

# Foreword

Remember way back when you were younger, and *had* to take high school art classes on the basics of color, layout, and form? And like so many of us you sat there thinking, “I am never going to use this in my life,” and promptly forgot what you might have learned. I know I sure do.

Many years down the track I built one of my first major applications with a significant user interface (UI); a run-of-the-mill business application: some data entry, some editing, we all know the drill. To this day, I still live in fear that the users of this application might find out where I live. If they do, I will have an angry mob of people at my door, keyboards and therapy bills in hand...well, you get the picture.

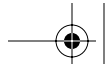
Why do I say that? Having had the privilege to review the book you are about to embark on, I decided that I would dig up that old application—well, I have to admit that I would have been part of that mob had I been using the application. It has the stale ring of “how many different doo-dahs can I squeeze into a simple application?”

The point here—and yes, I do have one—is that over the years I have certainly learned some of the essential user interface design techniques, but they had never been clearly solidified in my conscious mind until I read this book.

Now, way back when I was young and innocent—Visual Basic 1.0 had just hit the shelves—building a user interface was still a fairly rigid and non-creative task, and as such it wasn’t possible to do a lot of jiggling with the look and feel of applications. Then we experienced the Visual Basic wave, and everyone could easily write a Windows application.

It was fairly clean to start with, people followed the established trends, but soon enough people were bucking the system, and I have seen a wide range of unique and strange user interfaces, menu structures, and interaction metaphors—some good and some I would rather not mention in civilized company. Surely it couldn’t get worse than that. But it did: HTML, the Web browser, and easy-to-use design tools such as Microsoft FrontPage or HotDog or Insert your favorite Tool Here. Now, if you can use a word processor, you can publish Web pages and Web applications.





## xvi Foreword

---

One of the first chapters of this book covers color and its use—something I personally feel ought to be required reading for anyone who will be designing user interfaces. Why? Personally, I have observed a scary trend on the Web: The first time someone publishes some content, he or she seems to find the maximum number of animated GIFs, hideous background colors, background graphics, and text colors that will send you dyslexic.

Now the future is going to hold the capability to do some fantastic things with the user interface; the announcements made at the Microsoft Professional Developers Conference for the next generation of Windows, codenamed Longhorn, will give developers the capability to build and deliver user interfaces that can make the user experience a pleasure.

As with any set of technologies and tools, it is your capability that will eventually show what can be built and delivered. The next generation will provide some fantastic services, but also more than enough rope for catastrophic disasters.

The majority of my time is spent with Microsoft Enterprise customers and ISVs, providing them with assistance for the architecture and design of systems and the supporting technology to solve their business problems. As part of this, I have had the privilege to explore some fantastic user interfaces, and I am still getting therapy for some of the others!

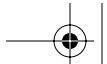
Now, personally, I don't want to see you or your application UIs being used as an example of how bad a user interface can be—although I can use it as a wonderful speaking example or story one day. I would rather see you get your hands on some well thought-out tools, and information to build a better UI—this book is an excellent place to start.

To be perfectly honest, if you are a professional user interface designer, this is not really the book for you. However, if you are a typical overworked, deadline-constrained, feature-creeping software developer, this book is going to make your life easier.

Now, when you get your cup of coffee/tea/Bonnox and settle down on the sofa to read this book, there is something I would like to suggest. First, don't skip the introductory chapters. One of the great strengths of Rebecca's writing is a clear, consistent structure, and this book is no exception. By building on the basics of design—spacing, color, text, and so forth—you will get a solid set of foundation topics that she discusses later. As you read further into the chapters, you will start to auto-magically think and recognize the concepts that are described—but only if you read them.

The remainder of the book pulls from the wealth of Rebecca's experience as an architect and designer. As I read the book, I caught myself flashing back to a tale of a customer project where a UI blunder was about to happen (or worse still, *had* happened), as the advice and guidance in this





book has been pulled from years of solid experience and has been refined to the point where you can use it as a guideline for your development projects.

I hope you enjoy reading this book as much as I have, and learn at least one new thing, although I firmly believe that you will learn many more than one because I certainly have. (But please don't tell Rebecca. She'll never let me live it down.)

If you learn nothing else, remember the number-one rule when putting an application together: It is the end user out there who is using the application. End users don't give a left dingo biscuit about your cool user interface widget; they care about getting the task done for their job. Your prime driver is making sure they can—quickly, efficiently, and without tears of frustration. This book is a step in the right direction.

Stephen W. Jeffries  
Principal Technical Specialist—Developer & E-Business  
Microsoft Corporation

