **build**.

result The exit-value of the command, an integer.

role The type of command executed, which will be one of the following values:

- **configure**: the CMake configure step
- **generate**: the CMake generate step
- **compile**: an individual compile step invoked during the build
- **link**: an individual link step invoked during the build
- **custom**: an individual custom command invoked during the build
- **build**: a complete **make** or **ninja** invocation. Only generated if **preBuild** or **postBuild** hooks are enabled.
- **cmakeBuild**: a complete **cmake --build** invocation

- **cmakeInstall**: a complete **cmake --install** invocation
- **install**: an individual **cmake -P cmake_install.cmake** invocation
- **ctest**: a complete **ctest** invocation
- **test**: a single test executed by CTest

target The CMake target associated with the command. Only included when **role** is **compile** or **link**.

targetType

The *TYPE* of the target. Only included when **role** is **link**.

targetLabels

The *LABELS* of the target. Only included when **role** is **link**.

timeStart

Time at which the command started, expressed as the number of milliseconds since the system epoch.

duration

The duration that the command ran for, expressed in milliseconds.

outputs

The command's output file(s), an array. Only included when **role** is one of: **compile**, **link**, **custom**.

outputSizes

The size(s) in bytes of the **outputs**, an array. For files which do not exist, the size is 0. Included under the same conditions as the **outputs** field.

source The source file being compiled. Only included when **role** is **compile**.

language

The language of the source file being compiled. Only included when **role** is **compile**.

testName

The name of the test being executed. Only included when **role** is **test**.

config The type of build, such as **Release** or **Debug**. Only included when **role** is **compile**, **link** or **test**.

dynamicSystemInformation

Specifies the dynamic information collected about the host machine CMake is being run from. Data is collected for every snippet file generated by CMake, with data immediately before and after the command is executed. Only included when enabled by the *v1 Query Files*.

beforeHostMemoryUsed

The Host Memory Used in KiB at **timeStart**.

afterHostMemoryUsed

The Host Memory Used in KiB at **timeStop**.

beforeCPULoadAverage

The Average CPU Load at **timeStart**.

afterCPULoadAverage

The Average CPU Load at **timeStop**.

Example:

```
{
  "version": 1,
  "command" : "/usr/bin/c++ -MD -MT CMakeFiles/main.dir/main.cxx.o -MF CMakeFiles",
  "role" : "compile",
  "return" : 1,
```

```

    "target": "main",
    "language" : "C++",
    "outputs" : [ "CMakeFiles/main.dir/main.cxx.o" ],
    "outputSizes" : [ 0 ],
    "source" : "<src>/main.cxx",
    "config" : "Debug",
    "dynamicSystemInformation" :
    {
        "afterCPULoadAverage" : 2.3500000000000001,
        "afterHostMemoryUsed" : 6635680.0
        "beforeCPULoadAverage" : 2.3500000000000001,
        "beforeHostMemoryUsed" : 6635832.0
    },
    "timeStart" : 1737053448177,
    "duration" : 31
}

```

v1 Index File

Index files contain a list of *v1 Snippet File*. It serves as an entry point for navigating the instrumentation data. They are generated whenever *Indexing* occurs and deleted after any user-specified *Callbacks* are executed.

version

The Data version of the index file, an integer. Currently the version is always **1**.

buildDir

The build directory of the CMake project.

dataDir

The full path to the **<build>/cmake/instrumentation/v1/data/** directory.

hook

The name of the hook responsible for generating the index file. In addition to the hooks that can be specified by one of the *v1 Query Files*, this value may be set to **manual** if indexing is performed by invoking **ctest --collect-instrumentation <build>**.

snippets

Contains a list of *v1 Snippet File*. This includes all snippet files generated since the previous index file was created. The file paths are relative to **dataDir**.

staticSystemInformation

Specifies the static information collected about the host machine CMake is being run from. Only included when enabled by the *v1 Query Files*.

- **OSName**
- **OSPlatform**
- **OSRelease**
- **OSVersion**
- **familyId**
- **hostname**
- **is64Bits**
- **modelId**
- **numberOfLogicalCPU**
- **numberOfPhysicalCPU**

- **processorAPICID**
- **processorCacheSize**
- **processorClockFrequency**
- **processorName**
- **totalPhysicalMemory**
- **totalVirtualMemory**
- **vendorID**
- **vendorString**

Example:

```
{
  "version": 1,
  "hook": "manual",
  "buildDir": "<build>",
  "dataDir": "<build>/cmake/instrumentation/v1/data",
  "snippets": [
    "configure-<timestamp>-<hash>.json",
    "generate-<timestamp>-<hash>.json",
    "compile-<timestamp>-<hash>.json",
    "compile-<timestamp>-<hash>.json",
    "link-<timestamp>-<hash>.json",
    "install-<timestamp>-<hash>.json",
    "ctest-<timestamp>-<hash>.json",
    "test-<timestamp>-<hash>.json",
    "test-<timestamp>-<hash>.json",
  ]
}
```

COPYRIGHT

2000-2025 Kitware, Inc. and Contributors