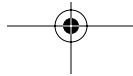


Foreword

IT'S WITH SOME amount of trepidation that I faced the challenge of composing a foreword to this particular book. Let's face it: The names on the cover inspire some amount of awe. It's humbling to know that one's words will introduce what one believes to be the seminal work on a given topic, and believe me, I'm relatively sure this book will meet that lofty goal. When approached with the invitation to grace the front matter of the book, my first response was to wonder what I could possibly add—couldn't they find some luminary at Microsoft to write the foreword? It seems, however, that an outside voice adds some credence to the proceedings, so, dear reader, I speak meekly in the presence of greatness.

First, a little about me (it's the last chance I'm going to get in this short piece): I've been lurking about, programming Office in its various guises, for more than ten years. I've written a lot about the wonders—and gotchas—of Office development, and survived the glory years surrounding Office 2000, when it looked like Office might finally make a successful, integrated development platform. Around 2001, it became clear that no matter how hard I and like-minded folks wanted Office to become a respected development standard, it just wasn't going to make it with VBA as the programming language.

With the release of Visual Studio Tools for Office 2003, it finally looked like we'd made some progress: No longer relegated to the 1990s, Office developers could embrace .NET and all its goodness, taking advantage of managed code, code-access security, xcopy deployment, and all the rest





Foreword

that .NET supplied. I loved this product, but it never really reached critical mass with the developer community. Most likely, the fact that you could only use COM-based controls on documents, and the fact that the product supplied no design-time experience at all, made it a slow starter.

Around that time, I very clearly remember sitting down at some Microsoft event and meeting Eric Carter. I didn't really know who he was at the time (and he certainly didn't know anything about me), but he seemed nice enough, and we chatted for several hours about Office development in general and about VSTO specifically. Only later did I learn that he was high up in the development side of the product. (I spent hours worrying that I'd said something really stupid while we were chatting. Hope not....) We began a long correspondence, in which I've more often than not made it clear that I've got a lot to learn about how .NET and Office interact. I've spent many hours learning from Eric Carter's blog, and Eric Lippert's blog is just as meaty. If you are spending time doing Office development, make sure you drop by both:

<http://blogs.msdn.com/ericlippert/>

http://blogs.msdn.com/eric_carter/

I spent some hours perusing a draft copy of this book and, in each chapter, attempted to find some trick, some little nugget, that I had figured out on my own but that didn't appear in the book. I figured that if I was going to write the foreword for the book, I should add something. The result: I was simply unable to find anything missing. Oh, I'm sure you'll find some little tidbit you've figured out that doesn't appear here, but in my reading of it, I wasn't able to. I thought for sure I'd catch them on something but, alas, I failed. (And, I suppose, that's a good thing, right?) Every time I thought I had them in a missing trick, there is was, right there in print. What that means is that you'll have the best possible reference book at your fingertips. Of course, you need to get your expectations set correctly: It's simply not possible, even in a 60-page chapter, to describe the entirety of the Excel or Word object model. But E&E have done an excellent job of pointing out the bits that make the biggest impact on .NET development.





If you're reading this foreword before purchasing the book, just do it. Buy the thing. If you've already bought it, why are you reading this? Get to the heart of the matter—skip ahead, and get going. You can always read this stuff later. There's a considerable hill ahead of you, and it's worth the climb. Office development using managed code has hit new strides with the release of Visual Studio 2005. I can't wait to take advantage of this book to build great applications.

—*Ken Getz, senior consultant*
MCW Technologies

