

Getting Started with the 4D Ajax Framework

By Larry Sharpe, 4D Developer, InfoService

Technical Note 07-33

Abstract

This Tech Note is meant to help 4D developers who are new to the Web 2.0 world. The goal is to help developers understand what the 4D Ajax Framework is and how to get started using it.

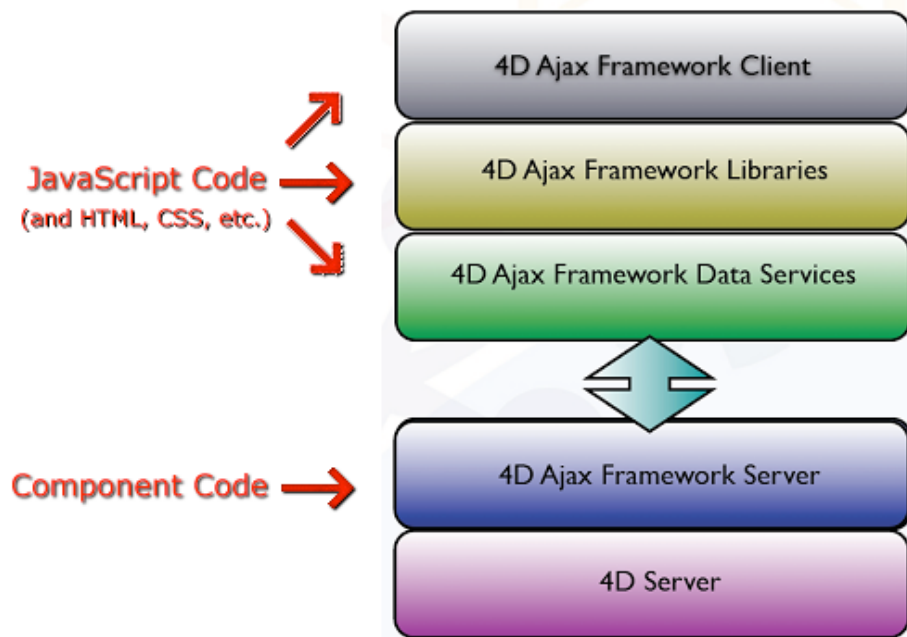
Introduction

4D Ajax Framework (4DAF) is part of the 4D Web 2.0 Pack, along with 4D Live Window and the 4D Ajax for Dreamweaver plug-in. Note that the 4D Web 2.0 Pack is made up of many facets and other products may have been added since the writing of this Tech Note.

What is the 4DAF?

From a 4D perspective a very simple definition of the 4DAF is a 4D Component along with some HTML, CSS, JavaScript, and image files.

The 4DAF is really built of several pieces. Here is a diagram to illustrate the different layers:



There are two layers that are the most important in terms of the Web interface: **4DAF Libraries**; and the **4DAF Client**. In very simple terms, the 4DAF Client is a web client implementation built on the 4DAF Libraries. This is the default client implementation, if you will. You can tweak some of the settings of the 4DAF Client and customize it to a certain degree, but to build a highly customized implementation you use the 4DAF Libraries.

What does the 4DAF do?

Once installed the 4DAF allows a web browser (Firefox, Safari, IE, etc.) to query, view, add, modify or delete 4D records via web pages served from your database using 4D's web server functionality. By comparison a very simple definition of the 4D Live Window plug-in is it allows you to bring web browser functionality into your 4D database. In other words 4DAF goes in the opposite direction.

All the necessary 4D code and HTML/JavaScript based web pages are supplied with the 4DAF Client for these actions giving you pretty much instant access to your database over the Internet. After you get up to speed with what is available you can add or modify sections of the 4D code and all the HTML/JavaScript pages as needed. You can add these functions to an existing 4D-based web site or just start with the basic 4DAF Client and go from there.

Why use the 4DAF?

4D has long supplied developers with a means of serving their databases as HTML-based data over the Internet using the 4D web server commands. However you had to write all the HTML and 4D code needed to handle that functionality. The 4DAF now gives you that capability with about 15 minutes of work. True, a more custom look-and-feel, or specific functionality may take a little time and effort, but the groundwork has been done for you. The 4DAF is designed and focused for building (or learning how to build) access to your database via the Internet. You could recreate all the 4DAF functions yourself, but you would have to design, write, test and debug all that code yourself.

Another great thing the 4DAF does is open the door to the Web 2.0 coding style that is becoming more and more common and requested by clients. Web 1.0 designed pages were much more static, data changes to a section of the page meant that the entire page would be reloaded. Web 2.0 design means only the updated section of the web page actually needs to be reloaded, not the entire page. The Web 1.0 to Web 2.0 switch is not really a technical specification change, but more of a technical philosophy change in the way developers can use the web interface in a more robust, functional manner. So, if nothing else, the 4DAF provides a great training ground for understanding and learning how to develop Web 2.0-based Ajax web sites.

To truly get into making significant changes to the functionality of a 4DAF-based web site you will need to work with several different programming languages and formats. It is not just 4D and basic HTML anymore. You will need to use 4D, HTML,

XML, JavaScript, CSS and 4DAF functions and have them all work together to make a web site with access to your database. However, if all you need is to quickly get your database up on the Internet then an afternoon's worth of work installing and understanding the basics of the 4DAF Client is all that is required.

Getting Started with the 4DAF

The following is a list of what information and tools are currently available and a proposed order in which to learn them for new users of the 4DAF. At the end of this Tech Note is a list of web sites with useful information that may help as well.

4D Ajax Framework Installation Guide (PDF)

This PDF document details how, and where, the various 4DAF pieces need to be installed into your existing 4D database. The list of 4DAF pieces are the Component, plugins folder, DAX folder, Extras/Support folder, adding the necessary calls to the DAX_??? methods to your database structure, and setting the 4D web server preferences.

4D Ajax Framework Admin Reference (PDF)

This document contains full documentation about how to use either the *4DAF Admin Client* (described next) or the web-based administration of the 4DAF. This is where you customize the look as well as the functions available to access your database when using the 4DAF Client.

4D Ajax Framework Admin Client (Compiled 4D application)

This compiled database is another way to setup and make changes to your 4DAF settings. Instead of using the web interface to set these preferences, you use this 4D application to do the same functions. This standalone application communicates with the web server in your web-enabled database much like a web browser but using a 4D interface instead.

4D Ajax Framework Developer Guide v1.0 and v1.1 (PDF)

This PDF document describes the various commands and functions that are available to be updated in the 4D code (i.e. the "back-end"). These are references to the 4D code changes only, not the HTML or JavaScript programming (i.e. the "front-end").

Daxipedia (Wiki Web Site)

This web site covers the JavaScript commands that are used with the 4DAF. This is an evolving web site, so check often for changes and additions:

<http://daxipedia.4d.com/>

Tech Notes/ Tech Tips (4D Partners web site)

There is a growing list of Tech Notes and Tech Tips available that will help with specific parts of the 4DAF. After going through this Tech Note, the others will make much more sense:

<http://www.4dpartnercentral.com/>

The PDF files mentioned above can be downloaded here:

<http://www.4d.com/support/documentation.html>

The 4DAF Admin Client application can be downloaded here:

http://www.4d.com/products/downloads_4d.html

Installing and using the 4DAF


The 4DAF starts with installing and modifying the following items: 4D component (using 4D Insider), 4D Pack plug-in, DAX folder, Extra/Support folder, adding calls to the DAX_??? methods in your 4D structure, and updating the 4D web server preferences. These steps are well described in the *4DAF Installation Guide*. Note: please do not forget to relaunch your database at the end of all these installation steps.

Once you have installed the various pieces of the 4DAF you should have access to your database using a web browser like Safari or Firefox. The interface is currently set to the default settings for accessing your database, which means that you can log-in to view your database over the web and use all the functions available via the 4DAF Client. Using the "Administrator" username (and password if needed) allows you into the "back door" of the 4DAF where you can set many of the options available. The *4DAF Administration Guide* offers full documentation on how to set up viewing and accessing your database, but we will cover some of the options in the next section of this Tech Note.

Setting up and adding functions to the 4DAF

Changing how your database looks and operates is done using either of two similar tools. The first is found using your web browser and entering the "back door" mentioned earlier.

This is accessed by clicking the "Show Control Panel" link on the "Welcome to 4D Ajax Framework" web page (circled below):



Welcome to 4D Ajax Framework!




The 4D Web 2.0 Pack is a set of applications, tools, plug-ins, and components that allow a 4D developer to quickly harness the power of Web 2.0 technology and create lightweight, agile web and widget based applications over a 4D framework.

Right now you are using the 4D Ajax Framework, a web-based toolkit that helps you develop fully functional Ajax-powered applications without having to invest a lot of time learning CSS and JavaScript.

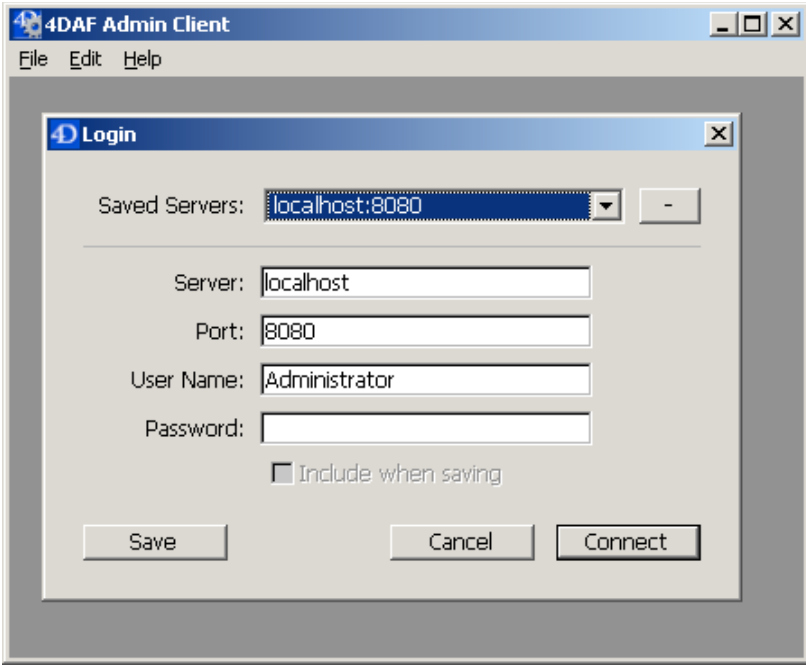
Use this application for fast prototyping of a new application, or to rapidly convert existing applications to a web framework. Integrate third party data with developer-defined windows and create new Mash-ups With the 4D Ajax Framework, you have the power.

What you do with it is up to you.

[Sign In](#) with your 4D Database account. Show [console window](#) Show [control panel](#)

Choose your language:   

The second method is by running the 4DAF Admin Client on your computer and setting up the "Login" window:



The screenshot shows the 4DAF Admin Client application window. It has a menu bar with 'File', 'Edit', and 'Help'. The main content area is titled 'Login'. It contains a 'Saved Servers' dropdown menu with 'localhost:8080' selected. Below this are input fields for 'Server' (localhost), 'Port' (8080), 'User Name' (Administrator), and 'Password'. There is a checkbox labeled 'Include when saving'. At the bottom are three buttons: 'Save', 'Cancel', and 'Connect'.

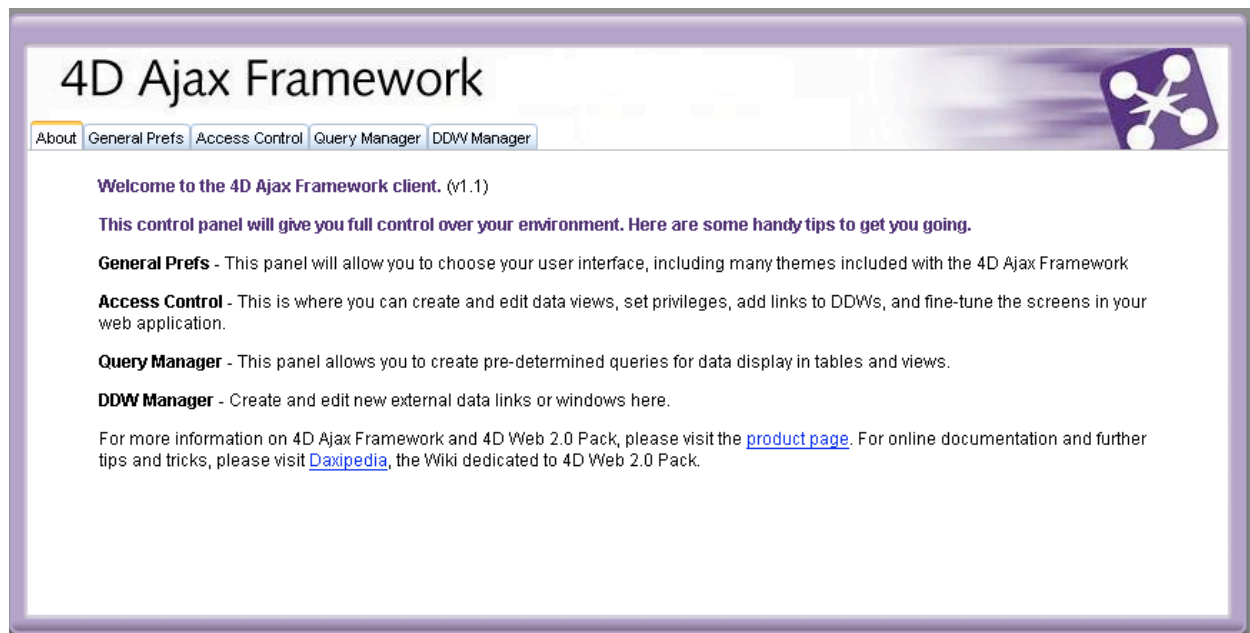
This tool is a merged, single-user 4D application that allows you to access and change all the 4DAF settings over the Internet in a standalone 4D application instead of a web browser. Once either tool is logged into the database the look and feel are very similar. Overall I prefer the 4DAF Admin Client over the web-based version because the user interface seems to function a little faster and has a few

more user interface options.

One thing to note is that the web version of the 4DAF settings will automatically log-in and log-out of the database and its Control Panel web page while the standalone 4D database does not. Using the standalone 4D database you will get a dialog that alerts you to that fact and the web pages will not always have your new changes until you log-out and log back in. The simplest way to work through this is that every time you switch applications remember to log-out of the current application, switch to the other application and log-in to it. Do this when switching applications in both directions and everything will workout fine.

Using the Control Panel (or 4D Admin Client)

You are shown five tabs in the Control Panel; About, General Prefs, Access Control, Query Manager, and DDW Manager:



We will briefly cover each of these tabs. See the *4DAF Administration Guide* for more detail.

About tab - This tab displays information about the currently installed version of the 4DAF, as well as a description of the other tabs.

General Prefs tab - This page lets you set where the opened View's buttons are placed and which theme to use when displaying the 4DAF web pages. It is pretty straight forward with no need for anymore comments here with the exception that with some work you can create your own themes.

Access Control tab - This is where you can change which tables and fields are displayed and how they are displayed. To word it in a way that 4D Developers are familiar with, you setup the output forms (Views) and once a form is defined you setup the linked input form (Sheet) to edit the data. Unlike 4D code, you can not change input forms on the fly, each View and Sheet (output and input form) is tied together. The first step is to define how you want your Views to look. Most of the settings are pretty straight forward but there are three major things to know about.

1. You can change the Style (how each View is shown) from the default Grid (similar to a standard output form) to a Calendar, Data Tree (think of a hierarchical list) or a Data/Image Matrix. This Tech Note will only cover setting up and using the Grid style.
2. You can setup the DDW (Developer Defined Windows) buttons, see the DDW Manager section a little later in the Tech Note for more info on the DDW.
3. When working with related tables you can setup the Parent Record. This column lets you pick which related tables/fields to display. When using the Data Tree style you can create a hierarchical list based on the One-to-Many relation between two tables. When using the Grid, you will show the parent record multiple times for each related child record. In other words, if you have a list of [People] records, each with multiple [Keywords] records, showing the related [Keywords] will show multiple instances of the [People] record for each [Keyword] record that is related to it.

Once you have setup the options for the View of a table, you can setup the options for which fields to display and how they function in that View. Each of your selected fields has each of the following options available to it. Again most of the settings are pretty straightforward with the exception of the following five items:

1. Which fields to show in the View window. Using the first two columns you can select which fields to show and in what order to show them. There are several very important things to keep in mind here:
 - a. Whatever fields you show, or do not show, in the View is also what shows, or does not show, in the Sheet window. There is no way to show 5 of the fields in a View and 30 of the fields in the related Sheet.
 - b. A table with 50 fields shown in the View will build a Sheet window with entry into all 50 fields with very little to no customization of the look or feel.
 - c. Keep in mind that when the field has been disabled from showing, using the Control Panel, and you are developing a separate HTML page using HTML/JavaScript and the 4DAF commands, the enable/disable check box setting is followed. This one took a few hours to figure out why a few fields were not showing up correctly in my separately designed web page.
2. The Break Level is used for the Data Tree style View. It is how you specify the hierarchical list views of your data.
3. OnLoad Event. This text entry area lets you enter the name of a 4D Project Method to run when a Sheet window is first opened. Remember that a Sheet

window is the same as an input form, you just do not have much control over what it looks like. The OnLoad Event method runs for both new record creation and modifying an existing record. You could use it to preset certain values into fields, etc. Each field can have its own method or you can use one method and handle the division of work within the code.

4. OnChange Event. This is another text entry area in which to enter the name of a 4D Project Method to run whenever the Delete, Cancel or Save buttons are clicked. Only when a field is edited and any of the three buttons is clicked will the method be run. It is up to your 4D code to figure out what was changed, which button was clicked, and what you want to do about it. These two 4DAF events, OnLoad and OnChange, are referred to as Callbacks in the documentation and are covered a little later in this Tech Note.
5. DDW Link. Again, see the DDW Manager section a little later in the Tech Note for more info on the DDW.

Query Manager tab - Each View has its own defined Queries that are associated with it. There are three types of Queries that can be defined by you.

1. *Custom Query* - This is very similar to the standard 4D Query Editor with a few additions. You pick the View you want to create the Query for, set the name of the Query tab which will show up in that View, define the default sort order, define the maximum number of records to show, and then define the Query itself. When defining the Query, you pick the fields, conditions, and values you are looking for. You can also use the Method check-box. If this is checked then you should place the name of the Project Method to execute into the Value portion of the Conditions list. What this means is that the Value data is now based on whatever the 4D Project Method returns in \$0 of that method. The other parts of that Query are still in effect, it is just using whatever the Project Method returns as the Value to search for.
2. *Dynamic Queries* - The second option is letting the 4DAF set up the Query buttons for you. You pick the View and the Field you want to create Dynamic Queries for and it does the rest. Be careful here since the Dynamic Queries are just that, when first loading the View it is finding all distinct values of the selected field and then building a Query button for each of those values. If you have a table of 10,000 People records and tell it to build a Dynamic Query based on the Last Name field, it will go and build a huge number of buttons across the top of the View window. Not the most practical use of Dynamic Query. However if you are looking for Query buttons based on Marital Status (Single, Married, Separated, Divorced, or Unknown) then you will have a much more usable interface.
3. *Templates* - The last Query type is picking from the built-in Queries that are available in the pop-up list. You can use the 4D Project Method named DAX_DevHook_QueryAdd to add additional Query templates to this pop-up if you would like.

DDW Manager tab - DDW stands for Developer Defined Window, which will open a new window showing the specified HTML web page. This window can be anything from a very simple external URL to a much more complicated database driven

HTML/JavaScript web page built on the fly. Once you have created a DDW you go back to the Access Control settings and determine where you want the DDW link to be placed. Also keep in mind that, currently, there is no way to modify an existing DDW. You will have to create a new, updated DDW and then reselect it in the appropriate places using the Access Control window. Do not forget to delete the older DDW if it is no longer correct or needed.

A DDW entry has four fields which all need to be completed:

1. *Title* - This is pretty straight forward, it is the name that shows up in the Access Control DDW selection pop-ups and what the user sees as the button's title. Do not make it a really long title or you may run out of space on the web page.
2. *Content* - Depending on what the DDW is meant to do, this will either be a URL (e.g., <http://www.4d.com/>) or a 4D Project Method (e.g., MyMethodName) which sends the appropriate HTML content to the new window.
3. *Type pop-up* - There are three Types displayed in the pop-up but the *4DAF Administration Guide* talks about four types of DDW. The pop-ups three names are different names than the documentation is describing but hopefully we will clear this up here.
 - a. *New Window - Link* - Allows you to create either a *Static Link DDW* or *Dynamic Link DDW*. By selecting a Static Link you will be opening a window pointing to the URL entered in the Content field. When using the Dynamic Link you are running a 4D Project Method, named in the Content field, in order to build the URL to open.
 - b. *New Window - HTML Method* - This is where you define the *HTML BLOB DDW* by using a 4D Project Method, named in the Content field, to create an HTML/JavaScript based web page which will then be opened in a new window. This expects a full HTML page, stored in a BLOB, not just a URL to an web page.
 - c. *Status Message - Text Method* - This is the *Text Message DDW* were you use a 4D Project Method, named in the Content field, which will access your database as programmed and return a text value that will be sent in a message to the user.
4. *Associated To pop-up* - Depending on what Type of DDW is being created this is where the DDW will be made available in order to be added to the 4DAF web pages. Again it slightly confusing as there is a choice of two locations, but those two choices are really for three locations and are where you will place the three different Types selected in the pop-up above which are really four different Types of DDW:
 - a. *Portal* - Obviously this is what you select if you want to add a button to the Portal section of the interface. This selection does not support a *Type of Status Message - Text Method* and it does not know what table or records you have selected in the 4DAF interface, but you can do whatever you would like within the code or link that is set here. Selecting this value also means that the DDW is automatically added to the Portal window, as a button, with no further work needed.
 - b. *Other* - This selection is really just setting up the option of adding a button in two (really three) places of a selected View; either as a button

at the top of a View, or a link on each row of a View (and as a button on its related Sheet). Selecting *Other* does not automatically place the DDW button on View for you. You will still have to go to the Access Control tab and select a View or a View's columns to specify the DDW to use. This part just makes the option available to you. Here is a graphic view of the various *Types* and *Associated To* options and their relationship to each other. Hopefully this will help it make more sense:

Type	New Window - Link		New Window - HTML Method	Status Message - Text Method	
	Static Link	Dynamic Link	HTML Blob	Text Message	
Associated To	A Static URL	Project Method returns Dynamic URL (as Text)	Project Method returns HTML (as BLOB)	Project Method returns Message (as Text)	Where it show up
Portal	X	X	X		Portal
Other	X	X	X	X	Top of a View (near Queries)
Other	X	X	X	X	Field Level (both for each Row in View and the appropriate field in Sheet)

Programming the 4DAF

All of the settings covered so far are pretty much handled using the 4DAF web pages. There were a few places that you can add the name of a 4D Project Method to perform a function, but most of the database functionality has all been set using the 4DAF Client. In this section of the Tech Note we will talk about the component Project Methods that are supplied with the 4DAF. The *4DAF Developer's Guide* covers what each 4D Project Method does and how you can use them to override the default behavior of the 4DAF. The documentation does a pretty good job of going through the changes you can make with the 4D code, but we will describe the basics here.

There are eight categories of 4DAF component Project Methods:

Preferences - This one method, DAX_DevHook_Preferences, is where you can change any of the global preference variables that used within the 4DAF.

Authentication and Sessions - There are four methods associated with this category. The main use of these methods is to provide customizable security functions to your database. By default the 4DAF provide all the functions needed using the standard Users and Groups setup within your 4D database. If you have written your own security functions and are not using the standard 4D tools, then these methods can be used to tie your functions into the 4DAF.

Data Modifications - These two methods allow you to check whether a record is, or is not, to be saved or deleted and if any other functions are needed to be completed at that time. As an example, when deleting a record you could check for any related records, delete them and then accept the command to delete the parent record, or cancel the entire function if appropriate without deleting anything.

Developer Created Selections (DCS) - Using the four methods in this category, you have the option of adding a custom View to the Portal that is defined and built entirely in your 4D code. It is up to you to define the arrays used and fill them with whatever data you would like to be shown to the user. You can write 4D code to change selected records, save any changes or delete elements of the arrays shown. This View is not customizable using the 4DAF Control Panel, it is up to the developer to make any changes required.

Developer Defined Windows (DDW) - This one method is very similar to creating a DDW from the 4DAF Control Panel, but here you can define the same four variables within your 4D code and not have to create them using the 4DAF Control Panel. You will still need to add them to the correct View using the Control Panel.

Callbacks - There are two places that callback methods can be used; when a record is first opened in the Sheet window and when the Delete, Cancel or Save buttons are clicked on a Sheet. These callback methods can be setup to handle when these events happen in either the 4DAF Control Panel, discussed briefly earlier, or you can update a 4D Project Method to define the callback. In either case, you create a method that is run when these events happen and within that method you can get which table, field, record id and action triggered the callback. Also note that the 4D Trigger method is run, when appropriate, so you do not need to duplicate that code within the callback methods.

Queries - Using these two methods you can create preset queries that will be available when setting up a 4DAF View. One method lets you define what Query buttons to show on the View and the other method allows you to filter records from the View before any records are actually shown. This last function, filtering, is very useful if using a common development practice of not actually deleting records but marking them as deleted instead. You can filter out those marked records so they never show up to the user.

Lists - A nice feature of the 4DAF is that it supports creating a selection list for field entry using the standard Lists Editor in 4D and assigning it to a field. You may want to build your own list that is built on the fly and not stored in the structure. In this case you can use this method to perform that function.

Using the 4DAF code but building your own web pages

At this point you can change the look and feel of the 4DAF pages, what tables and fields are available to view and edit, or other aspects of the standard 4DAF interface, but you still use the full 4DAF Client. You may want to use just one View

of your data as part of your company's web site, but not use the full 4DAF Client or functionality. You should still install, setup and define your 4DAF using the tools covered here to get it working. Once everything is working the way you would like it to it is time to add the customized interface to your web site.

Much of what you will need to configure and write is contained in HTML/JavaScript web pages that will be served by your 4DAF based database. The 4DAF Javascript commands are documented in the *Daxipedia web site*. Please refer the *07-21 4DAF Custom Data Grid Tech Note* which covers the basics of how to make this happen.

Conclusion

This Tech Note was meant to help 4D developers who are new to the 4DAF and who want to get their databases into the Web 2.0 world. When first learning about Web 2.0 coding and design, there are many different programming languages, documents and tools that all need to be used together to make this happen. With the 4D Ajax Framework you start with your existing database skills and add a few minutes of work to get a fully functional web accessible database. Once the basics are there you can continue on with customizing your web site with 4D code changes and/or HTML/JavaScript pages as needed.

A Note about 4D Web 2.0 Pack

The products in 4D Web 2.0 Pack are a departure from most other 4D products. Thus please note that this Technical Note is based on 4DAF 1.1. As new features are implemented this Technical Note may become obsolete (faster than most other 4D products). For the latest information on 4DAF consult the latest documentation and also check the 4D Web 2.0 Pack Wiki:

<http://daxipedia.4d.com>

For the latest news about 4D Web 2.0 Pack please see:

<http://www.4d.com/products/4dweb20pack.html>

Related Resources

What is Web 2.0?

http://en.wikipedia.org/wiki/Web_2.0

What is Ajax? (Ajax stands for Asynchronous Javascript and XML)

http://en.wikipedia.org/wiki/Ajax_%28programming%29

Why is 4DAF called a Framework?

<http://en.wikipedia.org/wiki/Framework>

<http://www.codeproject.com/gen/design/WhatIsAFramework.asp>

Larry Sharpe can be contacted at (831) 373-6266 or LSharpe@infoservice.com