

# CPP Compilation Database Discoverer 1.1

## On this page:

- [Extension ID](#)
- [Extension description](#)
  - [In what situation should you install this extension?](#)
- [What's new?](#)
  - [1.1.0 - funcrel](#)
  - [1.1.0 - beta 3](#)
- [CAST AIP compatibility](#)
- [Download and installation instructions](#)
- [Packaging your source code with the CAST Delivery Manager Tool](#)
  - [Packaging results](#)
  - [Errors during packaging](#)
    - [The directory 'xxx' used in commands inside the compile\\_config.json file is not matching the selected root 'yyy'](#)
      - [Explanation](#)
      - [Solution](#)

## Target audience:

CAST Administrators



**Summary:** Information and Release Notes about the **CPP Compilation Database Discoverer** for the CAST Delivery Manager Tool. This extension is a **file discoverer**.

## Extension ID

`com.castsoftware.dmtcppcompilationdatabasediscoverer`

## Extension description

This discoverer detects projects based on the presence of specific build compilation output files and their content. It can be used in conjunction with:

- [Objective-C extension](#)
- [C/C++ Analyzer](#) embedded in CAST AIP / [C and Cpp Analyzer extension](#)

## In what situation should you install this extension?

If your **C/C++ source code** is designed to compile on **Linux**, you can use a **compilation logging tool** called [Scan Build](#) (freely available from CAST - instructions included) while compiling the source code to output two files `compile_commands.json` and `compile_config.json` that will contain all the information necessary to compile your source code. These two Scan Build files should be delivered as input with your source code in the CAST Delivery Manager Tool: as long as you have also installed the **CPP Compilation Database Discoverer**, then the Scan Build output files will be taken into account when you package the source code and projects will be created based on what is found in the Scan Build files (`compile_commands.json` and `compile_config.json`).

## What's new?

### 1.1.0 - funcrel

Macros without a value (e.g.: `-DMACRO_NAME` instead of `DMACRO_NAME=MACRO_VALUE`) are now supported.

### 1.1.0 - beta 3

- The following specific include paths are now supported:
  - `-isystem`
  - `-iquote`
  - `-idirafter`

## CAST AIP compatibility

This extension is compatible with:

CAST AIP release	Supported
8.3.x	✓
8.2.x	✓
8.1.x	✗
8.0.x	✗
7.3.x	✗

## Download and installation instructions

Please see:

- [Download an extension](#)
- [Install an extension](#)



- This extension contains a **File discoverer** and you should take note of the specific instructions in the installation guide that explains how to package your source code with the CAST Delivery Manager Tool when you have an **existing Version**.
- The latest [release status](#) of this extension can be seen when downloading it from the CAST Extend server.

## Packaging your source code with the CAST Delivery Manager Tool

When you use the CAST Delivery Manager Tool to package the source code, please ensure that you tick the following option in the CAST Delivery Manager Tool to activate the extension:

Package configuration Package content

This tab allows you to define what source code will be extracted

▶ Where is your source code?

▶ Is the source code deployed in a specific folder in the development environment?

▼ Which project should we take into account for the analysis?

In this configuration section, you can filter the list of projects that will be analyzed. By default all projects found are included.

▼ Type of projects to discover in the package

Specify the project types that the CAST Delivery Manager Tool will discover. This is particularly useful for situations where your Source Code is not in a standard format.

◆ Project types discovered

- Power Builder project
- Eclipse no nature project
- Visual Basic.NET project
- Visual C# project
- ASP.Net Web Site
- Visual Basic 6 project
- ASP
- C/C++ Compilation Database project
- Visual C++ 2010-2012 project
- Visual C++ 2002-2008 project
- SHELL
- SAP project
- Maven Java project
- Eclipse Java project
- Web JSP
- Mainframe project



Note that:

- this option will be **disabled** (unticked) in all pre-existing packages - you need to check it before starting the packaging process otherwise it will be ignored.
- for new packages created after the extension's installation, the option will be **enabled** by default

## Packaging results

The use of the discoverer will not result in a specific **CPP Compilation Database Discoverer** "project" - instead, based on what has been delivered in the [Scan Build](#) tool output files, projects may be identified (such as Visual Studio projects), or may not, in which case you will need to manually create an Analysis Unit for your analysis requirements.

**i** Note also that:

- if your source code includes **vcproj** files and the **Visual Studio C++ discoverers** (embedded in the CAST Delivery Manager Tool) are active (ticked) as well as the **CPP Compilation Database Discoverer**, then you may need to carefully check the results of the packaging.
- you may need to create **additional** source code packages in the CAST Delivery Manager Tool to deliver the **header folders**

## Errors during packaging

### The directory 'xxx' used in commands inside the compile\_config.json file is not matching the selected root 'yyy'

The following **error** may be visible following a packaging action when the **CPP Compilation Database Discoverer** is installed:

```
The directory 'xxx' used in commands inside the compile_config.json file is not matching the selected root 'yyy'.  
The source code must be packaged in its development location or you should define a value for the Development root folder.
```

Time	Message Type	Message ID	Message	What To Do
2017-12-20 12:56:22.626	INFO	cast.commandLine.appInfoVersion	DMT Discoverer version 0.5	
2017-12-20 12:56:24.408	ERROR	cast.dmt.discover.cpp.compilationdatabase.invalidR...	The directory '/cygdrive/e/temp/googletest/build...	The source code must be packaged in its ...
2017-12-20 12:56:25.470	WARNING	cast.dmt.discover.cpp.vcproj.vsPropsFileNotFound	Inherited property sheet 'src/googlemock/msvc/2...	
2017-12-20 12:56:25.470	WARNING	cast.dmt.discover.cpp.vcproj.vsPropsFileNotFound	Inherited property sheet 'src/googlemock/msvc/2...	
2017-12-20 12:56:25.470	WARNING	cast.dmt.discover.cpp.vcproj.vsPropsFileNotFound	Inherited property sheet 'src/googlemock/msvc/2...	
2017-12-20 12:56:25.611	INFO	cast.commandLine.processFinished	The execution has finished without a fatal error	

## Explanation

This error occurs if you are packaging the code (i.e. running the CAST Delivery Manager Tool) on a machine different from the one where the compilation took place and indicates that the commands found by the CAST Delivery Manager Tool in the **compile\_config.json** file (output of the **Scan Build** tool) do not match the path defined in the **Root Folder** field.

## Solution

The solution to this error is as follows:

- Either run the CAST Delivery Manager Tool on the machine on which the compilation took place
- Or, use the **Development Root Folder** option in the CAST Delivery Manager Tool to remap the paths:

Package configuration | Package content

This tab allows you to define what source code will be extracted and therefore included in the Source Package. Changes in the configuration will require you to generate the package.

**Where is your source code?**

Choose the type of SCM or File system you want to target in the drop down list. Depending on the choice you make, the configuration section will change.

SCM or File system targeted: Folder on your local file system

Root folder:  [Browse...](#)

**Advanced extraction settings**

**Is the source code deployed in a specific folder in the development environment?**

When the source code is not directly packaged from the development environment, it may be necessary to define the root folder normally used to deploy the source code, so that file/folder references (either full path or relative path outside the package) can be found. If you do not change the root path, the CAST Delivery Manager Tool will not be able to resolve all paths and as such you may see "Missing file" or "Missing folder" alerts.

Development root folder: