# Discovery Options properties

Use the **Discovery Options** properties panel to control how information required to build your project is discovered, enhance search and Content Assist functionality, and automatically enhance your makefile, such as include paths and symbol definitions.



**Discovery Options preferences**

| **Group** | **Description** |
| --- | --- |
| **Configuration** | [Click here for a description.](http://docs.google.com/cdt_u_prop_all.htm) |
| **Manage Configurations...** | [Click here for a description.](http://docs.google.com/cdt_u_prop_all.htm) |
| **Discovery Profiles Scope** | Specifies the type of profile to set for discovery:   * **Per Language** - Enables the association of different profiles with different resource types (different tools and input types), to have different settings discovered, e.g. for C and C++ source files and for different tools used by the project. In addition, selecting this option lets you have different profile settings for different folders; however, only project profile types are allowed. * **Configuration-wide** - The Eclipse CDT uses only one profile for discovering scanner information for the entire project (configuration). This means that both the project and per-file discovery profiles are allowed. |
| **Tools** (language specific list) | Shows a list of language specific compilers. Select a language from the list. |
| **Automated discovery of paths and symbols** | |
| **Automate discovery of paths and symbols** | Scans the build output to populate the path and symbol tables, such as symbol definitions, system include directories, local include directories, macros, and include files. |
| **Report path detection problems** | Sets the notification of diagnostic errors for include paths that the Eclipse CDT is unable to resolve as it parses the build output. |
| **Discovery profile** | Indicates the discovery profile to use for paths and symbol detection. The type of **Configuration** and **Discovery Profile Scope** you specify determine what which **Discovery Profile** options display on this tab. |
| **Discovery profile options** | |
| **Enable build output scanner info discovery** | Configures the scanner to parse the output from a build for compiler commands with options that specify the definition of preprocessor symbols, and include search paths (for GCC compiler, -D and -I respectively). This button is only visible when **Configuration** is set to Release and the **Discovery Profiles Scope** is set to **Configuration-wide**. |
| **Load build output from file** | Specifies the name of the file you selected to load the build output from. |
| **Load** | Lets you load a file to discover paths and symbols based on a previous builds' output. To activate the discovery select a build log file and then the **Load** button. This button is only visible when **Configuration** is set to Release and the **Discovery Profiles Scope** is set to **Configuration-wide**.  **Note:** Click **Variables** to open the Select Variables window to define a build output file. |
| **Browse...** | Click to locate a previously built output file to load the build output from. This button is only visible when **Configuration** is set to Release and the **Discovery Profiles Scope** is set to **Configuration-wide**. |
| **Variables...** | Click to specify an argument for, or create and configure simple launch variables which you can reference in launch configurations that support variables. This button is only visible when **Configuration** is set to Release and the **Discovery Profiles Scope** is set to **Configuration-wide**. |
| **Enable generate scanner info command** | Enables the retrieval of information from the scanner. If it is not selected, the includes will be populated with default gcc system includes; Eclipse gathers the compiler settings based on the specified toolchain. This means that the Eclipse CDT can obtain the default gcc system includes to associate with the project.  When selected, you can specify any required compiler specific commands in the **Compiler invocation command** field. |
| **Compiler invocation command** | Indicates the compiler specific command used to invoke the compiler (such as gcc, or more complicated commands used to perform all necessary steps for compiling C/C++ source files into .o files, and linking the object files and libraries into an executable program. For example, the command **gcc -E -P -v hello.c | hello.cpp** reads a compiler's configuration file and prints out information that includes the compiler's internally defined preprocessor symbols and include search paths. The information is complementary to the scanner configuration discovered when the output is parsed (if you've enabled the **Enable build output scanner info discovery** option), and is added to the project's scanner configuration. You can click **Browse** to locate this command, if required.  **Note:** The parsing of build output for scanner information is compiler specific. For example, the GNU toolchain compilers (gcc and g++) use -I for include paths, and -D for symbol definitions. Consult your compiler specific documentation for more information about scanner information commands, such as the following gcc commands:   * -D *name* * -I * -U *name* * -I- * -nostdinc * -nostdinc++ * -include *file* * -imacros *file* * -idirafter *dir* * -isystem *dir* * -iprefix *prefix* * -iwithprefix *dir* * -iwithprefixbefore *dir* |
| **Browse...** | Click to browse for a file to include in the compiler invocation command. This button is only visible when **Configuration** is set to Release and the **Discovery Profiles Scope** is set to **Configuration-wide**. |
| **Compiler invocation arguments** | Enter additional arguments to send to the chosen compiler. |



* [Build Project preferences, Macros](http://docs.google.com/cdt_o_mproj_pref_macro.htm)
* [Project Properties, Resource page](http://docs.google.com/cdt_u_prop_resource.htm)
* [Project Properties, Builders page](http://docs.google.com/cdt_u_prop_builders.htm)
* [Project Properties, C/C++ Build category](http://docs.google.com/cdt_u_prop_build.htm)
  + [Discovery options page](http://docs.google.com/cdt_u_prop_build_discovery.htm)
  + [Environment page](http://docs.google.com/cdt_u_prop_build_environment.htm)
  + Settings page
    - [Tool Settings tab](http://docs.google.com/cdt_u_prop_build_settings_tool.htm)
    - [Build Steps tab](http://docs.google.com/cdt_u_prop_build_settings_steps.htm)
    - [Build Artifact tab](http://docs.google.com/cdt_u_prop_build_settings_artifact.htm)
    - [Binary Parsers tab](http://docs.google.com/cdt_u_prop_build_settings_binparser.htm)
    - [Error Parsers tab](http://docs.google.com/cdt_u_prop_build_settings_errparser.htm)
  + [Tool chain editor page](http://docs.google.com/cdt_u_prop_build_toolchain.htm)
  + [Variables page](http://docs.google.com/cdt_u_prop_build_variables.htm)
* Project Properties, [C/C++ General category](http://docs.google.com/cdt_u_prop_general.htm)
  + [Documentation page](http://docs.google.com/cdt_u_prop_general_doc.htm)
  + [File Types page](http://docs.google.com/cdt_u_prop_general_typ.htm)
  + [Indexer page](http://docs.google.com/cdt_u_prop_general_idx.htm)
  + [Language Mapping page](http://docs.google.com/cdt_u_prop_general_lng.htm)
  + Paths and Symbols page
    - [Includes tab](http://docs.google.com/cdt_u_prop_general_pns_inc.htm)
    - [Symbols tab](http://docs.google.com/cdt_u_prop_general_pns_sym.htm)
    - [Libraries path tab](http://docs.google.com/cdt_u_prop_general_pns_libpath.htm)
    - [Source Location tab](http://docs.google.com/cdt_u_prop_general_pns_src.htm)
    - [References tab](http://docs.google.com/cdt_u_prop_general_pns_ref.htm)
* [C/C++ Project Properties, Project References page](http://docs.google.com/cdt_u_prop_ref.htm)
* [C/C++ Project Properties, Refactoring History page](http://docs.google.com/cdt_u_prop_refactoring_history.htm)
* [C/C++ Project Properties, Run/Debug Settings page](http://docs.google.com/cdt_u_prop_rundebug.htm)

