

# Search

After you have installed and started running TeamCity, it collects the information on builds, tests and so on and indexes it. In TeamCity you can search builds by build number, tag, build configuration name and other different parameters specifying one or several keywords and use [Lucene](#) search query syntax to get more precise results.

For complete list of available search fields (keywords) please refer to [Complete List of Available Search Fields](#) section.

In this section:

- [Search Query](#)
  - [Differences from Lucene Syntax](#)
  - [Performing Fuzzy Search](#)
  - [Boolean Operators and Wildcards Support](#)
- [Complete List of Available Search Fields, Shortcuts, and Keywords](#)
  - [Search Fields](#)
  - [Shortcuts](#)
    - [Using Double-Colon](#)
  - ["Magic" Keywords](#)

## Search Query

In TeamCity you can search for builds using the Lucene query syntax. Though in TeamCity search query has [two major differences](#), please refer to the [Lucene](#) documentation for complete syntax rules description.

To narrow your search and get more precise results you can use available search fields - indexed parameters of each build. Please refer to the [Complete list of available search fields](#) for more details.

## Differences from Lucene Syntax

When using search query in TeamCity, please pay attention to the following major differences in query syntax from Lucene native syntax:

1. By default, TeamCity uses `AND` operator in query. That is, if you type in the following query: "failed @agent123", then you will get a list of all builds that have keyword "failed" in any of its search fields, and were run on build agent, which name is "agent123".
2. By default, TeamCity uses "prefix search", not exact matching like Lucene. For example, if you search for "c:main", TeamCity will find all builds of the build configuration which name starts with "main" string.

## Performing Fuzzy Search

You also have a possibility to perform fuzzy search using the tilde, "~", symbol at the end of a single word term which to search items similar in spelling.

## Boolean Operators and Wildcards Support

You can combine multiple terms with Boolean operators to create more complex search queries. In TeamCity, you can use `AND`, `+`, `OR`, `NOT` and `-`.

 When using Boolean operators, type them ALL CAPS.

- `AND` (same as a plus sign). All words that are linked by the "AND" are included in the search results.

 This operator is used by default.

- `NOT` (same as minus sign in front of the query word). Exclude a word or phrase from search results.
- `OR` operator helps you to fetch the search terms that contain either of the terms you specify in the search field.

TeamCity also supports usage of "\*" and "?" wildcards in the build query.

 Please do not type an asterisk sign, \*, at the beginning of the search term because it can take a significant amount of time for TeamCity to search its database. For example, \*onfiguration search term is incorrect.

# Complete List of Available Search Fields, Shortcuts, and Keywords

## Search Fields

When using search keywords, use the following query syntax:

```
<search field name>:<value to search>
```

| Search Field  | Shortcut | Description   | Example                            |
|---------------|----------|---|------------------------------------|
| agent         |          | Find all builds that were run on specific agent.  | agent:unit-77, or agent:agent14*   |
| build         |          | Find all builds that include changes with specific string.  | build:254 or build:failed          |
| changes       |          | Find all builds that include changes with specific string.  | changes:(fix test)                 |
| comitters     |          | Find all build that include changes committed by specific developer.  | comitters:ivan_ivanov              |
| configuration | c        | Find all builds from the specific build configuration.  | configuration:IPRC:(Nightly Build) |
| file_revision |          | Find all builds that contain a file with specific revision.   | file_revision:5                    |
| files         |          | Find all build that include files with specific filename.   | files:                             |
| labels        | l        | Find all builds that include changes with specific VCS label.   | label:EAP<br>l:release             |
| pin_comment   |          | Find all builds that were pinned and have specific word (string) in the pin comment.  | pin_comment:publish                |
| project       | p        | Find builds from specific project.  | project:Diana<br>p:Calcutta        |
| revision      |          | Find all builds that include changes with specific revision (e.g., you can search for builds with specific changelist from Perforce, or revision number in Subversion, etc.). | revision:4536                      |
| stamp         |          | Find all builds that started at the specific time (search by timestamp).  | stamp:200811271753                 |
| status        |          | Find all builds with specific status.   | status:failed                      |
| tags          | t        | Find all builds with specific tag.  | tags:buildserver<br>t:release      |
| tests         |          | Find all builds that include specific tests.  | tests:                             |
| triggerer     |          | Find all builds that were triggered by specific user.   | triggerer:ivan.ivanov              |
| vcs           |          | Find builds that have specific VCS.   | vcs:perforce                       |

## Shortcuts

In addition to above mentioned search fields, you can use two following shortcuts in your query:



Please pay attention that when you use these shortcuts, you should not insert colon after it. That is, the query syntax is as follows: <shortcut><value to search>

| Shortcut | Description  | Example                            |
|----------|--|------------------------------------|
| #        | Search for build number.                             | #<number>, e.g. #1234              |
| @        | Find all builds that were run on the specific agent. | @<agent's name>, e.g. @buildAgent1 |

## Using Double-Colon

You can use double-colon sign (: :) to search for project and/or build configuration by name:

- `pro::best` — search for builds of configurations with the names starting with "best", and in the projects with the names starting with "pro".
- `mega::` — search for builds in all projects with names starting with "mega"
- `::super` — search for builds of build configurations with names starting with "super"

## "Magic" Keywords

TeamCity also provides "magic" keywords (for the list see table below). These magic keywords are formed of the '\$' sign and a word itself. The word can be shortened up to the one (first) syllable, that is, the \$labeled, \$l, and \$lab keywords will be equal in query. For example, to search for pinned builds of the "Nightly build" configuration in the "Mega" project you can type any of the following queries:

- `configuration:nightly project:Mega $pinned`
- `c:nigh p:mega $pin`
- `M::night $pin`

| Magic word  | Description   |
|-------------|---|
| \$tagged    | Search for builds with tags. For example, <code>Calcutta::Master \$t</code> query will result in a list of all builds marked with any tag of build configurations which name is started with "Master" from projects with names started with "Calcutta". |
| \$pinned    | Search for pinned builds.   |
| \$labeled   | Search for builds that have been labeled in VCS. For example, to find labeled builds of the Main project you can use following queries: <code>p:Main \$labeled</code> , or <code>project:Mai \$l</code> , or <code>m:: \$lab</code> , etc.              |
| \$commented | Search for builds that have been commented.   |
| \$personal  | Search for personal builds. For example, using <code>-\$p</code> expression in your query will exclude all personal builds from search results.   |