
Technical Note

Demystifying Dispose

Author(s): R&D Department

Publication date: December 5, 2006



Demystifying Dispose

Overview

The purpose of this document is to provide you with basic information on the **IDisposable.Dispose** method and to advise you to call **Dispose** (when it exists) on objects that you instantiate.

What is Dispose

Dispose is a method whose purpose is to allow resource cleanup, much like C++ destructors, but without freeing the objects' memory. The **Dispose** method is never called directly by the .NET garbage collector.

Dispose is the ideal place to perform any cleanup that needs to be done in a timely fashion, like closing database connections, closing files, releasing bitmaps, etc. Unmanaged resources should be released using **Dispose**, because the garbage collector has no knowledge of anything not allocated on the managed heap. For example, a **File** object that encapsulates a 2 MB data file reports only its managed size (i.e. the size of the managed object) to the garbage collector. The garbage collector has no knowledge of the 2 MB of unmanaged data, thus will not collect it from memory. By calling **Dispose**, you tell the **File** object that you are finished with it, and it will release the file itself and any locks placed on it.

Conclusion

When you write code that uses an object that defines a **Dispose** method, you should make sure that the object's **Dispose** method is called when you are finished with the object. You can do this with the C# **using** statement or by implementing a *try/finally* block in other languages that target the common language runtime.

The whole purpose of the **IDisposable** interface is to signal to you that this class allocates system resources that should be released promptly; the **Dispose** method enables you to tell the object that it is time to release those system resources.

Call **Dispose** on objects that you create that have a **Dispose** method: do not wait for the garbage collector.

References

IDisposable.Dispose Method

[http://msdn2.microsoft.com/en-gb/library/system.idisposable.dispose\(VS.80\).aspx](http://msdn2.microsoft.com/en-gb/library/system.idisposable.dispose(VS.80).aspx)

Demystifying Dispose

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

.NET GC Best Practice -- ALWAYS Call Dispose

<http://weblogs.asp.net/pwilson/archive/2004/02/20/77435.aspx>