

## *foreword*

---

In my foreword to the first edition, I wrote about the process of creating software at Microsoft, specifically the first version of SQL Server Reporting Services. Although the organic style of software development we used for the initial release of the product had proven to be a success, creating the follow-on version would have its own set of challenges.

After launching the product in early 2004, we turned to planning for the new release, which was synchronized with the rest of SQL Server. The release was on track for a delivery in 2005, which meant we had comparatively little time for development of new features. In addition, joining the SQL Server mainline product required the team to adopt new versions of Visual Studio and the .NET Framework, and we had to merge our setup with a new, integrated setup engine. To accommodate this accelerated schedule, our original plan was to provide a small set of incremental improvements over the original version. The broad adoption of the product had already given us a good amount of feedback about what customers felt was missing from the initial release. From an architectural standpoint, we wouldn't change the core of Reporting Services, allowing us to safely add selected features.

At the same time, we also realized that something major was missing from the first version of the product. While developers and IT professionals liked the fact that Report Designer integrated fully into Visual Studio, the most frequent question we received was, "How can nondevelopers build their own reports?" We knew we had to address this need with a tool that was easy to use and that didn't require users to understand a database query language. Fortunately, we didn't have to start from scratch and were able to acquire a small company called ActiveViews to provide the core technology. The result of this acquisition was Report Builder (discussed in chapter 7). As we had in our adoption of the .NET Framework in the first release, we took a gamble again in adoption of the new ClickOnce technology for Report Builder.

The last piece of the puzzle was to continue our investment in a rich platform for reporting. Many customers told us that they wanted to easily embed reporting functionality into their applications. So we separated the report viewing components from the Report Server and provided a rich set of report controls in the release of Visual Studio 2005. These are covered in depth in chapter 11. We actually rebuilt both the

Report Manager web application and the Report Designer to leverage the new controls. The end user of these tools will see little difference in the new release, but building them with the new controls helped us validate their functionality and usability.

Even more than with the first release, books such as Bret Updegraff's *SQL Server 2005 Reporting Services in Action* are critical for helping you get the most out of Reporting Services. As the capabilities of the product have increased, the information and guidance that this book provides will help you leverage the Reporting Services platform to the fullest in your own environment. There are many parts of the product that we weren't fully able to expose or document, and this book will help you unlock some of these hidden gems.

BRIAN WELCKER  
*Group Program Manager*  
*Microsoft SQL Server Reporting Services*