Filtering by object type

This page describes the filters that allow you to filter by the object type.

For a summary of how to use them, see Using filters to find objects. For a description of which filters are most useful, see Suggestions for using filters.

Objects that are disposable

Show objects that implement the IDisposable interface. This filter is identical to the **Referenced by: System.IDisposable** filter.

Use this filter to check whether objects which should be disposable (whether or not they have actually been disposed) are still in memory.

Objects that *are / are not* disposed

When set to *are,* show only objects on which Dispose() has been called, but which cannot be garbage-collected because a reference to the object still exists in memory.

This filter is useful because disposed objects should not normally be kept in memory. When you have identified a disposed object with this filter, display it on the object retention graph, then follow the chains of references to identify the objects keeping the disposed object in memory.

This filter is not available when you are profiling a .NET 1.1 application.

Objects that *are / are not* GC roots

Show only GC root objects, or hide GC root objects

If you are investigating a memory leak, it may be useful to hide GC root objects, because they are not generally the source of leaks.

If you are performing a general check on memory usage, it may be useful to show only GC root objects. Show the class list to see details of classes with instances which are GC roots, and then use the class reference explorer to investigate the relationships between these classes and other classes with instances in memory.

To look for specific types of GC roots, combine this filter with the **Kept in memory exclusively by GC roots of type** filters on the **Filter by Reference** tab.

Objects that implement *class/interface*

Show objects that:

- inherit from the selected class
- implement the selected interface
- are objects of the selected class/interface.

If more than one class/interface is selected, objects are shown only if they implement all of the selected classes/interfaces.

This filter is useful if you believe that the selected class/interface may be being kept in memory by other type which implements it.