



# Detecting Memory Leaks with Memcheck

You can use the Memcheck tool included in the Valgrind tool suite to detect problems that are related to memory management in applications. You can use the tool together with the GDB debugger. When a problem is detected, the application is interrupted and you can debug it.


**Note:** You can install and run Memcheck locally on Linux. You can run it on a remote host or device from any development machine. On Windows, you can use the [Heob](#) heap observer to receive similar results.

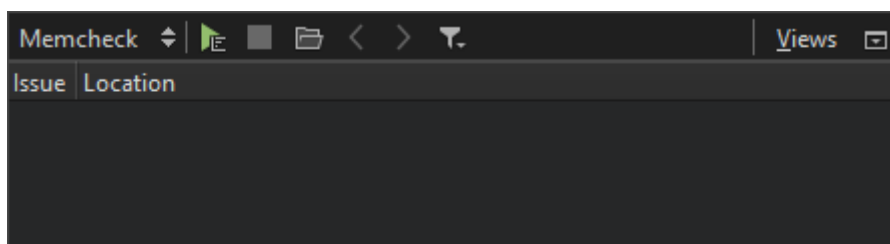
After you download and install Valgrind tools, you can use Memcheck from Qt Creator.

To analyze applications:

1. In the **Projects** mode, select a debug build configuration.
2. Select **Debug** to open the **Debug** mode, and then select **Memcheck** on the toolbar.
3. Select the  button to start the application.
4. Use the application to analyze it.
5. Select the  button to view the results of the analysis in the **Analysis** view.


While the application is running, Memcheck checks all reads and writes of memory and intercepts calls that allocate or free memory or create or delete memory blocks. The results are displayed when you stop Memcheck. Click a line to view where a memory leak occurred and a stack trace that shows what caused it.

As an alternative to collecting data, you can select  to load an external log file in XML format into the **Memcheck** view.



Move the mouse on a row to view more information about the function.

To move between rows, select  or .

To filter the results, select , and then select the types of issues to display in the view. You can view and hide definite and possible memory leaks, uninitialized values, invalid calls to `free()`, and external errors.

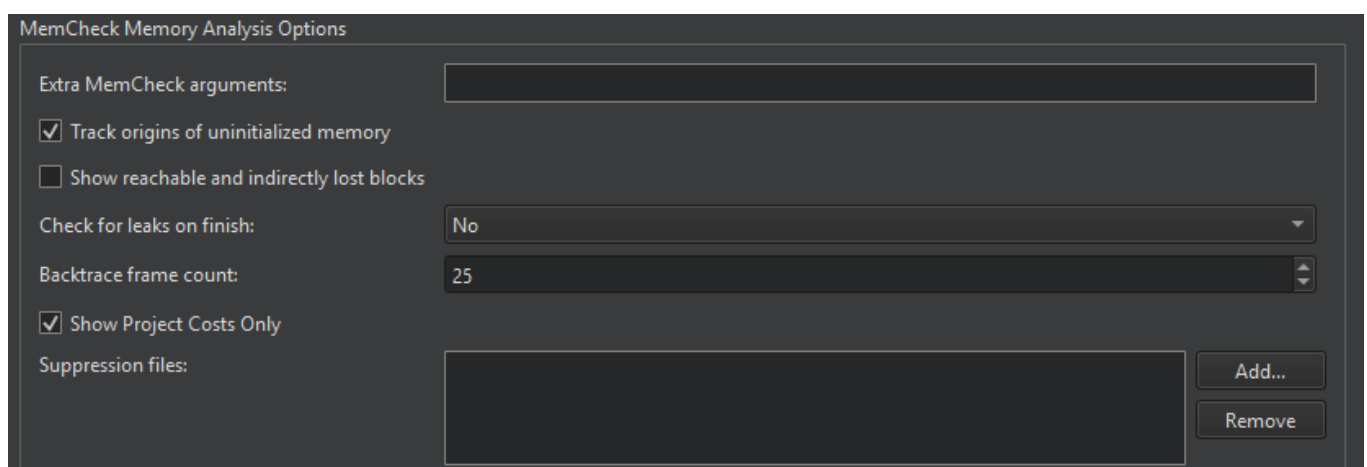
## Selecting Options for Memory Analysis

You can specify analyzer settings either globally for all projects or separately for each project in the [run settings](#) of the project.

To specify global settings for Valgrind, select **Edit > Preferences > Analyzer**. The **Memcheck Memory Analysis Options** group contains Memcheck options.

In **Extra Memcheck arguments**, specify additional arguments for launching the executable.

Stack traces can get quite large and confusing, and therefore, reading them from the bottom up can help. If the stack trace is not big enough or it is too big, select **Edit > Preferences > Analyzer** and define the length of the stack trace in the **Backtrace frame count** field.



Memcheck also reports uses of uninitialised values, most commonly with the message **Conditional jump or move depends on uninitialised value(s)**. To determine the root cause of these errors, the **Track origins of uninitialized memory** check box is selected by default. You can deselect it to make Memcheck run faster.

Memcheck searches for memory leaks when the client application finishes. To view the amount of leaks that occurred, select **Summary Only** in the **Check for leaks on finish** field. To also view details of each leak, select **Full**.

## Showing Reachable and Indirectly Lost Blocks

*Reachable* blocks are blocks that are pointed at by a pointer or chain of pointers and that might have been freed before the application exited. *Indirectly lost* blocks are considered lost because all the blocks that point to them are themselves lost. For example, all the children of a lost root node are indirectly lost.

By default, Memcheck does not report reachable and indirectly lost blocks. To have them reported, select the **Show reachable and indirectly lost blocks**.

## Suppressing Errors

Memcheck detects numerous problems in the system libraries, such as the C library, which come pre-installed with your OS. As you cannot easily fix them, you want to suppress them. Valgrind reads a list of errors to suppress at startup. A default suppression file is created by the `./configure` script when the system is built.

You can write your own suppression files if parts of your project contain errors you cannot fix and you do not want to be reminded of them. Click **Add** in the **MemCheck Memory Analysis** dialog to add the suppression files. For more information about writing suppression files, see [Suppressing Errors](#) in the Valgrind documentation.



© 2022 The Qt Company Ltd. Documentation contributions included herein are the copyrights of their respective owners. The documentation provided herein is licensed under the terms of the [GNU Free Documentation License version 1.3](#) as published by the Free Software Foundation. Qt and respective logos are trademarks of The Qt Company Ltd in Finland and/or other countries worldwide. All other trademarks are property of their respective owners.



Contact Us

Company

- About Us
- Investors
- Newsroom
- Careers
- Office Locations

Support

- Support Services
- Professional Services
- Partners
- Training

Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

Licensing

- Terms & Conditions
- Open Source
- FAQ

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success