CHAPTER 8

Unsafe Codeand Interoperability

Few real-world applications are composed strictly of managed code. Instead, they frequently make use of in-house or 3rd party libraries implemented in native code. The .NET Framework offers several mechanisms to interoperate with native code that is implemented in a number of widespread technologies:

- P/Invoke: enables interoperability with DLLs exporting C-style functions.
- COM Interop: enables consumption of COM objects by managed code as well as exposing .NET classes as COM objects to be consumed by native code.
- C++/CLI language: enables interoperability with C and C++ via a hybrid programming language.

In fact, the Base Class Library (BCL), which is the set of DLLs shipped with the .NET Framework (mscorlib.dll being the main one) that contain .NET Framework's built-in types, uses all of the aforementioned mechanisms. It can therefore be argued that any non-trivial managed application is actually a hybrid application under the covers, in the sense that it calls native libraries.

While these mechanisms are very useful, it is important to understand the performance implications associated with each interop mechanism, and how to minimise their impact.