Foreword

In 1998 I left the World Wide Web Consortium to work on a small project at Microsoft that was code-named “Lightning,” later known as the Common Language Runtime. Our goal was to create the programming platform for the twenty-first century, and Brad Abrams was tasked with designing the classes that are now known as the Microsoft .NET Framework. While Brad worked on the libraries, I worked on the virtual machine.

As time went on, Microsoft decided to standardize much of our work to enable broad adoption of the new programming platform. Brad and I took on this endeavor together, working with Anders Hejlsberg to produce a self-consistent framework that encompassed the most common programming tasks—sort of an updated version of the C Runtime library, but based on a modern object-oriented approach. This framework, along with a description of the architecture, type system, and file format for a virtual machine capable of running the framework, was submitted to ECMA International (formerly known as European Computer Manufacturers Association and refered to simply as ECMA herein), an international standards organization, in October 2000. The original submission included almost 700 classes, divided roughly equally between an abstract operating system API, a common programming library, higher-level programming constructs, low-level virtual machine APIs, and miscellaneous other items.

ECMA accepted the submission and created Task Group 3 of Technical Committee 39 (TC39/TG3) to convert the submission into an ECMA Standard. Brad Abrams was Microsoft’s primary representative to the committee for the work on the frameworks, while I was overall editor for the standard. Brad took on the gargantuan task of coordinating changes to the framework proposed by the committee with changes being made by the Microsoft product team, as well as integrating the text of the documentation into the formal XML format being prepared for submission as the eventual standard.

Working under a self-imposed deadline of one year to release the first version of the standard, the committee decided to reduce the scope of the libraries to allow a complete and careful review, whittling the initial submission of 700 classes down to a more manageable 253 classes. They also provided a structure around the framework that allows a variety of implementations based on the size and capabilities of the system on which it will run—the “profiles” and “libraries” of the final standard.

At the end of the year, in October 2001, the standard was approved and ultimately published as ECMA-335, Second Edition, in December 2002. This was then submitted by ECMA to ISO, which adopted a revised version as ISO/IEC 23271:3003(E) about one year later.

Shortly after the adoption of the Standard, Brad and I agreed to provide a more readable version of the standard as part of a series of books being prepared by Addison-Wesley. Little did we know that we'd be producing a trilogy! Susann Ragsdale and I wrote *The Common Language Infrastructure Annotated Standard* (Addison-Wesley, 2004), which deals with the virtual machine. Brad and the .NET Framework Team wrote Volume 1 of the current book, covering roughly two-thirds of the standardized framework. The current book completes this by providing the definitive description of the remaining classes in the framework, including networking, reflection, XML, and infrastructure support.

As with Volume 1, this book goes into considerably more depth than the standard itself. It includes large numbers of examples, as well as information about the intention behind the classes and methods. It provides comments by members of the committee and the Microsoft product team, and discusses differences between the standardized framework and the Microsoft implementation.

In the two years since the adoption of the ECMA Standard, TC39/TG3 has continued to work. At the time of this writing, we anticipate a revised version of the standard to be submitted for consideration within nine months. This version will include a number of additional classes, as well as support for a new underlying technology known as "generics" that allows the framework to provide type-safe abstractions. Look for the new standard, but even more importantly, help me urge Brad to write Volume 3!

James S. Miller
Software Architect
Common Language Runtime
Microsoft Corporation