

Re: Finalizer vs. Dispose

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.clr/2005-03/0211.html>

From: Chris Lyon [MSFT] (clyon_at_online.microsoft.com)

Date: 03/18/05

Date: Fri, 18 Mar 2005 01:45:02 GMT

Brock makes a good point. You can read more on Dispose guidelines here:

<http://blogs.msdn.com/clyon/archive/2004/09/21/232445.aspx>

<http://blogs.msdn.com/clyon/archive/2004/09/23/233464.aspx>

Hope that helps

-Chris

|
| > Mostly all .NET classes which implements IDisposable, also implements
| > an finalizer.

|
| I disagree. The Finalizer is really there for non-managed resources that
| the GC has no idea how to properly cleanp. Think Win32 handles that your
| code went out thru interop to acquire. If you don't have those sorts of
| resources,
| then don't implement a Finalizer as it slows your app down and the memory
| for your objects hang around longer than they need to. I'd say the
| converse

| is more accurate: "All classes that have a Finalizer should also
| implement
| IDisposable".

|
| IDisposable, as already mentioned, is just a pattern that alerts the
| client
| of your code that Dispose needs to be called as soon as you're done with
| this object. This allows more aggresive cleanup of resources (both
| managed
| and unmanaged). If all you hold are managed resources, then IDisposable
| is
| sufficient, yet important.

| -Brock

| DevelopMentor

| <http://staff.develop.com/ballen>

|
|