

Chart FX Internet

Chart FX Client Server

Chart FX Financial

Image Toppings

SoftwareFX
Any Chart, Anywhere!

myChartFX.com

Map Web Service

Chart FX for .NET

Pocket Chart FX

Chart FX Real-Time

Chart FX Wireless

**The Truth About
Client Controls**

Chart FX Internet

Chart FX Client Server

Chart FX Financial

Image Toppings

WebBar FX

WebTree FX

myChartFX.com

Map Web Service

Chart FX for .NET

Pocket Chart FX

Chart FX Real-Time

Chart FX Wireless

www.softwarefx.com

Using client controls on intranet applications to maximize analytical capabilities and to enhance user satisfaction.

Security concerns, browser dependence and negative publicity have separated ActiveX and other client controls from the Internet spotlight resulting in many server-based Internet products producing standardized output such as DHTML and server-side image generation like JPG, PNG and GIF.

It was back in 1997 when most vendors were pushing the use of ActiveX controls that Chart FX introduced server-side image generation as way to complement its ActiveX approach. Back then; ours was the only server-based charting product capable of responding appropriately based on the browser's ability to handle client controls. Today, sever-side image generation is so popular that it is probably the single most important feature in a charting tool. In the meantime, it is unfortunate that client-side charting controls have become nearly extinct.

Nevertheless, with the popularity of the Internet came a new breed of applications. Developers started to take advantage of the server-based nature of the Internet, and as a result, intranets flourished in the corporate world.



As a way to meet the increasing needs for image interactivity on the browser, Chart FX incorporated new formats (SVG) and navigational features (image maps and drilldown capabilities). However, these images were not enough for intranet applications that required the same rich interfaces users were accustomed to in desktop applications. Because intranet developers were not faced with the same constraints as Internet sites (security, dependence, etc), they quickly realized the value of client controls, making Chart FX an essential front-end analysis and reporting tool for many intranets across the globe.

Of course, if you are developing a site geared to web surfers there's little we can do to convince you about the benefits of using client controls on the browser. Furthermore, we strongly encourage you to inspect the great image generation abilities in Chart FX. However, if you are developing mission-critical intranet applications, you should know there are many advantages and features at your disposal in the client controls that are embedded with any of our web based products: Chart FX Internet (COM) or Chart FX for .NET.

This paper is geared toward developers using or planning to use Chart FX and looking to enhance the interactivity of their intranet applications through the use of client controls.

How do Chart FX client controls work?

For those of you who are not familiar with Chart FX, you should know that, during installation, a series of components and virtual directories are created on your web server. Once installed, they allow chart integration in ASP, web form projects and ASP.NET pages by simply using the Chart FX API or design-time components for Visual Studio.

When a browser accesses a page containing a chart, Chart FX activates a browser detection feature that allows it to respond appropriately based on the platform and the browser's ability to use client controls¹. This mechanism makes the Chart FX server-side component generate a universally accessible image (JPG, PNG) or a binary file (OLE) that can be read exclusively by our client controls.

For example, if a Mac or Unix-based system accesses a page containing a chart, the Chart FX Server component will inspect the user agent coming from the browser and will quickly realize such a system can't handle ActiveX or .NET thus generating an image that can be displayed on the browser. In contrast, if a Windows-based system with Internet Explorer or Netscape, accesses the same page, Chart FX will generate a binary file that can only be read by a Chart FX client control. Finally, clients need to download and install signed cabinet files that will reside permanently on the client computer for current and future chart display and interaction.

As with any other development product, of course, you can override this behavior and force (via code) any file format. You could, for example, force image generation without taking into consideration the browser ability to handle client controls or, if you're in a controlled environment where all systems are Windows-based, you could force the use of client controls.

At first, it almost looks like there are not enough reasons or advantages in using such controls on your web-based applications. After all, these controls are platform dependent and use proprietary files, which prevent them from being universally accessible. Also, they must be downloaded and properly installed plus many developers think that these controls post significant security risks for the organization.

¹ This is the default behavior in Chart FX Internet (COM). Chart FX for .NET changed its behavior by producing images by default. Nevertheless, automatic browser detection and client controls are also available in Chart FX for .NET

Deployment, Security and Version Control

Essentially, client controls can be embedded in a web page, downloaded across the net, and run on a local machine. This creates a significant deployment advantage over desktop applications that require manual installation and registration in every client on which they run.

Furthermore, users can download client-side controls with exceptional ease — sometimes without even knowing it — and while non-trusted controls can be terribly dangerous, most vendors guarantee their code through digital signatures, a kind of encryption technology that unequivocally identifies its author. In some cases, .NET and Java controls aren't even allowed to perform many activities that compromise security.

In the end, it's important to remember most security mechanisms are geared to protect web surfers from bringing over executable code from untrustworthy sources that could potentially damage them or compromise security and privacy. In a controlled environment, like an intranet, users will not be allowed to download controls that haven't been certified by their licensees or publishers. Therefore, I would like, once and for all, to downplay the security issues associated with the use of client controls on intranets.

Similarly, there are those who would say that using client controls would contribute to DLL Hell on the client machine. In reality, when a vendor, like Software FX, releases new versions of their ActiveX controls, users receive the update automatically. The new version can be installed immediately, or pre-cached for offline installation later. In other words, professional software doesn't cause or contribute to this problem because it always checks versions before stomping on old or previously installed DLLs.

In particular, with every Chart FX patch or upgrade, new files are installed on your server allowing the CODEBASE portion of any server-side generated charts to be stamped with a version number. Then, the Internet Component Download mechanism downloads and installs the file only if the specified version number is more recent than any existing version of the same file currently installed in the system.

As you can see, many deployment issues have been elegantly addressed and have been available to intranet developers since the early days. Now consider new deployment advantages offered by .NET [e.g. No impact, secure and private components, side-by-side versioning, XCOPY deployment and replication] and there is almost no excuse for excluding these controls based on security and deployment.

Server Performance & Scalability

By definition, a client control is code that gets downloaded and runs locally on every client accessing a browser-based application. So, if the client provides the processing load, how can server performance improve when a client control is being used? The answer to this question is subject to the client control and the context on which it is being used. For a server-based charting solution, client controls can make a big difference on server performance, allowing your intranet application to accommodate a growing number of users and to give each user satisfactory levels of responsiveness. I'll explain:

Consider that when the ASP code is running, the Chart FX Server component will essentially create a chart for every page hit that it receives. When a chart image is produced, the server must paint this chart in a memory device context and, finally, save the chart to disk for immediate access and display. Bear in mind, this process has to take place every time a page gets accessed. In other words, if the underlying data or chart attributes haven't changed, the same chart is being created and painted over and over affecting server performance.

In contrast, client controls enhance server performance since the server generates a binary file that contains property settings and chart attributes that the client control will use to paint and display the chart, concentrating much of processing load on the client. Similarly, binary files are much more compact and easier to save and transmit than raster images generated on the server helping to keep disk space and bandwidth use at a minimum.

Also, a common practice among web developers is to accommodate a chart size that can be useful on a browser. After all, a chart is a data analysis tool. However, this practice can be an important factor on how your server behaves and performs under heavy load. Essentially, a bigger chart means a larger image that needs to be processed, generated, stored and finally downloaded; affecting, in one way or another, the overall application's performance. Therefore, you must be careful when choosing the final chart image if server performance is a real concern.

Quite the opposite happens when a control is being used since charts are being painted and displayed on the client. In other words, developers are no longer forced to set a chart size but they can simply specify a percentage as a final size for their chart; this will allow the browser to accommodate charts to the different screen resolutions client systems are using without requiring trips back to the server.

A recent performance test² unveiled that servers were capable of producing 5 times more charts per second when client controls are used to process and display charts on the browser. Therefore, if you need to make performance improvements or do capacity planning, you can definitely consider client controls as part of your strategy.

2 See Chart FX Performance and Scalability test at www.softwarefx.com

Client Interactivity

In previous pages, I wanted to shed light on some of the most controversial topics about using client controls on your intranet initiatives. However, you will find that most advantages mentioned so far are attributable to the ActiveX specification rather than Chart FX. However, the most important attribute a client control can provide is client interactivity and the features end users will be able to access without further intervention from the developer and one that, in my opinion, Chart FX outrivals any other competitor.

A decade ago Software FX introduced the Chart FX User Interface (UI) as a way to help developers cut development time by providing end users the ability to change chart types, colors and access tools like editors, legends, scroll bars and zooming capabilities thus improving the data analysis capabilities on top of their image counterparts.

You will learn that of all the advantages client controls will provide to your intranet applications none would be more appreciated than the Chart FX User Interface. Chart FX provides a wealth of tools that end users can access, among which I would like to highlight the following:

ToolBar and MenuBar

One of the most useful tools in Chart FX is the ToolBar; many programmers appreciate this feature because they are not required to add additional lines of code in their applications to give end users more freedom to customize their charts.

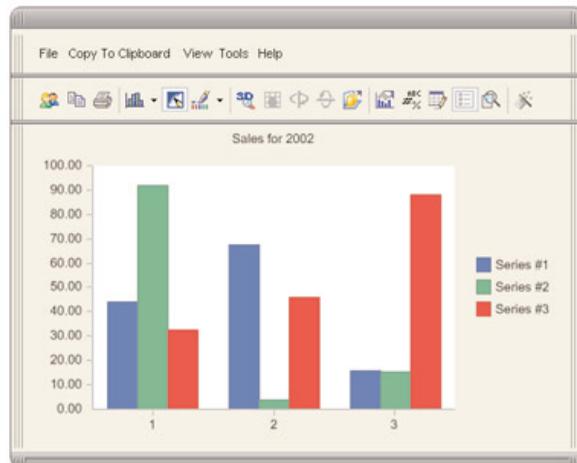


Figure 1. Toolbar and MenuBar integrate smoothly with the chart.

By default, the Chart FX ToolBar and MenuBar provide the most common options required by an end-user, such as switching from 2D/3D, color change, zooming & rotation and show/hide tools like data editor and legends. However, it can be completely customized by the developer according to application and user needs, by adding or removing buttons from the toolbar and even changing the tooltips displayed by the buttons.

Context Sensitive User Interface

While the ToolBar provides one-click access to many chart options, the user may find it difficult to access specific features such as axis and point customization. For that purpose, the Chart FX client components provide a comprehensive set of context sensitive UI menus and dialog boxes that allow the end user to customize many of the options of the chart by right-clicking on specific elements and using the Context Sensitive User Interface.

Depending on where on the chart the end user presses the right-button, a menu will show up with the features inherent to that specific element. Chart FX provides UI menus for customizing general chart features, titles, axis, each particular series, legends & data editor, constant lines and stripes.



Figure 2. Context Sensitive User Interface helps users locate element specific features.

Some of the UI menus provide a Properties option for further customization. This option will bring up a properties page dialog box that allows customizing of general chart features, series and axes. Depending on whether the end user selected a particular series or axis, the dialog box will change accordingly.

Zooming and Scrolling

One of the clearest advantages our client component provides is the ability to zoom and scroll. By using the zooming capabilities of Chart FX, end-users can see a more detailed portion of chart, making the chart more useful as it can show both general and detailed information with the same data.

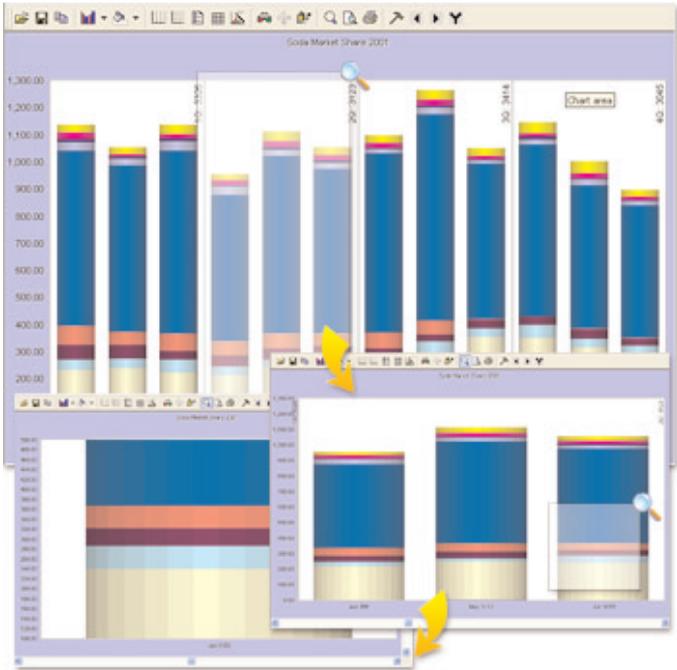


Figure 3. Smart zooming capabilities are provided by Chart FX client components.

Additionally, Chart FX provides scrolling capabilities, which allows for a better readability of the chart when it has unusually large data sets. This feature is also enabled when the chart is enlarged to allow further views of data during a specific zoom state.

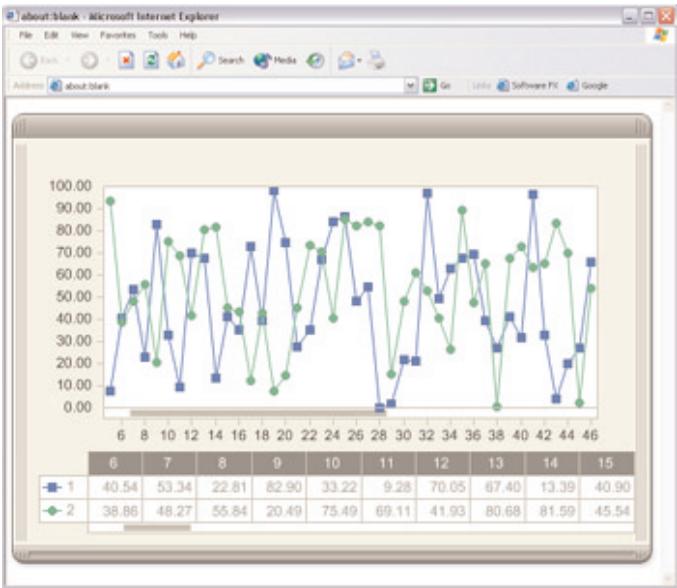


Figure 4. Horizontal and vertical scroll is available for the chart, as well as, the data editor.

Rotation and Perspective

Another nice feature of the Chart FX Client components is the ability to change the rotation angle and perspective of a 3D chart. It can be done directly from the toolbar or from a dialog box for more accuracy.

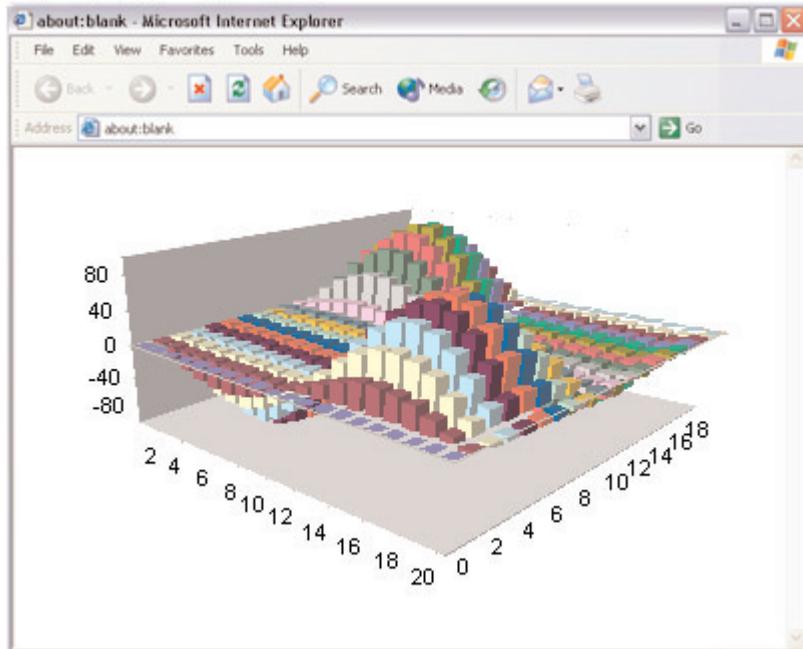


Figure 5. Charts can be rotated to any angle or degree of perspective.

Personalization

As commercial web sites and corporate intranets become more and more sophisticated, users require and demand information on an individual basis. This is one the reasons why sites allow visitors to customize pages with individual topics of interest and other personal information.

When it comes to Chart FX, there's a lot the user can customize about charts. Picture this, a particular visitor likes the graphical capabilities of Chart FX in your site; there is, however, one minor issue that annoys him, which is the color scheme you have chosen for your charts. There is another user asking you to change your default bar chart for a line chart. You instruct them to use the Chart FX Client component UI to change the visual attributes of the chart every time they log on to your site.

But, wouldn't it be nice if your site could provide charts with visual attributes they like but with new data for subsequent visits? Using the client component, Chart FX can deliver personalized charts to your users. This way, the chart will look exactly as each user wants it, regardless the data displayed at any given moment.

Annotation

Chart FX provides the ability to add Annotation objects to a chart programmatically, so you can highlight values in a certain scenario. But the annotation extension of Chart FX also provides the end user the option to add his/her own annotation objects, from a simple interface using the Annotation Toolbar.

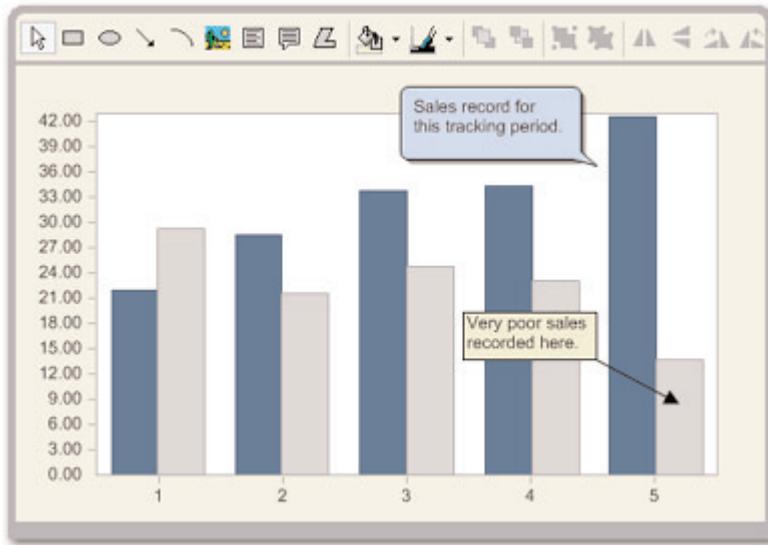


Figure 6. End users can add their own annotations to a chart using the annotation objects offered by Chart FX.

Printing

Another advantage of using Chart FX Client component over an image is the ability to print only the chart in a page that has other elements, and more important, print the chart using the printer's resolution instead of the screen resolution, which provides a much more readable printed chart.

Additional Functionality

Last but not least, Software FX has built a complete line of products for vertical markets; such as Chart FX Financial and Chart FX Real-Time, which rely, heavily, on client controls to provide the necessary functionality required in these types of applications.

Consider the case of Chart FX Financial where analysts require the addition of hundreds of different studies to properly make investment recommendations. Some of these studies not only perform complex computations and require rich user interfaces but they need to be performed repeatedly until a trend is finally detected.

Similarly, Chart FX Real-Time uses client controls to display real-time charts on the browser by establishing asynchronous connections to a real-time data server using industry standard protocols such as XML, HTTP and SOAP.

For these types of products, client controls provide the following advantages:

1. Since users have the expertise or require control to further analyze data on the browser, there are very few things the developer can do to customize the chart. This entails the application to use client controls that provide a complex UI that allows users to customize the chart and to perform additional data analysis without additional intervention from the programmer or server.
2. Client controls allow load distribution on many of the complex calculations that are required, making the intranet application suitable for performance and scalability. In other words, producing server-side images will concentrate much of the processing load on the server thus affecting performance and scalability.
3. In many situations, controls are mandatory since they use system related resources on the client machine. For example, establishing an asynchronous connection to a server and allowing a multitude of clients to connect to the same server requires the client code to invoke and access local resources that aren't simply available by the browser. Similarly, repainting and scrolling a real-time chart requires the use of the client machine's GDI resources.

Client Control Advantages (cont'd)

4. The only UI permissible in a browser are html forms which makes it impossible or impractical for certain features that client controls handle very easily through natural and intuitive interfaces. For example, when displaying a financial study (e.g. Fibonacci Arc) the user will be forced to input two dates which are not usually labeled in the chart; to overcome this problem the client control allows the user to position the mouse on top of the starting date and drag the mouse to the desired ending date, as soon as the mouse is released the study will be immediately redrawn without requiring additional trips back to the server. In contrast, when producing a server image, the user will be forced to write the dates, hit a submit button and download a new image from the server. This obviously makes it impractical when the task must be performed repeatedly as the user must locate and write dates that aren't clearly labeled on the chart.
5. Client controls allow selective refreshing. In other words, when producing a server-side image, the whole page must be refreshed before the new image can be displayed on the browser. This is a major inconvenience. As the rest of the page elements that were not affected by the change must also be downloaded, making certain applications (real-time charts) impossible to achieve without the help of client controls.

My goal was to show you that client-side controls can be deployed effortlessly; will significantly contribute to enhance server performance and allow your application to provide a level of interactivity that is practically impossible to achieve through images. Without them, you may be failing to provide your users a great analytical experience that can be a competitive advantage for you and your organization.

Hopefully, you've learned that having access to even one of these great advantages can tip the scale in favor of client controls. If you decide to stick with server-side image generation, at least it will be an informed decision.

Chart FX Internet

Chart FX Client Server

Chart FX Financial

Image Toppings

WebBar FX

WebTree FX

myChartFX.com

Map Web Service

Chart FX for .NET

Pocket Chart FX

Chart FX Real-Time

Chart FX Wireless

Chart FX Internet

Chart FX Client Server

Chart FX Financial

Image Toppings

WebBar FX

WebTree FX

myChartFX.com

Map Web Service

Chart FX for .NET

Pocket Chart FX

Chart FX Real-Time

Chart FX Wireless

About Software FX

Software FX began in 1993 with one idea in mind, creating the most powerful, yet easy to use, data analysis and reporting solutions for developers. Today, Chart FX positions itself as the worldwide leader in helping developers integrate and display graphical information between and among diverse markets, platforms and environments.

Software FX

5200 Town Center Cir., Suite 450
 Boca Raton, FL 33486
 (800) 392-4278
 (561) 999-8888
 tech support (561) 392-2023
 fax (561) 998-2383
www.softwarefx.com
www.mychartfx.com

©2002 Software FX, Inc. All rights reserved. All other brand names are trademarks of their respective owners. Software FX is not liable for any errors or omissions. Some information may change without notice.

www.softwarefx.com