EspressChart®

Technical Overview

Architecture, Technology, and Detailed Feature Specifications For the Powerful, and Versatile Java Charting Tool

Quadbase Systems Inc.

Contents:

I. Introduction 3

II. Chart Types 3

A. Column/Bar Charts 3
B. Stack Charts 4
C. XY Charts 4
D. High - Low Charts 5
E. Pie & Doughnut Charts 5
F. Dial Charts & Gauges 6
G. Combination Charts 6

III. Customizable Chart Attributes 7

A. Chart Axes 7
B. Labels 7
C. Lines & Points 7
D. Legends 7

IV. Statistical & Analysis Features 8

V. Chart Designer 9

VI. Chart Data 9

A. Database Data 10
B. XML Data 10
C. Java Object/Array Data 11
D. SOAP Data Source 12

VII. Chart API 12

A. Using Applets 13
B. Using Images 14

VIII. Technology & Architecture 14

A. EspressChart Components 15
B. EspressChart Configuration 15

IX. Conclusion 16

I. Introduction

A picture is worth… Everyone knows how the euphemism ends, but if the picture is an interactive, dynamic chart, it's probably worth a great deal more than 1000 lines of data. In fact nothing conveys salient information better than a well-designed chart or graph. EspressChart® from Quadbase Systems provides a feature-rich, powerful Java™ solution that allows users to easily enhance Web pages and applications with dynamic charts.

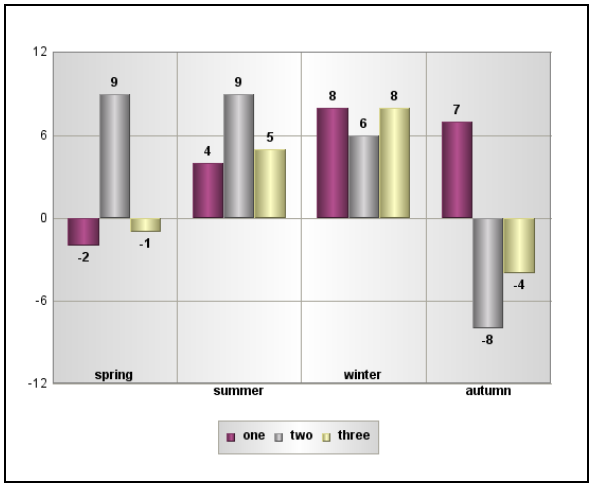
Written in Pure Java, EspressChart easily deploys in most application environments, and can generate over 30 different types of charts for applications and Web content. The powerful Java API allows charts to be created and modified programmatically. With hundreds of customization features, EspressChart makes it easy to create powerful data presentations for virtually any situation.

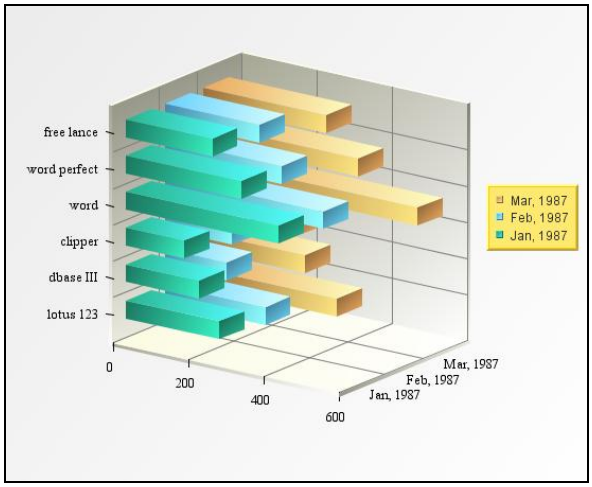
This paper gives a technical background of EspressChart, highlights some of the powerful features of the product, and explains modes of usage.

II. Chart Types

EspressChart supports over thirty different chart types in both two-dimensional and three-dimensional layouts. Chart types allow users to display up to four dimensions of data in a single plot and even more using combination/composite chart options. Unlike many charting tools, EspressChart renders charts with true 3D allowing users to pan, zoom, rotate, and even adjust the shading and ambient light source effects for charts.

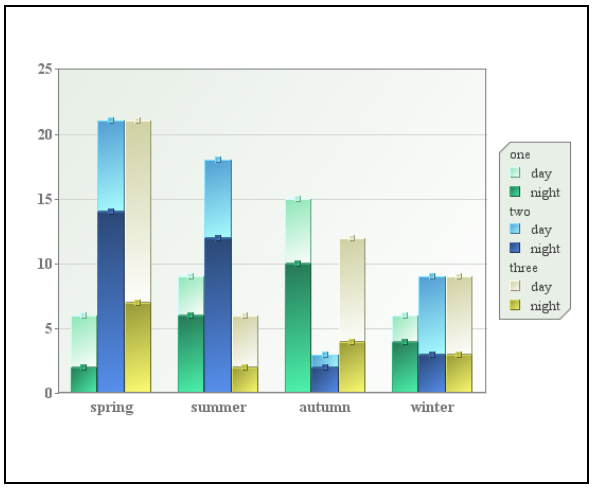
A. Column/Bar Charts

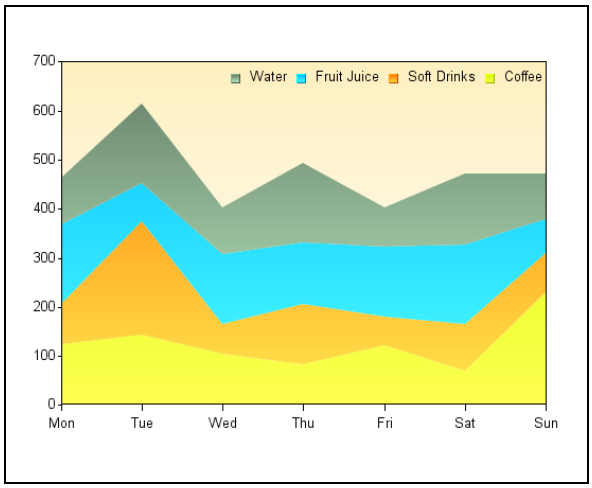




Basic chart layouts in EspressChart consist of a discrete number of categories plotted on one axis and a numeric value for each category plotted on the other. This layout is often shown in a columnar form as pictured above, but can also be rendered as bars, lines, and shaded areas. An additional dimension of data called a series can be added to these charts. A series will plot 2 or more values for each category.

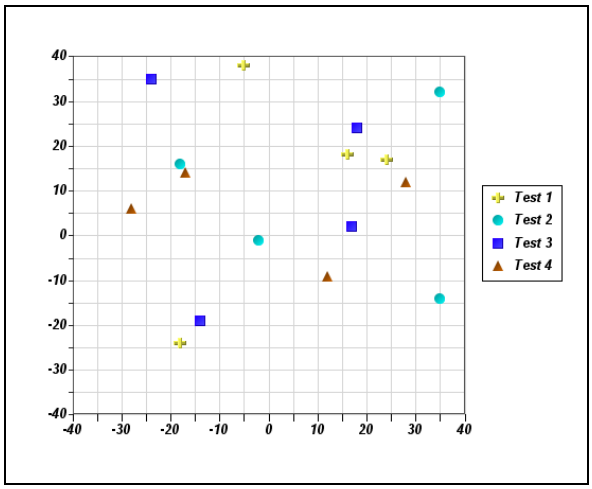
B. Stack Charts

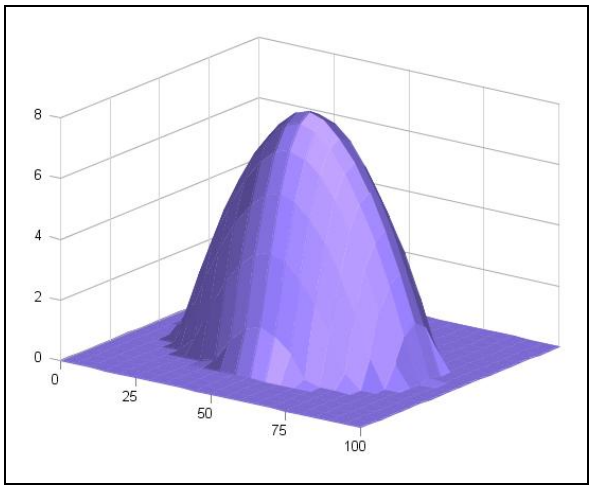




Stack charts allow an additional dimension of data called a sum by. The sum by values is aggregated to show the total for each category. EspressChart supports column, bar, and area stacking. A special type of chart in this category is a 100% column chart. This chart, instead of showing the aggregated value, shows each sum by value as a percentage of the total.

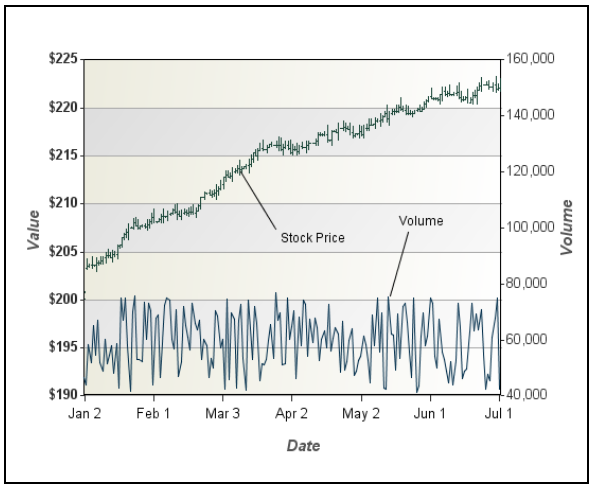
C. XY Charts

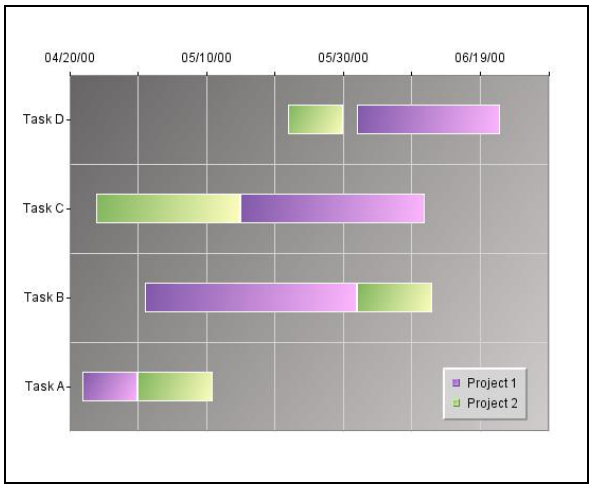




XY charts allow users to plot points in two-dimensional or three-dimensional space. Most common among these are scatter plots, where each point has an XY coordinate, or an XYZ coordinate. EspressChart allows users to add lines or columns to scatter plots.

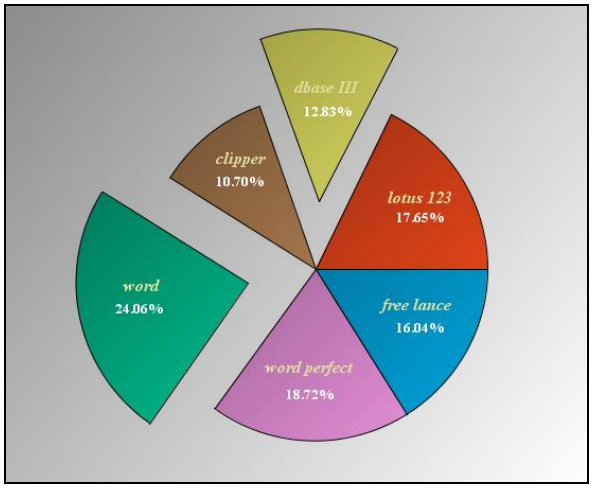
D. High - Low Charts

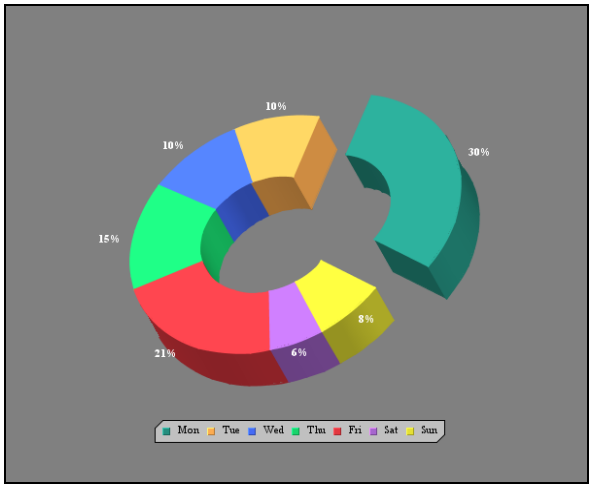




Useful for charting pricing data over time, high-low charts are most commonly used for plotting stock price data. High-low charts allow two to four values to be associated with each category and plot a connecting line, point, or candlestick to show the points. A Gantt layout is also supported that allows each category to have time-based data points, for task start and end times.

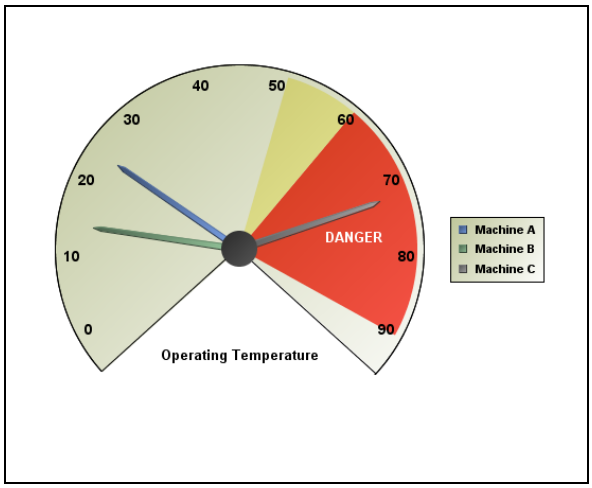
E. Pie & Doughnut Charts

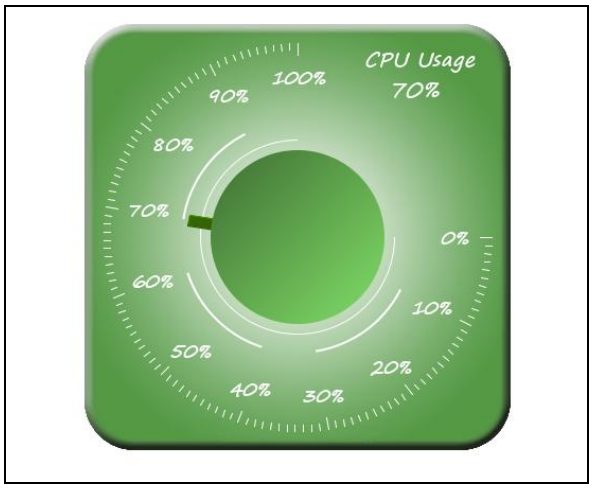




Rather than plotting points on a two-dimensional or three-dimensional area, pie charts plot points as percentages of a whole. This common business chart type can be drawn as a single pie, or plot a data series as multiple pies, one for each category.

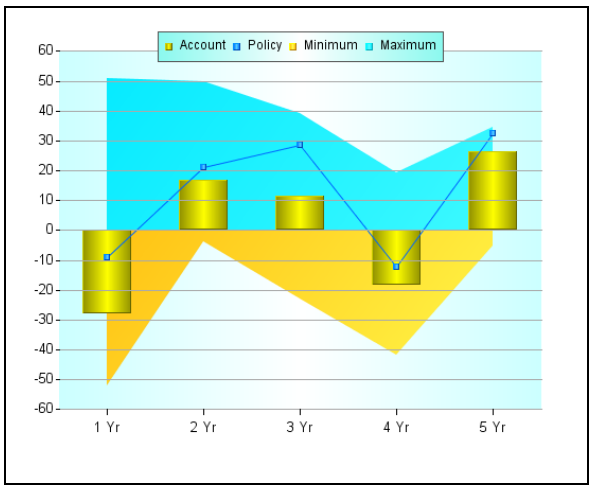
F. Dial Charts & Gauges

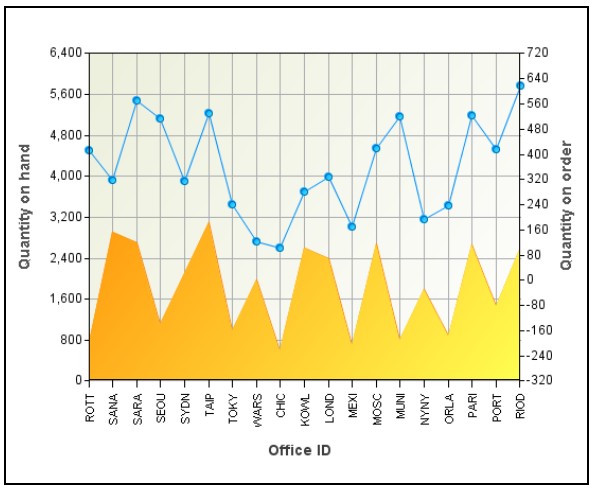




EspressChart also allows data to be drawn as hands on a dial or gauge. This representation is useful for dashboard applications, and other monitoring software.

G. Combination Charts





In addition to the many chart types, users can easily create their own custom combination charts. Combination charts allow multiple values for each category to be plotted as a different type on one or more axes. Using these multiple chart representations, users can create chart plots for virtually any applications.

III. Customizable Chart Attributes

EspressChart allows users to modify over 240 different chart attributes. Everything from axis tickers to legends, to extensive element annotation can be added and modified. Chart attributes can be set in a template at design-time or programmatically through the API.

A. Chart Axes

Chart axes support automatic or manual scaling. Under automatic scaling the axis will scale to the best range to fit the data supplied to the chart. The scaling will adjust as the data changes. Users can also set fixed or logarithmic scale. Beyond the range and step for the axis, EspressChart also supports origin shifting. Users can specify any value for the chart origin; they are not restricted to 0,0. For charts with multiple axes, users can set the scales independently, or align them with the same scale.

Other axis formatting options allow users to draw grid lines across the chart plot, modify the axis appearance, and set the intervals and layout for tickers and labels.

B. Labels

Virtually any font element can be customized for chart labels. Users can control the font, style, size, and even angle for all labels in the chart. Value and category labels are generated automatically for any chart. In addition, users can add custom titles, and annotation to specific chart elements, as well as show the specific values for each data point in the chart.

C. Lines & Points

EspressChart allows users to draw lines and points for data points regardless of chart type. Lines can be formatted in a number of ways. Users can control line thickness, and create custom dash patters for dotted lines. A variety of point shapes and sizes are supported, and users can select different point shapes for series elements.

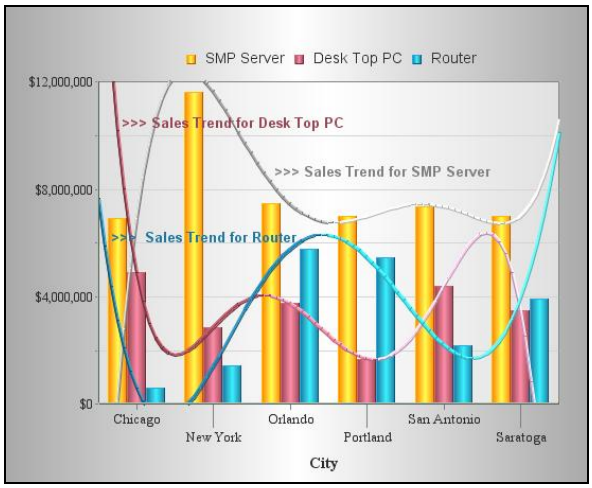
Users can draw additional floating lines anywhere on a plot, or anchor fixed horizontal/vertical lines to specific X or Y axis values.

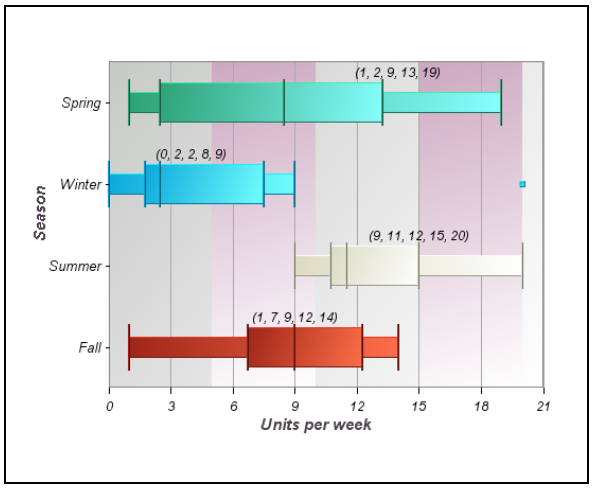
D. Legends

Chart legends can be modified to use vertical or horizontal layout. Users can draw a border or background for the legend. Users can control which chart elements to display in the legend. For ultimate flexibility, users can create their own legends using an API interface.

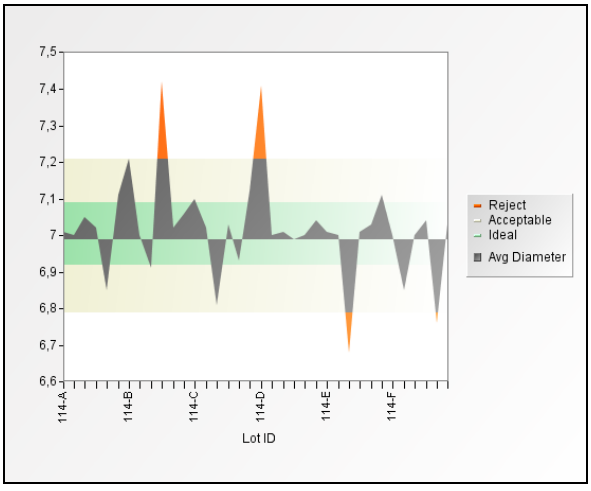
IV. Statistical & Analysis Features

EspressChart supports a number of statistical and control features in charts. Statistical chart options include a box and whiskers plot that shows the distribution of data for each category with any outliers. Users can also track frequency data using Pareto and histogram plots. In addition to these plot options, users can add many types of trend lines to charts. Trending options range from moving averages, to logarithmic functions, to normal distribution curves.





EspressChart also allows users to setup control lines and areas. These features are useful for SPC-type applications. They allow users to see when data falls into a particular range, or past a threshold with just a glance. Control areas can be drawn on a chart plot, or used to highlight chart data points.

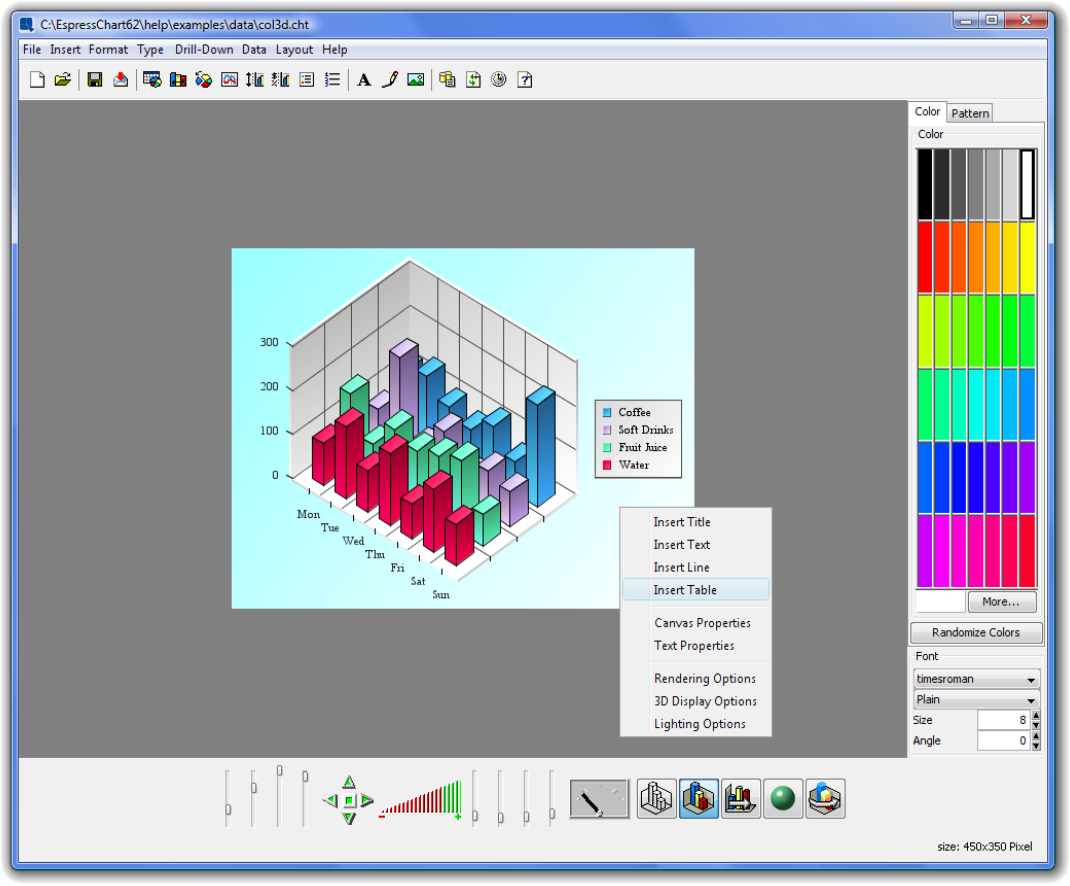


Built-in drill down capability allows users to create linked series of charts. Users can click a particular data point and drill into another chart with details for that particular category/series value. Drill down charts can aggregate data at the top level, and show more details of the chart data as users click through, or they can link different charts with different data using query parameters.

For charts that plot data over time, EspressChart provides a time-series zooming feature. Using this feature, uses can aggregate data points into specific intervals. For example, a user could plot a year's worth of data, by showing the average of all the points for each month. This feature provides a clean view of charts that may otherwise contain too much data.

V. Chart Designer

EspressChart includes a powerful design tool that allows users to create charts in a point-and-click visual environment. The Chart Designer allows users to separate the chart design process from application development. In addition the designer allows developers to have a visual reference when creating charts. Most of EspressChart's features are accessible in the designer, allowing users to implement complete charts, not just basic layouts. Templates created in the Chart Designer can be run directly in applets or applications, or their appearance properties can be applied to other charts.



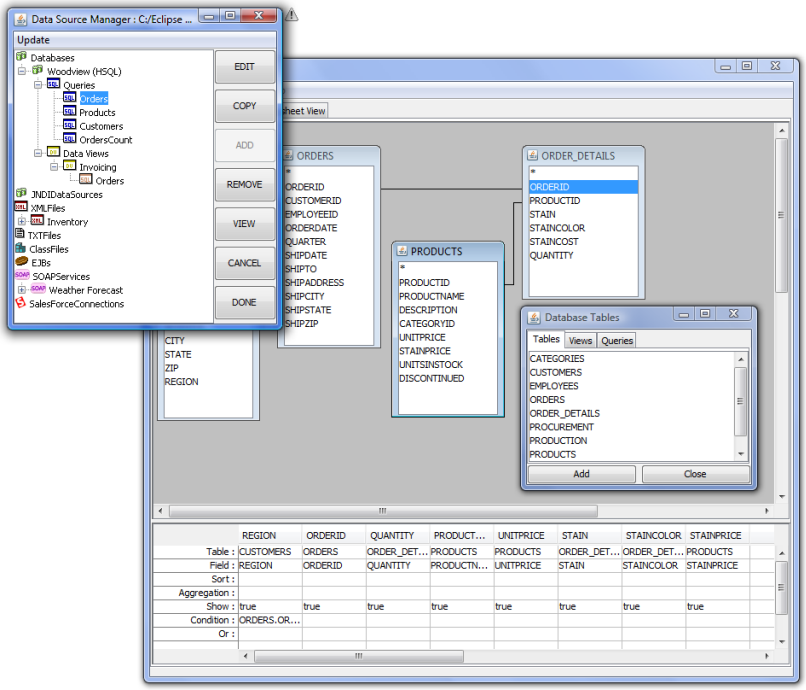
The Chart Designer is a pure Java application, which allows it to run on any platform. In addition it can also be installed on a Web server, and run through a client browser with zero install on the client. For maximum flexibility API hooks are provided for the Designer, allowing it to be launched programmatically in a number of different configurations.

VI. Chart Data

EspressChart provides a number of different options for supplying a chart with data. Chart templates are directly associated with a data source. This allows users to run a chart in their code without having to re-specify a data source. Charts will automatically connect to the source with which they were created, and retrieve the latest data. Users who would prefer to feed data to the chart at run-time can do so as well. Several API interfaces allow users to define chart data and pass it to a chart object.

A. Database Data

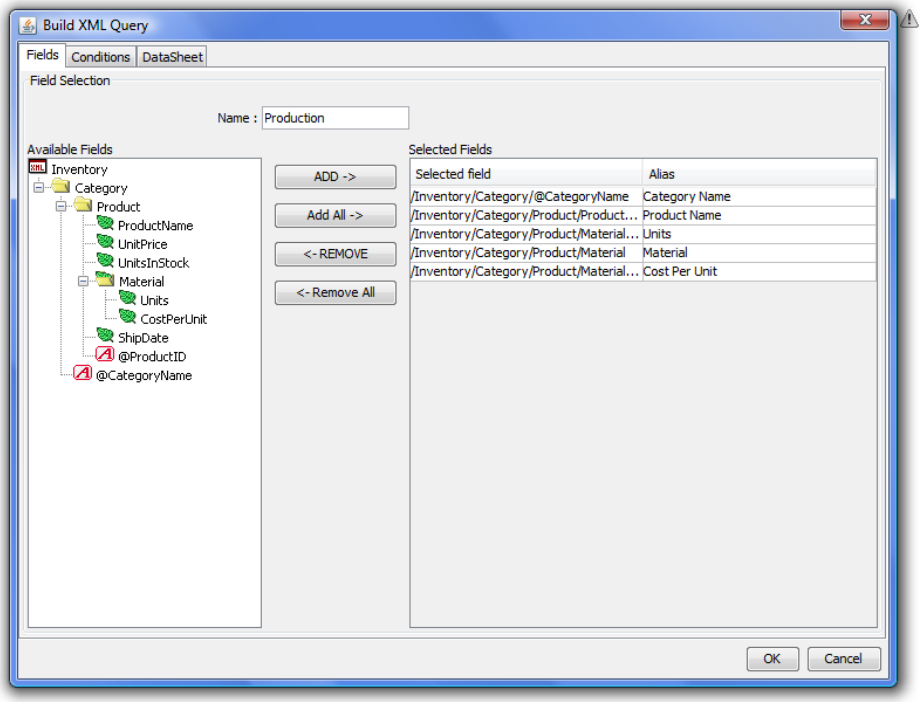
EspressChart can directly query relational databases via JDBC or JNDI data sources. Users can run a user-provided query, or create one using the graphical query builder that is integrated with the Chart Designer. The query builder allows users to design queries in a QBE style format that allows them to select fields, conditions, joins, aggregation, and even build complex expressions using imported and custom database functions.



EspressChart also allows users to define dynamic parameters for chart queries. At run-time parameter values can be supplied programmatically, or submitted by the user. EspressChart includes a unique feature that generates an HTML form with parameter options automatically, and returns the chart in the desired format using a servlet.

B. XML Data

EspressChart can also retrieve data in XML form. It includes several interfaces to set up XML data definitions and query them. Based on a DTD or XML Schema, users can generate queries against an XML file or an XML output stream from an application. EspressChart can also retrieve data from delimited text files.



C. Java Object/Array Data

In addition to directly retrieving data from relational databases or XML sources, EspressChart lets you pass data directly into a chart from Java objects or arrays, or by connecting to EJBs. At run-time data can be passed directly into the chart, and at design-time object/array data can be pulled from a Java class file. EspressChart provides several different interfaces for users to import their application data.

Code Sample: Class to return a simple data array to a chart

```
import java.awt.*;
import quadbase.util.*;
import quadbase.ChartAPI.*;

public class dataSource implements IDataSource {

        // Create a two-dimensional data array
        String dataTypes[] = {"string","double","double"};
        String fieldNames[] = {"Products","Orders","Units Sold"};
        String records[][] = {{"Chair","4","24"},{"Table","2","12"},
                              {"Dresser","1","6"},{"Cabinet","2","10"}};

        DbData data = new DbData(dataTypes, fieldNames, records);

        // Create an empty constructor
        public dataSource() {};

        // Implement getResultSet
        public IResultSet getResultSet() { return data; }

}
```

D. SOAP Data Source

EDAB allows the user to retrieve data using SOAP (Service Oriented Architecture Protocol). Two types of SOAP data source are supported, namely WSDL compliant and Salesforce. To connect to a SOAP data source using WSDL, the user does not have to know URLs for the services, SOAP actions, operation names and parameters. All that is required is the location of WSDL file, which contains all the necessary information. A couple of dialogs in Data Source Manager facilitate setting a WSDL SOAP data source easily.

Likewise for existing Salesforce users, Salesforce SOAP data sources can be set up in a couple of simple dialogs in Data Source Manager. The user must have a valid Salesforce account and have access to a Salesforce account from a trusted network. A SOQL (Salesforce Object Query Language) needs to be included to complete the data source set up. Query parameters can be added to the query in the same manner as database queries.

VII. Chart API

The included Java API (or application programming interface) is used for most chart deployments. The API contains handles to all of the features in EspressChart and allows users to implement charts with a near endless degree for flexibility. Using the API, developers can incorporate charting functionality into servlets, JSP, and applications. Users can create charts on the fly, or run templates created in the Chart Designer interface. Charts can be displayed in applets, or exported to a variety of image formats for integration with Web content.

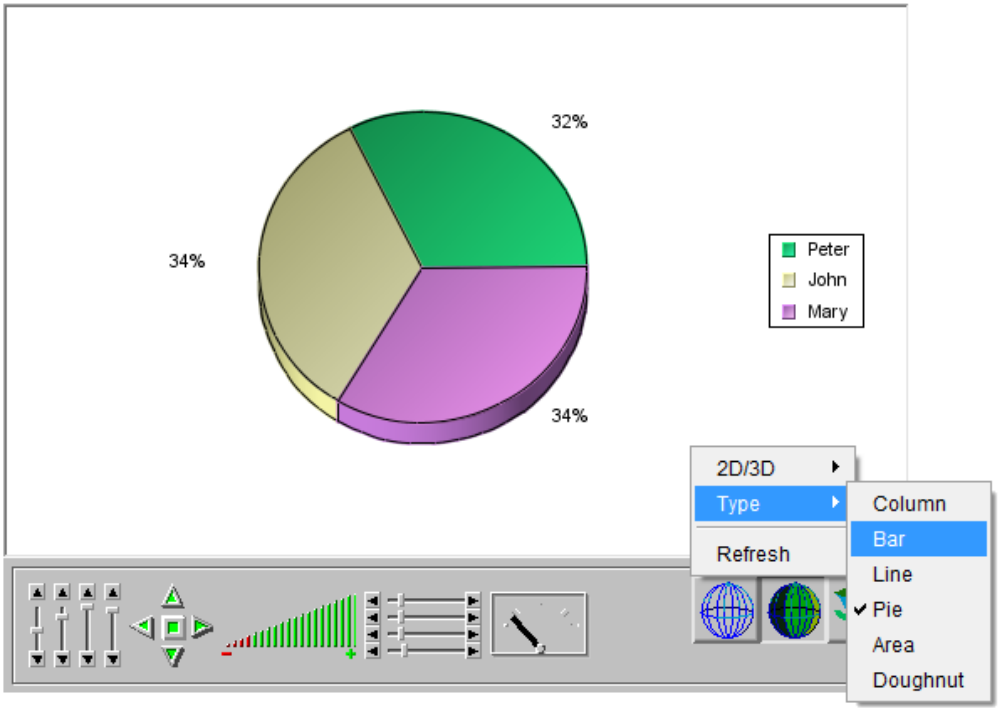
Code Sample: Create a chart and stream it to the client as a GIF image

```
import java.awt.*;
import java.applet.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import quadbase.util.*;
import quadbase.ChartAPI.*;

public class chartServlet extends HttpServlet implements SingleThreadModel {

       public void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {

               res.setContentType("image/html");
               OutputStream toClient = res.getOutputStream();
               QbChart.setEspressManagerUsed(false);

               try {

                       // Create a two-dimensional data array
                       String dataTypes[] = {"string","double","double"};
                       String fieldNames[] = {"Products","Orders","Units Sold"};
                       String records[][] = {{"Chair","4","24"},{"Table","2","12"},
                              {"Dresser","1","6"},{"Cabinet","2","10"}};

                       DbData data = new DbData(dataTypes, fieldNames, records);

                       // Map the data to the chart
                       ColInfo colInfo = new ColInfo();
                       colInfo.category = 0;
                       colInfo.value = 1;
                       colInfo.subvalue = 2;
```

```
                        // Create a column chart
                        QbChart chart = new QbChart((Applet)null, QbChart.VIEW2D,
QbChart.COL, data, colInfo, null);

                        // Export the chart to a GIF image
                        chart.export(QbChart.GIF, toClient);

                } catch (Exception ex) {

                        ex.printStackTrace();

                }

                toClient.flush();
                toClient.close();

        }

}
```

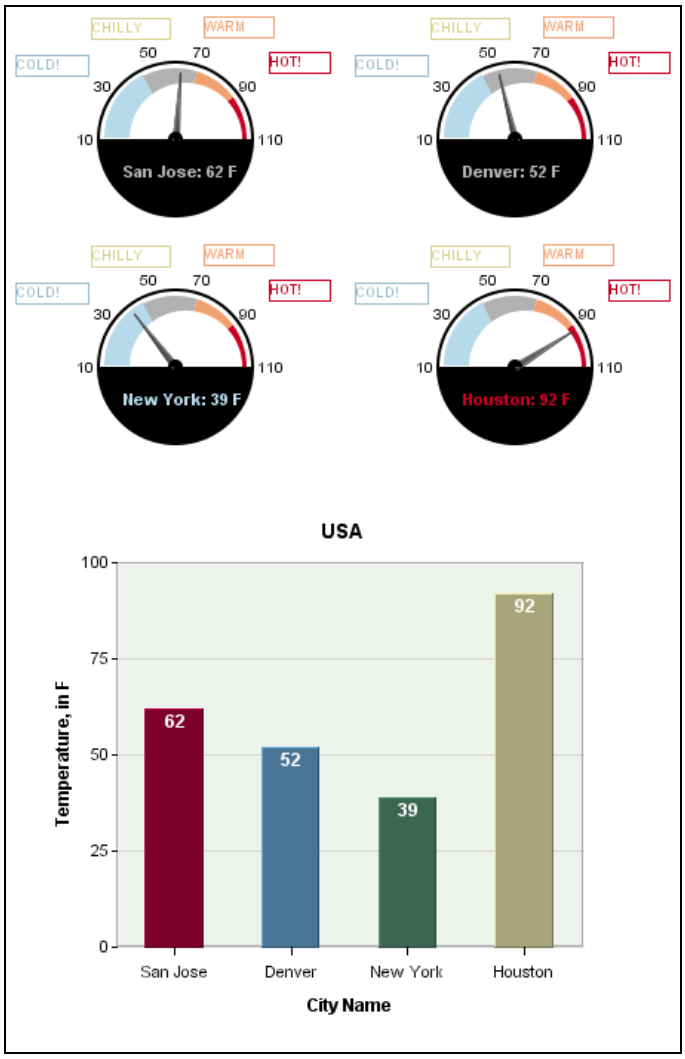
A. Using Applets

EspressChart provides several viewer interfaces that can be integrated into applet or application content. These viewer components load the chart directly, allowing users to interact with the chart in real-time, and showing live up-to-the-minute data. There are three different viewer components: Chart Viewer, Mini Viewer, and Micro Viewer. Each sacrifices some functionality and has a smaller client footprint.



B. Using Images

Charts can also be integrated with thin-client Web content by exporting them to image formats. EspressChart can generate GIF, JPEG, PNG, WMF, Flash, SVG, and BMP images as well as PDF files. Although charts in these formats are generated on the server-side they still support many interactive features. Users can generate image maps along with the exported images, allowing them to include hyperlinks, drill-down and mouse-over pop-up labels with generated charts.



VIII. Technology & Architecture

EspressChart is a pure Java product, giving it the flexibility to run on nearly any platform. It runs on most JVM's equivalent to JDK 1.4 and higher. EspressChart is compatible with most application servers and servlet/JSP containers such as WebLogic™, WebSphere™ and Apache Tomcat. It can directly connect to any database using JDBC, or using a JNDI lookup name for deployed J2EE data sources.

A. EspressChart Components

EspressChart has four associated components:

Chart Designer: Chart Designer is a GUI tool that allows users to build and design charts in a point-and-click environment. Included with the Chart Designer are interfaces to access data and query databases. The Chart Designer can run as a client application or in a client-server configuration with the Designer loaded as an applet through a Web browser.

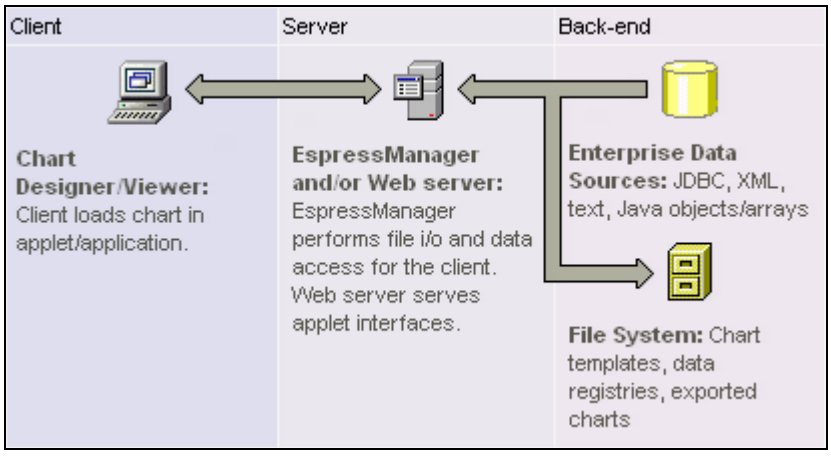
Chart API: Chart API is an easy-to-use application programming interface that allows users to imbed charting functionality into their applications, servlets, or JSPs, either on the server-side or client-side. Because it is pure Java it can run on most platforms with few or no changes. Charts can be created programmatically, or users can run pre-defined templates. A chart can be created and deployed with just a few lines of code.

Chart Viewer: Chart Viewer is an integrated applet that allows charts to be viewed remotely. Chart Viewer provides full user interaction on the chart, and can display live data. EspressChart includes two other smaller applets for interactive chart viewing: Mini Viewer and Micro Viewer. Each sacrifices some functionality and has a successively smaller client footprint.

EspressManager: EspressManager serves as the "back end" to the Chart Designer interface. It supports Chart Designer running as an applet by handling the data access and file I/O activities on the server-side. In addition, EspressManager provides database connection and data buffering.

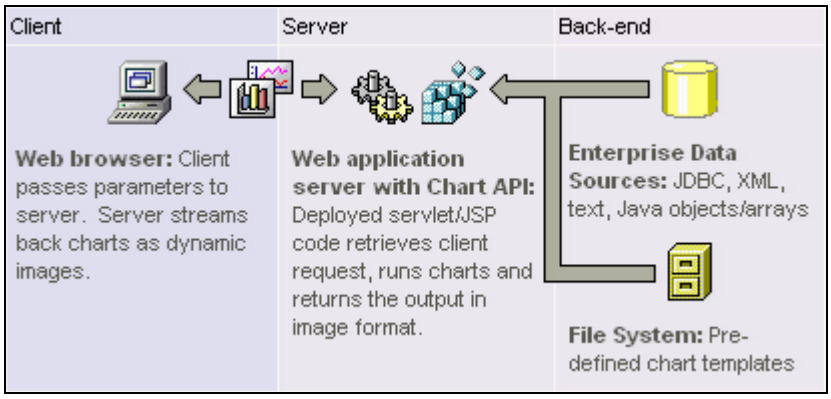
B. EspressChart Configuration

EspressChart has a number of different configurations in which it can run both at design-time, and at run-time. At design-time the Chart Designer GUI interface, which includes data access tools can be loaded as an application on a client machine, or as an applet in a client server architecture. The following diagram illustrates EspressChart running with the Chart Designer.



When Chart Designer is running, the EspressManager component runs on the server-side. The EspressManager performs the data access and the file I/O that is prevented by the client applet due to security restrictions. The EspressManager also provides connection and data buffering for database connections, as well as concurrency control for a multi-user development environment. The EspressManager must be run in conjunction with the Chart Designer.

At run-time, the EspressManager does not need to be running. EspressChart is designed to run embedded within other application environments, and can have a minimal deployment of only the API classes. The following diagram illustrates EspressChart running in a Web application environment.



When running in an application server, the Chart API can be used to generate charts within servlets and JSPs and stream the generated charts as images to the client browser. Clients can also view charts, by loading the Chart Viewer applet.

IX. Conclusion

EspressChart is a complete Java charting solution. With flexible API-driven chart creation, users can easily integrate charting functionality into application environments on any platform. Users can deploy charts in thick-client applications using the provided viewer interfaces, or including dynamic charts in Web content, running charts on the server-side and streaming them to the client.

With over thirty different chart types, and hundreds of customizable chart attributes, EspressChart can create a chart for just about any requirement, and the integrated visual designer allows users to create charts in an intuitive point-and-click environment. EspressChart's full feature set and flexible deployment make it the standard for Web-based charting.