



## Foreword

---

I'M WRITING THIS foreword in December of 2007, a little more than a year after the first release of Windows Communication Foundation hit the streets as part of .NET Framework 3.0 and less than a month after we shipped significant additions to the platform as part of .NET Framework 3.5. Saying there's a lot to absorb in those two releases is something of an understatement.

One of the goals of WCF was to unify the programming experience for building all types of distributed applications on the Microsoft platform. We wanted a core set of concepts that were simple and approachable, yet expressive enough to model the underlying semantics of all the technologies we intended to replace. The Microsoft stacks that came before us (ASMX, Remoting, COM+, MSMQ, and WSE) had strong benefits as well as significant limitations; our mission was to leverage ideas that had worked well in the past and learn from those that didn't. If we succeeded, developers would be able to write many different types of distributed applications without having to learn many different (and often wildly discontinuous) programming models.

To make the vision of a unified developer experience successful in the real world, we needed a highly flexible runtime architecture that matched the richness of the programming model. Key areas of variability had to be identified and isolated into generalized extensibility mechanisms to avoid unnecessarily restricting the capabilities of our new platform. Our goal with the runtime was to make sure that, if our default behavior didn't meet the needs of a particular application or we lacked a feature required by a



specific scenario, there would be a natural point in the runtime into which an external developer could plug some customization to address the issue.

The most exciting part about WCF for me is the amazing breadth of scenarios to which this technology can be applied. Nothing demonstrates that more concretely than the feature set we delivered in .NET 3.5. This release had two parallel thrusts covering very different types of distributed application scenarios. One thrust was about integrating WCF with the power of Windows Workflow Foundation to provide a substrate for long-running, declarative, connected business processes. The other thrust was about extending the reach of WCF to address the needs of today's evolving Web. Both of these scenarios impose unique requirements on the runtime and programming model, and the fact that we were able to address these requirements via extensions to WCF *without requiring significant changes to the existing implementation* is a strong indicator that the WCF architecture will be able to address the evolving needs of distributed applications for a long time to come.

Now that we're a year out from shipping our first bits, it's exciting to see real customers make big bets on our platform. It's more exciting to hear about the gains they see in the areas of developer productivity, performance, and interoperability as a result of betting on WCF. We judge the success of our platform first and foremost by the success of our customers, and by that metric WCF will be a very successful platform indeed.

Really, this is all just a long-winded way of saying that the time you invest in learning WCF is well spent. To that end, you're very lucky to be holding this book in your hands right now. Rich, Chris, and Steve have done a fantastic job distilling the broad story of WCF down to the essential elements required to be productive on our platform. The authors' unique combination of technical acumen, field experience, and close relationship with the product team has yielded a book that will undoubtedly hold a distinguished place on every WCF developer's bookshelf. I'm incredibly happy to have these guys telling the technical story of our product. By the end of this book, I'm sure you'll feel the same way.

Steve Maine

Seattle, Washington

December 2007