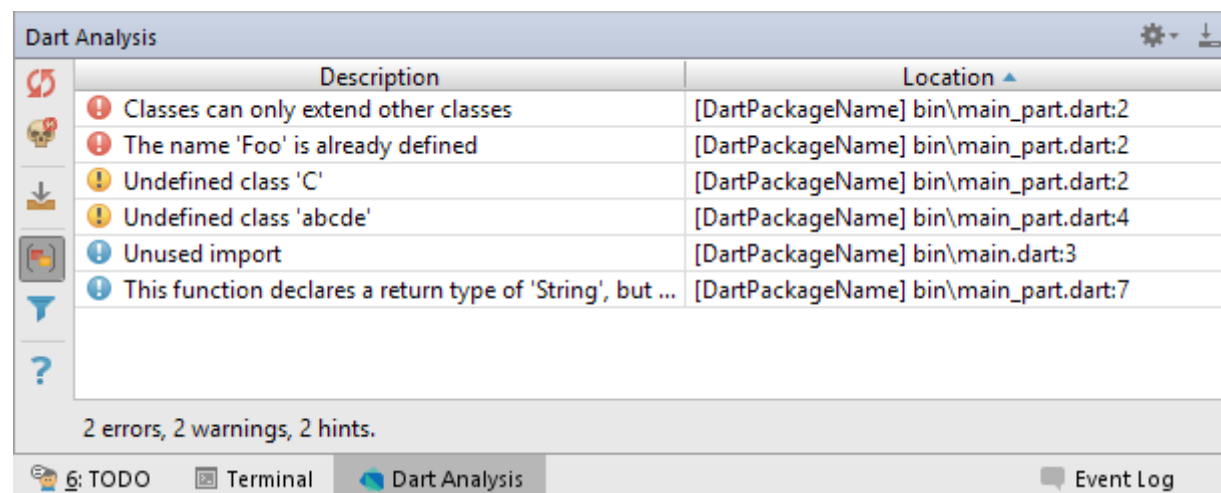


Using Integration with the Dart Analysis Server

PyCharm provides interface for using the [Dart Analyzer tool](#) . The tool is intended for static analysis of your Dart source code. All inconsistencies and potential problems are reported in the dedicated **Dart Analysis** tool window with the possibility to navigate to the fragment of the source code where the problem was detected.



On this page:

- [Messages List](#)
- [Grouping and Sorting](#)
- [Navigation and Speed Search](#)
- [Configuring the Scope of Files to View Problem Reports From](#)

Messages List

All the problem reports are displayed in a table which consists of two columns, **Description** and **Location**, and a summary below.


- The **Description** column shows the message itself, which can be an **error message**, a **warning**, or a **hint**.
- The **Location** column shows the following data:
 - The name of the Dart package in which the problem arose. The Dart package name corresponds to the `project name` from `pubspec.yaml`.
 - The relative from the the Dart package root to the file where the problem arose.
 - The line number where the problem arose.
- The **Summary** area below the table shows the number of detected errors, warnings, or hints and the **filter** status.

Grouping and Sorting

The messages in the **Dart Analysis** tool window can be grouped and sorted. By default, the messages are primarily sorted by their **severity**, that is, the error messages are shown at the top of the list, then come warnings, and finally hints are displayed. In each severity group, messages are grouped by the Dart package name, which is super handy for projects with multiple `pubspec.yaml` files. In each package group, the messages are grouped by their file paths. Finally, in each file group, the problems are sorted by the line number where they occurred.

You can re-configure the secondary sorting:


- Click the **Location** column header to switch between the ascending and descending order of the **Dart package name + file path** sorting. Note that this does not affect the order of severity groups, that is, errors are always shown first. Line numbers are not taken into consideration either, for each particular file problems are always shown in ascending order of line numbers.
- Click the **Description** column header to perform the secondary sorting alphabetically by the problem description. Note that this does not affect the primary grouping by severity.

To disable the primary grouping by severity, release the **Group by Severity** toggle button  on the toolbar. After that the severity of a problem is not taken into consideration at all and problem reports are sorted only by their **Description** or **Location (Dart package name + file**

path). This sorting method guarantees that all the problems for each particular file are grouped together in the table, regardless of their severity.


Navigation and Speed Search

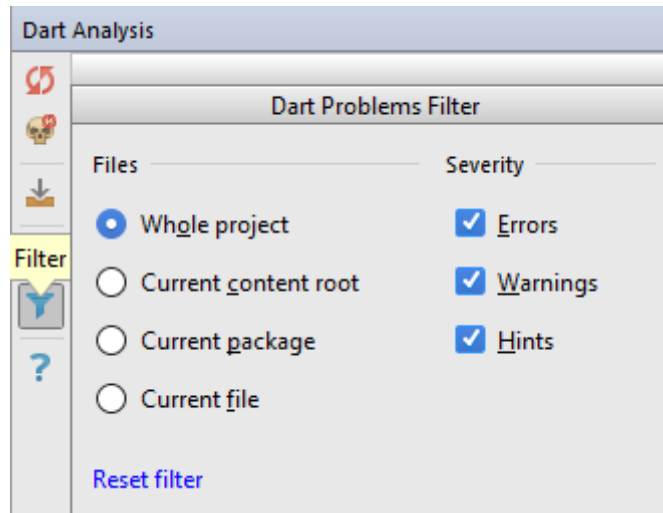
PyCharm provides speed search through the messages in the **Dart Analysis** tool window with the possibility to navigate from a problem report to the fragment of the source code where the problem was detected.

- To locate problems for a specific file or to find a particular message, move the focus to the table and start typing the file name or the text of the message. PyCharm detects the matching row and selects it.
Fuzzy search with partial and middle matching is also supported. For example, if you start typing `ain`, the messages related to the `main.dart` will be also detected.
- To navigate from a problem report to the source code where the problem occurred, select the message in question and choose **Jump to Source** on the context menu of the selection.
- To have PyCharm automatically open the source file with the problem reported in the selected row, press the **Autoscroll to Source** toggle button  on the toolbar.

Configuring the Scope of Files to View Problem Reports From

You can configure the criteria according to which a problem report is displayed in the tool window or not:

1. Click the **Filter** button  on the toolbar.
2. In the **Dart Problems Filter** pop-up window, specify the following:
 - In the **Severity** area, specify the types of messages to be shown, the available types are **Errors**, **Warnings**, and **Hints**. To have the problems of a severity level displayed, select the check box next to this severity level.
 - In the **Files** area, choose the scope for which you want to see problem reports. The available options are:
 - **Whole project**
 - **Current content root**
 - **Current package:** when this option is selected but no `pubspec.yaml` file is detected up the folder hierarchy starting from the current file, then filtering is performed according to the content root where current file is located.
 - **Current file**
 - Click the **Reset all filters** link to restore the default filter settings.



Filtering is applied immediately as soon as you change the current settings.

Last modified: 20 April 2016

