

## Web Reporting with Ease, Flexibility, and Power

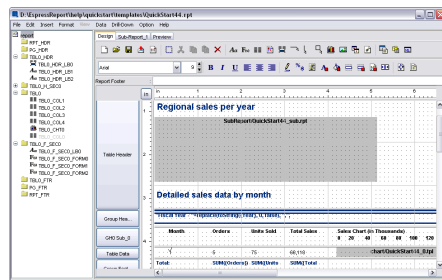
The explosive growth of Web technologies, and the resulting glut of new information has fundamentally changed the way in which data is presented and delivered. Companies now find that customers, partners and suppliers expect to receive high-value business information via the Web. Within organizations, thick-client stand-alone reporting tools are making way for thin-client browser-based solutions. Users expect to be served custom, interactive information directly rather than designing their own reports in a desktop environment.

EspressReport®, from Quadbases Systems, is designed to meet these new reporting challenges. Written in pure Java™, it offers the flexibility of platform independence, a powerful feature set, and a robust architecture designed for enterprise-level server-side deployment. By combining ease of use, flexibility, and power in a reporting engine, EspressReport offers a new standard in Java-based Web reporting. The visual Report Designer, robust API and integrated data management/retrieval mechanisms make a complete reporting package that is accessible to developers, as well as end users.

## Designing Reports

EspressReport provides a number of interfaces that make it easy for developers and end users to create polished, professional information presentations.

**Report Designer:** The integrated Report Designer tool allows users to create reports in an easy-to-use point and click interface. The familiar banded-style interface is similar to many legacy reporting tools and provides users a short learning curve. Most of the advanced reporting features are accessible through the designer, and can be implemented without any additional coding. API hooks allow the Report Designer to be customized and launched programmatically.



Report Designer

**Full Data Connectivity:** EspressReport allows users to build reports from virtually any data source. Users can draw data from relational databases, XML files, SOAP, text file, and Java objects/arrays. EspressReport provides three different query interfaces, giving users full SQL control. Advanced users can simply write, or import a SQL statement. Intermediate users can employ the integrated Query Builder which provides a point and click QBE style interface. For users with limited database knowledge, EspressReport provides the Data View Interface. It insulates users from database complexity, by allowing administrators to pre-define tables, joins, fields, and even filters creating in effect a local schema. End-users can then select (typically aliased) fields, and perform ordering, sorting, filtering, and aggregation without any knowledge of the underlying database structures.

**Extensive Data Visualization:** EspressReport supports a wide range of charting features. Users can draw data in one of over 30 different 2D and 3D chart types. The Chart Designer interface allows users to customize over 240 different chart attributes. Advanced features include time-series zooming, data drill-down, run-time text substitution, and mouse-over pop-up labels. Statistical features allow users to draw trend lines, normal curves, histograms, Pareto charts, box charts, and control lines and areas. Charts can be embedded in reports or deployed on their own, allowing users to stream dynamic chart images to existing Web content.

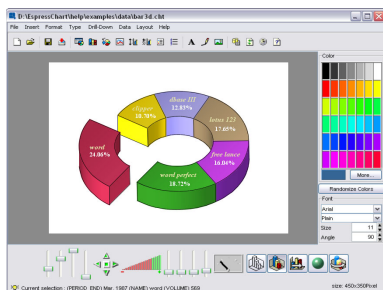


Chart Designer

## Key Features

- **Pure Java Architecture:** Deploy on any platform and run with most Java-supported application servers.
- **Full Data Connectivity:** Draw data from databases, XML files, SOAP, Text files, and Java arrays/objects/classes/EJBs. Unique Data View feature allows easy database access/querying.
- **Numerous Output Options:** Run reports in applets or generate HTML/DHTML, PDF, Excel, RTF, XML, text, or CSV files.
- **Visual Designer:** Powerful and easy-to-use design tool. Can be integrated into 3rd party applications.
- **High-Powered Data Visualization:** Draw over 30 different 2D and 3D charts with true 3D rendering. Embed charts in reports or deploy them independently.
- **Robust API:** Deep level Java API integrates reporting into applications, applets, servlets, and JSPs.
- **Flexible Configuration:** Run as stand-alone design tool. Run Designer on server or through API to integrate Report Designer in Applications. Run API classes server-side and embed powerful reporting functionality into Web applications.



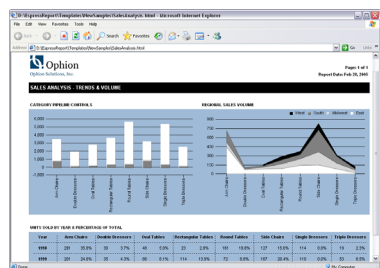
**Quadbases Systems Inc.**  
275 Saratoga Ave., #105  
Santa Clara, CA 95050  
Tel: (408) 982-0835  
<http://www.quadbases.com>

**Advanced Reporting Features:** EspressReport supports built-in drill-down mechanisms for charts and reports, allowing users to show summarized information in a top-level presentation, and drill into specific elements for more detail. Powerful sub-report capabilities allow users to build reports with multiple tables from different data sources, and establish relationships (joins) across them. Parameterization features allow users to implement query/Java class parameters to filter report data including multi-value array and cascading parameters. Scripts and formula parameters allow run-time report customization.

**Security and Template Re-Use:** Report security features allow users to develop several different views of the same report within a template. Depending on the view select, automatic filters can be applied, and report elements can be modified/hidden. Advanced internal scripting library allows users to embed custom logic in report templates.

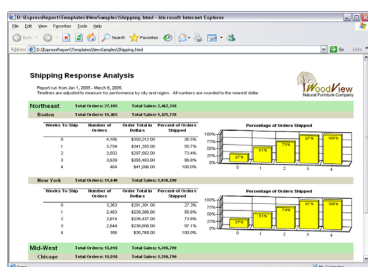
## Running Reports

**Report API:** Report deployment is generally handled through the powerful Java application programming interface. With only a few lines of code developers can easily integrate reporting into applications, applets, servlets and JSPs. The API can be used to create reports on the fly, or to run templates created in the Designer. Reports can also be customized programmatically. Using

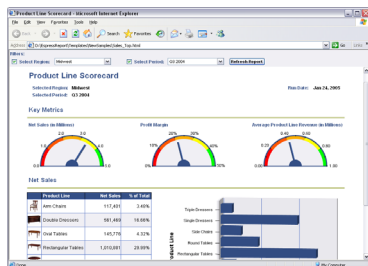


the API, developers can access and modify virtually any report element.

**Output Formats:** EspressReport easily integrates with Web and application content, and can generate reports in a number of different formats including tabular HTML, DHTML style sheet layouts, print-quality PDF, Microsoft Excel, and Rich Text. Report data can be exported to XML, CSV, and text formats.



**Using Applets:** EspressReport includes several lightweight applet viewers for showing reports in thick-client/application interfaces. Viewers allow reports and charts to be loaded directly, and allow users to interact/manipulate the report on the client. The Page Viewer component uses a powerful page serving technology that loads pages on the client only as they are requested. This allows large reports to be deployed quickly without client performance issues.



**Scheduling Reports:** A built-in scheduling interface allows users to schedule report templates, or command line operations. Reports can be generated to the file system or delivered via email. API hooks into the scheduler are provided allowing users to create and customize schedule jobs programmatically.

## Sample Code: Generate a report as HTML content

```
import java.awt.*;
import java.applet.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import quadbase.reportdesigner.ReportAPI.*;
import quadbase.reportdesigner.ReportElements.*;
import quadbase.reportdesigner.ReportViewer.*;
import quadbase.reportdesigner.util.*;
import quadbase.reportdesigner.lang.*;

public class SampleServlet extends HttpServlet implements
SingleThreadModel {

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

        // Set the "content type" header of the
        // response
        res.setContentType("text/html");

        // Get the response's OutputStream to
        // return content to the client.
        OutputStream toClient =
            res.getOutputStream();

        try {

            // Open up specified Report
            QbReport report = new
            QbReport((Applet)null,
                "templates/SampleReport.rpt");

            // Export (Stream) the report to
            // DHTML
            report.export(QbReport.DHTML,
                toClient);

        } catch(Exception e) {

            e.printStackTrace();

        }

        // Flush the outputStream
        toClient.flush();

        // Close the writer, the response is
        // done.
        toClient.close();

    }

    public String getServletInfo() {

        return "SampleServlet servlet for
        EspressReport";

    }

}
```