macromedia® white paper

Building Rich Internet Applications with Macromedia Flash MX and ColdFusion MX

by Andrew Guldman

Fluid, Inc.
www.fluid.com
info@fluid.com

May 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, Inc.
600 Townsend Street
San Francisco, CA 94103
415-252-2000

Contents

Table of Figures

1. Executive Summary

1.1. Beyond HTML

The World Wide Web has grown up in a hurry. What was initially a collection of static pages has evolved into a system of sophisticated and highly functional Internet applications. Unfortunately, not all of the advancements have been beneficial to the end user. With the delivery of more advanced applications, users are increasingly forced to endure slow and confusing interfaces. The most successful Internet applications are tailored to the specific needs of the end user: they are engaging, responsive, and provide useful functionality.

This white paper describes the next generation of Internet application development. The application architecture outlined in this document relies on the use of the ubiquitous Macromedia Flash(tm) Player to deliver engaging, responsive, and powerful interactivity akin to that of desktop software systems. Because Macromedia Flash Player is installed on more than 98 percent of today's web browsers, it is ideally suited for all classes of Internet applications, from public-facing websites to employee intranets.

The presentation logic of these applications is written using Macromedia Flash MX authoring software. At run time, Macromedia Flash Player 6 communicates directly with a server application that handles business logic and data access. The server application described in this paper is built with Macromedia ColdFusion MX(r) Server, although other server technologies can be used as well. Macromedia Flash Player can effectively exchange data with an application server at run time over HTTP and HTTPS.

1.2. Benefits

The main motivation for building applications using Macromedia Flash MX is to provide a highly interactive and enjoyable user experience that is difficult to achieve using HTML. Macromedia Flash MX also helps reduce server load and network bandwidth use by offloading presentation logic to the client and eliminating the need for continual page refreshes, which are unavoidable when using HTML by itself.

1.2.1. Improved Interactivity

While HTML provides a great way to display textual and tabular data in linear application workflows, Macromedia Flash MX applications are not burdened by the HTML page model and can offer richer interactivity. With Macromedia Flash, Internet applications can behave like desktop applications, instead of merely being a series of linked pages. Macromedia Flash facilitates the development of insightful, engaging, and highly interactive views into complex data, through techniques such as the filtering or aggregation of data into visual components such as graphs. Unlike HTML applications, Macromedia Flash does not incrementally load individual pages from the server; instead, Macromedia Flash applications move instantaneously from one view to another without visually jarring delayed page redraws. Macromedia Flash applications can also increase end-user productivity by combining disparate business systems into a streamlined, single-screen application.

1.2.2. Reduced Server Load

The entire presentation layer of a Macromedia Flash MX application runs in Macromedia Flash Player 6, which means that no application server processing resources are spent on data formatting. Because the server focuses strictly on the delivery of raw content, rather than on formatting, the processing load on the server is reduced.

1.2.3. Conservation of Bandwidth

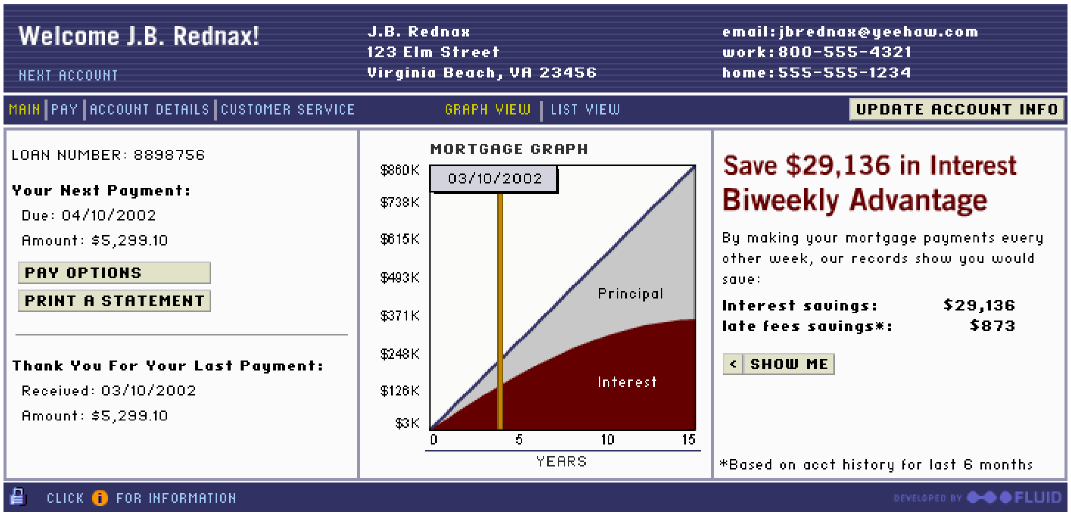
Because the application's presentation layer resides in Macromedia Flash Player 6, repetitive page redraws are avoided, and run-time communication between the client and server is kept to a minimum. Only immediately relevant data is sent from the application server to Macromedia Flash Player, and it does not include any formatting information. This data can then be cached in Macromedia Flash Player for future use, resulting in less network traffic. In addition, the Macromedia Flash file format is thoroughly compressed and automatically streamed to Macromedia Flash Player. These factors combine to further reduce the network bandwidth required.

1.3. This White Paper

This white paper discusses how enterprises can effectively use Macromedia Flash MX and ColdFusion MX to develop Internet-based business applications. We look closely at the key differentiators between Internet applications built using Macromedia Flash and those built using HTML. We also examine the relationship between Macromedia Flash Player and the application server tier.

2. Application Example

The Home Mortgage Application presented as an example in this section illustrates the benefits of building Internet applications with Macromedia Flash MX and ColdFusion MX. To view this sample application, please visit Macromedia's Business Center at www.macromedia.com/resources/business.



Welcome J.B. Rednax!

NEXT ACCOUNT

J.B. Rednax
123 Elm Street
Virginia Beach, VA 23456

email:jbrednax@yeehaw.com
work:800-555-4321
home:555-555-1234

LOAN NUMBER: 8898756

Your Next Payment:
Due: 04/10/2002
Amount: $5,299.10

Thank You For Your Last Payment:
Received: 03/10/2002
Amount: $5,299.10

Save $29,136 in Interest
Biweekly Advantage

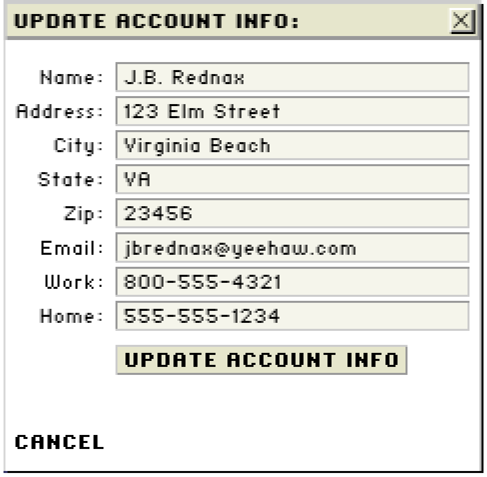
By making your mortgage payments every other week, our records show you would save:

Interest savings:	$29,136
late fees savings*:	$873

*Based on acct history for last 6 months

Figure A: *Home Mortgage Application*

The main view of the Home Mortgage Application (Figure A) contains four main panels. The top (blue) panel shows account information for a home mortgage loan. This information can be edited in a pop-up window (Figure B) by clicking the Update Account Info button.

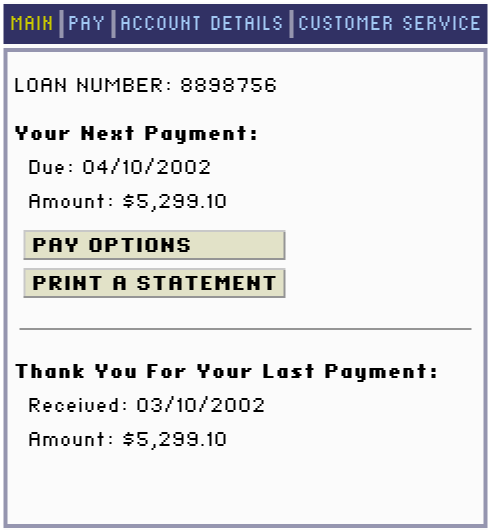


UPDATE ACCOUNT INFO:

Name:	J.B. Rednax
Address:	123 Elm Street
City:	Virginia Beach
State:	VA
Zip:	23456
Email:	jbrednax@yeehaw.com
Work:	800-555-4321
Home:	555-555-1234

Figure B: *Home Mortgage Application Update Account Window*

The left section of the main application view shows detailed payment and customer service information in four separate panels. The Main panel (Figure C) contains information about the most recently received payment, as well as about the next payment due. The Pay panel lets the user see the impact of making additional principal payments to the loan and gives the user the opportunity to actually make a payment. The Account Details panel shows total and year-to-date account summaries. The Customer Service panel displays phone numbers, provides e-mail links, and offers access to live chat support.



LOAN NUMBER: 8898756

Your Next Payment:

Due: 04/10/2002

Amount: $5,299.10

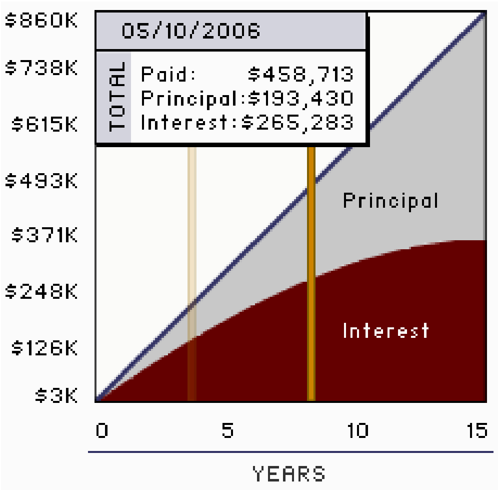
Thank You For Your Last Payment:

Received: 03/10/2002

Amount: $5,299.10

Figure C: *Home Mortgage Application Main Payment Window*

The center panel presents an interactive chart containing detailed information about historical and future payments for the loan. The cumulative payment amount is displayed on the y-axis; the number of years elapsed since the inception of the loan is displayed on the x-axis. When the mouse is rolled over the chart, the vertical gold bar moves with the mouse to the nearest payment in the graph. When there is a pause in mouse movement, detailed payment information is displayed in a payment box (Figure D).



05/10/2006

Paid: $458,713
Principal: $193,430
Interest: $265,283

Figure D: *Home Mortgage Application Chart Rollover*

The right panel of the main application view describes a promotion available for loan holders called the Biweekly Advantage, which allows the borrower to pay off the loan early and save interest payments by paying every other week instead of monthly. To clearly see the financial benefits of the Biweekly Advantage, the borrower can click Show Me; the projected biweekly payment schedule is overlaid on the current loan graph (Figure E), and savings data is displayed in an expanded version of the payment box when the mouse pauses over the chart.

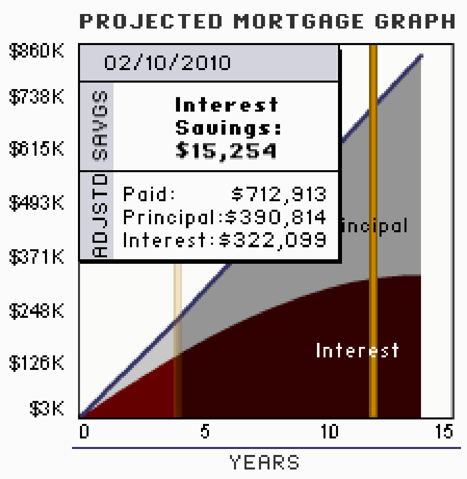
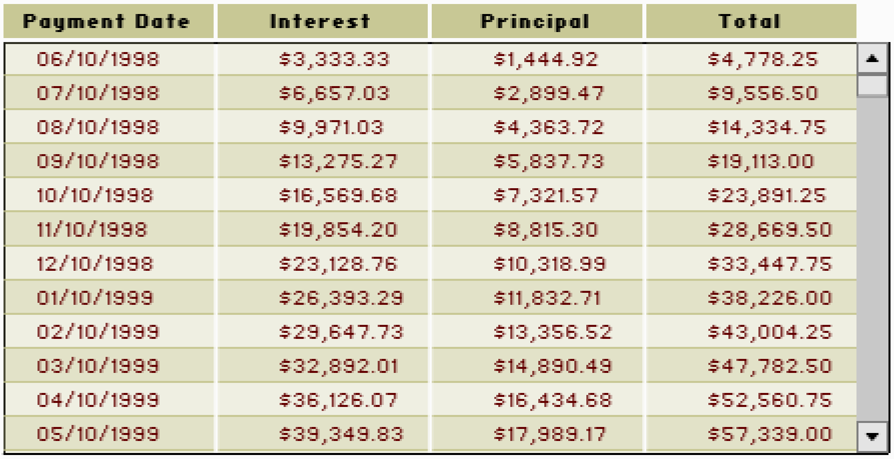


Figure E: *Home Mortgage Application Biweekly Advantage Overlay*

In the center panel, by clicking List View, borrowers can view loan payments in a list instead of a chart (Figure F). This area of the application provides an alternate view of the same data that is displayed in the chart. The user can scroll up and down through the list of payments.



Payment Date	Interest	Principal	Total
06/10/1998	$3,333.33	$1,444.92	$4,778.25
07/10/1998	$6,657.03	$2,899.47	$9,556.50
08/10/1998	$9,971.03	$4,363.72	$14,334.75
09/10/1998	$13,275.27	$5,837.73	$19,113.00
10/10/1998	$16,569.68	$7,321.57	$23,891.25
11/10/1998	$19,854.20	$8,815.30	$28,669.50
12/10/1998	$23,128.76	$10,318.99	$33,447.75
01/10/1999	$26,393.29	$11,832.71	$38,226.00
02/10/1999	$29,647.73	$13,356.52	$43,004.25
03/10/1999	$32,892.01	$14,890.49	$47,782.50
04/10/1999	$36,126.07	$16,434.68	$52,560.75
05/10/1999	$39,349.83	$17,989.17	$57,339.00

Figure F: *Home Mortgage Application Payment List*

The presentation tier of the application loads all the loan data dynamically from a ColdFusion MX application server, which in turn pulls the data from a relational database. Macromedia Flash Player displays an interactive user interface designed to help the end user understand and manipulate the data.

3. Macromedia Flash MX Application Architecture Overview

Macromedia Flash MX is a perfect environment for developing sophisticated, easy-to-use Internet applications that enhance end-user productivity and integrate with a variety of back-end systems. These Internet applications run in standard web browsers through Macromedia Flash Player 6. Macromedia Flash Player exchanges data with an application server that provides business logic and access to data stored in relational databases and legacy systems. Because Macromedia Flash Player is installed on virtually all web browsers (more than 98 percent), and because Macromedia Flash applications can integrate into virtually any server architecture, applications built with Macromedia Flash fit nicely into existing corporate infrastructures.

Macromedia Flash can be used to build a wide range of sophisticated Internet business applications, ranging from visual analysis tools to transactional data processing systems. The architecture particularly excels when building applications with the following requirements:

- High-productivity applications: These applications involve building a small number of finely tuned data entry screens that are used repetitively. The application must draw pages quickly, must constantly show accurate and timely information, must quickly and helpfully validate data, and must interact with multiple back-end systems.
- Extremely usable applications: These applications must be easy for untrained individuals to use effectively. The user interface must be simple and consistent and provide feedback promptly. Complex tasks must be simplified.
- Data analysis applications: These applications help the user gain insight into complex data. Data must be presented intuitively and interactively, often in multiple modes. Large quantities of data must be condensed and presented quickly to the user.

The Home Mortgage Application meets several of these requirements. It targets end users with limited computer and financial experience, and it must display a large amount of complex financial data in a clear and concise manner. The user interface makes it possible for the user to interact with loan data in both graphical and textual modes.

3.1. Comparing HTML and Macromedia Flash MX Architectures

Why do we need a new architecture to build these web applications? Can't we perform these same tasks using an application server to deliver HTML to a web browser? This section of this white paper compares and contrasts HTML and Macromedia Flash MX architectures, particularly in the context of applications geared toward productivity, usability, or data analysis.

Obviously, the HTML application architecture has its strengths. Prominent among them are the ability of HTML to run consistently and securely on leading operating systems and browsers, and the minimal administrative overhead that HTML requires. There is also a large and well-established community of HTML developers. However, Macromedia Flash applications share these advantages and also offer some additional benefits.

3.1.1. Usability

HTML provides a standard set of simple form widgets (such as check boxes and lists); however, it does not elegantly support more complex user interface controls such as tree views, menus, tabbed panels, and pop-up windows. Further, HTML doesn't easily allow the customization of standard widgets. To create advanced or customized user interface controls, HTML developers must use supplemental technologies such as JavaScript(tm), dHTML, Java applets, and ActiveX(r) controls. Each of these technologies is challenged by some combination of cross-browser incompatibility, poor security, slow performance, development complexity, and limited functionality.

Because HTML sites are page based, they demand complete page redraws when forms are submitted to a server as well as when the user navigates from one section of an application to another. Even if only a small portion of a page actually changes, the entire page must be retrieved from the server and redrawn. This constant redrawing slows the navigation between sections of an application and leads to a discordant user experience.

Macromedia Flash MX provides an alternate paradigm for constructing Internet application user interfaces. In contrast to an HTML application's series of linked and separately downloaded pages, a Macromedia Flash user interface is a single, unified application that manages the user experience in an uninterrupted manner. Interactive capabilities like drag and drop; mouse and keystroke trapping; server communication; animation; pop-up windows; and fine control over fonts, colors, and shapes help developers put together exceptionally interactive and usable systems. Moreover, since Macromedia Flash doesn't require that pages be loaded from a server as the user moves from one section of a web application to another, new pages snap up immediately. These characteristics raise the usability of a Macromedia Flash application to the level of a client-server desktop application, while retaining the administrative advantages of a distributed thin-client web application.

Sections of the Home Mortgage Application clearly illustrate the usability advantages that Macromedia Flash MX offers. The interactive chart, for example, could not be easily created in HTML. With HTML, the end-user would be required to click a Submit button, forcing the browser to reload a newly generated HTML page with the data that the user requested. In the Macromedia Flash application, the chart display is refreshed when the data set is updated. This client-side processing allows data to be displayed and manipulated in a flexible way. For example, the overlay of Biweekly Advantage data on the chart dynamically updates the interface, with the supplemental projected payment schedule layered over it. For the end user, the result feels as seamless as a desktop application; the user no longer has to wait for the HTML page to reload.

When the Macromedia Flash Player 6 tier does interact with a server, it sends and receives only essential data, which results in smaller data transfers than those that typically occur with HTML web applications. Formatting and presentation information is notably absent from the data that the application server provides to the client. The application server manages data; Macromedia Flash Player is responsible for its presentation.

3.1.2. Performance and Scalability

Macromedia Flash MX applications impose less of a load on an application server than do HTML applications. The two primary reasons that Macromedia Flash applications perform better and scale better are the alternatives they offer to the HTML page load problem and the management of session state data on the server.

3.1.2.1. The HTML Page Load Problem

All pages displayed in an HTML application must be downloaded one at a time from an application server. This means that, during the course of a typical user session, a large number of HTTP connections are opened to download many pages and images. Each of these downloads creates network traffic. Much of the data that is downloaded is repetitive, such as navigation bars, headers, and footers that are the same on many of the pages. Web browsers effectively cache images and URLs but not these repetitively used HTML fragments.

When Macromedia Flash MX is used, an entire application containing many complex screens and animations can be retrieved in a single HTTP connection to a server. Shared design elements such as navigation bars are downloaded once and can be used throughout an application without additional downloads. These characteristics reduce network traffic in Macromedia Flash applications. Lower network traffic carries two primary benefits. First, a web application is more responsive and user friendly if there are fewer trips to the server. Second, the load on the servers and network infrastructure at the hosting facility is reduced, which minimizes the cost of hosting the web application.

3.1.2.2. The Management of Session State Data

A session is a sequence of page views that comprise a single visit to a website. The session state is the information that is accessible to all of the pages in a session, the existence of which helps ensure a smooth user experience. Consider the example of a login form on a secure website. After a user logs in, subsequent pages often display the user ID. The user's login information typically is stored in the session so that it can be easily used in subsequent pages in that session.

HTML applications offer three primary mechanisms for tracking the session state: the application server, cookies on the client, and the URLs of page requests. HTML applications typically store much of their session state data on the application server because this approach is secure and offers good cross-platform compatibility. However, this technique affects the scalability of web applications because the data associated with each session consumes memory on the application server, a problem exacerbated by large numbers of concurrent sessions.

Since Macromedia Flash Player stays loaded throughout a Macromedia Flash MX application session, the state associated with that session can easily be maintained in Macromedia Flash Player. This trims the quantity of data that is stored in each session on the server, which decreases memory consumption on the server and allows each application server to accommodate more concurrent clients.

The Update Account Info section of the Home Mortgage Application illustrates how the use of Macromedia Flash reduces network traffic and thereby improves the performance and scalability of a web application. When the user clicks the Update Account Info button, the edit screen pops up instantly. When the form is submitted, the updated information is immediately visible in the application. Macromedia Flash Player manages all of the session state associated with the form. The result is a responsive application that conserves application server resources.

3.1.3. Development Workflow

A fundamental problem facing Internet application development teams is separating the design of the presentation layer from the development of the business logic. Ideally, the designers create the user interfaces, and the engineers build the technical underpinnings of a site. Various technologies have attempted to address this issue, but none have been truly successful. All existing HTML-based development languages and environments are cursed with an intermingling of code and HTML. They generally merge business logic into the presentation layer using customized scripting and tag mechanisms.

Macromedia Flash MX simplifies the server development of web applications by clearly separating the construction of the presentation layer in Macromedia Flash from the implementation of business logic on an application server. All interactions between Macromedia Flash Player and the application server are conducted through a set of well-defined messages that exchange data without any presentation formatting. When server engineers and Macromedia Flash developers agree on the structure and content of the messages sent between the tiers, they have thereby drawn clean boundaries between their areas of responsibility.

The development of the Home Mortgage Application involved four primary roles: information architect, Macromedia Flash designer, Macromedia Flash application developer, and ColdFusion developer. The information architect and Macromedia Flash designer collaborated to create the visual aspect of the user interface. The Macromedia Flash application developer brought the user interface to life using ActionScript. And the Macromedia Flash application developer worked closely with the ColdFusion developer to define the messages sent between Macromedia Flash and ColdFusion. The ColdFusion developer was responsible for the entire server implementation, including creation of the database, database querying, and the return of data in a structure that Macromedia Flash Player can use.

3.2. Integration into Existing Environments

Macromedia Flash MX can be used to build applications that integrate into a wide variety of IT infrastructures. This white paper describes using the ColdFusion MX Application Server to build the back end to support a Macromedia Flash application. ColdFusion is an eminently suitable platform to use for the following reasons:

- It streamlines integration with Macromedia Flash MX via Macromedia Flash Remoting.
- It offers a highly productive development environment.
- It offers open integration with databases and other server resources.
- It is a proven, high-performance architecture.

Other server technologies such as J2EE, ASP, .NET, PHP, and Perl can also be used to effectively build the server component of Macromedia Flash applications. An enterprise can reap the benefits of Macromedia Flash applications without needing to adopt any new server technologies.

One benefit of Macromedia Flash applications is that they provide simplified access to existing IT systems. For example, ColdFusion can easily interact with systems built on a variety of technologies, including CORBA, Java, and COM. Macromedia Flash Player can then present the data gathered from these different systems in a straightforward and unified way for the end user. Consequently, a developer can build Macromedia Flash applications that can pull dynamic information from databases without disrupting any part of an existing technical infrastructure.

3.2.1. Macromedia Flash Player 6 Deployment

Macromedia Flash Player 6 can be deployed in many ways, which adds another dimension of flexibility in the development of Macromedia Flash MX applications. When it runs in a browser, Macromedia Flash is usually embedded in an HTML page. This page might be a mere shell with the single purpose of running the Macromedia Flash file, or it might be a full-featured HTML page. In the latter case, this would allow the Macromedia Flash application to cooperate with a complex HTML application. Macromedia Flash provides bi-directional communication capabilities with JavaScript so it can interact with the HTML page on which it resides.

Macromedia Flash Player can also run outside a browser in a stand-alone player. Stand-alone players offer enhanced access to the resources of the client computer, such as the file system and the registry. They also permit flexibility in the way that executable Macromedia Flash movies are packaged. For example, movies can be screen savers, they can have custom icons, and they can have transparent backgrounds.

4. Macromedia Flash MX

This section of the white paper focuses on the design of user interfaces of business web applications using Macromedia Flash MX. We look at different types of user interfaces commonly found in business applications. We also discuss techniques for efficiently interacting with an application server: how to send data back and forth, and how to use that data effectively.

Macromedia Flash and the ActionScript programming language put many sophisticated software development techniques in the hands of developers so they can quickly and easily build user interfaces for complex business systems. Some useful techniques include the following:

- The decoupling of visual components from the data that they represent. The visual portion of Macromedia Flash user interfaces (movie clips) can be separated from the data that they represent (ActionScript objects). This separation breaks the system into smaller components that can be more easily built, maintained, and integrated.
- ActionScript supports object-oriented development techniques such as inheritance, interfaces, delegation, and composition. Macromedia Flash applications can be intelligently divided into components that interact with each other using these mechanisms.

4.1. Forms

Forms are used to solicit information from the user. They can consist of standard text input fields or custom widgets such as sliders or tree views. Macromedia Flash MX excels at building forms, particularly those with complex widgets or interactive forms that change as the customer uses them. Macromedia Flash boasts full-featured text fields that can be manipulated with ActionScript as well as a rich set of user interface components such as radio buttons, tree view controls, and list boxes.

4.2. Charts and Graphs

Macromedia Flash MX is unique in its ability to build interactive charts and graphs to present complex data. The chart in the center panel of the Home Mortgage Application, for example, helps the user visualize and interact with a large quantity of complex mortgage data. Macromedia Flash MX includes standard drag-and-drop chart components for bar, line, and pie charts.

4.3. Animation and Multimedia

The appropriate use of animation and multimedia capabilities within Macromedia Flash MX can improve the usability of business applications. Subtle touches like simple sounds and the fading in or out of user interface components can make an application more engaging.

4.4. Transferring Data

To display and process useful information, Macromedia Flash Player 6 must retrieve data from and send data to an application server. Macromedia Flash Player 6 normally sends and receives messages using the standard HTTP protocol on port 80. Since corporate firewalls normally open port 80 to their servers, this means that Macromedia Flash applications can typically operate in a standard IT infrastructure. Macromedia Flash can also use HTTPS with SSL encryption for applications that demand extra security, such as e-commerce, financial, or medical applications. These communications occur on the standard HTTPS port, 443, which corporate firewalls also typically open.

5. Macromedia Flash Remoting

Macromedia Flash Remoting is a new remote-method invocation technology from Macromedia that facilitates the connection of Macromedia Flash MX applications to application servers such as Macromedia ColdFusion and Macromedia JRun. The typical sequence of steps involved in using Macromedia Flash Remoting at run time is as follows

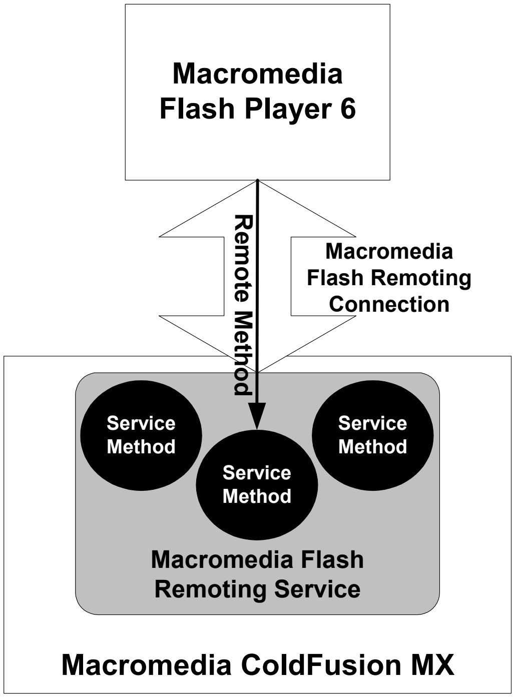

Macromedia Flash Player 6
Remote Method
Macromedia Flash Remoting Connection
Service Method
Service Method
Service Method
Macromedia Flash Remoting Service
Macromedia ColdFusion MX

Figure G: *Macromedia Flash Remoting Run-Time Connections*

- Macromedia Flash Player creates a persistent connection to a remote service on a Macromedia Flash Remoting server.
- Macromedia Flash Player invokes a method on the remote service. Parameters are serialized and sent to the server in a new optimized binary format called Action Message Format (AMF). The speed of the serialization, de-serialization, and transmission of Action Message Format messages is a key component of Macromedia Flash Remoting.
- The Macromedia Flash Remoting server de-serializes the parameters and then performs the indicated operation. The return value from the operation is sent back to Macromedia Flash Player, again in optimized Action Message Format.

- Macromedia Flash Player automatically de-serializes the return value from the remote method invocation and processes the result in an appropriate manner.

Before Macromedia Flash Remoting existed, server interactivity with Macromedia Flash had to be implemented using the loadVariables function or XML data. Macromedia Flash Remoting offers two substantial benefits over these techniques.

First, Macromedia Flash Remoting saves development time. Parameters and return values transmitted through Macromedia Flash Remoting are ActionScript objects, which means that rich hierarchical data can be sent back and forth without having to build, parse, or navigate XML trees on the client or the server. The debugging tools provided with Macromedia Flash Remoting further cut development time. Macromedia Flash Remoting has been architected to fit into the Data Provider API used by many of the Macromedia Flash MX components, which further streamlines development. Because of this foresight, return values from Macromedia Flash Remoting methods can plug right into Macromedia Flash MX UI and charting components using connective technologies such as DataGlue.

The second major benefit of Macromedia Flash Remoting is improved run-time performance. Macromedia Flash Remoting serializes data in Action Message Format, which is an optimized binary format. Data serialized in Action Message Format is smaller than equivalent data serialized in XML, which leads to less data transmission time, as well as less time spent serializing and de-serializing the data.

Macromedia Flash Remoting consists of three main pieces: the code that runs in Macromedia Flash Player, the services defined in an application server, and extensions to the Macromedia Flash MX authoring environment.

5.1. Macromedia Flash Player 6

The standard Macromedia Flash Player 6 includes all of the functionality necessary to use Macromedia Flash Remoting. No additional software needs to be downloaded or installed on the client. A visitor to a page will never even know that Macromedia Flash Remoting is being used. All of the networking, serialization, and de-serialization capabilities are part of the standard Macromedia Flash Player 6.

5.2. Application Server

Macromedia Flash Player connects to an application server through Macromedia Flash Remoting to access services implemented on the application server. The Macromedia Flash Remoting server runs on Macromedia ColdFusion MX, Macromedia JRun 4, Microsoft .NET, and Java. Each of these platforms has its own syntax for creating Macromedia Flash Remoting services. The Macromedia Flash Remoting server is installed automatically as a part of Macromedia JRun 4 and ColdFusion MX, and stand-alone versions are available for other implementations.

5.3. Macromedia Flash MX Authoring

Installing Macromedia Flash Remoting enhances the Macromedia Flash MX authoring environment. Several ActionScript files are installed, as are two important debugging tools. Together, these additions enable the Macromedia Flash MX developer to build applications that connect to Macromedia Flash Remoting services on the server.

The key top-level ActionScript files are NetServices.as, RecordSet.as, and NetDebug.as. Including NetServices.as in a movie gives the movie the basic Macromedia Flash Remoting functionality. RecordSet.as lets Macromedia Flash MX manipulate query objects that are returned from ColdFusion MX through Macromedia Flash Remoting. NetDebug.as activates the debugging tools that are included with Macromedia Flash Remoting.

Macromedia Flash Remoting includes two visual debugging tools: the NetConnect Debugger and the Service Browser. The NetConnect Debugger displays Macromedia Flash Remoting activity such as connections made, messages sent, responses received, and errors that have occurred. The Service Browser displays the available Macromedia Flash Remoting services and the methods offered by those services.

6. Macromedia ColdFusion MX

The main role of an application server like ColdFusion MX in a Macromedia Flash MX application is to exchange data with Macromedia Flash Player 6. To satisfy requests from Macromedia Flash, the application server sends and retrieves data to and from relational databases and enterprise systems. In contrast to what it does in HTML architecture, the application server does not format the data for presentation.

Macromedia Flash MX can be used in conjunction with all the leading web application server development platforms, including ColdFusion, IIS/ASP, .NET, J2EE, PHP, and Perl. We elected to develop the Home Mortgage Application using ColdFusion MX because of its native support for Macromedia Flash Remoting, and because it is an excellent platform for rapidly developing sophisticated web applications. The most prominent strengths of ColdFusion are explored in the following sections.

6.1. Integration

ColdFusion MX facilitates integration with a variety of systems, including enterprise databases and legacy applications. As a result, developers can easily connect a variety of back-end systems to their Macromedia Flash MX applications.

6.1.1. Relational Databases

ColdFusion can connect to the leading relational database platforms, including, but not limited to, Oracle(r), Microsoft SQL Server(tm), and Sybase(r) using ODBC, OLE DB, or optimized native drivers. Retrieving data from a database is extremely easy in ColdFusion.

6.1.2. Java

ColdFusion provides a variety of ways to integrate Java code with ColdFusion applications. JavaBeans(tm), EJBs, applets, servlets, and Java custom tags can all be woven into ColdFusion pages. Each of these mechanisms can be used to connect to internal or external systems with Java APIs. These systems might include distributed applications such as internal enterprise systems, partners' B2B extranet systems, online payment systems, or content management repositories.

6.1.3. COM Objects

The COM capabilities of ColdFusion Server allow the developer to leverage the large universe of products that similarly feature a COM interface. These products include data analysis tools, scheduling systems, mail servers, desktop productivity tools, and a world of other applications.

6.1.4. CORBA

The Common Object Request Broker Architecture (CORBA) is a leading infrastructure for communicating with legacy systems, particularly in large enterprises. As with other object standards, developers can easily reuse CORBA objects from within ColdFusion and thereby interact with legacy mainframe systems.

6.2. Unobtrusive Architecture

ColdFusion MX can be installed alongside virtually any existing corporate IT infrastructure without any adverse effects. ColdFusion can be used to extract data from a disparate range of existing systems and to present that unified data to an application running in Macromedia Flash Player 6. Macromedia Flash MX applications can be developed for private intranet use, where they will run on existing workstations, as well as for public Internet use (Figure H).

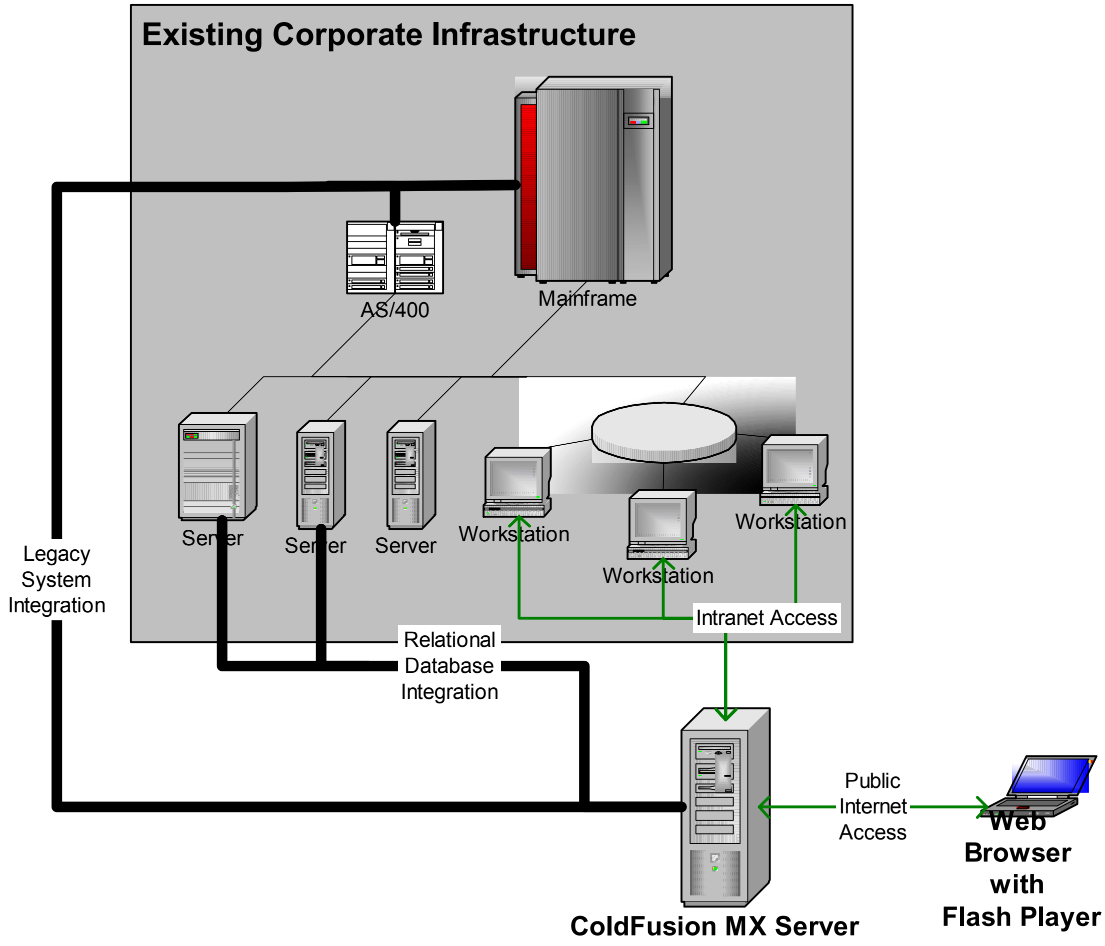


Figure H: *Macromedia Flash MX Application in an Existing IT Infrastructure*

6.3. Rapid Development

ColdFusion MX provides standard tags to perform many common web development tasks: sending e-mail, transferring files via FTP, interacting with LDAP servers, scheduling tasks, and much more. In addition, ColdFusion facilitates the development of reusable custom tags in ColdFusion Markup Language, Java, and C++. Common application-specific tasks can be handled consistently with minimal effort by creating appropriate custom tags. The Home Mortgage Application uses custom tags to calculate mortgage payment schedules.

6.4. Scalable and Reliable

ColdFusion MX supports the major server operating systems, including Windows(r), Linux(r), and Solaris(tm). It provides a high-performance engine that has been proven at many of the Internet's largest sites, including Half.com, OneWorld Alliance, Hertz, and FAO Schwartz.

In addition to delivering high-performance processing, ColdFusion MX enables companies to build highly reliable systems. ColdFusion MX Server includes an integrated load-balancing and failover solution, while also providing open integration with hardware-based load-balancing solutions, such as Cisco(r) LocalDirector.

Finally, ColdFusion MX Server provides built-in management tools that enable companies to monitor the health of their applications. Using this capability, administrators can be notified at the first sign of a problem, such as a failed disk drive, giving them time to react before the application goes down.

7. Managing Macromedia Flash MX Development Projects

IT managers with experience overseeing HTML projects will be on familiar ground when supervising Flash MX development projects. The developer roles, testing strategies, deployment, and hosting of Flash MX applications are similar to those for HTML web applications.

7.1. Resource Planning

The roles involved in a Macromedia Flash application project are similar to those in an HTML project, with three main differences. First, since the bulk of the user interface is developed in Macromedia Flash rather than HTML, Macromedia Flash expertise is needed instead of HTML skills for interface design. Second, there is a cleanly marked boundary between the responsibilities of user interface designers and server application developers. This separation allows both groups to work more efficiently. Finally, the work of server application developers is streamlined because they no longer need to include presentation formatting in their pages. They normally interpret and create messages that consist of structured data rather than formatted HTML pages.

7.1.1. Information Architect

Information architects determine the structure of the content of the system as a whole and of the specific screens within the application. Their work takes place mostly at the beginning of the project: they play a big role in determining the functional components of the system and the relationships among them. This role is the same in a Macromedia Flash MX application development project as it would be in an HTML project.

7.1.2. User Interface Designer

User interface designers create the assets that make up the application presentation and stitch those assets together into a cohesive whole. They often work in Macromedia Flash MX, but may also use a variety of other multimedia authoring tools. Since designers can usually work more comfortably and effectively in Macromedia Flash than in HTML, they often assume more responsibility in a Macromedia Flash application development project.

7.1.3. Macromedia Flash Application Developer

Macromedia Flash application developers are technically savvy individuals who typically have experience programming in languages such as JavaScript, SQL, Visual Basic, or Java. Macromedia Flash application developers are responsible for implementing technical capabilities in Macromedia Flash MX, such as form processing, interactive widget creation, object model implementation, and data transfer. Macromedia Flash provides vehicles such as shared library files, components, templates, and externalized ActionScript files that facilitate collaboration between Macromedia Flash application developers and designers. While it is different in the technical details, this role corresponds to that of the HTML engineer who focuses on complex JavaScript and dHTML in a traditional HTML web application.

7.1.4. Database Engineer

Database engineers are responsible for designing and implementing the database for the system, as well as for connecting to external databases. They back up the databases, create schemas as necessary, and allow appropriate access to the databases. This role is the same as in traditional HTML development projects.

7.1.5. Application Developer

Application developers write the code on the server to respond to requests being sent from Macromedia Flash Player 6 to the application server. Requests are deciphered and messages assembled in response. Typical skills required for this position include aptitude in programming languages such as CFML, ASP, JSP, JavaScript, PHP, Perl, Java, and C++. In a Macromedia Flash MX project, application engineers focus their development efforts on interacting with enterprise databases and legacy systems to exchange data with Macromedia Flash Player.

7.1.6. Systems Engineer

Systems engineers configure the servers used in the system. They take care of tasks like networking, clustering, installing software, and allowing developer access to servers. This role remains the same as in traditional HTML development projects.

7.2. Quality Assurance

Testing Macromedia Flash MX applications is similar to testing HTML applications. Under most circumstances, it requires similar resources, skills, and tools. However, developers can leverage the strong cross-platform support that Macromedia Flash Player 6 offers. When compared to the tribulations encountered testing Java applets, JavaScript, or dHTML on different browsers and operating systems, the cross-platform support of Macromedia Flash really shines. One detail to note: for some functionality such as opening URLs, Macromedia Flash relies on the host browser to perform the task. In these areas, Macromedia Flash inherits some browser discrepancies.

7.3. Deployment

Deployment of a Rich Internet Application follows the same guidelines as for deployment of an HTML application. A reliable and full-featured configuration management tool is invaluable for coordinating the work of multiple developers, ensuring that the right versions of the software are on the appropriate servers, and supporting the inevitable emergency version rollbacks.

The application typically runs on a development server during development. Milestone builds are pushed to a staging server for greater scrutiny. Final versions of the application are then pushed live to the production server.

7.4. Hosting

Hosting a Macromedia Flash MX application is fundamentally the same as hosting an HTML application. Web, application, and database servers must be installed, integrated, configured, and customized. Network infrastructure, including firewalls, routers, DNS records, and clusters must be set up. Macromedia Flash itself places no demands on the hosting infrastructure. The Macromedia Flash run-time files (SWF files) merely sit on a web server, just as HTML files do.

8. New in Macromedia MX

The Macromedia MX family of products includes Macromedia Flash MX, Dreamweaver MX, Fireworks MX, ColdFusion MX, and Macromedia Studio MX. The Macromedia MX family has been designed to interoperate efficiently so that sophisticated applications can be more easily built and maintained.

8.1. New in Macromedia Flash MX

The Home Mortgage Application was originally developed in Macromedia Flash 5 and was later upgraded to Macromedia Flash MX. Macromedia Flash MX includes many new features that make it easier to develop applications than it was in prior versions. This section of the white paper discusses the concrete benefits of some of the new features that were used in the Home Mortgage Application. This section does not attempt to enumerate all of the important new capabilities of Macromedia Flash MX; we don't even mention features such as video, templates, Unicode, accessibility, dynamic loading of images, and sound, among many others.

8.1.1. Movie Clip Drawing API

The home mortgage payment chart was originally created in Macromedia Flash 5 by manipulating a large number of triangular and rectangular movie clip instances at run time. Many, many duplicates of these clips were stretched, rotated, colored, and positioned to form the chart. Slight visual irregularities in the chart were difficult to remove. The calculations involved in arranging the sub-clips were quite challenging. We had to carefully manipulate the size of the data set that was being graphed to balance having enough points to ensure a reasonably good appearance with having as few points as possible to keep performance tolerable.

Macromedia Flash MX includes a set of new movie clip functions that allow developers to draw lines, curves, and fills in movie clip instances using ActionScript at run time. This drawing API dramatically simplified the process of drawing the chart. The main benefits of using the drawing API were that we were able to develop the chart more quickly, and that it became easier to maintain. There were, however, some important corollary advantages as well. The performance at run time was much faster, making it feasible to graph much more data than was previously possible. It also became easy to implement translucent backgrounds, something that we had not previously been able to resolve to our satisfaction.

8.1.2. Components

Macromedia Flash MX features a new component framework for working with customizable reusable user interface widgets. Macromedia Flash MX ships with components such as radio buttons, check boxes, lists, combo boxes, scroll panes, and push buttons.

Developers can create their own custom components. They can also conveniently install and use components that have been developed by third parties or Macromedia. Macromedia has already released components such as a calendar; a tree view; a ticker; and line, pie, and bar charts. Visit the Macromedia Exchange at www.macromedia.com/exchange to acquire these and many other useful components.

When we originally developed the Home Mortgage Application, we built a custom scroll bar to allow the user to scroll the list of textual payments in the list view. We were able to adapt a scroll bar that we had written for a previous project, but we still spent quite a bit of time modifying the code. While the scroll bar was attractive, neither its appearance nor its behavior was completely standard.

When we updated the application to Macromedia Flash MX, we substituted a Macromedia Flash component for our custom scroll bar in only a few minutes. We were able to use the same payment list display clip we previously had used, which minimized the work we had to do and maintained visual consistency with the prior version of the application. We now have a scroll bar that looks and behaves in a standard, full-featured way. We were able to conveniently customize the appearance of the scroll bar via ActionScript. Refer to Figure F to see the scroll bar component in action.

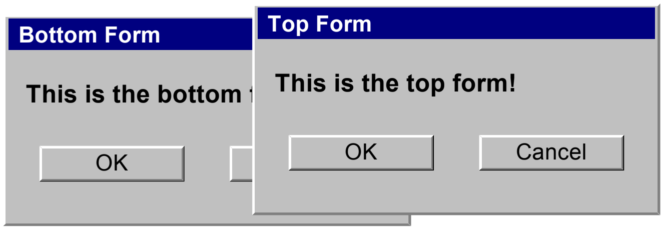
8.1.3. Movie Clip Event Handlers

Macromedia Flash MX adds some important new functions that make it easier to create Rich Internet Applications. Movie clips have several new event handlers, including onDragOver, onEnterFrame, onKeyDown, onSetFocus, onRelease, onMouseDown, and onRollOver, among many others. These new event handlers make it easier for developers to process specific user activities. They also make is easier to intuitively organize and document the code associated with movie clips.

The Home Mortgage Application chart demands sophisticated mouse event processing. For example, we need to perform specific processing when the mouse has been over the chart for 375 ms and, after some initial motion, the mouse has been still over the chart for 250 ms. Macromedia Flash MX simplifies the implementation of this processing through new methods that automatically filter irrelevant events and can be more intuitively associated with the appropriate clips.

8.1.3.1. Layered User Interfaces

Modern operating systems like Windows and the Macintosh operating system allow multiple windows to appear on the screen at once, but only the topmost window responds to mouse clicks. In the example in Figure I, a click on the OK button in the top form would not normally trigger processing of the obscured Cancel button in the bottom form, even if the mouse pointer were in fact over the hidden Cancel button.



Bottom Form

This is the bottom f

Top Form

This is the top form!

Figure I: *Layered User Interface*

Achieving this expected behavior was surprisingly challenging in prior versions of Macromedia Flash. Macromedia Flash MX makes this formerly perplexing task simple because the new mouse event handlers respect the visual layering of movie clip instances at run time. Mouse clicks no longer "fall through" to obscured clips.

8.1.4. Text Fields

On the Pay tab in the left portion of the Home Mortgage Application, the user can enter an additional monthly principal payment amount. As the user types in this field, the system incrementally processes the amount. If the value is negative or larger than the remaining principal, then the background is shaded pink to indicate an error. Otherwise, a button appears allowing the user to graphically view the impact that this hypothetical payment would have on the loan repayment schedule.

When we originally developed the Home Mortgage Application in Macromedia Flash 5, we relied on undocumented features to process the contents of the text field dynamically as described earlier. Macromedia Flash MX introduces the new TextField object. Text fields existed in prior versions of Macromedia Flash, but they are now objects with properties, events, and methods that can be thoroughly observed and manipulated using ActionScript. Their functionality has expanded to include change handlers and listeners as well as enhanced formatting, scrolling, selection, and font capabilities.

The Macromedia Flash MX version of the Home Mortgage Application uses the new formatting and change handler capabilities of the TextField object to provide the same interactivity as the prior version, but with a smaller amount of better organized code.

8.1.5. Native XML, Array, and String Objects

In Macromedia Flash 5, the XML, Array, and String objects were written in ActionScript. In Macromedia Flash MX, these objects have been rewritten using native code, improving performance. Consequently, existing Macromedia Flash applications run noticeably faster in Macromedia Flash MX, even if not a line of code is changed.

8.1.6. Macromedia Flash Remoting

Macromedia Flash MX supports Macromedia Flash Remoting, a new means of connecting to selected application servers such as Macromedia ColdFusion MX. Macromedia Flash Remoting speeds the development of Macromedia Flash applications because hierarchical data can be transmitted to and from the server in native ActionScript objects through simple remote method calls. Helpful debugging tools and integration with Macromedia Flash MX UI components further speed development. As an added benefit, Macromedia Flash Remoting improves the run-time performance of systems since the data being steamed back and forth is transmitted in an optimized binary format.

The server interactivity in the original version of the Home Mortgage Application was implemented using XML. The creation, parsing, and navigation of XML documents on both Macromedia Flash Player and on the ColdFusion server consumed a disproportionate of our development time. With the switch to Macromedia Flash Remoting, the volume and complexity of code required to communicate with the server was greatly reduced. At the same time, the performance of the system improved.

8.2. New in Macromedia ColdFusion MX

The server portion of the original version of the Home Mortgage Application was written in Macromedia ColdFusion 5. Upgrading the server to Macromedia ColdFusion MX was a pleasure. The Macromedia ColdFusion MX administrative tools, cross-platform support, and performance have all been substantially improved. And, of course, the native support for Macromedia Flash Remoting has been added.

8.2.1. Components

Macromedia ColdFusion MX components allow functions to be grouped together into manageable units. Components can be used within web applications in a variety of ways: to organize functions in a traditional CFML application, to present functions as web services using SOAP, or to build Macromedia Flash Remoting services. Components are implemented using standard CFML tags and script but are saved in CFC (.cfc extension) files rather than CFML (.cfm extension) files. Components are structured using several new tags, including the cfcomponent tag to delimit components, the cffunction tag to describe functions, the cfargument tag to list parameters to the functions, and the cfreturn tag to specify return values. Read more in the related articles and tutorials at the ColdFusion MX Application Developer Center.

8.2.2. Macromedia Flash Remoting Services

Macromedia ColdFusion MX makes it easy to create Macromedia Flash Remoting services, particularly for Macromedia Flash developers with limited server development experience. Macromedia Flash Remoting services can be created in one of three ways: through CFML pages in a directory, through the functions in a ColdFusion component, or through the functions in a server-side ActionScript file.

8.2.2.1. Macromedia Flash Remoting As CFML Pages

The most familiar way for an experienced Macromedia ColdFusion 5 developer to define a Macromedia Flash Remoting service is as a collection of CFML pages in a folder. The name of the folder on the web server is the name of the service, and the CFML pages in the folder are the service methods. In the Home Mortgage Application, for example, the file `/mx/homemortgage/getLoanDetail.cfm` defines the getLoanDetail method in the mx.homemortgage Macromedia Flash Remoting service.

A Macromedia Flash Remoting service method implemented in this way uses standard ColdFusion tags to retrieve and format information and has two important new objects at its disposal. The flash.params object is an array that holds the parameters sent from Macromedia Flash Player to the service method. The flash.result object is serialized and sent back to Macromedia Flash Player after page processing is complete.

8.2.2.2. Macromedia Flash Remoting as ColdFusion Components

Macromedia Flash Remoting services can be implemented as ColdFusion components. In this model, the name of the service is the URL of the component (without the CFC extension). The service methods are the functions defined in the component.

If the functions in the component include the attribute `access="remote"` within the cffunction tag, the component methods can be invoked in a browser window using a URL as follows:

```
http://yourserver.com/directory/yourcomponent.cfc?method=yourMethod&p
    aram1=value1.
```

This capability is very handy during development; it allows you to quickly debug your server development work before integrating Macromedia Flash MX with the server. Once development is complete, you can remove the `access="remote"` attribute to improve security.

8.2.2.3. Macromedia Flash Remoting As Server-Side ActionScript

Macromedia ColdFusion MX allows server-side scripting for Macromedia Flash Remoting services using ActionScript instead of CFML. This feature allows Macromedia Flash developers to implement server applications without having to learn a new programming language. (Note that ActionScript can be used only to implement Macromedia Flash Remoting services; it cannot be used in other CFML pages.)

Macromedia Flash Remoting services implemented as server-side ActionScript are constructed much like ColdFusion MX components: as a group of functions in a file on the server. With server-side ActionScript, however, the file has an .asr extension, and the functions are defined entirely in ActionScript. As with components, the name of the Macromedia Flash Remoting service is the URL of the ASR file (without the extension), and the methods on the service are the functions defined in the file.

Server-side ActionScript provides two functions to assist with server implementation. The CF.query function allows database queries to be executed, and CF.http allows HTTP connections to be made to other servers. Because it runs on the server, server-side ActionScript cannot directly manipulate Macromedia Flash MX objects. For example, there are no Movie Clip or Key or Selection objects on the server.

8.2.3. Improved Java Integration

Macromedia ColdFusion MX runs on top of a J2EE server platform (such as Macromedia JRun), so Java integration is now even more powerful and seamless than it previously was. The prior mechanisms of Java integration (JavaBeans, EJBs, applets, servlets, and Java custom tags) can still be used and are now faster and more reliable. ColdFusion MX offers some important new Java integration capabilities:

- JSP tags can now be used directly within CFML pages.
- The same variables can be referenced by both Java and CFML.
- There is a simplified syntax for interacting with Java primitives, Java Beans, and Java collections.
- A single web application can now consist of CFML pages, JSP pages, and servlets.

9. Conclusion

Macromedia Flash MX is the next-generation platform for building sophisticated Rich Internet Applications. Macromedia Flash provides a highly interactive and fluid user experience not possible in HTML. Macromedia Flash applications can be built quickly, and they work within the confines of existing IT organizations without draining network resources.

Macromedia ColdFusion MX is an ideal platform for building the server side of Rich Internet Applications. It helps developers create server applications quickly, allows integration into existing IT environments, and enables applications to scale to serve a large numbers of users.

9.1. Broad Adoption and Industry Support

Since its introduction in 1996, Macromedia Flash has become the standard for developing rich Internet content and applications. Global enterprises such as General Motors, Disney, CNN, IBM, Yahoo, Boeing, Merrill Lynch, and Coca-Cola have deployed Macromedia Flash content because it provides the richest and most consistent viewing experience across all major platforms, browsers, and devices.

Macromedia Flash Player 6 is the enabling technology for these rich experiences and is widely available through successful distribution partnerships with industry leaders such as Microsoft(r), Apple(r), and AOL(r). Leading independent studies conducted in late 2001 show that more than 98 percent of the online audience can immediately view Macromedia Flash content, making Macromedia Flash Player the most widely distributed rich client technology on the web. Versions of Macromedia Flash Player are available on all leading operating systems and browsers, as well as on platforms such as handheld devices and web-enabled television.

Due to the Internet's accelerated evolution, users have high expectations for increasingly sophisticated and usable Internet applications that offer more and more capabilities. Macromedia Flash has been progressing at the same pace to meet the escalating demands of users. From its roots as a vector animation tool, Macromedia Flash has added scripting, data processing, and server interactivity features over the years to the point where it is now the leading platform for developing user interfaces for Rich Internet Applications.

9.2. Usability is King

Internet applications that are easy to use are much more likely to be successful, attracting people because they provide an engaging, productive user experience. Macromedia Flash MX offers some distinct advantages in building usable Internet applications:

- Rather than offering a breadcrumb trail of separately downloaded pages, a Macromedia Flash user interface is a single, unified application that manages the user experience in an uninterrupted manner. Navigation is much quicker and crisper.
- Macromedia Flash provides developers with interactive capabilities like drag and drop, mouse and keystroke trapping, pop-up windows, and much more. These features all help the end user interact with the application in intuitive and creative ways.
- Macromedia Flash gives designers a high degree of control over fonts, shapes, colors, and animation to create engaging and attractive user interfaces.
- Macromedia Flash can interact with application servers such as ColdFusion MX to retrieve and submit data. This exchange of data can be done without page redraws, providing a more fluid user experience.

9.3. Rapid Application Development Environment

Server-side Internet application development tends to be both expensive and challenging. Building applications with Macromedia Flash MX facilitates server-side development because all presentation layer formatting is done in Macromedia Flash rather than on the application server. ColdFusion MX simplifies development in the following ways:

- Provides smooth connectivity to relational databases, Java, COM, and CORBA.
- Supplies a rich feature set such as tags to send e-mail, connect to other servers, and schedule tasks, and much more.
- Offers an expansive library of mature third-party extensions.
- Facilitates the reuse of code through components, custom tags, and modules.
- Includes industry-leading debugging tools.
- Accommodates massive numbers of clients using advanced clustering techniques.

For more information about acquiring the tools and training necessary to build your own Macromedia Flash MX and ColdFusion MX applications, please visit the Macromedia Flash MX Application Developer Center at www.macromedia.com/desdev/mx/flash/ and the ColdFusion MX Application Developer Center at www.macromedia.com/desdev/mx/cf/.

10. Appendix

The sections that follow explore in more detail related topics that diverge from the central message of this white paper.

10.1. Managing Session State with Macromedia Flash MX

Earlier, we discussed how Macromedia Flash MX improves a web application's scalability by reducing the amount of session data that is stored on the server. This appendix takes this concept to its logical conclusion: completely removing the session state from the server.

The benefits achieved by accomplishing this are substantial. Memory consumption on each individual server is minimized so each can accommodate a larger number of concurrent clients. It also facilitates and optimizes the clustering of multiple servers in a single powerful site because the load balancer does not need to provide sticky routing of requests to the same server in the cluster. Complex session failover becomes irrelevant, since there is no server session in the first place.

Accomplishing this goal is simple: rather than attaching data to the session in the application server, the data is maintained in Macromedia Flash Player 6. Initially, the session data is usually sent from the application server to Macromedia Flash. If it is subsequently needed on the application server, it must be sent back from Macromedia Flash Player to the application server with the request.

The question of whether it is wise to move session state from the server is application specific and hinges on where the session data will be used. If session data were needed on the client for tasks like form validation, then removing it from the server would probably make sense. If, on the other hand, session data were used for rigorous authentication and authorization on the application server itself, then removing it from the server session would probably not be sensible. Although the clustering architecture would be simplified, the price would be extraneous transfer of sensitive user profile data over the Internet, and repetitive validation of that data on the application server.

10.2. Form Processing in Macromedia Flash MX

Form processing in Macromedia Flash MX is conceptually similar to form processing in an HTML application. We follow the same sequence of steps, listed in Figure J.

Form Processing
Initialize the data that will appear on the form.
Solicit `user input` through a series of input fields.
Validate the form data in Macromedia Flash Player using ActionScript. If validation fails, present a “failure” page, usually the input page adorned with error messages.
Validate the form data on the server. If validation fails, route the user to the same “failure” page shown when validation fails on the client.
Once the form data has been completely validated, process the form data on the server: save it to a database, distribute it to interested people, and so on.
Present a “success” page to the user.

Figure J: *Form Processing Steps*

Although the sequence of steps is the same, form processing in Macromedia Flash MX offers several tangible advantages over HTML. Steps 1, 4, 5, and 6 benefit because Macromedia Flash exchanges data with the server more efficiently and less jarringly than occurs in an HTML page reload. Step 2 benefits from more attractive widgets, which can be tuned for a specific input form. Step 3 benefits because Macromedia Flash conveniently tracks more of the data associated with the user's session and has a more sophisticated presentation layer architecture than HTML.

The developer can conveniently build a reusable framework for facilitating form processing on both Macromedia Flash Player and the application server. Form processing error messages can be sent between Macromedia Flash Player and the application server in a consistent way so they can be easily interpreted.

10.3. ActionScript Domain Object Model

Macromedia Flash MX can help developers manage complex user interfaces by dividing the presentation layer into visual components (movie clips) and a data model (ActionScript objects). This section explores the data model in more detail.

A domain object model (or DOM) is an object-oriented representation of an application's subject matter in a programming language. An application's DOM represents business-oriented concepts such as company, employee, apartment, or hospital. In the Home Mortgage Application, for example, the core members of the domain object model are Loan and LoanPayment.

A DOM facilitates the display and manipulation of data in a software system by organizing it in a way that maps cleanly to the real-world organism that the system models. It promotes the intelligent structuring of complex systems in a way that is intuitive, easily extensible, and maintainable.

The Macromedia Flash ActionScript programming language includes the object-based constructs required to effectively define and use a DOM. Properties and methods can be defined on constructors, objects can be created using these constructors, and relationships can be defined between objects. ActionScript even offers the powerful flexibility to redefine an object's behavior (and even its constructor) on the fly.

Loading data from an application server into a domain object model imbues the resulting objects with appropriate functionality. Rather than just being collections of data, the objects can perform application-specific tasks, such as calculating sales tax or validating user input. Figure K is a working example of a domain object model in action.

```
/**
 * The constructor for LoanPayment objects.
 * Each LoanPayment object represents a payment applied
 * to a loan.
 */
mmm.LoanPayment = function(aPaymentDate, anInterestPayment,
   aPrincipalPayment, aPenaltyPayment)
{
 /** The date that the payment was received */
 this.paymentDate = aPaymentDate;
 /** The interest portion of the payment */
 this.interestPayment = anInterestPayment;
 /** The principal portion of the payment */
 this.principalPayment = aPrincipalPayment;
 /** The penalty portion of the payment */
 this.penaltyPayment = aPenaltyPayment;
};
/**
 * A piece of useful behavior in a LoanPayment object.
 *
 * @return The total amount of the payment. Add the interest,
 * principal, and penalty values together to arrive at this
 * number.
 */
mmm.LoanPayment.prototype.getTotalPayment = function()
{
 return this.interestPayment + this.principalPayment +
  this.penaltyPayment;
};
// Now use the LoanPayment constructor to create a
// LoanPayment object.
var aPayment = new Object.mmm.LoanPayment(new Date(), 1000, 100, 25);
// Use the object we just created to do something useful
trace(aPayment.getTotalPayment());
```

Figure K: Domain Object Model in ActionScript

Most of the techniques used to transfer objects from the application server to Macromedia Flash Player create generic objects (that is, new Object( ) elements) rather than the domain-specific objects that we want to use in our applications. Fortunately, ActionScript lets us change the prototype of an object, and hence the standard methods associated with the object, on the fly after the object has been created. The code in Figure L performs just such a prototype swap.

```
// Create a generic object that we will subsequently transmogrify
// into a LoanPayment object
var aPayment = new Object();
aPayment.paymentDate = new Date();
aPayment.interestPayment = 1000;
aPayment.principalPayment = 100;
aPayment.penaltyPayment = 25;
// At this point our generic payment object has all the data
// that a normal LoanPayment object would have.
// Convert this object to be a true LoanPayment object
aPayment.__proto__ = mmm.LoanPayment.prototype;
// Now call a useful LoanPayment method
trace(aPayment.getTotalPayment());
```

Figure K: *Dynamic ActionScript Prototype Swapping*