macromedia® white paper

Macromedia Flash MX–A next-generation rich client

by Jeremy Allaire

March 2002

The information contained in this document represents the current view of Macromedia on the issue discussed as of the date of publication. Because Macromedia must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Macromedia, and Macromedia cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for information purposes only. MACROMEDIA MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Macromedia may have patents, patent applications, trademark, copyright or other intellectual property rights covering the subject matter of this document. Except as expressly provided in any written license agreement from Macromedia, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or other intellectual property.

Macromedia, Inc.
600 Townsend Street
San Francisco, CA 94103
415–252–2000

Contents

With the introduction of Macromedia Flash MX, Macromedia is ushering in the next major innovation in client-side Internet content and applications. Macromedia Flash MX was built from the ground up to provide a 'rich client' environment for Internet content and applications that will radically improve the quality of end-user applications, making the Internet more relevant and useful to businesses and consumers.

The Internet and client-side applications

In the mid-1990s, explosive growth in the Internet and the World Wide Web drove widespread adoption of a new model for content and applications using personal computers connected to the Internet. Coined 'thin-client' computing, this new model promised to lower the cost of developing and delivering applications to end-user desktops, customers and business partners, and to increase the range of application types that could be delivered. This model centered on a very thin client based on HTML, and powerful application servers that dynamically composed and delivered 'pages' to web browsers.

So far this model has proven successful. However, it has also suffered from significant drawbacks and limitations, especially around the richness of the application interfaces, media and content, and the overall sophistication of the solutions that could be built and delivered. Indeed, for many traditional application developers, while the web has offered significant conveniences in terms of ease of deployment, the capabilities of the programming and user interaction models have forced users to suffer. In many respects, much of the web application development and deployment technology of the late 1990s has had to adapt to the challenges imposed by the architecture inherent in the web.

The Internet of 2002 will be different. End-users and businesses are demanding more from their investments in Internet technology. The ability to deliver true value to users is forcing many companies to look towards richer models for Internet applications; models that combine the media-rich power of the traditional desktop with the deployment and content-rich nature of web applications. Companies are also anticipating a growth in the use of web services, or reusable software components that are used as services over the network, and looking towards a world where applications will need to share functionality and data across many types of client devices. These trends are driving the industry towards next-generation rich clients.

This is the backdrop upon which Macromedia built Macromedia Flash MX and Macromedia Flash Player 6.

Rich clients and rich Internet applications

Before detailing the technical aspects of the Macromedia Flash MX client environment, it is important to note what we consider to be the crucial aspects of rich client technologies. Rich client technologies should:

- *Provide an efficient, high-performance runtime for executing code, content and communications.* The principle here is that the end-user experience of HTML-based web applications suffers from a variety performance related challenges. These include the request-response page rendering model; the need to dynamically generate large blobs of text for transmission of simple data; the lack of client-side data storage; the inability to easily invoke and use remote business logic, and even the basic graphics model of HTML. These all must be improved.
- *Integrate content, communications, and application interfaces into a common environment.* The end-user experience of the Internet today is fragmented into the HTML browser for textual content and basic application interfaces; multiple messaging clients for performing communications functions; and multiple media players for handling audio, video, and other forms of media. Rich clients need to provide deep integration for all of these types of interaction.
- *Provide powerful and extensible object models for interactivity.* While web browsers have progressed in terms of their support for interactivity through the Document Object Model (DOM), JavaScript, and DHTML, they still lack the richness needed for building serious applications. Rich clients need to provide a powerful, object-based model for applications and events. This common object model must integrate user interface, communications, and system level services.
- *Enable rapid application development through components and re-use.* Rich clients should support powerful component-driven development, enabling both third party and corporate developers to easily reuse visual components to accelerate development, and give junior developers access to complex functionality. These components should integrate seamlessly into the design-time environment for ease of development.
- *Enable the use of web and data services provided by application servers.* The promise of rich clients includes the ability to cleanly separate presentation logic and user interfaces from the application logic hosted on the network. Rich clients should provide a model for easily using remote services provided by back-end components, whether hosted in an application server or accessed as XML web services.
- *Embrace connected and disconnected clients.* While many users have gotten used to having to be online and in a web browser to perform work, the reality is that most applications would benefit from the ability to be used offline on occasionally connected devices such as personal digital assistants (PDAs) and laptops. Likewise, many applications require support for persistent connections with two-way, notification-based communications. Rich clients must enable both of these types of applications to be easily built and deployed.

- *Enable easy deployment on multiple platforms and devices.* Internet applications are all about reach. The promise of the web is one of content and applications anywhere, regardless of the platform or device. Rich clients must embrace and support all popular desktop operating systems, as well as the broadest range of emerging device platforms such as smart phones, PDAs, set-top boxes, game consoles, and Internet appliances.

Macromedia Flash MX attempts to address and enable all of these opportunities.

Provide an efficient, high-performance runtime for executing code, content and communications

Macromedia Flash MX provides a high-performing runtime for code, content and communications both in terms of real runtime performance, as well as perceived performance by virtue of the execution model of Macromedia Flash Player. This is accomplished through the following capabilities:

- **Compiled SWF files.** Macromedia Flash runtime code (SWF files) combines code, media and data into a compact, compiled file format that can be easily delivered. HTML pages must combine raw text markup, actual textual data, external binary image files, and internal or external script code, creating bloated files and non-optimized execution. Unlike HTML pages, Flash files are tuned to execute fast.
- **Compressed SWF files.** Even pre-complied files can sometimes grow large when they contain a lot of code and textual data. To solve this, Macromedia Flash MX includes a high-performance compression/decompression model. This allows developers to compress their code, media, and data at publish time, and then decompress at runtime on the local user's computer, taking advantage of the latent CPU power of most modern computers. Compression improves the end-user experience and lowers network costs for IT.
- **Efficient rendering through vector graphics.** The core graphics-rendering engine in Macromedia Flash, whether for full images, animations, or simple user interface controls, is a vector graphics-rendering engine. Unlike bitmapped interfaces that must send data for each pixel in a screen, vector-based interfaces need only send the mathematical description of the interface. The result is much smaller files and faster transmission. Another benefit is that vector graphics scale much more easily to a variety of different form-factors, whether smaller monitors with constrained screen space on a desktop, or new device formats that we have yet to consider, such as tablet PCs or PDAs.
- **Streaming, buffered execution of code and content.** Macromedia Flash SWF files load using a streaming model, where the first few frames of content are immediately available once they've loaded. This applies whether the file contains code, media, or data. As a result, developers can very easily design their applications and content to quickly load specific portions of the application while the remaining portions are loaded in the background. This improves the perceived and real application performance by giving immediate functionality and visual cues. For example, an application could load its "shell" and basic navigation, enabling the user to take action, while more advanced functions and media load in the background.

- **Code, media and data caching.** Unlike HTML applications that must be generated on the fly for each request, forcing new data and content transmissions, Macromedia Flash files can be cached for fast retrieval from the local disk. SWF files are cached on the local disk just like GIF or JPG files, except SWF files also contain code, media, and data. A complex application might load a dozen SWF files as modules throughout the use of the application. These modules will all be cached locally, including audio and video data contained in the SWF files. When the user returns to the application, it will load immediately without returning to the Internet. The Flash application can then query the server for any new data (say, new weather data) and transmit only that back to the client.
- **Locally persistent state and data.** Macromedia Flash MX includes a feature called SharedObjects, which can be used entirely on the client computer to store complex object data for reuse across or within application sessions. SharedObjects improve application performance by reducing the number of requests made to servers, and the amount of data needed to transmit before a user can begin working. They also enable offline applications that can synchronize data with servers whenever a user is online.
- **High-performance remote data integration.** Macromedia Flash supports integration of data and logic from remote application servers and web services through new technology for integrating application servers and web services with Macromedia Flash. This new model will make developing, debugging, and deploying rich Internet applications with Macromedia Flash easier and increase performance.
- **Dynamic code and media loading.** Like dynamic class file loading in Java, Macromedia Flash allows developers to partition their application and media into components or modules that can dynamically load at runtime. This application modularity supports a higher-performance end-user experience by only requiring an application to load the basic modules necessary to begin working. New in Macromedia Flash MX is the ability to load not only SWF files that contain media, but also raw media itself, such as MP3 and JPG files.
- **Visual transitions and visual state.** While largely a design convention in Macromedia Flash applications, the ability to build an application interface that can change the visual state of one part of the screen, even dynamically loading a new interface widget at runtime without refreshing the whole screen or page, is a major breakthrough in the perceived performance of Internet applications. The current expectation on the web, even for trivial end-user actions, is a jarring, multi-second blank screen as new pages refresh and re-render.

Integrate content, communications and application interfaces into a common environment

Macromedia Flash MX provides an unparalleled environment in terms of the richness of content, media, and application interface elements. Macromedia Flash MX also expands from media and application interfaces into communications. This integration enables the richest applications on the Internet today, and is made possible through the following capabilities:

- **Support for richly formatted text and graphics layout.** Unlike HTML, which relies on obscure formatting techniques such as GIF pixel positioning, frames, and table layout to emulate screen regions, or on style sheets that are plagued with cross-browser challenges, Macromedia Flash provides broad and fine-grained control over text formatting. Text formatting controls include leading, kerning and pixel-precise placement, as well as the ability to create and use custom fonts, system fonts and to map fonts between platforms, allowing for a best of breed experience on any platform. Likewise, graphics and user interface elements can be placed with pixel-level precision, and can be composed of bitmaps, vector drawings, and custom Macromedia Flash components. Strong drawing tools combined with powerful concepts such as levels and layers provide a depth of formatting that is unparalleled in content and applications on the Internet.
- **Standard user interface components.** Macromedia Flash MX includes a standard set of user interface components including scrollbars, scrollpanes, rich text areas, buttons, listboxes, combo boxes, radio buttons, and check boxes, enabling developers to quickly compose applications from standard building blocks, and even use skins to control the look and feel of the components. A range of additional components including windows, tree controls, graphs, tickers, and others are available from the Macromedia Exchange for Flash at the Macromedia website.
- **Easily create custom user interface components and behaviors.** Using the powerful drawing tools in Macromedia Flash MX, developers can easily imagine and create any user interface control that they can imagine, and then add behavior and dynamic data to the controls using ActionScript. There are already hundreds of free user interface components available from the developer community, including spin controls, dial controls, slider bars, tab controls, tree controls, and many others.
- **Streaming audio.** Macromedia Flash MX includes powerful audio support for both MP3 and voice-oriented audio. Macromedia Flash MX can dynamically load and play MP3 content, used for either basic sound events or full-fledged stereo music. Flash also includes an optimized codec for voice audio (Nelly Moser Voice Codec), enabling developers to easily attach the human voice to content and applications. Like other Macromedia Flash content, audio files can begin loading and playing before they are entirely downloaded.

- **Streaming video.** Macromedia Flash MX now includes powerful new video capabilities, allowing developers to add full-motion video to their content and applications. Video can be easily imported from any standard video format, and played back at a variety of speeds and frame-rates. The video support in Macromedia Flash MX has been optimized for low-bandwidth, high-quality playback through the use of the Sorenson Spark Codec.
- **Real-time communications.** Macromedia is introducing new server solutions that will work with Macromedia Flash Player for adding two-way communications, including voice, video and shared data between multiple users.

Provide a powerful and extensible object model for interactivity

Macromedia Flash MX includes one of the richest object models for interactive client applications on the Internet. The degree of control and flexibility provided to developers is enormous, yet still remains accessible to a wide range of skill sets. The object model includes:

- **ActionScript for programming and scripting.** Macromedia Flash MX includes an ECMAScript-compliant programming language called ActionScript. ActionScript provides an object-oriented scripting model for controlling and extending Macromedia Flash applications, as well as a strong base of built in objects unique to the Macromedia Flash environment. Developers familiar with other languages such as Visual Basic, JavaScript, Perl, Java and ColdFusion will feel comfortable with the syntax and model provided by ActionScript.
- **Dynamically modify visual objects.** Almost all of the properties of Macromedia Flash’s intrinsic objects (textfields, graphics, movie clips) can be dynamically modified at runtime using ActionScript. This makes it easy to create highly customizable and adaptable interfaces.
- **Support for prototype-based inheritance.** Developers can implement reusable classes in ActionScript using a technique known as prototype-based inheritance. This enables developers to build more modular applications and reuse code effectively across a team.
- **Event-based programming.** The Macromedia Flash MX object model provides support for creating custom events and event handlers for both visual and non-visual objects. Objects can broadcast events globally or specifically to a given application module, and other objects can listen for or watch events fired by these objects. This event model makes creating complex, high-performance user interfaces possible.
- **Built-in events.** Standard events are provided for mouse, button, audio, video, timeline, keyboard, microphone, camera, and data events, providing developers with the ability to dynamically adapt to changing state and to anticipate activities performed by users or the application itself.

- **Tabbing and Focus APIs.** Macromedia Flash MX includes a new set of APIs for controlling tabbing behavior between form and graphical elements, as well as a core Focus API that allows developers to easily change the cursor's focus on different elements on the screen. This makes applications more usable and productive by helping to guide end-users through the application interface for input and selection.
- **Shape Drawing API.** This is a new API that enables developers to dynamically construct and draw graphical objects at runtime. The Shape Drawing API exposes some of the core rendering internals of the Macromedia Flash runtime to developers, enabling powerful graphical applications such as graphing and reporting for data visualization, games, or even simple graphics creation tools.
- **Accessibility API.** Delivering accessible applications for the hearing or visually impaired is made possible in Macromedia Flash MX with the introduction of a powerful set of APIs for exposing content and applications to screen readers. Corporate and government developers who need to build applications that are Section 508-compliant can rely on Macromedia Flash MX to help accelerate their development.

Enable rapid application development through components and re-use

Modern application development demands common patterns and behaviors to be easily encapsulated into components that can be shared and re-used by other developers. Until now, building true visual components for Internet applications has been nearly impossible. With Macromedia Flash MX, we've introduced a new Macromedia Flash Component model that enables powerful capabilities to be shared and used by developers of any skill set. The Macromedia Flash Component model provides an end-to-end approach for design, development and deployment of components.

- **Create reusable visual components with properties, methods and events.** Macromedia Flash Components can define customizable properties, methods, and events. Component properties can be set at design-time. These could be visual properties such as style-oriented properties, or logic properties such as custom behaviors for the component. Macromedia Flash Component methods provide well-defined APIs for developers to easily integrate and manipulate the component in a customizable way within their own applications. Component events enable developers to watch for changes in the state or behavior of the components and trigger custom actions.
- **Package components for drag-and-drop use in visual development.** Macromedia Flash Components can be easily redistributed through the Macromedia Exchange for Flash, and can be installed and visible to end users in the Macromedia Flash MX Components Panel. Developers can then drag-and-drop components into their interfaces, and even drag components onto other existing interface elements to apply behaviors dynamically. Once the components are dragged into the interface, developers can easily set properties and parameters for them at design time.

- **Customizable design-time interfaces.** Component developers can easily build custom design-time interfaces for their components. Macromedia Flash Components can include a custom Property Inspector that provides a rich configuration interface for the component's settings and behaviors. Macromedia Flash Components can also include a LivePreview SWF file that provides an in-line rendering of how the component will look in the application.
- **Data-bound components.** Developers can easily make data-bound components using XML or new server-data integration technology. Component end users can then connect the visual components to their own server-side scripts to provide custom dynamic data. More details on building data-bound components will be available in coming months.

Enable the use of web services and data services provided by application servers

Rich clients are made much more valuable when combined with logic and data delivered from application servers and XML web services. Indeed, the new services-centric application model being ushered in by web services is the perfect compliment to rich-client environments like Macromedia Flash MX. Over the coming months, Macromedia will provide details on new server solutions for building rich Internet applications with Macromedia Flash, Macromedia ColdFusion, and other application servers:

- **Develop Macromedia Flash services using Macromedia ColdFusion MX, Java and Microsoft .Net.** Macromedia Flash MX applications can easily invoke logic and get data from services built and deployed on Macromedia ColdFusion MX Server. Macromedia Flash can also use services built with standard Java Servlets, Java Classes, EJBs, and Microsoft's .NET components.
- **Coherent model for connecting rich clients to web services.** The services-based application model provided by XML web services promises to marshal in a new wave of loosely coupled applications, where component re-use and application integration flourish. Rich clients such as Macromedia Flash MX provide a complementary model to web services by combining user interfaces with behaviors and data hosted by services in the network. With Macromedia Flash MX and ColdFusion MX, developers will be able to easily connect their rich client applications with SOAP-accessible web services hosted over the Internet.
- **Real-time messaging between client and server.** As part of forthcoming communications server technology, Macromedia Flash MX applications will be able to integrate two-way, real-time communications and data.
- **Support for integrating data via HTTP, sockets and XML.** Macromedia Flash MX continues its support for loading data from an URL using the loadVariables command, as well as including improved support for loading and using XML documents. With these basic forms of integration, Flash MX can work with data generated from almost any server-side application environment.

Embrace connected and disconnected clients

While the Internet of the past five years has been largely about connected web applications, where the user is always online, using a web browser, and connected to a web server, it has lacked some common application models that were prevalent in the desktop world, such as local desktop applications. Furthermore, it has lacked the future potential of occasionally connected and notification-based applications. With Macromedia Flash MX, a new realm of client-side application models are made possible through a number of innovations:

- **Ability to run as a stand-alone, offline application.** Macromedia Flash MX applications can be built to include both the application and its data, as well as the full Macromedia Flash Player runtime, enabling developers to create stand-alone applications that can operate offline, outside of the context of a web browser. While this has been possible in past versions of Macromedia Flash, when combined with local data storage it becomes much more powerful.
- **Local data storage via client-side SharedObjects.** Offline or occasionally connected applications can use SharedObjects to store and work with data on the local disk. End users have control over how much disk space to allow the application to use, and developers can use that disk space to store complex object data.
- **Persistent, real-time connections.** As part of a new communications server technology, Macromedia Flash MX will be able to keep open connections for real-time shared data and two-way voice and video. Developers can also use XML sockets to keep live connections to servers for building applications with persistent connectivity.

Enable easy deployment on multiple platforms and devices

The promise of the Internet is content and applications anywhere and on any device. The incredible growth of the Internet was fueled by the fact that web sites and applications can be viewed on any desktop operating system. In the coming years, we are seeing an explosion in the types of devices that are connected to the Internet, creating even greater challenges for content and application developers. Macromedia Flash Player provides the widest support possible for development and deployment of content and applications on multiple client devices:

- **A common runtime across operating systems, browsers and chip architectures.** Internet client devices typically consist of a combination of operating system, microchip architecture, and additional Internet software such as a web browser. The Macromedia Flash Player runtime environment works the same across any combination of these components, ensuring that developers can easily create applications that work the same across these client platforms. The programming model and runtime work exactly the same for developers, regardless of the device platform.

- **Support for major desktop platforms.** Macromedia Flash Player includes broad support for all major versions of Windows, the Macintosh, Linux, and Solaris, addressing all popular desktop and workstation environments. Developers seeking to build rich client-side content and applications that work across multiple versions of Windows and Macintosh can do so easily with the reliable deployment of Macromedia Flash Player.
- **Broad support for industry-leading devices.** Macromedia Flash Player extends well beyond desktop computers into new categories of device platforms. Macromedia Flash Player is available for Windows CE, PocketPC, Embedded Linux, QNX, Symbian's Epoch OS, OpenTV, and a variety of other device platforms. Based on support for these platforms, Macromedia Flash Player is finding its way into many popular consumer and corporate devices. With support from leading vendors like Microsoft, Sony, Motorola, Samsung, Symbian, Moxi, and Nokia, Macromedia Flash Player is enhancing smart-phones, PDAs, set-top boxes, tablet computers, and gaming consoles.
- **Easy to design interfaces for multiple form-factors.** While the runtime is consistent across platforms, often the form-factors of devices are quite different. For example, the screen and user input model on a personal computer differs radically from that of a personal digital assistant. With Macromedia Flash MX, developers can use templates that constrain the design environment to the physical properties of a device. Templates can provide default screen sizes, color constraints, and can even include common libraries of assets that can be shared across different device types.

Conclusion

Macromedia Flash MX is a breakthrough in delivering effective experiences to end users, enabling rich Internet applications that blend content, application logic and communications. As rich clients emerge to make the Internet more usable and enjoyable, Macromedia Flash MX provides a solid architecture for developers embracing the future.