



## Foreword

---

**B**EING A DEVELOPER isn't easy. Generally this is a good thing, or the job would get boring very quickly. Each project brings new requirements and new technical challenges, and the backdrop of products and platform versions is constantly evolving.

Amid all the excitement, though, there's still a considerable amount of déjà vu in development: the feeling that you've written the exact same code many times before. This happens because real-world needs often require developers to write generic code at a higher level of abstraction than provided by base platform capabilities—often for application infrastructure or cross-cutting features that feature in enterprise line-of-business applications.

Not only is it boring writing more or less the same code again and again, but it's also inefficient and can lead to consistency and quality problems. To address this, for a number of years the Microsoft patterns & practices team has been producing reusable, extensible, and configurable source code components called *application blocks*. We started off with a very simple data access helper class that we called the Data Access Application Block, but we quickly moved to many other areas, including exception management, logging, authorization, caching, and offline smart client support. These blocks demonstrate proven practices and are designed to meet the most common enterprise application requirements out-of-the-box, but they are also customizable and extensible to meet each organization's or project's unique requirements.



**xxvi** ■ **Foreword**

While the original generation of application blocks was generally very well received by the .NET developer community, most would agree that Enterprise Library marked the beginning of a new era for code-based guidance from the patterns & practices team. Enterprise Library includes functionality similar to that previously released as standalone blocks, but internally it was completely overhauled to feature much-improved consistency across blocks; extensibility to easily support new requirements; ease of use through simpler APIs, documentation, and samples; and integration between blocks that are often used together in applications.

In addition to the changes within the deliverable itself, the other big improvement that arrived with Enterprise Library was the formation of a community—thousands of developers, testers, and architects—sharing problems, solutions, experiences, and extensions with each other and with the patterns & practices team, communicating using our GotDotNet community site, blogs, Webcasts, and events. This community has dramatically increased the value of Enterprise Library by delivering insights and feedback on experiences far beyond what the patterns & practices team could possibly come up with on our own.

Even though Lenny Fenster is a Microsoft employee, it was through this public community that the patterns & practices team really got to know him. Lenny's role involves working with customers every day, on real projects, and this gave him both the opportunity to use Enterprise Library on these projects, as well as the insight into new customer challenges that weren't addressed by Enterprise Library but seemed to be asking for a similar solution. Armed with this insight, Lenny created the Data Mapping Application Block as an extension to Enterprise Library and shared it with the community—and now, with readers of this book as well!

When the time came to update Enterprise Library for the upcoming release of .NET Framework 2.0, the patterns & practices team was lucky enough to be able to “borrow” Lenny from his customer work so he could work as a core member of the Enterprise Library development team. Not only did this allow the patterns & practices team to benefit from his recent customer experiences, but it also gave Lenny first-hand experience with what it takes to build a patterns & practices deliverable. While this book focuses on the .NET Framework 1.1 release of Enterprise Library, most of

the concepts are still relevant for the new .NET Framework 2.0 release. Lenny's expertise and experience on the development team have put him in a unique position to teach developers how to use Enterprise Library most effectively.

So thanks to Lenny—and to all of you as well—for your support of Microsoft patterns & practices and your willingness to give back to the community!

—Tom Hollander, product manager  
*Microsoft patterns & practices*  
<http://msdn.microsoft.com/practices>  
<http://blogs.msdn.com/tomholl>