Foreword

Flash has become an important and pervasive feature of the Web. Consider the numbers: In June 2003 an NPD Research study showed that 97.4 percent of Web users have Flash installed on their computer. The study was conducted roughly a year after the release of the Flash Player 6 and showed that version 6 of the player was already available on more than 86 percent of Web users' computers. The Flash Player is preinstalled in Internet Explorer and other browsers. But that doesn't explain why more than 86 percent of Web users had version 6 a year after its release. At the time, IDC Research estimated there were 448 million people using the Web. A year after the release of the version 6 player, something like 386 million of those people had version 6. The statistics don't show how many people downloaded the player and how many found that it came with their browser, but it's a safe bet that hundreds of millions of player downloads had to take place to get the player onto so many machines so quickly.

The pervasiveness of the Flash Player has a lot to do with how widely Flash is used. But it is not the only driving force. In 2003, during the course of history-making world events, large media sites made extensive use of Flash to provide multimedia coverage of these events and background information. At the same time, individuals created Flash animations and games that were used to express views across the political spectrum. Flash was then, and is now, an important vehicle for providing an expressive and dynamic Web. It is used for almost everything. Advertisers love Flash advertisements because they are difficult to ignore. Educators use Flash to create compelling online learning objects that enliven courses and provide online simulations. Artists are always finding new uses for Flash, and institutions such as museums rely on it more and more. Cartoonists continue to discover Flash and are constantly pushing the envelope of animation length and quality. Flash front-ends to online applications are popping up everywhere and allow you to do everything from booking a hotel reservation to making banking transactions.

The capacity of Flash to deliver compelling experiences on the Web encourages people to download the latest player. In turn, the near ubiquity of the Flash Player encourages designers and developers to make regular use of Flash. The two trends continue to feed off and reinforce each other.

Internet technologies are constantly appearing, evolving, and in some cases disappearing. Each new technology takes time to learn and apply effectively. As each technology evolves, keeping up with it takes more time and effort, as does learning how to apply it to old problems and how to use it to solve new ones. At some point designers and developers have to make big choices. It just isn't possible to learn every Web technology in depth. After the baseline requirements of learning HTML and style sheets, creating effective Web images, and learning some JavaScript, what comes next? Just keeping up with the abbreviations can be irritating. In no particular order: XML, XML-RPC, XUL, MXML, SQLXML, SVG, EPML, SMIL, WSDL, SOAP, WAP, WMA, EJB, J2EE, SAX, JAX-RPC, LDAP, PHP, CFML, and on and on.

Web designers and developers naturally gravitate to systems that are capable of reaching the greatest number of people, provide the richest set of capabilities, and are still cost effective to use. In other words, designers and developers need systems that provide ubiquity, utility, and usability.

Flash has evolved from a vector-based animation tool to an amazingly rich, flexible, and integrated design and development environment. Today you can seamlessly blend unique vector-based animation with images, components such as the Datagrid and Tree components, progressive download and streaming audio and video, real-time communications, and connectivity to every type of enterprise-class computing system. Flash works with many of the technologies already mentioned and provides some of its own for good measure, such as AMF and RTMP. The power of the Flash authoring system does not lie only in the fact that it does one or two things well. It delivers so many features in one integrated development and delivery system with which both designers and developers can effectively work. As Flash continues to evolve, the need to bring together a complex variety of difficult-to-integrate systems and tools continues to diminish.

At the core of both the Flash development process and delivery system is the ActionScript language and interpreter. Designers often make extensive use of ActionScript to simplify their Flash movies and provide the kind of fine-grained control they need for visual elements. Developers use ActionScript to make increasingly full-featured Web-based applications with rich user interfaces and excellent responsiveness to the user's actions. ActionScript can accommodate the needs of both designers and developers because it is based on JavaScript (also known as ECMAScript).

ECMAScript/JavaScript is first and foremost a scripting language. It was designed from the beginning to provide easy-to-use object scripting capabilities. In Flash, that means it is relatively simple to do things such as attach a behavior to a button that in turns manipulates another object such as an animation running within a Movie Clip. The ActionScript code to add a behavior that manipulates an object is short and simple, and the Flash authoring environment makes it easier still. However, ActionScript is not restricted to scripting objects. It is possible to build large and complex applications using traditional object-oriented design and development methods that are realized in ActionScript. With the release of the 2004 Flash products, ActionScript has evolved further to increase support for advanced object-oriented development without sacrificing the language's utility for designers.

Learning to work with ActionScript is an essential Flash skill. How far you go with ActionScript and how you approach learning the language will vary depending on your background — especially how much computer programming you already know. However you approach learning ActionScript, you will constantly be working with both the language and the objects it allows you to manipulate — especially Movie Clips. *Flash MX 2004 ActionScript Bible* provides an excellent roadmap to the language and the variety of objects it allows you to create and manipulate. The two authors bring a great deal of practical experience to the book and are also accomplished teachers and learners. I first met Robert Reinhardt when he was a first-year student at Ryerson University. Robert was the kind of student professors pray for. He is intelligent, inquisitive, and remarkably energetic. As I have with so many students who have graduated from Ryerson, I completely lost track of Robert after he got his degree. Some years later when I finally decided to learn something about Flash I asked one of my colleagues which book I should read. I was told to buy Reinhardt and Lentz's *Flash 4 Bible*. When I realized that

I knew one of the authors, I was delighted. But I was not surprised by the high quality of Robert's work. Some time later Robert met Joey Lott while teaching a class on Flash. Joey stood out for Robert in much the same way Robert stood out for me. Joey has moved on to write a number of books on different facets of programming in Flash and with Robert has produced a book I'm happy to sneak my name into by writing these few pages. *Flash MX 2004 ActionScript Bible* shows a concern for the needs of designers and developers who want to learn ActionScript through experimentation and exploration. Enjoy.

Brian Lesser
Assistant Director, Teaching and Technology Support
Computing and Communications Services
Ryerson University, Toronto, Canada