An Introduction to Ajax

By Joe Resuello, Technical Marketing Engineer, 4D Inc.

Technical Note 07-44

Abstract

Ajax has become a popular phenomenon in the world of web development. However, it can be a bit daunting to 4D Developers who may not be familiar with developing for the web. This Technical Note is an introduction to Ajax. It will define what Ajax is, it will dispel common misconceptions, it will provide examples of Ajax in action, and it will point out characteristics of Ajax applications. Once you gain a familiarity with Ajax, you will be introduced to the 4D Ajax Framework so you can get a glimpse of how your database could look and perform on the web.

If you are a 4D Developer that knows nothing about Ajax this is a good place to start.

Introduction

This Technical Note is broken up into the following sections:

- What is Ajax?
- Ajax: A Brief History
- What are the Characteristics of an Ajax Application?
- How Does Ajax Do It?
- Enter the 4D Ajax Framework
- Conclusion

What is Ajax?

Before we begin discussing what Ajax is, perhaps it is best to start discussing what Ajax isn't. Ajax is a commonly misunderstood phenomenon, and now would be a good time to dispel some common misconceptions and myths about it.

What Ajax isn't

Here are some common misconceptions as to what many believe Ajax is:

A Technology
Ajax is not a new technology. In fact, Ajax uses common technologies that existed long before it came to be. Thus, there is no Ajax Plug-in. Users do not need special software such as browser plug-ins or desktop applications in order to use an Ajax application.

A Language
To the relief of many, Ajax is not a new programming language. Developers cannot *code* in Ajax so, fortunately, many do not have to learn a new technology. However, as will be explained further later, Ajax does use a scripting language and other languages and if developers are not already familiar with these existing technologies then building an Ajax application from scratch will require some time to get the hang of.

Proprietary
Although it is a highly popularized buzzword, Ajax is not the name of a company or a product. In this respect many confuse Ajax with a proprietary technology such as Flash, which can also be used to created interactive web applications. One particular downside for proprietary technologies is that users need to download additional software in order to view and use it. Fortunately for Ajax, this is not the case.

What Ajax is

With those misconceptions out of way, let's unravel the truth behind what Ajax is:

A Technique
Ajax is a *way* of using existing technologies to create interactive web applications. These existing technologies are JavaScript, XML, HTML, and CSS. As mentioned earlier, these technologies existed long before the Ajax phenomenon came to be. Thus, Ajax is more of a novel and effective technique of using these technologies to create interactive web applications.

It uses open standards
Without getting into too much detail about web standards and such, the point here is that Ajax uses open technologies. Thus, Ajax applications can be implemented in any browser, free of legal constraints. Any recent browser is Ajax-compatible (IE 5.0+, Mozilla 1.0+, Firefox 1.0+, Netscape 7.0+, and Apple added it to Safari 1.2+).

Ajax: A Brief History

On February 18, 2005 a gentleman by the name of Jesse James Garrett coined the term Ajax, and thus forever immortalized himself in the annals of web application development history. He is an information architect and founder of a user experience firm named Adaptive Path. It was on Adaptive Path's website where he published his infamous essay, *Ajax: A New Approach to Web Applications* on that fateful February day.

Ajax as a phenomenon was creating quite a buzz at the time as Google had released some Ajax applications such as Google Maps and Google Suggest. But it was at this point and time that people finally got their hands on a term to give this phenomenon a tangible name. Mr. Garrett clearly explained the underlying

technologies of an Ajax application and gave it a representative name. Ajax, in his essay, stood for *Asynchronous JavaScript and XML.*

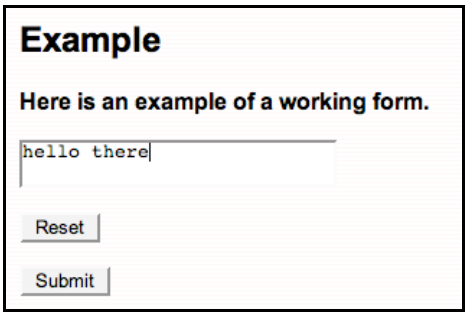
A photo of Jesse James Garrett himself

What are the characteristics of an Ajax application?

Ajax applications are interactive. In the world of web applications, this is a novel idea. Before Ajax came into the picture, the line between web applications and desktop applications was quite distinct. Now that line is becoming more and more grayed-out as Ajax web applications now have the GUI features with much of the seamlessness and responsiveness of desktop applications.

No Page Reloads

Web applications from the old days are heavy on page reloading. Make any request to the web server, and the web page would have to be reloaded so that you can witness the changes.

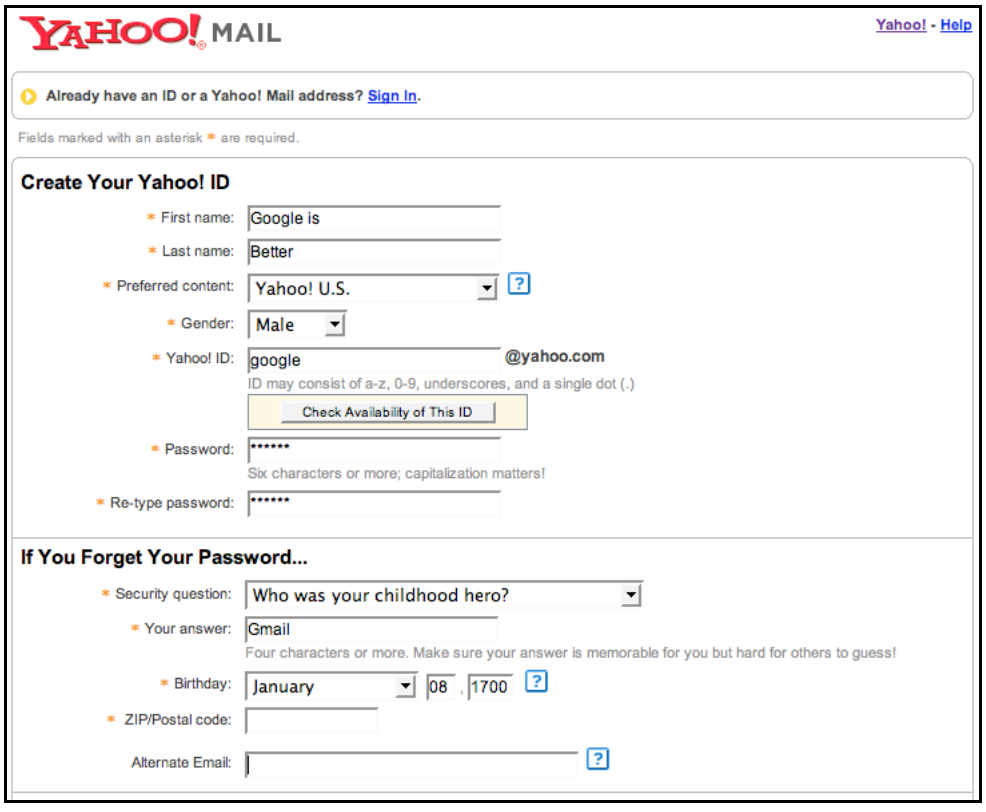


The above picture is an example a web form that does not use Ajax. Once the user is done entering information they can hit the *Submit* button. Once they hit the *Submit* button, the page reloads to present something like the following:

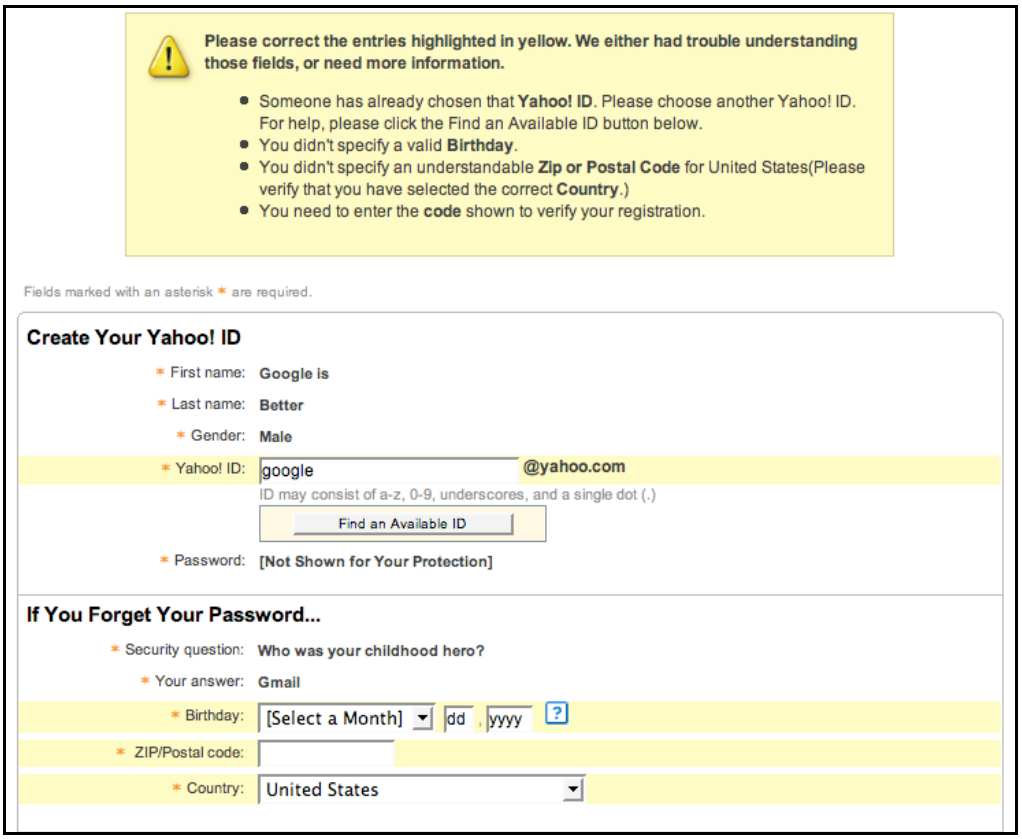
The form is fine as it is, but it isn't great. There are some noticeable pitfalls to having a web application such as this one.

What if the user entered incorrect information?

Suppose the user entered too many digits for a phone number field. Suppose the email address they entered was badly formed. The user would not have been informed of their mistakes until after they hit they *Submit* button.
Take a look at the following example:



This is part of the registration form for creating a new *Yahoo! Mail* user account. Now suppose we had made mistakes in this form. The following is what we would see after a hit of the *Submit* button and a page reload:



We would be presented with the same form but with indicators showing us where we messed up. The downside of such an application is that the user had wasted valuable time during the page reload. It would have been nice to have been informed of the mistakes as the fields were being entered. Also, this makes for an unpleasant and rather dry user experience. The 1-2-3 Step of *1) Enter Information*, *2) Submit Information*, and then *3) Cross Fingers and Hope the Information was Well Received* shows that there is clearly a lack of interactivity going on. An Ajax application, on the other hand, could validate information as it was entered.

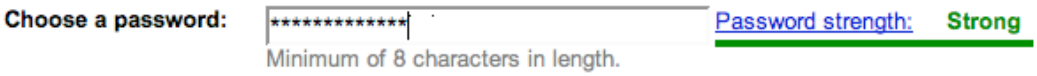
Ajax to the Rescue

Speaking of page reloads and email registration sites, let's take a look at the email registration form for *Gmail*. We will notice that it has some Ajax capabilities. Here is the field for entering your email password:

In the above example no characters have been entered, so the "Password strength" indicator shows nothing. Let's enter some data:

Notice that, *as I type*, a notification appears to the right. The above example one tells me my password length is too short.

Here I am being told that my password is not that strong. Hackers may be able to figure it out quite easily.

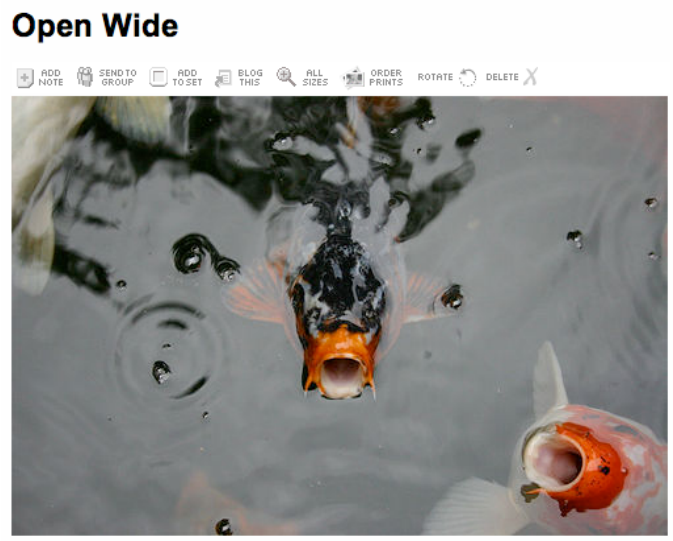


Now my password is satisfactory. I noticed that as I was typing in possible passwords, I did not have to wait for a page reload to see these notifications. These notifications appeared in real time, asynchronously as I was typing. This makes the experience interactive, and it is made possible by Ajax.

Graphical responsiveness

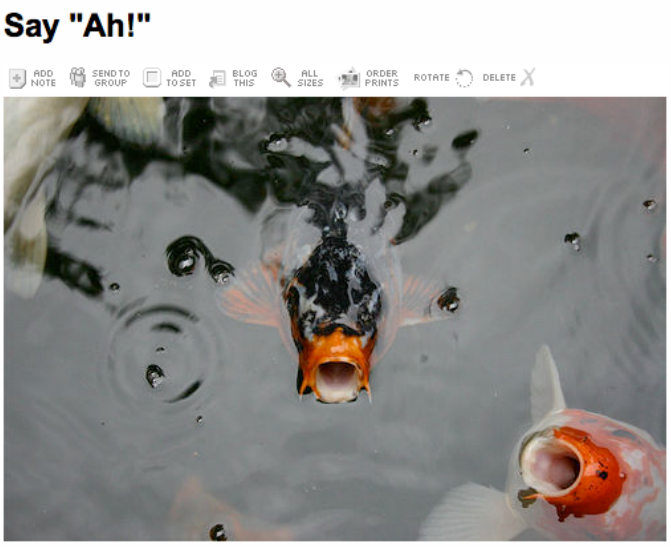
Indicators presented in the example above show how Ajax can create an interactive user experience. Those examples, however, represent only the *tip of the iceberg* when it comes to the graphical responsiveness that Ajax applications are capable of. Here now are more graphically rich examples of Ajax applications.

Flickr
Flickr is a popular photo management website and they use Ajax as their photo display interface. Here is an example photo in Flickr:



The title of the photo is up at the top and the image is below it. If I simply click the title…

… it suddenly becomes editable, and buttons to *Save* or *Cancel* appear. Again, no page reload occurred. I can edit the title, save it, and then *voila!*



The changes appear on the page with no delay. No page reload occurred. Editing text on an HTML page never before was so intuitive and, dare I say, fun.

Google Maps

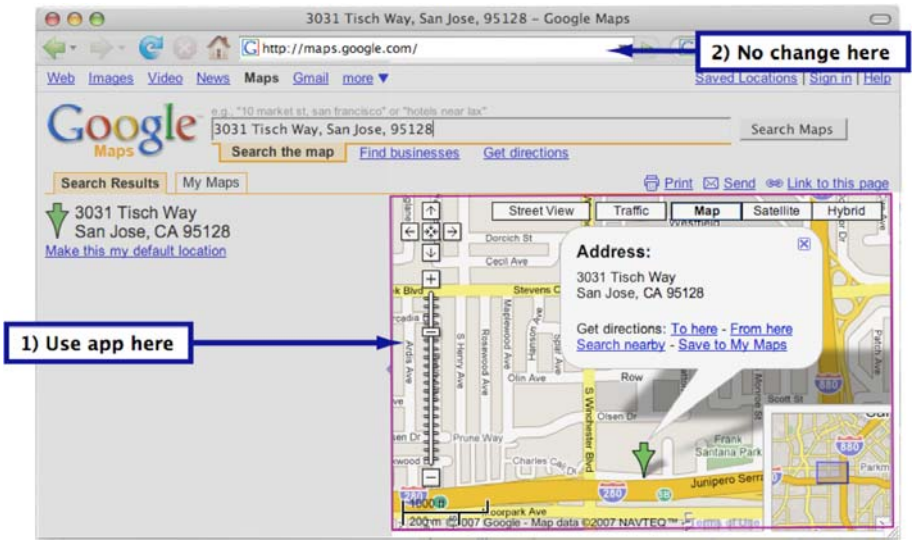
Google Maps is another example of a web application that employs a healthy dose of Ajax. Here is an example Google map:

Enter any address and you will be presented with a page like this one. The map area (indicated within the purple box and green arrow) is where much of the Ajax goodness happens. Here the interactivity is quite rich as you are able to:

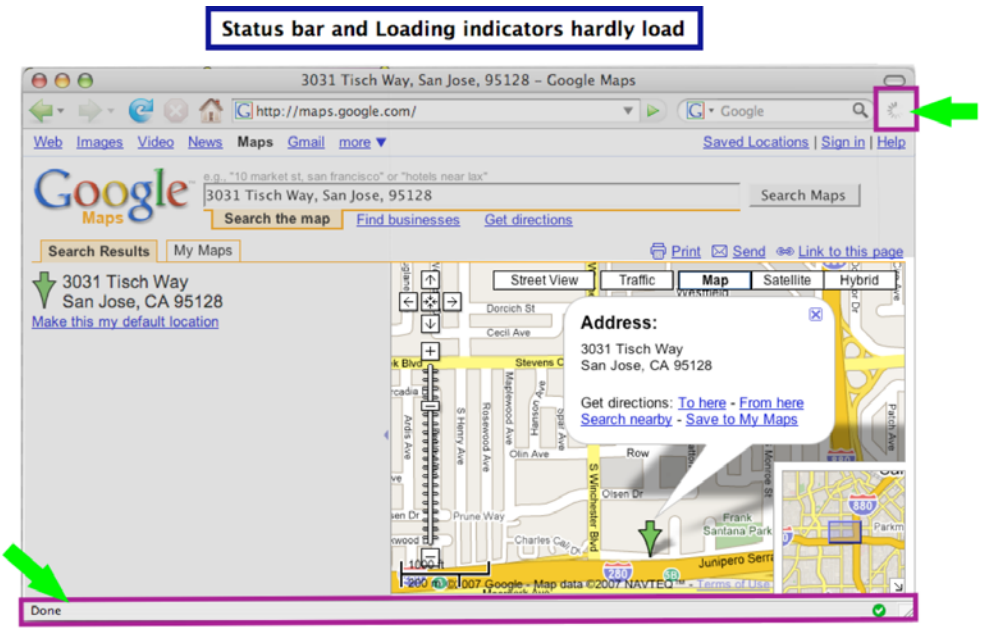
- Zoom in and zoom out of the map
- Drag the map around
- Switch to other views such as Satellite View, Street View, and even a Hybrid View

Ajax Characteristics

There are a couple of things that should be really noted when seeing an Ajax application in action:



- The address bar does not change: Since only parts of the web page are being updated on the fly, there is no need for a new HTML page to appear. The power of Ajax means that only specific elements on the page are changed.



- There is minor indication of loading while interacting with this the Ajax application. This makes the user experience asynchronous. Progress bars may indicate some activity between the web page and the web server, but that activity does not interrupt the user.

How Does Ajax do it?

I know what you are thinking and the answer to your question is, “No, black magic is not what makes Ajax applications work the way they do”. In this section we will get into some detail about the technologies that make an Ajax application run.

XHTML and CSS

This is what makes the user interface of the web application. Why XHTML and CSS? Because they represent an interface that any browser can display. Being compatible on any browser (whether it be proprietary like Internet Explorer or open source like Firefox) is one of the core strengths of an Ajax application.

DOM

DOM stands for Document Object Model, and it is a widely accepted programming interface that allows you to update specific areas on the page.

XML

XML is a tag language, much like HTML but it allows for more explicit tags. XML is the format that information is passed between the web application and the web server.

XMLHttpRequest

This is an object supported by most browsers that allows the application to make requests to the server without having to reload the page in order to process the request.

JavaScript

JavaScript is the scripting language that acts as the glue that puts all of these pieces together. Ajax applications rely heavily on this technology. One of the downsides of JavaScript is that 4D developers may not be familiar enough with this language to create Ajax applications.

NOTE: *Javascript is not Java. Our 4D Technical Support team has corresponded to many 4D developers who have been under the assumption that JavaScript is like Java, or even that JavaScript is Java. This is incorrect. JavaScript and Java are totally different technologies and the only similarity they have is the similarity in their name and syntax.*

Enter the 4D Ajax Framework

As a 4D Developer, the 4D Ajax Framework (4DAF) makes the business of building Ajax applications easy with minimal effort. A developer is not required to know the inner workings of an Ajax application in order to have Ajax objects integrated into their web applications. 4D developers can have much of the responsiveness, speed, interactivity, and usability that Ajax applications are suited to possess.

Why use the 4D Ajax Framework?

As great as Ajax is, it does have its downsides. It is quite difficult to create an Ajax application from scratch. The Javascript expertise required is not trivial to acquire. With the 4DAF, much of the JavaScript functionality is done for you.

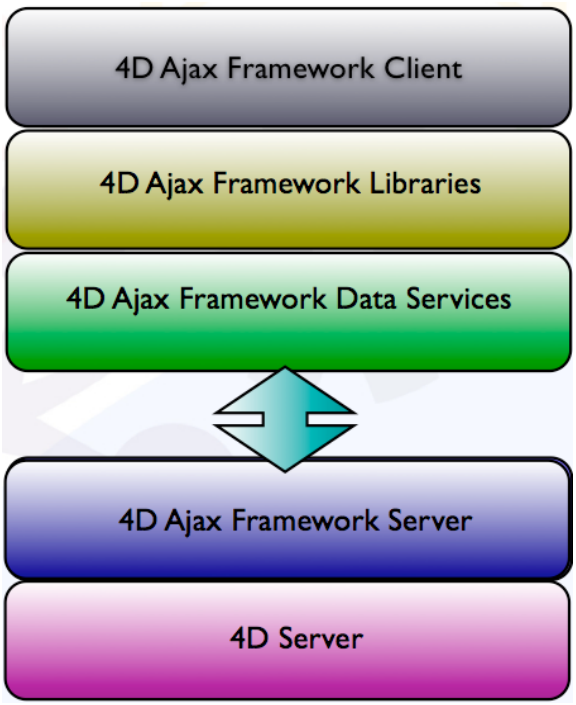
The 4D Ajax Framework is a set of pre-built programming libraries that wrap Ajax technologies into discrete pieces of code and objects ready at your disposal.

Customizability

One great thing about the 4D Ajax Framework is its customizability. It appeals to users who have different levels of comfort when it comes to web application development. For those who wish to *plug-and-play* right out of the box, they can use the 4D Ajax Framework Client.

For those more comfortable with creating their own custom web applications they can embed objects from the framework into their web pages.

This image identifies the different layers of the 4DAF:



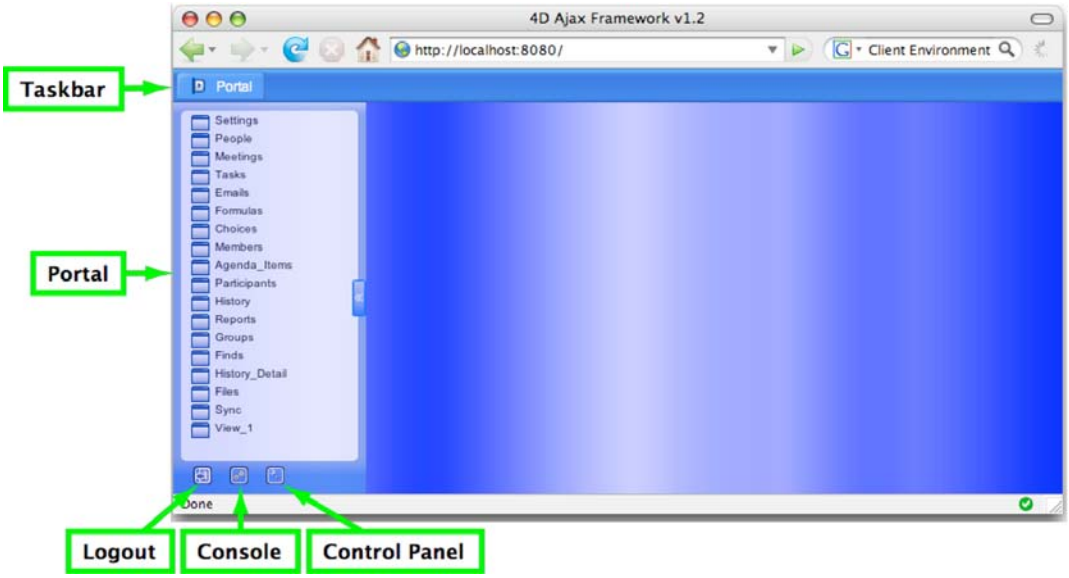
The “4D Ajax Framework Client” layer is the default client implementation of the 4DAF. This is the *plug-and-play* level of the 4DAF.

Custom web applications will use the 4DAF at the “4D Ajax Framework Libraries” layer.

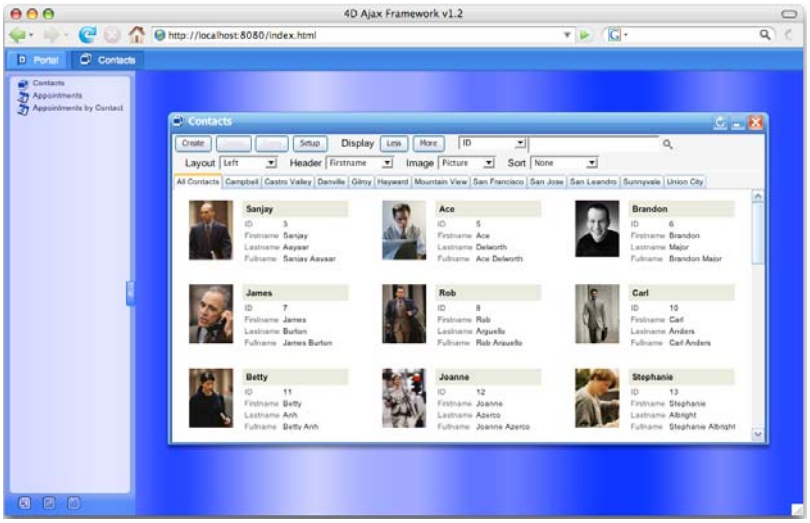
The “4D Ajax Framework Data Services” and “4D Ajax Framework Server” layers are the communication layer for the 4DAF. The former executes in the web browser, while the latter executes in the 4D database.

Example of 4DAF Client

Let's take a look at the 4D Ajax Framework Client:

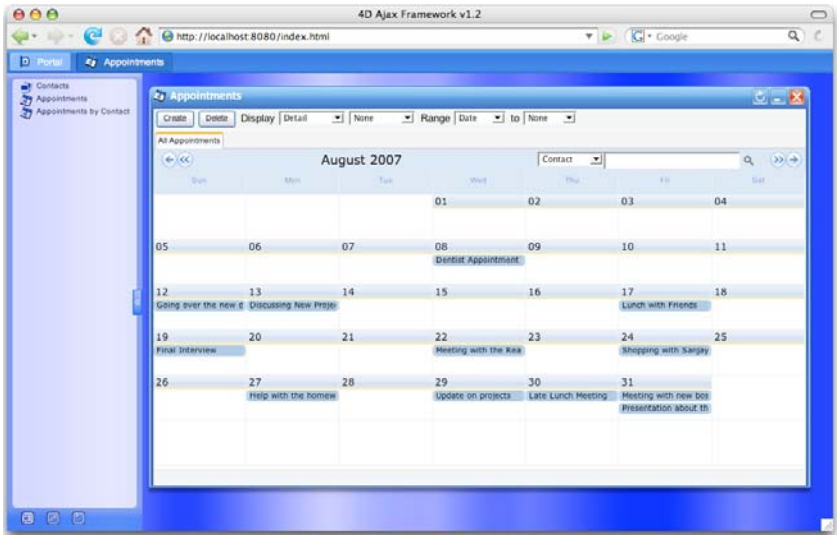


In this environment, 4D tables are listed in the Portal area to the left. Click on the table name to load it:



A table can be represented graphically, just like the Image Matrix style shown above. It represents your records visually. Now remember that this is an Ajax environment, so your users can interact with these objects asynchronously. They can perform live searches, they change styling and formatting, they can modify records, etc. Everything is asynchronous. Everything is updated live.

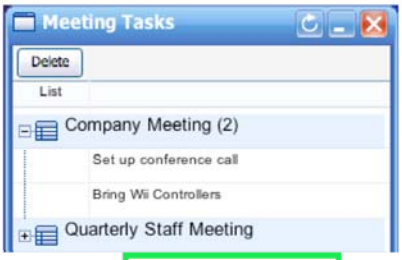
Here is an example of another Ajax object, the Calendar:



As long as your table has date fields, they can be represented in this Calendar object. What is nice is that these objects (the Calendar, the Image Matrix, etc.) are already predefined for you. All that you have to do is decide what object style best represents your information.

Here is a look at some 4D Ajax Framework objects:

Data Tree



Data Grid

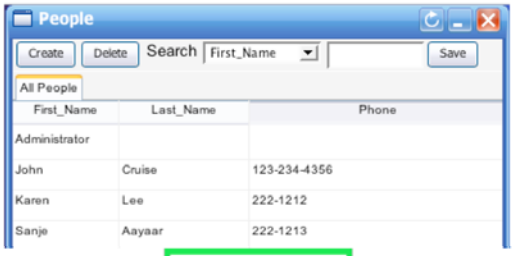
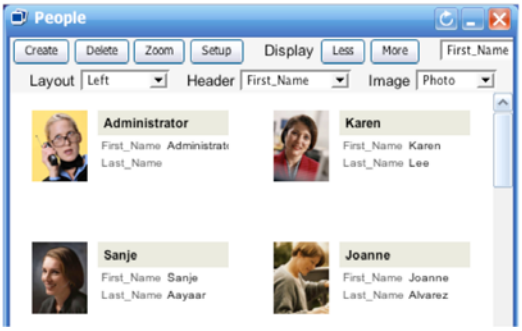
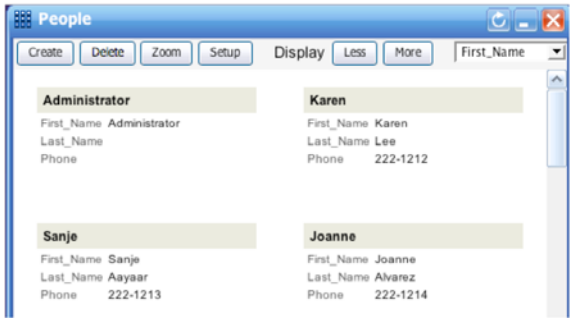
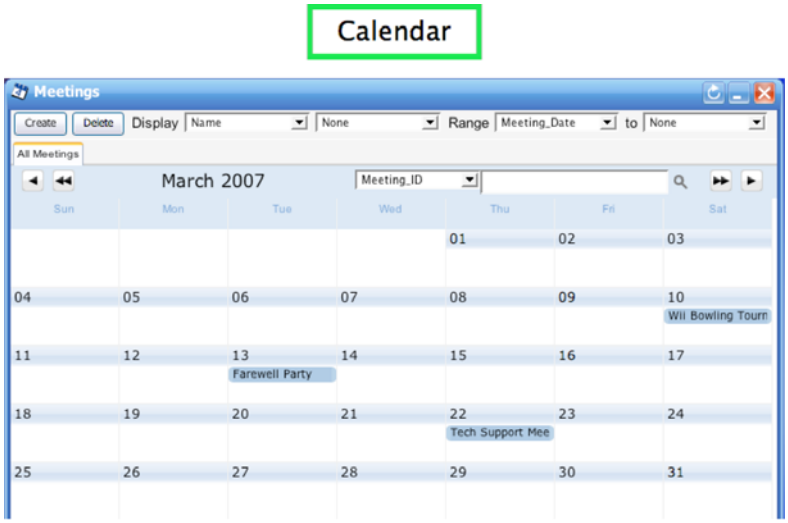


Image Matrix



DataMatrix



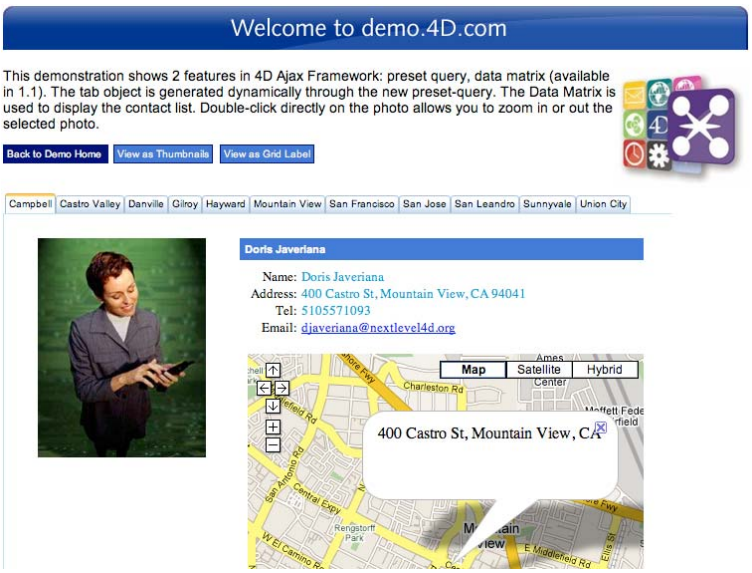


Example Custom Ajax Apps

For those more daring, they can embed these 4D Ajax Framework objects (such as the Calendar and the Image Matrix object) into their own HTML pages. Here are some examples.

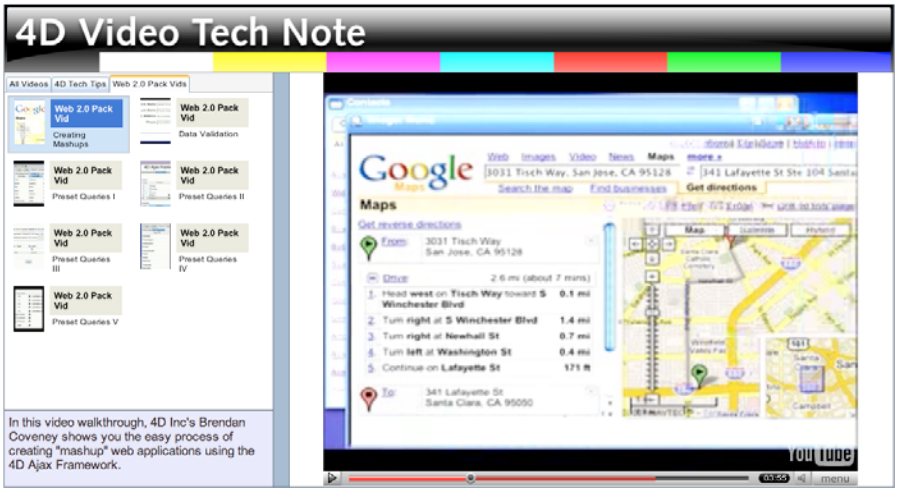
Contacts Database

Included with the 4D Ajax Framework is the Contacts database. It is a Custom Framework Application in that it takes an HTML page and embeds a 4D Ajax Framework object into it. Below you can see an Image Matrix object:



4DAF Video Player Technical Note

Available to members of the 4D Partner program is Technical Note 07-24, “4DAF Video Player”. The sample database in this Technical Note embeds an Image Matrix object as well, as seen on the left side:



Building Custom Framework Applications

From a developer standpoint, what is attractive about the 4D Ajax Framework is that embedding these objects only really takes a line or two of JavaScript code. This is much less than if you were to create an Ajax application from scratch. Also, no lines of code are necessary at all for those who have Dreamweaver. There is a “4D Ajax for Dreamweaver” extension that allows you to embed 4D Ajax Framework objects into web pages in a point-and-click fashion. No code necessary.

Conclusion

This Technical Note is aimed at giving 4D Developers some clarity about Ajax and the core technologies it is comprised of. With much of the introductory material out of the way, hopefully the idea of porting a 4D Database to the web is not so overwhelming. For more information on Ajax there are many resources on the web. For more information on the 4D Ajax Framework, there are many Tech Tips and Technical Notes that explain how to use the framework as well as how to perform advanced functions such as creating custom applications. Also, Daxipedia.com is a good resource for referencing many of the custom commands available for the framework. Now that you now have your feet wet, take the plunge into 4DAF development!