NAME

cmake-file-api − CMake File-Based API

INTRODUCTION

CMake provides a file−based API that clients may use to get semantic information about the buildsystems CMake generates. Clients may use the API by writing query files to a specific location in a build tree to request zero or more *Object Kinds*. When CMake generates the buildsystem in that build tree it will read the query files and write reply files for the client to read.

The file−based API uses a **<build>/.cmake/api/** directory at the top of a build tree. The API is versioned to support changes to the layout of files within the API directory. API file layout versioning is orthogonal to the versioning of *Object Kinds* used in replies. This version of CMake supports only one API version, *API v1*.

API V1

API v1 is housed in the **<build>/.cmake/api/v1/** directory. It has the following subdirectories:

query/ Holds query files written by clients. These may be *v1 Shared Stateless Query Files*, *v1 Client Stateless Query Files*, or *v1 Client Stateful Query Files*.

reply/ Holds reply files written by CMake whenever it runs to generate a build system. These are indexed by a *v1 Reply Index File* file that may reference additional *v1 Reply Files*. CMake owns all reply files. Clients must never remove them.

Clients may look for and read a reply index file at any time. Clients may optionally create the **reply/** directory at any time and monitor it for the appearance of a new reply index file.

v1 Shared Stateless Query Files

Shared stateless query files allow clients to share requests for major versions of the *Object Kinds* and get all requested versions recognized by the CMake that runs.

Clients may create shared requests by creating empty files in the **v1/query/** directory. The form is:

```
<build>/.cmake/api/v1/query/<kind>-v<major>
```

where **<kind>** is one of the *Object Kinds*, **−v** is literal, and **<major>** is the major version number.

Files of this form are stateless shared queries not owned by any specific client. Once created they should not be removed without external client coordination or human intervention.

v1 Client Stateless Query Files

Client stateless query files allow clients to create owned requests for major versions of the *Object Kinds* and get all requested versions recognized by the CMake that runs.

Clients may create owned requests by creating empty files in client−specific query subdirectories. The form is:

```
<build>/.cmake/api/v1/query/client-<client>/<kind>-v<major>
```

where **client−** is literal, **<client>** is a string uniquely identifying the client, **<kind>** is one of the *Object Kinds*, **−v** is literal, and **<major>** is the major version number. Each client must choose a unique**<client>** identifier via its own means.

Files of this form are stateless queries owned by the client **<client>**. The owning client may remove them at any time.

v1 Client Stateful Query Files

Stateful query files allow clients to request a list of versions of each of the *Object Kinds* and get only the most recent version recognized by the CMake that runs.

Clients may create owned stateful queries by creating **query.json** files in client−specific query subdirectories. The form is:

```
<build>/.cmake/api/v1/query/client-<client>/query.json
```

where **client−** is literal, **<client>** is a string uniquely identifying the client, and **query.json** is literal. Each client must choose a unique **<client>** identifier via its own means.

query.json files are stateful queries owned by the client **<client>**. The owning client may update or remove them at any time. When a given client installation is updated it may then update the stateful query it writes to build trees to request newer object versions. This can be used to avoid asking CMake to generate multiple object versions unnecessarily.

A **query.json** file must contain a JSON object:

```
{
  "requests": [
    { "kind": "<kind>" , "version": 1 },
    { "kind": "<kind>" , "version": { "major": 1, "minor": 2 } },
    { "kind": "<kind>" , "version": [2, 1] },
    { "kind": "<kind>" , "version": [2, { "major": 1, "minor": 2 }] },
    { "kind": "<kind>" , "version": 1, "client": {} },
    { "kind": "..." }
  ],
  "client": {}
}
```

The members are:

requests
A JSON array containing zero or more requests. Each request is a JSON object with members:

kind Specifies one of the *Object Kinds* to be included in the reply.

version
Indicates the version(s) of the object kind that the client understands. Versions have major and minor components following semantic version conventions. The value must be

- a JSON integer specifying a (non−negative) major version number, or
- a JSON object containing **major** and (optionally) **minor** members specifying non−negative integer version components, or
- a JSON array whose elements are each one of the above.

client Optional member reserved for use by the client. This value is preserved in the reply written for the client in the *v1 Reply Index File* but is otherwise ignored. Clients may use this to pass custom information with a request through to its reply.

For each requested object kind CMake will choose the *first* version that it recognizes for that kind among those listed in the request. The response will use the selected *major* version with the highest *minor* version known to the running CMake for that major version. Therefore clients should list all supported major versions in preferred order along with the minimal minor version required for each major version.

client Optional member reserved for use by the client. This value is preserved in the reply written for the client in the *v1 Reply Index File* but is otherwise ignored. Clients may use this to pass custom information with a query through to its reply.

Other **query.json** top−level members are reserved for future use. If present they are ignored for forward compatibility.

v1 Reply Index File

CMake writes an **index−*.json** file to the **v1/reply/** directory whenever it runs to generate a build system. Clients must read the reply index file first and may read other *v1 Reply Files* only by following references. The form of the reply index file name is:

```
<build>/.cmake/api/v1/reply/index−<unspecified>.json
```

where **index−** is literal and <**unspecified**> is an unspecified name selected by CMake. Whenever a new index file is generated it is given a new name and any old one is deleted. During the short time between these steps there may be multiple index files present; the one with the largest name in lexicographic order is the current index file.

The reply index file contains a JSON object:

```
{
  "cmake": {
    "version": {
      "major": 3, "minor": 14, "patch": 0, "suffix": "",
      "string": "3.14.0", "isDirty": false
    },
    "paths": {
      "cmake": "/prefix/bin/cmake",
      "ctest": "/prefix/bin/ctest",
      "cpack": "/prefix/bin/cpack",
      "root": "/prefix/share/cmake−3.14"
    },
    "generator": {
      "multiConfig": false,
      "name": "Unix Makefiles"
    }
  },
  "objects": [
    { "kind": "<kind>",
      "version": { "major": 1, "minor": 0 },
      "jsonFile": "<file>" },
    { "...": "..." }
  ],
  "reply": {
    "<kind>−v<major>": { "kind": "<kind>",
                         "version": { "major": 1, "minor": 0 },
                         "jsonFile": "<file>" },
    "<unknown>": { "error": "unknown query file" },
    "...": {},
    "client−<client>": {
      "<kind>−v<major>": { "kind": "<kind>",
                           "version": { "major": 1, "minor": 0 },
                           "jsonFile": "<file>" },
      "<unknown>": { "error": "unknown query file" },
      "...": {},
      "query.json": {
        "requests": [ {}, {}, {} ],
        "responses": [
```

```
            { "kind": "<kind>",
              "version": { "major": 1, "minor": 0 },
              "jsonFile": "<file>" },
            { "error": "unknown query file" },
            { "...": {} }
          ],
          "client": {}
        }
      }
    }
  }
```

The members are:

cmake A JSON object containing information about the instance of CMake that generated the reply. It contains members:

version
A JSON object specifying the version of CMake with members:

major, minor, patch
Integer values specifying the major, minor, and patch version components.

suffix A string specifying the version suffix, if any, e.g. **g0abc3**.

string A string specifying the full version in the format **<major>.<minor>.<patch>[-<suffix>]**.

isDirty A boolean indicating whether the version was built from a version controlled source tree with local modifications.

paths A JSON object specifying paths to things that come with CMake. It has members for **cmake**, **ctest**, and **cpack** whose values are JSON strings specifying the absolute path to each tool, represented with forward slashes. It also has a **root** member for the absolute path to the directory containing CMake resources like the **Modules/** directory (see **CMAKE_ROOT**).

generator
A JSON object describing the CMake generator used for the build. It has members:

multiConfig
A boolean specifying whether the generator supports multiple output configurations.

name A string specifying the name of the generator.

platform
If the generator supports **CMAKE_GENERATOR_PLATFORM**, this is a string specifying the generator platform name.

objects A JSON array listing all versions of all *Object Kinds* generated as part of the reply. Each array entry is a *v1 Reply File Reference*.

reply A JSON object mirroring the content of the **query/** directory that CMake loaded to produce the reply. The members are of the form

<kind>-v<major>
A member of this form appears for each of the *v1 Shared Stateless Query Files* that CMake recognized as a request for object kind **<kind>** with major version **<major>**. The value is a *v1 Reply File Reference* to the corresponding reply file for that object kind and version.

<unknown>
A member of this form appears for each of the *v1 Shared Stateless Query Files* that CMake did not recognize. The value is a JSON object with a single **error** member containing a string with an error message indicating that the query file is unknown.

client−<client>
A member of this form appears for each client−owned directory holding *v1 Client Stateless Query Files*. The value is a JSON object mirroring the content of the **query/client−<client>/** directory. The members are of the form:

<kind>−v<major>
A member of this form appears for each of the *v1 Client Stateless Query Files* that CMake recognized as a request for object kind **<kind>** with major version **<major>**. The value is a *v1 Reply File Reference* to the corresponding reply file for that object kind and version.

<unknown>
A member of this form appears for each of the *v1 Client Stateless Query Files* that CMake did not recognize. The value is a JSON object with a single **error** member containing a string with an error message indicating that the query file is unknown.

query.json
This member appears for clients using *v1 Client Stateful Query Files*. If the **query.json** file failed to read or parse as a JSON object, this member is a JSON object with a single **error** member containing a string with an error message. Otherwise, this member is a JSON object mirroring the content of the **query.json** file. The members are:

client A copy of the **query.json** file **client** member, if it exists.

requests
A copy of the **query.json** file **requests** member, if it exists.

responses
If the **query.json** file **requests** member is missing or invalid, this member is a JSON object with a single **error** member containing a string with an error message. Otherwise, this member contains a JSON array with a response for each entry of the **requests** array, in the same order. Each response is

- a JSON object with a single **error** member containing a string with an error message, or
- a *v1 Reply File Reference* to the corresponding reply file for the requested object kind and selected version.

After reading the reply index file, clients may read the other *v1 Reply Files* it references.

v1 Reply File Reference
The reply index file represents each reference to another reply file using a JSON object with members:

kind A string specifying one of the *Object Kinds*.

version
A JSON object with members **major** and **minor** specifying integer version components of the object kind.

jsonFile
A JSON string specifying a path relative to the reply index file to another JSON file containing the object.

v1 Reply Files

Reply files containing specific *Object Kinds* are written by CMake. The names of these files are unspecified and must not be interpreted by clients. Clients must first read the *v1 Reply Index File* and and follow references to the names of the desired response objects.

Reply files (including the index file) will never be replaced by files of the same name but different content. This allows a client to read the files concurrently with a running CMake that may generate a new reply. However, after generating a new reply CMake will attempt to remove reply files from previous runs that it did not just write. If a client attempts to read a reply file referenced by the index but finds the file missing, that means a concurrent CMake has generated a new reply. The client may simply start again by reading the new reply index file.

OBJECT KINDS

The CMake file−based API reports semantic information about the build system using the following kinds of JSON objects. Each kind of object is versioned independently using semantic versioning with major and minor components. Every kind of object has the form:

```
{
  "kind": "<kind>",
  "version": { "major": 1, "minor": 0 },
  "...": {}
}
```

The **kind** member is a string specifying the object kind name. The **version** member is a JSON object with **major** and **minor** members specifying integer components of the object kind's version. Additional top−level members are specific to each object kind.

Object Kind codemodel

The **codemodel** object kind describes the build system structure as modeled by CMake.

There is only one **codemodel** object major version, version 2. Version 1 does not exist to avoid confusion with that from **cmake−server(7)** mode.

codemodel version 2

codemodel object version 2 is a JSON object:

```
{
  "kind": "codemodel",
  "version": { "major": 2, "minor": 2 },
  "paths": {
    "source": "/path/to/top-level-source-dir",
    "build": "/path/to/top-level-build-dir"
  },
  "configurations": [
    {
      "name": "Debug",
      "directories": [
        {
          "source": ".",
          "build": ".",
          "childIndexes": [ 1 ],
          "projectIndex": 0,
          "targetIndexes": [ 0 ],
          "hasInstallRule": true,
          "minimumCMakeVersion": {
            "string": "3.14"
```

```
        },
        "jsonFile": "<file>"
      },
      {
        "source": "sub",
        "build": "sub",
        "parentIndex": 0,
        "projectIndex": 0,
        "targetIndexes": [ 1 ],
        "minimumCMakeVersion": {
          "string": "3.14"
        },
        "jsonFile": "<file>"
      }
    ],
    "projects": [
      {
        "name": "MyProject",
        "directoryIndexes": [ 0, 1 ],
        "targetIndexes": [ 0, 1 ]
      }
    ],
    "targets": [
      {
        "name": "MyExecutable",
        "directoryIndex": 0,
        "projectIndex": 0,
        "jsonFile": "<file>"
      },
      {
        "name": "MyLibrary",
        "directoryIndex": 1,
        "projectIndex": 0,
        "jsonFile": "<file>"
      }
    ]
  }
 ]
}
```

The members specific to **codemodel** objects are:

paths A JSON object containing members:

source A string specifying the absolute path to the top−level source directory, represented with forward slashes.

build A string specifying the absolute path to the top−level build directory, represented with forward slashes.

configurations

A JSON array of entries corresponding to available build configurations. On single−configuration generators there is one entry for the value of the **CMAKE_BUILD_TYPE** variable. For multi−configuration generators there is an entry for each configuration listed in the **CMAKE_CONFIGURATION_TYPES** variable. Each entry is a JSON object containing members:

name A string specifying the name of the configuration, e.g. **Debug**.

directories
A JSON array of entries each corresponding to a build system directory whose source directory contains a **CMakeLists.txt** file. The first entry corresponds to the top−level directory. Each entry is a JSON object containing members:

source A string specifying the path to the source directory, represented with forward slashes. If the directory is inside the top−level source directory then the path is specified relative to that directory (with **.** for the top−level source directory itself). Otherwise the path is absolute.

build A string specifying the path to the build directory, represented with forward slashes. If the directory is inside the top−level build directory then the path is specified relative to that directory (with **.** for the top−level build directory itself). Otherwise the path is absolute.

parentIndex
Optional member that is present when the directory is not top−level. The value is an unsigned integer 0−based index of another entry in the main **directories** array that corresponds to the parent directory that added this directory as a subdirectory.

childIndexes
Optional member that is present when the directory has subdirectories. The value is a JSON array of entries corresponding to child directories created by the **add_subdirectory()** or **subdirs()** command. Each entry is an unsigned integer 0−based index of another entry in the main **directories** array.

projectIndex
An unsigned integer 0−based index into the main **projects** array indicating the build system project to which the this directory belongs.

targetIndexes
Optional member that is present when the directory itself has targets, excluding those belonging to subdirectories. The value is a JSON array of entries corresponding to the targets. Each entry is an unsigned integer 0−based index into the main **targets** array.

minimumCMakeVersion
Optional member present when a minimum required version of CMake is known for the directory. This is the<**min**> v ersion given to the most local call to the **cmake_minimum_required(VERSION)** command in the directory itself or one of its ancestors. The value is a JSON object with one member:

string A string specifying the minimum required version in the format:

```
<major>.<minor>[.<patch>[.<tweak>]][<suffix>]
```

Each component is an unsigned integer and the suffix may be an arbitrary string.

hasInstallRule
Optional member that is present with boolean value **true** when the directory or one of its subdirectories contains any **install()** rules, i.e. whether a **make install** or equivalent rule is available.

jsonFile
A JSON string specifying a path relative to the codemodel file to another JSON file containing a *"codemodel" version 2 "directory" object*.

This field was added in codemodel version 2.3.

projects
A JSON array of entries corresponding to the top−level project and sub−projects defined in the build system. Each (sub−)project corresponds to a source directory whose **CMakeLists.txt** file calls the **project()** command with a project name different from its parent directory. The first entry corresponds to the top−level project.

Each entry is a JSON object containing members:

name A string specifying the name given to the **project()** command.

parentIndex
Optional member that is present when the project is not top−level. The value is an unsigned integer 0−based index of another entry in the main **projects** array that corresponds to the parent project that added this project as a sub−project.

childIndexes
Optional member that is present when the project has sub−projects. The value is a JSON array of entries corresponding to the sub−projects. Each entry is an unsigned integer 0−based index of another entry in the main **projects** array.

directoryIndexes
A JSON array of entries corresponding to build system directories that are part of the project. The first entry corresponds to the top−level directory of the project. Each entry is an unsigned integer 0−based index into the main **directories** array.

targetIndexes
Optional member that is present when the project itself has targets, excluding those belonging to sub−projects. The value is a JSON array of entries corresponding to the targets. Each entry is an unsigned integer 0−based index into the main **targets** array.

targets A JSON array of entries corresponding to the build system targets. Such targets are created by calls to **add_executable()**, **add_library()**, and **add_custom_target()**, excluding imported targets and interface libraries (which do not generate any build rules). Each entry is a JSON object containing members:

name A string specifying the target name.

id A string uniquely identifying the target. This matches the**id** field in the file referenced by **jsonFile**.

directoryIndex
An unsigned integer 0−based index into the main **directories** array indicating the build system directory in which the target is defined.

projectIndex
An unsigned integer 0−based index into the main **projects** array indicating the build system project in which the target is defined.

jsonFile
A JSON string specifying a path relative to the codemodel file to another JSON file containing a *"codemodel" version 2 "target" object*.

codemodel version 2 directory object
A codemodel "directory" object is referenced by a *"codemodel" version 2* object's **directories** array. Each "directory" object is a JSON object with members:

paths A JSON object containing members:

source A string specifying the path to the source directory, represented with forward slashes. If the directory is inside the top−level source directory then the path is specified relative to

that directory (with **.** for the top−level source directory itself). Otherwise the path is absolute.

build A string specifying the path to the build directory, represented with forward slashes. If the directory is inside the top−level build directory then the path is specified relative to that directory (with **.** for the top−level build directory itself). Otherwise the path is absolute.

installers

A JSON array of entries corresponding to **install()** rules. Each entry is a JSON object containing members:

component

A string specifying the component selected by the corresponding **install()** command invocation.

destination

Optional member that is present for specific **type** values below. The value is a string specifying the install destination path. The path may be absolute or relative to the install prefix.

paths Optional member that is present for specific **type** values below. The value is a JSON array of entries corresponding to the paths (files or directories) to be installed. Each entry is one of:

- A string specifying the path from which a file or directory is to be installed. The portion of the path not preceded by a **/** also specifies the path (name) to which the file or directory is to be installed under the destination.
- A JSON object with members:

  from A string specifying the path from which a file or directory is to be installed.

  to A string specifying the path to which the file or directory is to be installed under the destination.

In both cases the paths are represented with forward slashes. If the "from" path is inside the top−level directory documented by the corresponding **type** value, then the path is specified relative to that directory. Otherwise the path is absolute.

type A string specifying the type of installation rule. The value is one of the following, with some variants providing additional members:

file An **install(FILES)** or **install(PROGRAMS)** call. The **destination** and **paths** members are populated, with paths under the top−level *source* directory expressed relative to it. The **isOptional** member may exist. This type has no additional members.

directory

An **install(DIRECTORY)** call. The **destination** and **paths** members are populated, with paths under the top−level *source* directory expressed relative to it. The **isOptional** member may exist. This type has no additional members.

target An **install(TARGETS)** call. The **destination** and **paths** members are populated, with paths under the top−level *build* directory expressed relative to it. The **isOptional** member may exist. This type has additional members**tar getId**, **targetIndex**, **targetIsImportLibrary**, and **targetInstallNamelink**.

export An **install(EXPORT)** call. The **destination** and **paths** members are populated, with paths under the top−level *build* directory expressed relative to it. The **paths** entries refer to files generated automatically by CMake for installation, and their actual values are considered private implementation details. This type has additional members **exportName** and **exportTargets**.

script An **install(SCRIPT)** call. This type has additional member **scriptFile**.

code An **install(CODE)** call. This type has no additional members.

importedRuntimeArtifacts
An **install(IMPORTED_RUNTIME_ARTIFACTS)** call. The **destination** member is populated. The **isOptional** member may exist. This type has no additional members.

runtimeDependencySet
An **install(RUNTIME_DEPENDENCY_SET)** call or an **install(TARGETS)** call with **RUNTIME_DEPENDENCIES**. The **destination** member is populated. This type has additional members **runtimeDependencySetName** and **runtimeDependencySetType**.

isExcludeFromAll
Optional member that is present with boolean value **true** when **install()** is called with the **EXCLUDE_FROM_ALL** option.

isForAllComponents
Optional member that is present with boolean value **true** when **install(SCRIPT|CODE)** is called with the **ALL_COMPONENTS** option.

isOptional
Optional member that is present with boolean value **true** when **install()** is called with the **OPTIONAL** option. This is allowed when **type** is **file**, **directory**, or **target**.

targetId
Optional member that is present when **type** is **target**. The value is a string uniquely identifying the target to be installed. This matches the **id** member of the target in the main "codemodel" object's **targets** array.

targetIndex
Optional member that is present when **type** is **target**. The value is an unsigned integer 0−based index into the main "codemodel" object's **targets** array for the target to be installed.

targetIsImportLibrary
Optional member that is present when **type** is **target** and the installer is for a Windows DLL import library file or for an AIX linker import file. If present, it has boolean value **true**.

targetInstallNamelink
Optional member that is present when **type** is **target** and the installer corresponds to a target that may use symbolic links to implement the **VERSION** and **SOVERSION** target properties. The value is a string indicating how the installer is supposed to handle the symlinks: **skip** means the installer should skip the symlinks and install only the real file, and **only** means the installer should install only the symlinks and not the real file. In all cases the **paths** member lists what it actually installs.

exportName
Optional member that is present when **type** is **export**. The value is a string specifying the name of the export.

exportTargets
Optional member that is present when **type** is **export**. The value is a JSON array of entries corresponding to the targets included in the export. Each entry is a JSON object with members:

id A string uniquely identifying the target. This matches the**id** member of the tar - get in the main "codemodel" object's **targets** array.

index An unsigned integer 0−based index into the main "codemodel" object's **targets** array for the target.

runtimeDependencySetName
Optional member that is present when **type** is **runtimeDependencySet** and the installer was created by an **install(RUNTIME_DEPENDENCY_SET)** call. The value is a string specifying the name of the runtime dependency set that was installed.

runtimeDependencySetType
Optional member that is present when **type** is **runtimeDependencySet**. The value is a string with one of the following values:

library Indicates that this installer installs dependencies that are not macOS frameworks.

framework
Indicates that this installer installs dependencies that are macOS frameworks.

scriptFile
Optional member that is present when **type** is **script**. The value is a string specifying the path to the script file on disk, represented with forward slashes. If the file is inside the top−level source directory then the path is specified relative to that directory. Otherwise the path is absolute.

backtrace
Optional member that is present when a CMake language backtrace to the **install()** or other command invocation that added this installer is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

backtraceGraph
A *"codemodel" version 2 "backtrace graph"* whose nodes are referenced from **backtrace** members elsewhere in this "directory" object.

codemodel version 2 target object
A codemodel "target" object is referenced by a *"codemodel" version 2* object's **targets** array. Each "target" object is a JSON object with members:

name A string specifying the logical name of the target.

id A string uniquely identifying the target. The format is unspecified and should not be interpreted by clients.

type A string specifying the type of the target. The value is one of **EXECUTABLE**, **STATIC_LIBRARY**, **SHARED_LIBRARY**, **MODULE_LIBRARY**, **OBJECT_LIBRARY**, **INTERFACE_LIBRARY**, or **UTILITY**.

backtrace
Optional member that is present when a CMake language backtrace to the command in the source code that created the target is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

folder Optional member that is present when the **FOLDER** target property is set. The value is a JSON object with one member:

name A string specifying the name of the target folder.

paths A JSON object containing members:

source A string specifying the path to the target's source directory, represented with forward slashes. If the directory is inside the top−level source directory then the path is specified relative to that directory (with **.** for the top−level source directory itself). Otherwise the path is absolute.

build A string specifying the path to the target's build directory, represented with forward slashes. If the directory is inside the top−level build directory then the path is specified relative to that directory (with **.** for the top−level build directory itself). Otherwise the

path is absolute.

nameOnDisk
Optional member that is present for executable and library targets that are linked or archived into a single primary artifact. The value is a string specifying the file name of that artifact on disk.

artifacts
Optional member that is present for executable and library targets that produce artifacts on disk meant for consumption by dependents. The value is a JSON array of entries corresponding to the artifacts. Each entry is a JSON object containing one member:

path A string specifying the path to the file on disk, represented with forward slashes. If the file is inside the top−level build directory then the path is specified relative to that directory. Otherwise the path is absolute.

isGeneratorProvided
Optional member that is present with boolean value **true** if the target is provided by CMake's build system generator rather than by a command in the source code.

install Optional member that is present when the target has an **install()** rule. The value is a JSON object with members:

prefix A JSON object specifying the installation prefix. It has one member:

path A string specifying the value of **CMAKE_INSTALL_PREFIX**.

destinations
A JSON array of entries specifying an install destination path. Each entry is a JSON object with members:

path A string specifying the install destination path. The path may be absolute or relative to the install prefix.

backtrace
Optional member that is present when a CMake language backtrace to the **install()** command invocation that specified this destination is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

link Optional member that is present for executables and shared library targets that link into a runtime binary. The value is a JSON object with members describing the link step:

language
A string specifying the language (e.g. **C**, **CXX**, **Fortran**) of the toolchain is used to invoke the linker.

commandFragments
Optional member that is present when fragments of the link command line invocation are available. The value is a JSON array of entries specifying ordered fragments. Each entry is a JSON object with members:

fragment
A string specifying a fragment of the link command line invocation. The value is encoded in the build system's native shell format.

role A string specifying the role of the fragment's content:

- **flags**: link flags.
- **libraries**: link library file paths or flags.
- **libraryPath**: library search path flags.
- **frameworkPath**: macOS framework search path flags.

lto Optional member that is present with boolean value **true** when link−time optimization (a.k.a. interprocedural optimization or link−time code generation) is enabled.

sysroot Optional member that is present when the **CMAKE_SYSROOT_LINK** or **CMAKE_SYSROOT** variable is defined. The value is a JSON object with one member:

path A string specifying the absolute path to the sysroot, represented with forward slashes.

archive

Optional member that is present for static library targets. The value is a JSON object with members describing the archive step:

commandFragments

Optional member that is present when fragments of the archiver command line invocation are available. The value is a JSON array of entries specifying the fragments. Each entry is a JSON object with members:

fragment

A string specifying a fragment of the archiver command line invocation. The value is encoded in the build system's native shell format.

role A string specifying the role of the fragment's content:

- **flags**: archiver flags.

lto Optional member that is present with boolean value **true** when link−time optimization (a.k.a. interprocedural optimization or link−time code generation) is enabled.

dependencies

Optional member that is present when the target depends on other targets. The value is a JSON array of entries corresponding to the dependencies. Each entry is a JSON object with members:

id A string uniquely identifying the target on which this target depends. This matches the main **id** member of the other target.

backtrace

Optional member that is present when a CMake language backtrace to the **add_dependencies()**, **target_link_libraries()**, or other command invocation that created this dependency is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

sources

A JSON array of entries corresponding to the target's source files. Each entry is a JSON object with members:

path A string specifying the path to the source file on disk, represented with forward slashes. If the file is inside the top−level source directory then the path is specified relative to that directory. Otherwise the path is absolute.

compileGroupIndex

Optional member that is present when the source is compiled. The value is an unsigned integer 0−based index into the **compileGroups** array.

sourceGroupIndex

Optional member that is present when the source is part of a source group either via the **source_group()** command or by default. The value is an unsigned integer 0−based index into the **sourceGroups** array.

isGenerated

Optional member that is present with boolean value **true** if the source is **GENERATED**.

backtrace

Optional member that is present when a CMake language backtrace to the **target_sources()**, **add_executable()**, **add_library()**, **add_custom_target()**, or other

command invocation that added this source to the target is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

sourceGroups
Optional member that is present when sources are grouped together by the **source_group()** command or by default. The value is a JSON array of entries corresponding to the groups. Each entry is a JSON object with members:

name A string specifying the name of the source group.

sourceIndexes
A JSON array listing the sources belonging to the group. Each entry is an unsigned integer 0−based index into the main **sources** array for the target.

compileGroups
Optional member that is present when the target has sources that compile. The value is a JSON array of entries corresponding to groups of sources that all compile with the same settings. Each entry is a JSON object with members:

sourceIndexes
A JSON array listing the sources belonging to the group. Each entry is an unsigned integer 0−based index into the main **sources** array for the target.

language
A string specifying the language (e.g. **C**, **CXX**, **Fortran**) of the toolchain is used to compile the source file.

languageStandard
Optional member that is present when the language standard is set explicitly (e.g. via **CXX_STANDARD**) or implicitly by compile features. Each entry is a JSON object with two members:

backtraces
Optional member that is present when a CMake language backtrace to the **<LANG>_STANDARD** setting is available. If the language standard was set implicitly by compile features those are used as the backtrace(s). It's possible for multiple compile features to require the same language standard so there could be multiple backtraces. The value is a JSON array with each entry being an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

standard
String representing the language standard.

This field was added in codemodel version 2.2.

compileCommandFragments
Optional member that is present when fragments of the compiler command line invocation are available. The value is a JSON array of entries specifying ordered fragments. Each entry is a JSON object with one member:

fragment
A string specifying a fragment of the compile command line invocation. The value is encoded in the build system's native shell format.

includes
Optional member that is present when there are include directories. The value is a JSON array with an entry for each directory. Each entry is a JSON object with members:

path A string specifying the path to the include directory, represented with forward slashes.

isSystem
Optional member that is present with boolean value **true** if the include directory is marked as a system include directory.

backtrace
Optional member that is present when a CMake language backtrace to the **target_include_directories()** or other command invocation that added this include directory is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

precompileHeaders
Optional member that is present when **target_precompile_headers()** or other command invocations set **PRECOMPILE_HEADERS** on the target. The value is a JSON array with an entry for each header. Each entry is a JSON object with members:

header Full path to the precompile header file.

backtrace
Optional member that is present when a CMake language backtrace to the **target_precompile_headers()** or other command invocation that added this precompiled header is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

This field was added in codemodel version 2.1.

defines Optional member that is present when there are preprocessor definitions. The value is a JSON array with an entry for each definition. Each entry is a JSON object with members:

define A string specifying the preprocessor definition in the format **<name>[=<value>]**, e.g. **DEF** or **DEF=1**.

backtrace
Optional member that is present when a CMake language backtrace to the **target_compile_definitions()** or other command invocation that added this preprocessor definition is available. The value is an unsigned integer 0−based index into the **backtraceGraph** member's **nodes** array.

sysroot Optional member that is present when the **CMAKE_SYSROOT_COMPILE** or **CMAKE_SYSROOT** variable is defined. The value is a JSON object with one member:

path A string specifying the absolute path to the sysroot, represented with forward slashes.

backtraceGraph
A *"codemodel" version 2 "backtrace graph"* whose nodes are referenced from **backtrace** members elsewhere in this "target" object.

codemodel version 2 backtrace graph
The **backtraceGraph** member of a *"codemodel" version 2 "directory" object*, or *"codemodel" version 2 "target" object* is a JSON object describing a graph of backtraces. Its nodes are referenced from **backtrace** members elsewhere in the containing object. The backtrace graph object members are:

nodes A JSON array listing nodes in the backtrace graph. Each entry is a JSON object with members:

file An unsigned integer 0−based index into the backtrace **files** array.

line An optional member present when the node represents a line within the file. The value is an unsigned integer 1−based line number.

command
An optional member present when the node represents a command invocation within the file. The value is an unsigned integer 0−based index into the backtrace **commands** array.

parent An optional member present when the node is not the bottom of the call stack. The value is an unsigned integer 0−based index of another entry in the backtrace **nodes** array.

commands
A JSON array listing command names referenced by backtrace nodes. Each entry is a string specifying a command name.

files A JSON array listing CMake language files referenced by backtrace nodes. Each entry is a string specifying the path to a file, represented with forward slashes. If the file is inside the top−level source directory then the path is specified relative to that directory. Otherwise the path is absolute.

Object Kind cache
The **cache** object kind lists cache entries. These are the CMake Language Variables stored in the persistent cache (**CMakeCache.txt**) for the build tree.

There is only one **cache** object major version, version 2. Version 1 does not exist to avoid confusion with that from **cmake−server(7)** mode.

cache version 2
cache object version 2 is a JSON object:

```
{
  "kind": "cache",
  "version": { "major": 2, "minor": 0 },
  "entries": [
    {
      "name": "BUILD_SHARED_LIBS",
      "value": "ON",
      "type": "BOOL",
      "properties": [
        {
          "name": "HELPSTRING",
          "value": "Build shared libraries"
        }
      ]
    },
    {
      "name": "CMAKE_GENERATOR",
      "value": "Unix Makefiles",
      "type": "INTERNAL",
      "properties": [
        {
          "name": "HELPSTRING",
          "value": "Name of generator."
        }
      ]
    }
  ]
}
```

The members specific to **cache** objects are:

entries A JSON array whose entries are each a JSON object specifying a cache entry. The members of each entry are:

name A string specifying the name of the entry.

value A string specifying the value of the entry.

type A string specifying the type of the entry used by **cmake−gui(1)** to choose a widget for editing.

properties
A JSON array of entries specifying associated cache entry properties. Each entry is a JSON object containing members:

name A string specifying the name of the cache entry property.

value A string specifying the value of the cache entry property.

Object Kind cmakeFiles

The **cmakeFiles** object kind lists files used by CMake while configuring and generating the build system. These include the **CMakeLists.txt** files as well as included **.cmake** files.

There is only one **cmakeFiles** object major version, version 1.

cmakeFiles version 1

cmakeFiles object version 1 is a JSON object:

```
{
  "kind": "cmakeFiles",
  "version": { "major": 1, "minor": 0 },
  "paths": {
    "build": "/path/to/top−level−build−dir",
    "source": "/path/to/top−level−source−dir"
  },
  "inputs": [
    {
      "path": "CMakeLists.txt"
    },
    {
      "isGenerated": true,
      "path": "/path/to/top−level−build−dir/.../CMakeSystem.cmake"
    },
    {
      "isExternal": true,
      "path": "/path/to/external/third−party/module.cmake"
    },
    {
      "isCMake": true,
      "isExternal": true,
      "path": "/path/to/cmake/Modules/CMakeGenericSystem.cmake"
    }
  ]
}
```

The members specific to **cmakeFiles** objects are:

paths A JSON object containing members:

source A string specifying the absolute path to the top−level source directory, represented with forward slashes.

build A string specifying the absolute path to the top−level build directory, represented with forward slashes.

inputs A JSON array whose entries are each a JSON object specifying an input file used by CMake when configuring and generating the build system. The members of each entry are:

path A string specifying the path to an input file to CMake, represented with forward slashes. If the file is inside the top−level source directory then the path is specified relative to that directory. Otherwise the path is absolute.

isGenerated
Optional member that is present with boolean value **true** if the path specifies a file that is under the top−level build directory and the build is out−of−source. This member is not available on in−source builds.

isExternal
Optional member that is present with boolean value **true** if the path specifies a file that is not under the top−level source or build directories.

isCMake
Optional member that is present with boolean value **true** if the path specifies a file in the CMake installation.

Object Kind toolchains

The **toolchains** object kind lists properties of the toolchains used during the build. These include the language, compiler path, ID, and version.

There is only one **toolchains** object major version, version 1.

toolchains version 1

toolchains object version 1 is a JSON object:

```
{
  "kind": "toolchains",
  "version": { "major": 1, "minor": 0 },
  "toolchains": [
    {
      "language": "C",
      "compiler": {
        "path": "/usr/bin/cc",
        "id": "GNU",
        "version": "9.3.0",
        "implicit": {
          "includeDirectories": [
            "/usr/lib/gcc/x86_64-linux-gnu/9/include",
            "/usr/local/include",
            "/usr/include/x86_64-linux-gnu",
            "/usr/include"
          ],
          "linkDirectories": [
            "/usr/lib/gcc/x86_64-linux-gnu/9",
            "/usr/lib/x86_64-linux-gnu",
            "/usr/lib",
            "/lib/x86_64-linux-gnu",
            "/lib"
          ],
          "linkFrameworkDirectories": [],
          "linkLibraries": [ "gcc", "gcc_s", "c", "gcc", "gcc_s" ]
        }
      },
      "sourceFileExtensions": [ "c", "m" ]
    },
    {
```

```
      "language": "CXX",
      "compiler": {
        "path": "/usr/bin/c++",
        "id": "GNU",
        "version": "9.3.0",
        "implicit": {
          "includeDirectories": [
            "/usr/include/c++/9",
            "/usr/include/x86_64-linux-gnu/c++/9",
            "/usr/include/c++/9/backward",
            "/usr/lib/gcc/x86_64-linux-gnu/9/include",
            "/usr/local/include",
            "/usr/include/x86_64-linux-gnu",
            "/usr/include"
          ],
          "linkDirectories": [
            "/usr/lib/gcc/x86_64-linux-gnu/9",
            "/usr/lib/x86_64-linux-gnu",
            "/usr/lib",
            "/lib/x86_64-linux-gnu",
            "/lib"
          ],
          "linkFrameworkDirectories": [],
          "linkLibraries": [
            "stdc++", "m", "gcc_s", "gcc", "c", "gcc_s", "gcc"
          ]
        }
      },
      "sourceFileExtensions": [
        "C", "M", "c++", "cc", "cpp", "cxx", "mm", "CPP"
      ]
    }
  ]
}
```

The members specific to **toolchains** objects are:

toolchains
: A JSON array whose entries are each a JSON object specifying a toolchain associated with a particular language. The members of each entry are:

  language
  : A JSON string specifying the toolchain language, like C or CXX. Language names are the same as language names that can be passed to the **project()** command. Because CMake only supports a single toolchain per language, this field can be used as a key.

  compiler
  : A JSON object containing members:

    path Optional member that is present when the **CMAKE_<LANG>_COMPILER** variable is defined for the current language. Its value is a JSON string holding the path to the compiler.

    id Optional member that is present when the **CMAKE_<LANG>_COMPILER_ID** variable is defined for the current language. Its value is a JSON string holding the ID (GNU, MSVC, etc.) of the compiler.

version
Optional member that is present when the **CMAKE_<LANG>_COMPILER_VERSION** variable is defined for the current language. Its value is a JSON string holding the version of the compiler.

target Optional member that is present when the **CMAKE_<LANG>_COMPILER_TARGET** variable is defined for the current language. Its value is a JSON string holding the cross−compiling target of the compiler.

implicit
A JSON object containing members:

includeDirectories
Optional member that is present when the **CMAKE_<LANG>_IMPLICIT_INCLUDE_DIRECTORIES** variable is defined for the current language. Its value is a JSON array of JSON strings where each string holds a path to an implicit include directory for the compiler.

linkDirectories
Optional member that is present when the **CMAKE_<LANG>_IMPLICIT_LINK_DIRECTORIES** variable is defined for the current language. Its value is a JSON array of JSON strings where each string holds a path to an implicit link directory for the compiler.

linkFrameworkDirectories
Optional member that is present when the **CMAKE_<LANG>_IMPLICIT_LINK_FRAMEWORK_DIRECTORIES** variable is defined for the current language. Its value is a JSON array of JSON strings where each string holds a path to an implicit link framework directory for the compiler.

linkLibraries
Optional member that is present when the **CMAKE_<LANG>_IMPLICIT_LINK_LIBRARIES** variable is defined for the current language. Its value is a JSON array of JSON strings where each string holds a path to an implicit link library for the compiler.

sourceFileExtensions
Optional member that is present when the **CMAKE_<LANG>_SOURCE_FILE_EXTENSIONS** variable is defined for the current language. Its value is a JSON array of JSON strings where each each string holds a file extension (without the leading dot) for the language.

COPYRIGHT